J. J. Rivkind, T. I. Liszenko,
L. A. Csernyikova, V. V. Sakotyko

Informatika

11

Akadémiai és profil szint

Tankönyv az általános oktatási rendszerű tanintézetek 11. osztálya számára

Ajánlotta Ukrajna Oktatási és Tudományos Minisztériuma

Львів
Видавництво «Світ»
2011

УДК 681.51(075)
ББК 32.81я721
І-74

Перекладено за виданням:
Інформатика: Підруч. для 11 кл. загальноосвіт. навч. закл. : академ. рівень, профільн. рівень / Й. Я. Ривкінд, Т. І. Лисенко, Л. А. Чернікова, В. В. Шакотько ; за заг. ред. М. З. Згуровського. – К. : Генеза, 2011

Рекомендовано Міністерством освіти і науки України (наказ МОН України № 235 від 16.03.2011 р.)

Видано за рахунок державних коштів.
Продаж заборонено

Наукову експертизу проводив Інститут кібернетики імені В.М. Глушкова НАН України

Психолого-педагогічну експертизу проводив Інститут педагогіки НАПН України

Ривкінд Й. Я.

І-74 **Інформатика :** підруч. для 11 кл. загальноосвіт. навч. закл. з навчанням угорською мовою : академ. рівень, профільн. рівень / Й. Я. Ривкінд, Т. І. Лисенко, Л. А. Чернікова, В. В. Шакотько ; переклад угорською мовою Г. Г. Семере. – Львів : Світ, 2011. – 304 с. : іл.

ISBN 978-966-603-716-2

Навчальний матеріал підручника поділено згідно з програмою на 4 розділи. В основу викладення навчального матеріалу покладені об'єктний та алгоритмічний підходи.

На початку кожного пункту наведено запитання для актуалізації знань учнів, на які спирається вивчення нового матеріалу. Для кращого сприйняття та засвоєння учнями навчального матеріалу підручник містить багато рисунків, таблиць, схем, зображень екрана монітора та інших наочних матеріалів. Для підвищення інтересу до вивчення предмета підручник, крім основного матеріалу, містить рубрики: «Для тих, хто хоче знати більше», «Це цікаво знати», «Цікаві факти з історії», «Додаткові джерела інформації».

Наприкінці кожного пункту наведено запитання для самоконтролю та практичні завдання, які розподілені за рівнями навчальних досягнень. Підручник містить 20 практичних робіт, виконання яких, згідно з програмою, є обов'язковим для всіх учнів.

УДК 681.51(075)
ББК 32.81я721

ISBN 978-966-603-716-2 (угор.)
ISBN 978-966-11-0068-7 (укр.)

© Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В., 2011
© Видавництво «Генеза», оригінал-макет, 2011
© Семере Г.Г., переклад угорською мовою, 2011

Kedves tizenegyedikesek!

Már néhány éve tanultok informatikát. Ez alatt az idő alatt megismertetek jó néhány szoftvert, megtanultátok alkalmazni az információs technológiát a mindennapi életben. Aktívan dolgoztatok az informatikai kompetenciátok fejlesztésén, ugyanakkor az információs technológia is rengeteget fejlődött. Ez utóbbi egyre inkább áthatja az emberi tevékenység valamennyi színterét, az intellektuális tevékenység nélkülözhetetlen eszközévé válik.

A 11. osztályban folytatjuk az informatika tanulmányozását. Megismerkedünk a **Microsoft Publisher** kiadványszerkesztővel, a **Microsoft Access 2007** adatbázis-kezelővel, megtanulunk videoklipeket készíteni és megismerkedünk a **web2.0.** technológiával. A legtöbb figyelmet azonban az informatika egyik legfontosabb részére, az algoritmizálásra és a programozásra fordítjuk majd.

A szerzők bíznak benne, hogy e könyv segítségével tovább folytatjátok majd az emberiség egyik legújabb tevékenységi területének megismerését, és gyakorlott számítógép-felhasználókká lesztek, a későbbiekben pedig az így megszerzett tudás segítségével jól boldogultok majd az információs társadalomban.

Tankönyvünk az iskolák 11. osztályosainak szól. A könyv tartalma és mélysége megfelel a nemzeti alaptantervben megfogalmazott tantárgyi követelményeknek.

A tankönyv négy részből áll, amelyek mindegyike néhány alpontra bomlik. Minden rész rövid bevezetővel kezdődik, amelyből megtudhatjátok, hogy mivel ismerkedünk majd meg az adott rész tanulmányozása során.

Az egyes pontok felépítése hasonló. Mindegyik kérdésekkel kezdődik, amelyek megválaszolása segít az új információk eddigi ismereteink közé történő beillesztésébe. Ezután az elméleti rész következik, majd az adott szoftver alkalmazásának módja, az alapvető tevékenységek algoritmusainak ismertetése kerül sorra. Minden pont végén ellenőrző kérdéseket, valamint az új ismeretek elsajátítását segítő gyakorlatokat találunk.

A kérdések és gyakorlatok nehézségi szintjét a következőképpen jelöljük:

- ○ alapszint, középszint;
- ● megfelelő szint;
- * emelt szint.

A feladatokat úgy válogattuk, hogy azok elvégzése egymásra épülve segítse a szoftverek alkalmazásának elsajátítását, ezáltal egyre alaposabb felhasználói ismeretekre tehessetek szert, megismerjétek a szoftverek működésének logikáját.

A tankönyv 20 kötelező gyakorlati munkát tartalmaz, amelyek elvégzése magas szintű felhasználói ismereteket feltételez. A könyv végén található tárgymutató az ismeretek aktualizálását hivatott segíteni.

Az ismeretek könnyebb és gyorsabb elsajátítását segítendő a szerzők rengeteg táblázatot, vázlatot, képernyőképet helyeztek el a könyvben.

Az *Aki többet szeretne tudni*, *Érdekességek*, *Érdekes történelmi tények*, *Kiegészítő információk* címekkel jelölt kiegészítő információk segítenek a tananyag alaposabb elsajátításában.

Az **www.allinf.at.ua** webhelyen a diákok és tanárok a gyakorlati munkák elvégzéséhez szükséges fájlokat, valamint kiegészítő anyagokat találhatnak. Ugyanezen a webhelyen a fórum és a vendégkönyv segítségével megoszthatjuk véleményünket a szerzőkkel, valamint más felhasználókkal.

Sok sikert kívánunk az információs technológiák elsajátításában!

A szerzők

1. fejezet

INFORMÁCIÓS TECHNOLÓGIÁK AZ OKTATÁSBAN

Ebben a fejezetben megismerkedünk:

- a matematika és a természettudományos tárgyak oktatásában alkalmazható szoftvereszközökkel;
- a matematikai programcsomagok működésével, valamint annak alkalmazásával a számítások automatizálása során;
- azzal, hogyan alkalmazzuk a matematikai programcsomagot a grafikonok rajzolása és egyenletrendszerek megoldása során;
- a természettudományok oktatásában alkalmazható elektronikus tankönyvekkel, virtuális laboratóriumokkal, interaktív modellekkel.

1.1. Oktatási szoftverek, matematikai programcsomagok

1. Milyen oktatási szoftvereket ismerünk?
2. Milyen multimédiás kurzusokkal találkoztunk már a különböző tantárgyak tanulmányozása során?
3. Mik azok az elektronikus oktatási eszközök? Milyen lehetőségeket biztosítanak felhasználóik részére?
4. A matematikaórákon milyen témakörök tanulmányozása során alkalmaztunk számítógépes eszközöket? Milyen feladatok megoldása során alkalmazták ezeket az eszközöket?
5. Milyen szabályokat kell betartanunk a képletek beírása során az **Excel 2007**-ben?

Oktatási szoftverek

Amikor a házi feladatokat készítjük, referátumot írunk, dolgozathoz készülünk információforrásokra van szükségünk: tankönyveket, feladatgyűjteményeket, atlaszokat, szótárakat, fényképalbumokat használunk. Feladataink megoldása során különböző elektronikus eszközöket is használhatunk. Ezek lehetnek internetes információforrások, elektronikus enciklopédiák, képgyűjtemények, de általános célú szoftverek, mint a **Számológép**, a **Word** szövegszerkesztő, vagy a **Paint** rajzprogram.

Léteznek olyan szoftvereszközök is, amelyeket oktatási céllal hoztak létre. Ezeket **oktatási szoftvereknek** nevezzük. Ilyen eszközök például a **Таблиця Менделєєва** szoftver, a **Біологія 8–9** elektronikus tankönyv, a **Система лінійних рівнянь** szoftverrendszer, amelyeket a kémia, biológia, illetve matematikaórákon használhattunk (1.1. ábra).

Az oktatási szoftverek nagy előnye a tradicionális oktatási eszközökkel szemben a fejlett vizualizációs eszközökben rejlik: a jelenségeket, objektumokat és folyamatokat azok változása közben szemléltetik, a kísérletek, azok eredményei és a feladatok megoldásai is szemléletesen ábrázolhatók. A szoftverek másik előnyös tulajdonsága a felhasználó és a rendszer közötti gyors visszacsatolás lehetősége. Ezt a tulajdonságot röviden **interaktivitásnak** nevezzük. Az **interaktivitás** a kommunikáció olyan formája, ahol a befogadó azonnal reagálhat a közlő üzenetére. Ennek következtében minden tanuló a saját maga számára megfelelő ütemben végezheti el a feladatait.

1.1. ábra. Elektronikus atlasz használata multimédiás eszközökkel felszerelt földrajzteremben

Az oktatási szoftverek a következőképpen csoportosíthatók:

- ***elektronikus tankönyvek.*** Ezek általában a tankönyveket kiegészítő elektronikus kiadványok, amelyek a hagyományos tananyagot vagy annak kibővített változatát tartalmazzák, jellemzően multimédiás formában. Ilyenek például a **Фізика, 10-11** (1.2. *a* ábra) vagy az **Алгебра, 11 клас.** Néhány magyar nyelven elérhető szoftver: a Realika digitális foglalkozásgyűjtemény és oktatási szoftver (http://realika.educatio.hu/), az iDoctum interaktív tananyag (http://www.interaktivtananyag.hu), vagy a GeoGebra matematika oktatási segédeszköz (letölthető a http://www.szoftverbazis.hu vagy a http://letoltés.com oldalakról).
- Virtuális gyakorlóeszközök – elektronikus feladatgyűjtemények, példatárak, ezen belül:
  - *Virtuális laboratóriumok,* például **Віртуальна хімічна лабораторія. 8-11 кл.,** (1.2. *b* ábra) vagy a **Програмно-методичний комплекс «DG – динамічна геометрія».** Érdemes kipróbálni a Yenka fizikai–kémiai szoftver (http://www.yenka.com/en/HU_Downloads/), a Physion virtuális fizikai laboratóriumot (http://physion.net), vagy az Euklides geometriai szerkesztőprogramot (http://matek.fazekas.hu/euklides/hun/euklides.htm)
  - *Elektronikus gyakorlóprogramok,* például a **«Майстер-клас». Клавіатурний тренажер з української мови,** vagy a hasonló rendeltetésű Gépírás200 (http://www.gepiras2000.hu/)
  - *Digitális feladatgyűjtemények,* például a **Електронний задачник «Фізика. 7-9».**
  - ***Megoldás-ellenőrző szoftverek,*** például a **«Школярик»** interaktív tesztrendszer, amelyek rendeltetése a tanulók tudását ellenőrző tesztek létrehozása és a teszteredmények dokumentálása.

a *b* *c* *d*

1.2. ábra. Pedagógiai szoftvereszközök

- *multimédiás tájékoztató és illusztrációs eszközök:*
  - *elektronikus térképek* – navigációs és keresési eszközökkel ellátott digitalizált kép- térképgyűjtemények, például az ***«Економічна і соціальна географія світу, 10-11 клас»*** (1.2. c ábra);
  - *elektronikus szöveggyűjtemények* – digitalizált szépirodalmi, történelmi, zenei, képzőművészeti művek, például **«Шкільна колекція». Фонохрестоматія для 10-12 класів з англійської мови;**
  - *elektronikus enciklopédiák* – digitális tájékoztató kiadványok, amelyek egy adott tevékenységi területről vagy tudományterületről tartalmaznak szócikkeket, audio- vagy videóanyagokat. Például **Електронна база знань «Людина, суспільство і світ», Електронна енциклопедія «Міфи народів світу»,** Magyar virtuális enciklopédia (http://www.enc.hu/enc.htm), Kis mitológiai enciklopédia (http://www.literatura.hu/lexikon/mitosz/mitolog.htm), Horgász e-enciklopédia (http://www.sejt.hu/);
  - *digitális szótárak* – szótárak digitalizált, keresővel ellátott, esetlegesen hangos változatai, például **Навчальне середовище «1001 слово»** (1.2. *d* ábra).

Léteznek ezenkívül összetett oktatási szoftvereszközök, amelyek különböző típusú eszközöket tartalmaznak, például az **Інтегрований електронний комплекс «Економічна та соціальна географія світу».** Ezeket általában **multimédiás** vagy **interaktív tananyagoknak** nevezik.

Az oktatási szoftvereszközök jellemző részei a következők:

- **menürendszer,** amely bemutatja a tananyag tartalmát, és elérhetővé teszi a szoftver fontosabb funkcióit;
- **navigációs rendszer,** amelynek segítségével a tananyag nagyobb egységei valamint mellékletei között lépkedhetünk;
- **keresőrendszer** a szükséges anyagrész gyors megkeresésére;
- **súgó,** amely segíti a szoftvereszköz használatának elsajátítását.

Ezenkívül az oktatási szoftver részei lehetnek:

- szöveges információs egységek;
- grafikus gyűjtemények;
- audio- és videógyűjtemények;
- gyakorlófeladatok;
- ellenőrző egységek stb.

Az oktatási szoftverek némelyike kliens–szerver architektúrájú. Ez ezt jelenti, hogy a tanári gépen a program szerver-része, a tanulók gépein pedig annak kliens része helyezkedik el.

Az oktatási szoftverek között a kitűzött cél alapján választunk. Ha például földrajzórára készítünk egy rövid beszámolót, akkor elektronikus tankönyvet, enciklopédiát, szöveggyűjteményt vagy térképet használunk. Egy kémiadolgozat előtt a virtuális kémiai laboratórium vagy egy virtuális kémiai feladatgyűjtemény lehet segítségünkre.

Az oktatási szoftverek tartalma nem feltétlenül egyezik meg egy tantárgy ismereteinek alaptanterv szerinti követelményeivel, jellemzően inkább kibővíti azt. Tartalmazhat például kiegészítő elméleti anyagokat, emelt szintű feladatokat, tanulmányi kutatási feladatokat és eszközöket. Ezeknek akkor látjuk igazán hasznát, ha tanulmányi versenyekre készülünk vagy kutatási feladatokat végzünk.

Sokféle tantárgyi oktatási szoftver ingyenesen letölthető és szabadon felhasználható. Egy ilyen webhely az **Острів знань** (http://www.ostriv.in.ua).

Az oktatási szoftvereket, akárcsak az egyéb szoftvertermékeket, használat előtt telepíteni kell a számítógépünkre. A számítógépes programokat szerzői jogi törvény védi. Telepítéskor mindig szem előtt kell tartani az Ukrán Oktatási és Tudományos Minisztérium által jóváhagyott *Számítógépes programok iskolai alkalmazásának szabályairól* szóló rendeletet. Csak legális számítógépes programokat használhatunk, amihez licencszerződéssel kell rendelkeznünk.

A szoftvert az eredeti hordozójáról (például CD-ről) átmásoljuk a számítógép merevlemezére. Ez általában a következő lépésekből áll:

1. Behelyezzük a telepítő CD lemezét a meghajtóba.
2. Megnyitjuk a telepítőlemez gyökérkönyvtárát.
3. Elindítjuk a telepítőprogramot (általában **setup.exe, install.exe** fájlokat).
4. Követjük a telepítőprogram utasításait.

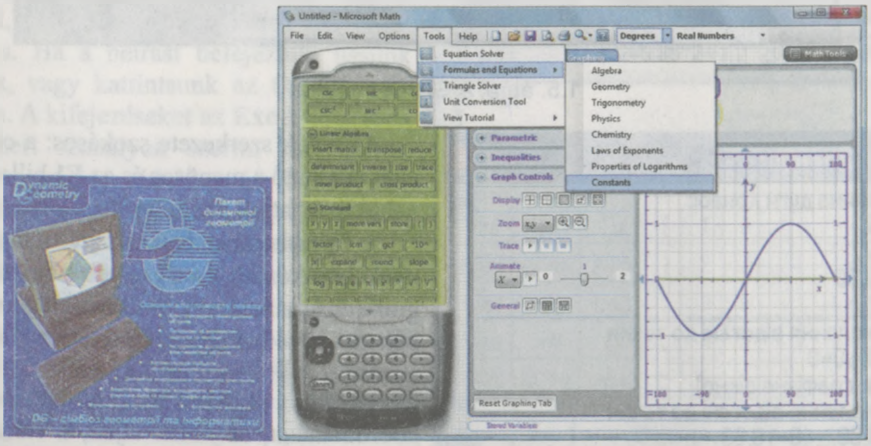
Matematikai programcsomagok

A matematika oktatását segítő szoftvereszközök között kiemelkedő jelentőséggel bírnak a *matematikai programcsomagok*. Ezek a következő lehetőségeket biztosítják a felhasználók számára:

- számkifejezések értékének meghatározása;
- függvénygrafikonok megrajzolása;
- függvénygrafikonok transzformációja;
- függvénygrafikonok metszéspontjának meghatározása az adott intervallumon belül;
- függvények zérushelyeinek és szélsőértékhelyeinek meghatározása;
- egyenletek és egyenletrendszerek gyökeinek közelítő meghatározása;
- egyenlőtlenségek és egyenlőtlenség-rendszerek grafikus megoldása;
- mértani testek felszínének és térfogatának meghatározása stb.

Az ismertebb matematikai programcsomagok a **GRAN,** a **DG,** (1.3. *a* ábra), a **Microsoft Mathematics** (1.3. *b* ábra), a **MathCad, MathLab,** az **EUREKA,** a **Maple, a DERIVE,** az **Advanced Grapher stb.** Ezek némelyike nagy teljesítményű, professzionális felhasználásra készült szoftver, mások kifejezetten középiskolai használatra készült szoftverek.

A matematikai programcsomagok alkalmazása lehetővé teszi, hogy a számítógépek gyorsaságát és pontosságát felhasználjuk a különböző összefüggések vizsgálatakor, a számítógépre hagyva a fárasztó rutinszámításokat, ugyanakkor megtartva a grafikus ábrázolás előnyeit. Ez segítheti a képi gondolkodást és a geometriai intuíciót.



a *b*

1.3. ábra. Matematikai programcsomagok

A GRAN matematikai programcsomag

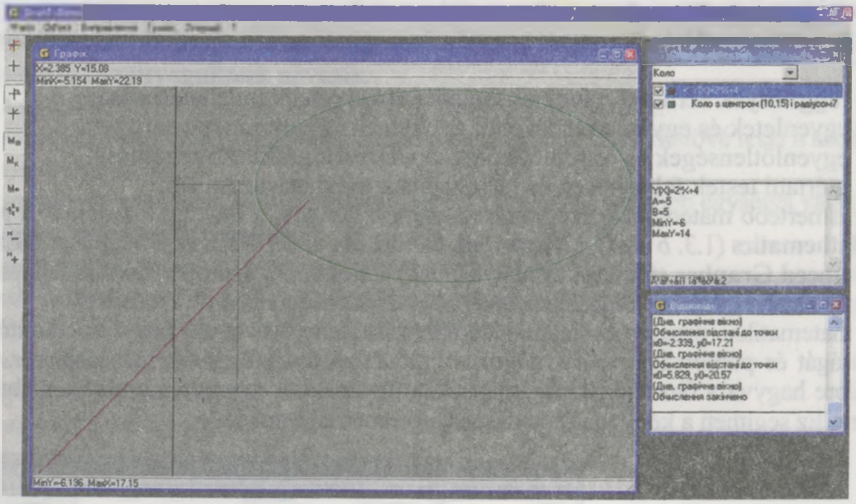
1.4. ábra.
M. I. Zsaldak

Ukrajnában az első pedagógiai programcsomagok egyike volt a matematika oktatását segítő **GRAN**, amelyet 1989-ben Miroszlav Ivanovics Zsaldak akadémikus vezetése alatt fejlesztettek ki.

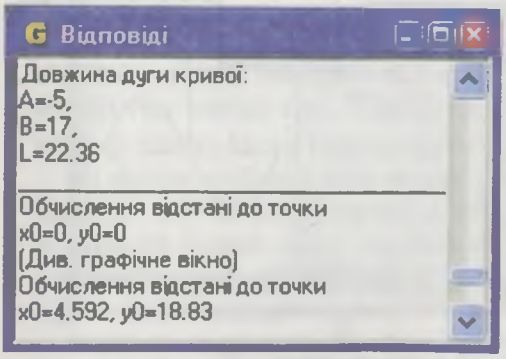
A **GRAN** programcsomag részei a **GRAN1**, a **GRAN-2D** és a **GRAN-3D** a planimetria, a sztereometria, a trigonometria, az algebra és az analízis elemei, a valószínűségszámítás, a matematikai statisztika valamint a középiskolai fizika tanulását hivatott segíteni az általános iskolák 7–11. osztályaiban.

Vizsgáljuk meg a matematikai programcsomagok használatát a **GRAN1** példáján, amelynek segítségével a **GRaphic ANalysis** segítségével ábrázolt függvények grafikus elemzését végezhetjük el.

A programot ***Пуск ⇒ GRAN1 ⇒ GRAN1*** menüpontok, vagy a parancsikon segítségével indíthatjuk el. Ezután megnyílik az 1.5. ábrán látható, három, **Графік, Список об'єктів, Відповіді** részből álló ablak.



1.5. ábra. A **GRAN1** ablaka



1.6. ábra. A **Відповіді** ablak

Az ablak szerkezete szokásos: a címsor alatt található a menüsor és az **F1** billentyűvel aktiválhatjuk a tartalomfüggő súgót.

A **Графік** ablakban találjuk a függvények grafikonjait és az egyéb objektumok grafikus ábrázolásait. Amikor az egeret ebben az ablakban mozgatjuk, azzal együtt mozog a + koordináta-mutató is. Az ablak bal felső részében láthatjuk az aktuális koordinátákat, az ablak alján, a státussorban pedig a koordináták maximális és minimális értékeit.

A **Список об'єктів** ablak két részből áll. A felső legördülő menüben találhatjuk meg az ábrázolható objektumok listáját, ami alatt a már ábrázolt objektumok listája látható. Az aktuális objektumot a ✓ jel mutatja. Ez alatt találjuk az aktuális objektum paramétereit: a függvény, az intervallum, amelyen a függvényt vizsgáljuk, a függvény minimális és maximális értéke a vizsgált intervallumon és egyéb paraméterek.

A **Відповіді** ablakban (1.6. ábra) azoknak a műveleteknek az eredményeit láthatjuk, amelyeket az **Операції** menüből választhatunk ki. Itt olyan műveletek választhatók, mint: a ponttól való távolság meghatározása, az ív és töröttvonal hosszának kiszámítása. Az ablak tartalmát az ***Операції ⇒ Відповіді ⇒ Очистити*** menüvel törölhetjük.

A program ablakának bal oldalán találhatjuk az eszköztárat. Az eszközök rendeltetését az 1.1. táblázatban találhatjuk:

1.1. táblázat. **A GRAN1 eszközei**

Jelölés	*Parancs*	*Jelölés*	*Parancs*	*Jelölés*	*Parancs*
	Grafikon rajzolása		Megnyitás	$M_к$	Felhasználói méretarány
	Törlés		Mentés	M_a	Automatikus méretarány
	Derékszögű koordináta		Objektum létrehozása	$M_•$	Előző méretarány
	Poláris koordináta	M_- M_+	Méretarány csökkentése Méretarány növelése		Eredeti méretarány

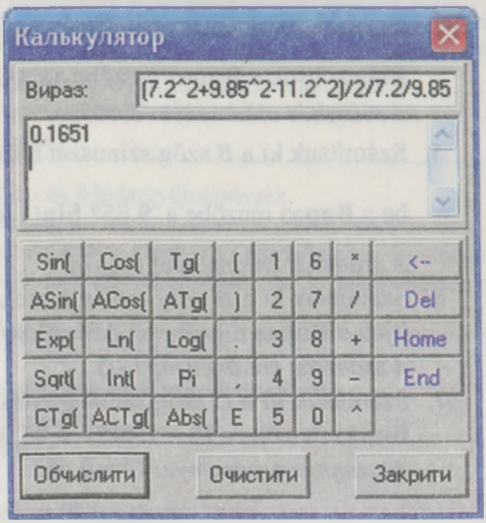
Számítások elvégzése GRAN1-ben

Gyakorlatilag minden matematikai programcsomagot használhatunk számítások elvégzésére is. A **GRAN1** e tekintetben kibővített lehetőségekkel bír.

A számításokat a **Калькулятор** ablakban végezzük, amelyet az ***Операції ⇒ Калькулятор*** menüvel aktiválunk. A kiszámítandó kifejezést beírhatjuk billentyűkkel, de a megfelelő gombok lenyomásával is. Ha a beírást befejeztük, üssünk **Enter**-t, vagy kattintsunk az **Обчислити** gombra. A kifejezéseket az **Excel 2007**-ben megismert szabályok szerint kell beírni. A **Калькулятор** ablakban (1.7. ábra) láthatjuk az összes függvényt, amelyet a kifejezések beírása során alkalmazhatunk.

A kifejezések szóközt nem tartalmazhatnak. A számok törtrészét az egész résztől tizedespont választja el. A műveletek prioritása megegyezik a matematikában megszokottal. A beírt képlet szerkesztését is a megszokott módon végezhetjük.

Az 1.2. táblázat a matematikai műveletek és függvények jelölését tartalmazza:



1.7. ábra. A **Калькулятор** ablak

1.2. táblázat. **A matematikai műveletek és függvények jelölése**

Jelölés	*Függvény*	*Jelölés*	*Függvény*	*Jelölés*	*Függvény*	*Jelölés*	*Matematikai művelet*
Sin	Szinusz	**Cos**	koszinusz	**Tg**	tangens	*	Szorzás
ASin	Arkusz szinusz	**ACos**	Arkusz koszinusz	**ATg**	Arkusz tangens	/	Osztás
Exp	e alapú hatvány	**Ln**	Természetes logaritmus	**Log**	Tetszőleges alapú logaritmus	+	Összeadás
Sqrt	Négyzet-gyök	**Int**	Egész rész	**Pi**	π	-	Kivonás
Ctg	Kotangens	**ACtg**	Arkusz kotangens	**Abs**	Modulus	^	Hatvány

Példaképpen nézzük meg, hogy használhatjuk a **Калькулятор**-t a következő feladat megoldása során.

Feladat. A háromszög oldalai AB = 7,2; BC = 11,2; AC = 9,85. Határozzuk meg a háromszög szögeit radiánban.

A feladat megoldása során a következő műveletet kell elvégezni:

1. Indítsuk el a **GRAN1**-et.
2. Indítsuk el a **Калькулятор**-t (***Операції ⇒Калькулятор***)!
3. Számítsuk ki az A szög koszinuszát a $\cos A = \frac{AB^2 + AC^2 - BC^2}{2 \cdot AB \cdot AC}$ képlet szerint.

   Ehhez írjuk be a **Вираз** mezőbe a **(7.2^2+9.85^2-11.2^2)/(2*7.2*9.85)** kifejezést (1.7. ábra)

   A számítás eredménye: 0.1651.
4. Számítsuk ki az A szög radiánmértékét az A = **arccos (cos (A))** képlet szerint. Ehhez írjuk a **Вираз** mezőbe az **ACos(0.1651)** kifejezést.

   A számítás eredménye: 1,405.
5. Számítsuk ki a B szög szinuszát a $\sin B = \frac{AC \cdot \sin A}{BC}$ képlet szerint! Ehhez írjuk be a **Вираз** mezőbe a **9.85* Sin(1.405)/11.2.** kifejezést.

   A számítás eredménye: 0,8674
6. Számítsuk ki a B szög radiánmértékét a B = **arcsin (sin(B))** képlet szerint! Ezért írjuk a **Вираз** mezőbe az **ASin(0.8674)** kifejezést.

   A számítás eredménye: 1,05.
7. Számítsuk ki a C szög radiánmértékét a $C = \pi - A - B$ képlet szerint. Írjuk be a **Вираз** mezőbe a **Pi – 1.405 – 1.05.** kifejezést!

   A számítás eredménye: C = 0,687.

Ellenőrző kérdések

1°. Mik azok az oktatási szoftverek?
2°. Nevezzünk meg oktatási szoftvereket!
3•. Az oktatási szoftverek milyen csoportjait ismerjük?
4•. Mik egy oktatási szoftver leggyakoribb elemei?
5°. Milyen lehetőségeket biztosítanak az oktatási szoftverek?
6•. Hogy zajlik egy szoftver telepítése?
7°. Milyen szoftverek segítik a matematika oktatását?
8. Mi egy matematikai programcsomag rendeltetése?
9°. Melyek az ismertebb matematikai programcsomagok?
10. Miért hasznos a matematikaoktatás szoftveres segítése?
11•. Ismertessük a **GRAN1** felhasználói felületét!
12•. Milyen szabályok érvényesek a kifejezések beírásakor **GRAN1**-ben?
13*. A matematikai programcsomagok funkcióinak melyik része van meg az **Excel 2007** táblázatkezelőben?
14*. Hasonlítsuk össze a kifejezések beírásának szabályait **GRAN1**-ben és **Excel 2007**-ben!

Végezzük el a feladatokat!

1•. Állapítsuk meg, hogy a leírások közül melyik milyen oktatási szoftverre érvényes:

	Oktatási szoftverek		*Leírás*
1	Elektronikus tankönyvek	A	Digitalizált szépirodalmi, történelmi, zenei, képzőművészeti művek
2	Elektronikus gyakorlók	B	Szótárak digitalizált, keresővel ellátott, esetlegesen hangos változatai
3	Elektronikus tudásellenőrzők	C	Digitális tájékoztató kiadványok, amelyek egy adott tevékenységi területről, vagy tudományterületről tartalmaznak szócikkeket, audio- vagy videoanyagokat
4	Digitális térképek	D	Általában a tankönyveket kiegészítő elektronikus kiadványok, amelyek a hagyományos tananyagot vagy annak kibővített változatát tartalmazzák, jellemzően multimédiás formában
5	Elektronikus szótárak	E	Navigációs és keresési eszközökkel ellátott digitalizált kép- és térképgyűjtemények
6	Digitális enciklopédiák	F	Elektronikus példa- és feladatgyűjtemények
7	Elektronikus olvasókönyvek	G	Interaktív tesztrendszer, amelyek rendeltetése a tanulók tudását ellenőrző tesztek létrehozása és a teszteredmények dokumentálása

2*. Válasszunk oktatási szoftvert a következő feladatok végrehajtásához:
a) kémiai laboratóriumi munka előkészítése;
b) referátum elkészítése matematikából;
c) gazdasági feladatok megoldása;
d) tudományos kutatómunka elkészítése földrajzból;
e) előadás készítése a XX. század tudományos felfedezéseiről!

3.* Keressük meg, milyen matematikai oktatási szoftvereket találhatunk az **Острів знань** (http://www.ostriv.in.ua) webhely ***Сервіси*** ⇒ ***Файловий архів*** ⇒ ***Електронні засоби навчання*** részében! Állapítsuk meg, az oktatási szoftverek milyen típusába sorolhatók ezek!

4. • Indítsuk el a számítógépünkre telepített matematika tankönyvet! Vizsgáljuk meg a menürendszerét! Olvassunk bele az első fejezet anyagába! Hozzunk fel példákat ezek alkalmazhatóságára!

5.• Nyissuk meg a **Математика для школи** (http://www.formula.co.ua) webhelyet! Ismerkedjünk meg az **Алгебра** és a **Геометрія** rovatok tartalmával! Keressük ki a *trigonometria* szó értelmezését! Alakítsuk át a 30°-os szögmértéket radiánná! Számítsuk ki a sin30°+cos30° értékét a beépített számológép segítségével! Mire alkalmazható még ez a webhely?

6.* Töltsük le az **Острів знань** (http://www.ostriv.in.ua) webhelyről a «**Динамічна геометрія**» programot! Telepítsük fel ezt a számítógépünkre! Ismerkedjünk meg működésével! Milyen típusú oktatási szoftver lehet ez? Nevezzük meg a program 3 lehetséges alkalmazását! Készítsünk prezentációt a szoftver működéséről!

7.• Számítsuk ki a **GRAN1** segítségével a következő kifejezések értékét:

a) $Y = \cos 2x$, ha $x = 0; 1; 1.57$

b) $Y = x2 - 4|x|3$, ha $x = -3; 3$

c) $Y = 1/(x2-3)$, ha $x = -2; 0; 15$!

8.• A **GRAN1** segítségével határozzuk meg annak a derékszögű háromszögnek kerületét és területét, melynek egyik befogója 7 cm, az ezzel szemben fekvő szöge pedig 60°!

9.• A **GRAN1** segítségével határozzuk meg annak a háromszögnek a szögeit, amelynek oldalai $a = 7$, $b = 5$ és $c = 8$!

1. számú gyakorlati munka **Számítások automatizálása**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

1. •A **GRAN1** segítségével számítsuk ki az alábbi kifejezések értékét:

a) $Y = 2\sin(x)$, ha $x = 0$;

b) $Y = |\,|x-1| - |x-2|\,|$, ha $x = -2; 0; 15$;

c) $Y = \cos 2x$, ha $x = 0$;

d) $Y = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, ha $a = 2, b = 8, c = 3$;

e) $Y = 2x$, ha $x = 0$;

f) $Y = \pi r^2$, ha $r = -3,9$;

g) $Y = (x-1)^2 / (x-2)^3$, ha $x = -3$.

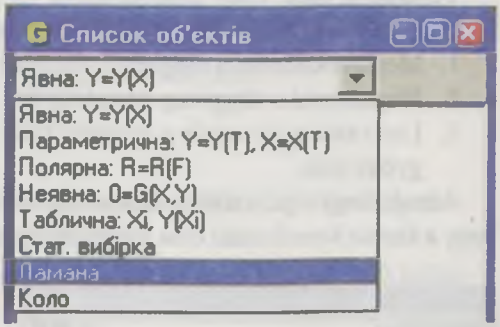
2. •A Hold sugara 1740 km. A **GRAN1** segítségével határozzuk meg a Hold felszínét és térfogatát! A számítások menetét és eredményét írjuk be a füzetbe!

1.2. A GRAN1 alkalmazása függvénygrafikonok ábrázolásakor

1. Milyen szabályok érvényesek a képletek beírásakor a **GRAN1**-ben?
2. Mi az egyes ablakoknak rendeltetése a **GRAN1**-ben?
3. Mit értünk a függvény a grafikonja alatt? Az egyenlet grafikonja alatt?
4. Hogy végezzük el egy függvény elemzését?
5. Ismertessük az egyenletek grafikus megoldásának menetét!

Grafikonok ábrázolása GRAN1 segítségével

A **GRAN1** segítségével 8 különböző típusú függvény grafikonját rajzolhatjuk meg (1.8. ábra). A grafikonok ablaka egyidejűleg maximálisan 5 grafikont tartalmazhat, amelyek színükben is eltérnek egymástól. Az objektumok ablakában láthatjuk, hogy melyik objektumhoz milyen szín tartozik.



1.8. ábra. Az összefüggések típusai

A függvénygrafikonok szerkesztésének főbb lépései a következők:

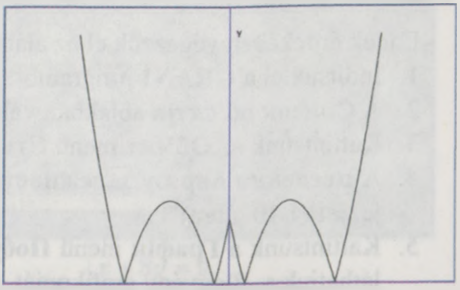
1. A **Список об'єктів** ablakban válasszuk ki az ábrázolandó függvény típusát!
2. Kattintsunk a **Об'єкт** menü **Створити** pontjára!
3. A **Введення виразу залежності** párbeszédablakba írjuk be a megfelelő kifejezést!
4. Kattintsunk a **Графік** menü **Побудувати графік** pontjára!

Tekintsük át a függvényábrázolás menetét a következő példán.

1. feladat. Ábrázoljuk az $y = |x^2 - 8|x| + 7|$ függvény grafikonját!

A fent vázolt algoritmus szerint a következőképpen járunk el:

1. Indítsuk el a **GRAN1** programot!
2. A **Список об'єктів** ablakban válasszuk ki a **Явна: Y=Y(X)** lehetőséget!
3. Kattintsunk az **Об'єкт** menü **Створити** pontjára!
4. A **Введення виразу залежності** párbeszédablakba írjuk be az **ABS(X^2-8*ABS(X)+7)** kifejezést!
5. Kattintsunk a **Графік** menü **Побудувати графік** pontjára!



1.9. ábra. A $y = |x^2 - 8|x| + 7|$ függvény grafikonja

A fenti eljárás során kapott grafikont az 1.9. ábrán láthatjuk.

A függvény grafikonját elemezve a következő megállapításokat tehetjük:

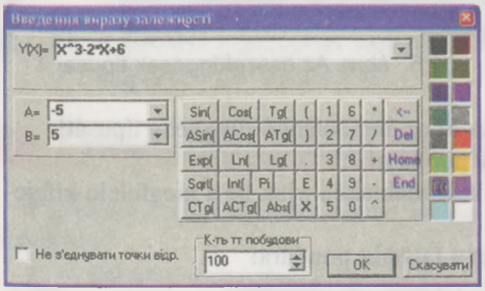
- *zérushelyek.* A zérushelyek meghatározása céljából helyezzük az egérkurzort a grafikon és az *Ox* tengely metszéspontjába, és olvassuk le a koordinátákat. Ezeket a **Графік** ablak felső részében láthatjuk;
- *szélsőértékhelyek és szélsőértékek.* Ahhoz, hogy ezeket leolvashassuk, helyezzük az egérkurzort a függvény legfelső, illetve legalsó pontjába. A függvények maximális és minimális értékét a **Список об'єктів** ablakban is láthatjuk;
- *a növekedés és csökkenés intervallumai.* A szélsőértékhelyek meghatározása után felírhatjuk a növekedés, illetve csökkenés intervallumait.

Egyenletek és egyenletrendszerek megoldása GRAN1-ben

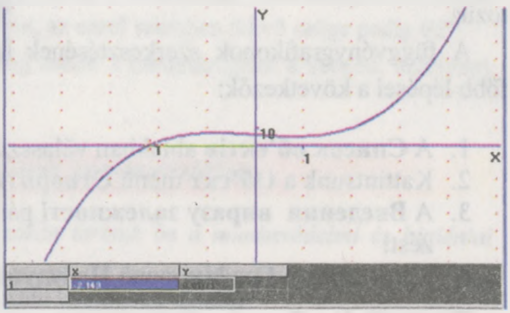
Vizsgáljuk meg, hogy használhatjuk fel a **GRAN1**-et egyváltozós egyenletek grafikus megoldására. A feladat megoldása három lépésből áll:

1. Megszerkeszteni a megfelelő függvény grafikonját.
2. Megkeresni a függvény zérushelyeit.
3. Leolvasni a zérushely x koordinátáját. Ez természetesen csak az egyenlet közelítő gyöke lesz.

Annak, hogy ilyen módon csak az egyenlet közelítő gyöke határozható meg, az az oka, hogy a kurzor koordinátáit csak bizonyos lépésenként lehet változtatni.



1.10. ábra. A **Введення виразу залежності** párbeszédablak



1.11. ábra. Egyenletek közelítő gyökének meghatározása

Lássunk egy példát az egyváltozós egyenlet grafikus megoldására.

2. feladat. Oldjuk meg grafikusan az $x^3-2x+6 = 0$ egyenletet!

Ennek érdekében végezzük el az alábbiakat:

1. Indítsuk el a **GRAN1** programot!
2. A **Список об'єктів** ablakban válasszuk ki a **Явна: Y=Y(X)** lehetőséget!
3. Kattintsunk az **Об'єкт** menü **Створити** pontjára!
4. A **Введення виразу залежності** párbeszédablakba írjuk be a **X^3-2*X+6** kifejezést (1.10. ábra)!
5. Kattintsunk a **Графік** menü **Побудувати графік** pontjára! A **Графік** ablakban láthatjuk a függvény grafikonját.
6. Kattintsunk a ***Графік*** ⇒ ***Список точок на графіку*** ⇒ ***Запис*** menüpontra!

7. Vigyük az egérkurzort a grafikon és az *Ox* tengely metszéspontja fölé (1.11. ábra)!
8. Olvassuk le a gyök közelítő érékét a **Графік** ablak alsó részéből!

Felelet: $x = -2{,}149$.

Amennyiben a függvény több pontban is metszi az *Ox* tengelyt, az egyenletnek több megoldása van, amelyek mindegyikét láthatjuk a Графік ablakban a koordináta-rendszer alatt.

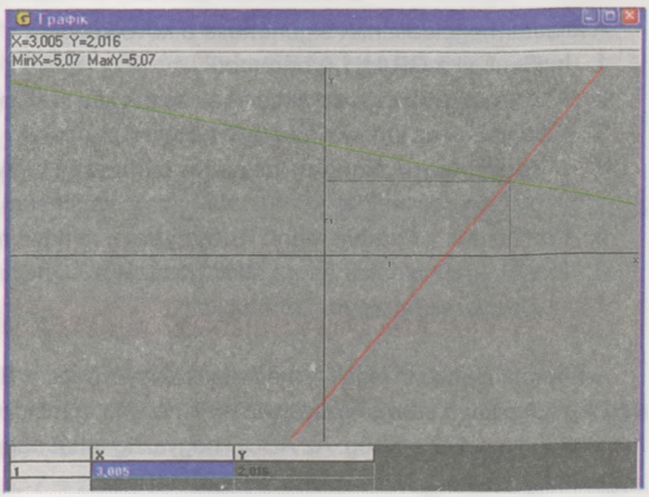
Egyenletrendszerek esetében a megoldás algoritmusa a következő:

1. Állítsuk elő mindkét egyenlet grafikonját!
2. Jelöljük meg a koordinátasíkon a grafikonok metszéspontjait!
3. Olvassuk le a pontok koordinátáit!

3. feladat. Oldjuk meg Gran1-ben a $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$ egyenletrendszert!

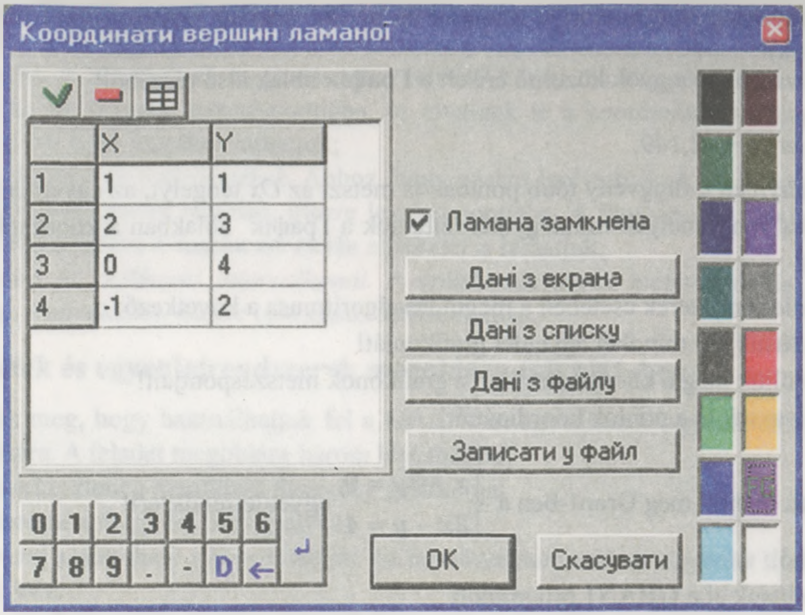
E célból:

1. Indítsuk el a **GRAN1** programot!
2. A **Список об'єктів** ablakban válasszuk ki a **Неявна: 0=G(Y,X)** lehetőséget!
3. Kattintsunk az **Об'єкт** menü **Створити** pontjára!
4. A **Введення виразу залежності** párbeszédablakba írjuk be az **X+3*Y-9** kifejezést!
5. Kattintsunk a **Графік** menü **Побудувати графік** pontjára!
6. A **Список об'єктів** ablakban válasszuk ki a **Неявна: 0=G(Y,X)** lehetőséget!
7. Kattintsunk az **Об'єкт** menü **Створити** pontjára!
8. A **Введення виразу залежності** párbeszédablakba írjuk be a **2*X–Y–4** kifejezést!
9. Kattintsunk a **Графік** menü **Побудувати** графік pontjára!
10. Kattintsunk a ***Графік ⇒ Список точок на графіку ⇒ Запис*** menüpontra!
11. Jelöljük ki az egérrel a grafikonok metszéspontját (1.12. ábra)!
12. Olvassuk le a metszéspont koordinátáit!



1.12. ábra. Az $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$ egyenlet grafikus megoldása

Felelet: (3,005; 2,016).



1.13. ábra. A **Координати вершин ламаної** párbeszédablak

Egyszerű síkmértani feladatok megoldása GRAN1-ben

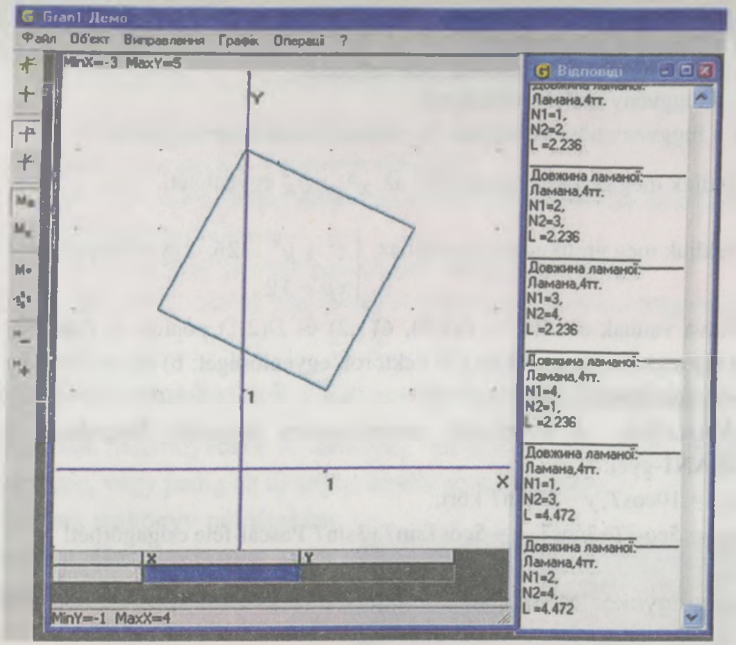
Vizsgáljuk meg, hogyan használhatjuk a **GRAN1**-et egyszerű síkmértani feladatok megoldása során.

4. feladat. Határozzuk meg az *ABCD* négyszög típusát, ha a csúcsok koordinátái *A*(1; 1), *B*(2; 3), *C*(0; 4) és *D*(–1; 2).

A feladat megoldásának algoritmusa a következő:

1. Indítsuk el a **GRAN1** programot!
2. A **Список об'єктів** ablakban válasszuk ki a **Ламана** lehetőséget!
3. Kattintsunk az **Об'єкт** menü **Створити** pontjára!
4. A **Координати вершин ламаної** párbeszédablakban (1.13. ábra) adjuk meg a négyszög csúcsainak koordinátáit, majd kattintsunk az **OK** gombra!
5. Kattintsunk a **Графік** menü **Побудувати графік** pontjára!
6. Mérjük le az oldalak és az átlók hosszát az ***Операції*** ⇒ ***Операції з ламаними*** ⇒ ***Довжина ламаної*** paranccsal!

A **Графік** ablakban látjuk a megrajzolt négyszöget, a **Відповіді** ablakban (1.14. ábra) pedig a következő értékek szerepelnek: a négyszög minden oldala 2,236 egység, az átlók hossza pedig 4,472 egység. Az *ABCD* négyszög tehát négyzet.



1.14. ábra. Síkmértani feladatok megoldása **GRAN1**-gyel

Ellenőrző kérdések

1. Milyen függvénytípusokkal dolgozhatunk a **GRAN1**-ben?
2. Egyidejüleg hány grafikon ábrázolására van lehetőség a **GRAN1**-ben?
3. Hogy törölhető a **Графік** ablak tartalma?
4. Hogyan ábrázolható egy függvény grafikonja **GRAN1**-ben?
5. Hogyan ábrázolható egy egyenlet grafikonja **GRAN1**-ben?
6. A függvényelemzés mely lépései végezhetők el a **GRAN1** segítségével?
7. Hogyan határozhatjuk meg egy függvény zérushelyeit? Szélsőértékhelyeit?
8. Hogy oldhatunk meg egyenleteket a **GRAN1**-gyel?
9. Hogy oldhatunk meg egyenletrendszereket a **GRAN1**-gyel?
10. Mi az oka annak, hogy az egyenletek grafikus és analitikus megoldásai különböznek egymástól?

Végezzük el a feladatokat!

1. • Ábrázoljuk az $y = x^4 - 5 \cdot x^2 + 3$ függvény grafikonját **GRAN1**-gyel!
2. • Ábrázoljuk az $Y = |x|$, $Y = |x + 1|$, $Y = |x| + 1$ függvények grafikonjait! Hogy változik a függvénygrafikon elhelyezkedése?
3. Ábrázoljuk az $Y = \sin x$, $Y = 2\sin x$, $Y = \sin 2x$, $Y = \sin(x+2)$ függvények grafikonjait! Hogy változik a függvénygrafikon elhelyezkedése?
4. • Ábrázoljuk az $y = |x^3| + x + 7$ függvény grafikonját! Állapítsuk meg:

a) a függvény paritását (páros-e vagy páratlan);
b) a függvény zérushelyeit;
c) a függvény előjeltartásának intervallumait;
d) a függvény szélsőértékhelyeit;
e) a függvény növekedésének és csökkenésének intervallumait!

5. • Oldjuk meg grafikus módszerrel az $x^3 = \sqrt{x}$ egyenletet!

6. * Oldjuk meg grafikus módszerrel az $\begin{cases} x^2 + y^2 = 25; \\ xy = 12 \end{cases}$ egyenletrendszert!

7. *Adva vannak az $A(0;1)$, $B(1;0)$, $C(1;2)$ és $D(2;1)$ pontok. A **GRAN1** segítségével: a) bizonyítsuk be az AB és CD vektorok egyenlőségét; b) bizonyítsuk be az AC és BD vektorok egyenlőségét!

8. *Ábrázoljuk a következő, paraméteresen megadott függvények grafikonjait a **GRAN1**-gyel:

a) az $y = 10\cos T$, $y = 10\sin T$ kört;
b) az $y = 5\cos^2 T + 3\cos T$, $y = 5\cos T \sin T + 3\sin T$ Pascal-féle csigagörbét!

2. számú gyakorlati munka — Függvények grafikonjainak szerkesztése és elemzése

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

1. •Ábrázoljuk a következő függvények grafikonjait:

a) $y = 2\sin x$; c) $y = 2x$; e) $y = ||x - 1| - |x - 2||$;

b) $y = \cos 2x$; d) $y = \dfrac{x-1}{x-2}$; f) $y = (1 + x)^x$.

2. Állapítsuk meg az előző feladatban megrajzolt függvényekről:
a) paritását (páros vagy páratlan függvényről van-e szó);
b) a függvény zérushelyeit;
c) a függvény előjeltartásának intervallumait;
d) a függvény szélsőértékhelyeit;
e) a függvények szélsőértékeit;
f) a függvény növekedésének és csökkenésének intervallumait!

3. számú gyakorlati munka — Egyenletek és egyenletrendszerek grafikus megoldása

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. A **GRAN1** segítségével határozzuk meg az alábbi egyenletek közelítő gyökeit grafikusan :

a) $y = x^2 - 4|x| + 3$; b) $y = \dfrac{1}{x^2 - 3}$; c) $y = \sqrt{||x| - 2|}$.

2. * Határozzuk meg az alábbi egyenletrendszerek közelítő gyökeit:

a) $\begin{cases} 3x - 7y = 9, \\ 5x + 13y = 7; \end{cases}$ b) $\begin{cases} 3x^2 - 4y^3 = 12, \\ 2x^3 - 5y^2 = 8; \end{cases}$ c) $\begin{cases} x + y = 1, \\ |xy| = \dfrac{1}{2}; \end{cases}$ d) $\begin{cases} x + y + 1 = 0, \\ \sin(x + y) = 0. \end{cases}$

1.3. A fizika, kémia és biológia tanítását segítő oktatási szoftverek

1. Mik azok az oktatási szoftverek?
2. Jellemezzük az oktatási szoftverek típusait!
3. Milyen oktatási szoftvereket alkalmaznak a matematikai és természettudományos tárgyak oktatásában?
4. Milyen lehetőségeket biztosít egy modern oktatási szoftver?
5. Milyen összetevői vannak egy oktatási szoftvernek? Mire szolgálnak?

Az elektronikus tankönyvek alkalmazásának sajátosságai

Az elektronikus tankönyveket a tananyag ismétlése, rögzítése, bővítése során célszerű alkalmazni, vagy pedig az új anyag önálló elsajátításakor.

Ez elektronikus tankönyv példájaként vizsgáljuk meg a **Педагогічний програмний засіб «Фізика»** kiadványt.

1.15. ábra. A **Педагогічний програмний засіб «Фізика»** főmenüje

A program főmenüje (1.15. ábra) az egyes fejezetek neveit tartalmazza. A nevek egyben hiperhivatkozásként is szolgálnak, melyek segítségével a fejezet elejére ugorhatunk.

A szoftver fizikai tematikájú szöveges információt tartalmaz. A szöveg minden oldala illusztrált. A szövegben található hivatkozásokra kattintva képi és videoanyagokat tekinthetünk meg, amelyek kiegészítik a tanult elméleti anyagot.

A fejezetekhez tartozó tesztkérdések megválaszolásával ellenőrizhetjük a tananyag elsajátításának mértékét. A tesztet a **Самоперевірка** gombra kattintva indíthatjuk el. A kérdéseket követő válaszok közül pontosan egy lesz helyes. Ha megválaszoltunk minden kérdést, a **Перевірити тест** gombra kattintva megkapjuk annak értékelését. Ha a **Правильні відповіді** gombra kattintunk, a program megjelöli a helyes válaszokat. A tesztet a **Повторити** gombra kattintva ismételhetjük meg. Ilyenkor természetesen mindenfajta jelölés törlődik a válaszok mellől. Amennyiben a kérdések megválaszolása során nehézségek merülnek fel, az **Інформаційна частина** gombra kattintva visszaléphetünk a kapcsolódó elméleti részhez.

A szoftver ezenkívül feladatokat is tartalmaz, melyeket a **Задачі** gombra kattintva nyithatunk meg. A feladatok feltételei illusztrációkat is tartalmaznak. A feladat megoldása után a **Перевірити** перевірити gombra kattintva ellenőrizhetjük a megoldás helyességét. Amennyiben a megoldás során számításokat kell végeznünk, használhatjuk a beépített számológépet. Ezt a **Калькулятор** gombra kattintva

indíthatjuk el. A számításokhoz szükséges adatokhoz a beépített tájékoztatóból (**Довідка**) juthatunk.

A tájékoztatót a **Допомога** gombbal vagy az **F1** billentyű leütésével aktiválhatjuk. A tájékoztató az adatokon kívül a különböző fizikai mennyiségek kiszámításához szükséges összefüggéseket, a fizikai mennyiségek mértékegységeit és táblázati értékeit is tartalmazza. De találhatunk itt fogalomtárat, valamint az ismert fizikusok életrajzi adatait is.

A fejezet áttanulmányozása után virtuális laboratóriumi munkát is végrehajthatunk. A laboratóriumi munkát a **Експеримент** gombra kattintva indíthatjuk el. A szükséges eszközöket és a kísérlet menetét interaktív számítógépes animáció szemlélteti. A kísérlet szükség esetén az objektumokat jellemző fizikai mennyiségek megváltoztatott értékei mellett is elvégezhető.

1.16. ábra. Képernyőképek a **Педагогічний програмний засіб «Фізика»** használatáról

Az 1.16. ábrán néhány képernyőképet láthatunk a vizsgált elektronikus tankönyv használatáról. A képek elméleti ismertető oldalt, feladatmegoldást és laboratóriumi munkát illusztrálnak.

A szoftver működésének jobb megismerését segíti a működését bemutató súgó **(Довідка)** is.

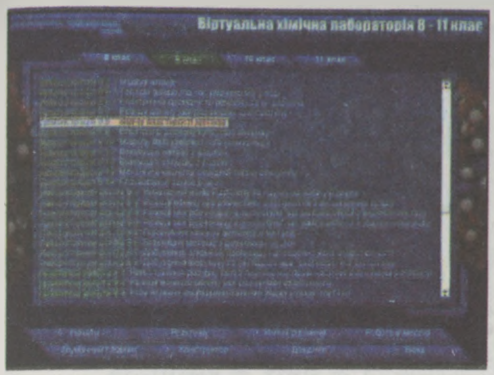
Virtuális gyakorlóeszközök alkalmazása

Vizsgáljunk most meg egy másik oktatási szoftvert, a **Віртуальна хімічна** лабораторія (Virtuális kémiai laboratórium)-ot. A program főmenüjét az 1.17. ábra szemlélteti. Osztályfokonkénti csoportosításban láthatjuk a megtekinthető bemutatókat és a kipróbálásra ajánlott kísérleteket valamint laboratóriumi munkákat.

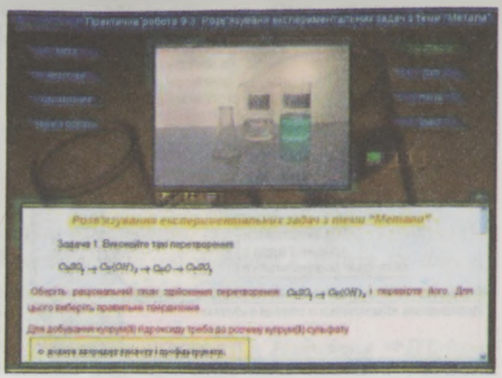
A főmenüből átléphetünk a tájékoztatóba **(Довідник)** és az értelmező szótárba **(Тлумачний словник)**, a vegyi anyagok listájába, de megismerkedhetünk a reakcióegyenletek felírásának szabályaival is. Ehhez csak a megfelelő gombra kell kattintanunk.

Ha kiválasztottunk egy fejezetet, kattintsunk a **Почати** gombra.

Ha a **Демонстрації** gombra kattintunk, megnyílik egy ablak, amelyben videofelvételen megtekinthetjük a kísérletet, és meghallgathatjuk a hozzá fűzött magyarázatot.



1.17. ábra. A **Віртуальна хімічна лабораторія** főmenüje



1.18. ábra. Egy gyakorlati munka ablaka

Ha kiválasztottuk a laboratóriumi kísérletet (**Лабораторний дослід**), vagy gyakorlati munkát (**Практична робота**), az megnyílik az ablakban (1.18. ábra). Ha ebben az ablakban a megfelelő gombra kattintunk, megismerkedhetünk a munka céljával, a szükséges eszközökkel, a betartandó munkavédelmi szabályokkal és a munka menetével.

A munkát videofelvételen is megtekinthetjük. A videofelvételt az ablak felső részében láthatjuk. A képméretet a videoablak tetszőleges pontjára kattintva növelhetjük. A lejátszást a gombokkal megállíthatjuk, folytathatjuk, újrajátszhatjuk.

Ha a megtekintés után folytatni kívánjuk a munkát, kattintsunk a **Далі** gombra, ha pedig másik munkát szeretnénk választani, akkor a **Назад** gombra.

Az ablak alsó részében olvashatjuk a munka menetét. Ha ezzel megismerkedtünk, ajánlott válaszolni a munka menetére vonatkozó kérdésekre. Jó válasz esetén a teszt szövege zöldre változik, rossz válasznál pedig pirosra. A reakcióegyenletek összeállítása (**Складіть рівняння реакції**) a füzetben történik. A felírt egyenlet ellenőrzését a **Перевірити** gombra kattintva kezdeményezhetjük. A következtetések részben ki kell választani azokat az állításokat, amelyek helyességéről a munka elvégzése során győződhettünk meg.

Vannak olyan gyakorlati munkák, amelyekhez teszt is tartozik. Ezt a tesztet a **Тестування** gombra kattintva indíthatjuk el. A tesztek végrehajtása során ki kell választani a helyes válaszokat, azaz össze kell párosítani a kérdéseket a helyes válaszokkal, majd a **Перевірити** gombra kattintva ellenőrizzük válaszaink helyességét.

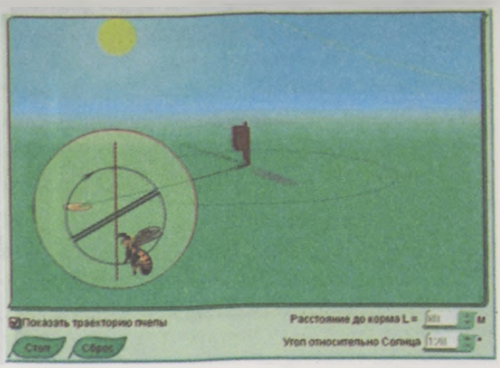
A főmenübe a **Зміст** gombra kattintva léphetünk vissza.

Az interaktív modellek alkalmazásának sajátosságai

Az ***interaktív modellek*** az órákon legnagyobb hatásfokkal felhasználható informatikai eszközökhöz tartoznak. Segítségükkel különféle jelenségek és folyamatok szimulálhatók, megfigyelhetők az egyes objektumok vagy azok részei, illetve ezek működése. Ezzel sok esetben kiegészíthetjük a valóságos objektumokon végzett kísérleteket.

Vizsgáljuk meg az interaktív modellek alkalmazását a **Відкрита Біологія** példáján. A program demo verziója letölthető a http://www.physicon.ru/demo/OpenBiology 2.5Demo.exe webhelyről. A program több biológiai témájú modellt tartalmaz. A klónozás (**Клонування**) segítségével (1.19. ábra) a klónozás folyamatát tanulmányozhatjuk, a méh tánca (**Танок бджоли**) segítségével pedig a méhek etológiai sajátosságait (1.20. ábra).

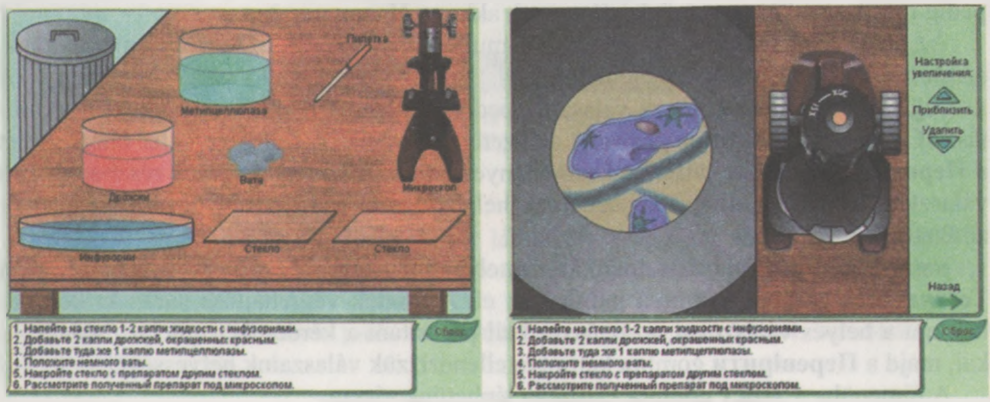
1.19. ábra. A klónozás tanulmányozása



1.20. ábra. A **méh tánca** interaktív modell

A modelleket a következő üzemmódokban használhatjuk: a) folyamatos megtekintés; b) lépésenkénti megtekintés; c) szünetek beszúrása e folyamat egyes stádiumai után.

A **Відкрита Біологія** nagyszámú interaktív modellt tartalmaz, amelyeket az órákon jól használhatunk virtuális kísérletek elvégzésére. A **Живлення інфузорії** (az ázalékállatok táplálkozása) modellen (1.21. ábra) virtuális preparátum segítségével mikroszkóp alatt vizsgálhatjuk a legegyszerűbb élő szervezetekben lezajló folyamatokat. Lehetőségünk van a kísérlet feltételeinek megváltoztatására és az eredmények felülvizsgálatára is.



1.21. ábra. A **Живлення інфузорії** (az ázalékállatok táplálkozása) interaktív modell

Ellenőrző kérdések

1°. Hozzunk fel példákat a természettudományos tárgyak és a matematika tanulását segítő oktatási szoftverekre!

2°. Mi az elektronikus tankönyv, és mire használjuk ezeket?

3°. Milyen eszközök tartoznak az interaktív gyakorlók közé? Mi ezek rendeltetése és alkalmazásának módja?

4°. Milyen eszközök tartoznak az interaktív modellek közé? Mi ezek rendeltetése és alkalmazásának módja?

5°. Milyen új lehetőségekhez juthat egy tanuló, ha a fizika, biológia, kémia, földrajz és csillagászat tanulása során oktatási szoftvereket is használ?

6°. Keressünk az interneten Ukrajna Oktatási Minisztériuma által ajánlott oktatási szoftvereket! Ezek közül melyeket használhatjuk az iskolánk profiljának megfelelő tárgyak oktatása során?

7*. Milyen online forrásokat alkalmazhatunk az iskolánk profiljának megfelelő tárgyak oktatása során? Keressük meg ezeket az interneten!

Végezzük el a feladatokat!

1°. Állapítsuk meg, milyen, a fizika, kémia és biológia tanítását segítő szoftvereket találhatunk az **Острів знань** (http://www.ostriv.in.ua) webhely ***Сервіси ⇒ Файловий архів ⇒ Електронні засоби навчання*** részében. Állapítsuk meg ezen oktatási szoftverek típusát!

2°. Töltsük le az **Острів знань** (http://www.ostriv.in.ua) webhelyen a ***Навчання ⇒ Шкільні предмети ⇒ Хімія ⇒ Електронні засоби для вивчення хімії*** linkről a **Таблиця Менделєєва** szoftvert. Telepítsük a programot a számítógépünkre. Ismerkedjünk meg a súgó segítségével a szoftver üzemmódjaival! Milyen típusba sorolható ez az oktatási szoftver? Hozzunk fel 3 példát, amikor ezt a szoftvert használnánk feladataink megoldása során! Készítsünk prezentációt a programról!

3°. Nyissuk meg a **Шкільна фізика** (http://sp.bdpu.org) honlapot, és állapítsuk meg, hogy az ott található anyagok közül melyek lennének hasznosak számunkra a fizika tanulása során. Vizsgáljuk meg a **Демонстрації** link alatt a ferde hajítás (рух тіла, кинутого під кутом до горизонту) interaktív modellt. Magyarázzuk meg a modell által bemutatott fizikai jelenséget!

4°. Töltsük le a **Google Earth** (http://earth.google.com) webhelyről a Google Earth ingyenes változatát, és telepítsük azt a számítógépünkre! Keressük meg településünk térképét és a nevezetességeiről készült fényképeket!

5°. Nyissuk meg a szumi egyetem Дистанційне навчання. **Демонстраційні фрагменти курсів** webhelyét (http://dl.sumdu.edu.ua/demo)! Ismerkedjünk meg az ott található interaktiv eszközökkel. Tanulmányozzuk az **Окислювально-відновлювальні реакції** (oxidációs-redukciós reakciók) oktatási szoftvert! Készítsünk prezentációt a szoftver működéséről!

4. számú gyakorlati munka — Oktatási szoftverek alkalmazása a fizika, kémia és biológia tanulása során

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

1. •Indítsunk el egy a számítógépünkön telepített oktatási szoftvert (interaktív multimédiás kurzust, elektronikus tankönyvet, elektronikus gyakorlóeszközt, digitalizált képgyűjteményt, elektronikus térképet stb.)!
   1) Ismerkedjünk meg a szoftver menürendszerével!
   2) Ismerkedjünk meg a szoftver által ismertetett tartalmakkal!
   3) Állapítsuk meg a program vezérlőeszközeinek rendeltetését!
   4) Állítsuk össze a program rövid ismertetőjét!
   5) Keressünk 3 példát, amikor a szoftvert célszerű a tanulásban felhasználni!
2. Nyissuk meg a **wiki.ciit.zp.ua** honlapot!
   1) Kattintsunk a **Интернет-ресурсы** linkre a **Свободная зона** részben!
   2) Ismerkedjünk meg az iskolánk szakmai profiljának megfelelő forrásokkal!
   3) Válasszunk ki ezek közül egyet, és nyissuk meg!
   4) Ismerkedjünk meg a kiválasztott forrás által nyújtott lehetőségekkel!
   5) Állapítsuk meg a kiválasztott forrás rendeltetését, a várható felhasználók körét, alkalmazhatóságát a tantárgy tanulása során!
   6) Írjuk le füzetünkbe megállapításainkat!

2. fejezet

ALGORITMIZÁLÁS ÉS PROGRAMOZÁS

Ebben a részben megismerkedünk:

- A modell fogalmával, a modellezés folyamatával és a modellek típusaival;
- Az információs modellek alkotásával;
- Az algoritmusok tulajdonságaival és az alapvető algoritmusszerkezetekkel;
- A számítógépes feladatmegoldás lépéseivel;
- A számítógépes modellezéssel;
- A számítógépes programokkal;
- A program megírásának és tesztelésének folyamata integrált fejlesztői környezetben;
- A számítógépes adatfeldolgozás módszereivel.

2.1. Modellek és modellezés. A modellek típusai

1. Hozzunk fel példát objektumokra! Milyen tulajdonságai vannak ezeknek az objektumoknak, és milyen értékeket vehetnek ezek fel?
2. Kik a modellezők? Mivel foglalkoznak?
3. Milyen modelleket ismerünk? Mi célból hozták ezeket létre? Milyen órákon használtunk már modelleket?
4. Milyen folyamatokat, illetve jelenségeket modellezhetünk az $y = kx$ egyenlettel?

A modell fogalma. Modellezés

Az egyes objektumok (tárgyak, folyamatok, jelenségek) tulajdonságainak és kölcsönhatásainak tanulmányozása során különféle vizsgálatokat végeznek. Nem mindig célszerű, és sokszor nem is lehet a tárgyakat, folyamatokat és jelenségeket közvetlenül vizsgálni. Ilyen esetekben nem magukat az objektumokat, hanem azok **modelljeit** vizsgálják. A modell szó a latin ***modulus*** kifejezésből ered, amelynek jelentése: *minta, analóg.*

Modellt alkotunk minden olyan esetben, amikor az objektum elérhetetlen, közvetlenül nem vizsgálható (például a Nagymedve csillagkép vagy egy vulkánkitörés), ha az objektum vizsgálata annak károsodásához vezethet (például egy híd), vagy ha az objektum elkészítése túlságosan nagy anyagi ráfordítást igényelne (egy új városrész felépítése).

Modelleket nemcsak akkor készítünk, ha az objektumok elérhetetlenek vagy elkészítésükhöz nem állnak rendelkezésre megfelelő anyagi eszközök, hanem például akkor is, ha az objektum tulajdonságainak csak egy részét vizsgáljuk. Ilyenkor olyan modellt ho-

zunk létre, amelyek a vizsgált tulajdonságokkal bírnak, az objektum egyéb tulajdonságaival pedig nem.

Tipikus modell a térkép és a földgömb, a mértani testek modelljei, az emberi szervek modelljei, az iskolai sportpálya tervrajza, egy gépkocsit ábrázoló rajz, egy ismeretlen ember szóbeli leírása.

2.1. ábra. Ruhamodellek

A ruhamodelleket azért hozzák létre, hogy felmérjék a vásárlók igényeit (2.1. ábra), a repülőgépmodelleket az aerodinamikai tulajdonságok vizsgálatának céljából, a gépkocsi-modelleket a váz teherbírásának vizsgálatáért, az építészeti modelleket azért, hogy tanulmányozhassák a tervezett és létező épületek összhatását, a kémiai modelleket pedig a vegyi folyamatok tanulmányozása céljából.

A fizikusok modellezik a testek különböző erők hatására történő mozgását, az atommagban zajló folyamatokat, az atomháború meteorológiai következményeinek tanulmányozására létrehozták az „atomtél" modellt. A szeizmológusok a földrengéseket modellezik, a közgazdászok a társadalom változásait, a mérnökök az űrhajó Marsra szállását, a paleontológusok pedig a dinoszauruszok kihalását.

A *vizsgálat tárgyát* azok az objektumok alkotják, amelyeknek a viselkedését és együttműködését tanulmányozzák.

A zoológia tudomány tárgyát például az állatok sokasága képezi, amikor pedig a matematikusok az oszthatóságot tanulmányozzák, akkor a kutatás tárgyát az egész számok képezik.

Az *objektummodell* egy olyan új objektum, amely az eredeti objektumnak az adott kutatás szempontjából fontos tulajdonságait tartalmazza.

Az objektumok tulajdonságainak modellek segítségével történő vizsgálatát *modellezésnek* nevezzük.

Fontos megjegyeznünk, hogy az egyes tudományok az objektumok különböző tulajdonságait vizsgálják. Ez az oka annak, hogy egy objektumnak több modellje van. Ez tehát attól függ, hogy az objektumok mely tulajdonságait vizsgáljuk. A fizikus, a vegyész, a biológus, az orvos, a divattervező különbözőképpen fogja modellezni az embert.

Másrész előfordulhat, hogy különböző objektumok közös modellel rendelkeznek. A derékszögű paralelepipedon a könyv, a szekrény, de sok egyéb objektum modellje a matematikában. Az $y = kx$ függvény az egyenes vonalú egyenletes mozgás modellje, ugyanakkor az egyenáram áramerőssége és feszültsége közötti összefüggés is, vagy a vásárolt burgonya mennyisége és ára közötti összefüggésé is.

Helytállónak fogadhatjuk el azt a kijelentést, miszerint az ember minden intellektuális tevékenysége során tárgyak, folyamatok és jelenségek képeivel foglakozik, amelyek lényegében modellnek tekinthetőek. Amikor egy konkrét objektumra gondolunk, valójában elkülönítjük annak tulajdonságai közül azokat, amelyek a vizsgálatunk szempontjából lényegesek.

Nyilvánvaló, hogy a modell helyes megalkotása meghatározza a tudományos vagy mérnöki következtetések helyességét és pontosságát. A modern tudományban és technikában a modellalkotást és a modellek elemzését is számítógépek és speciális szoftverek segítségével végzik.

Vannak speciális programok, amelyek megvalósítják az objektum modelljét. Ezeket a programokat az **objektum számítógépes modelljének** nevezik. Manapság a számítógépes modelleket széles körben alkalmazzák, **számítógépes kísérleteket** végeznek olyankor, ha a valóságos kísérlet lehetetlen, esetleg előreláthatatlan következményekkel járhat.

A modellek típusai

A modellek csoportosítását több szempont szerint végezhetjük. Ez lehet:

- Az előállítás módja
- Az alkalmazási terület szerint
- Az időbeliség szerint.

Ezenkívül elképzelhető másféle csoportosítás is.

Az **előállítás módja** szerint megkülönböztetünk **anyagi (tárgyi)** és **gondolati (eszmei)** modelleket.

A játékok, a kitömött állatok, a manökenek, a próbababák, a földgömb, a vízimalom modellje anyagi modellek. Az anyagi modelleket **gyakorlati vizsgálatok** céljából hoznak létre.

Ukrajna természetföldrajzi térképe, egy kémiai reakció egyenlete, egy matematikai függvény, egy elbeszélés a Dnyeper partjáról **gondolati modellek.** Az **eszmei modelleket** elméleti vizsgálat céljából hozzák létre.

Az *objektum anyagi modellje* egy valóságos tárgy, általános jellemzője valamilyen kézzelfogható megjelenés.

Az *objektum eszmei* modellje az objektum egyfajta leírása. A valóságot ez esetben gondolatilag modellezzük.

Az objektumnak a vizsgálat céljától függően különböző valóságos és információs modelljei lehetnek. A paradicsom gondolati modellje egy felvásárló számára olyan adatokat tartalmaz, mint a méretek, a betakarítás módja, az érés ideje, a tárolás maximális ideje. A termelő számára a paradicsom gondolati modellje olyan adatokat tartalmaz, mint a vetés ideje, a locsolási igény, a trágyázás alkalmazása stb.

A gondolati modellek megjelenítésének módjai a következők lehetnek:

- **szóbeliek** (szóban vagy írásban);
- **grafikusak** (rajzok, műszaki rajzok, piktogramok, térképek stb.);
- **szerkezetiek** (táblázatok, grafikonok, diagramok, vázlatok stb.);
- **algoritmikusak** (szabályok, cselekvési tervek);
- **matematikaiak** (képletek, egyenletek, egyenlőtlenségek, függvények stb.);
- **speciálisak** (kémiai képletek és egyenletek, kották, sakkjátszmák lejegyzései).

A gondolati modell létrehozása céljából:

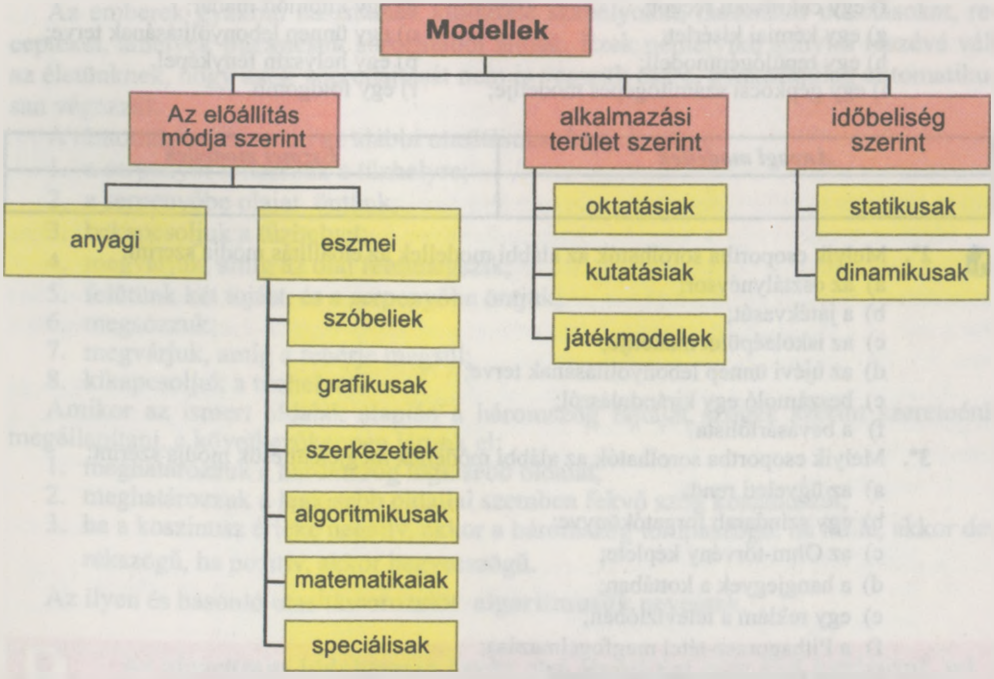
1. Meg kell határozni a vizsgálandó objektumot és a vizsgálat célját.
2. Meg kell határozni az objektum azon tulajdonságait, amelyek lényegesek az adott vizsgálat szempontjából.
3. Meg kell határozni a lényeges tulajdonságok között fennálló összefüggéseket és kifejezni azokat valamely fentebb leírt alakban.

Alkalmazási terület szerint a modellek lehetnek:

- **oktatásiak** (illusztrációk, számítógépes programok, szimulátorok, például repülőgép-szimulátorok, amelyek a pilóták és űrhajósok kiképzésére szolgálnak;
- **kutatásiak** (a valóságos objektumok másolatai, például egy gőzhajó modellje a hajó manőverezési képességének tanulmányozására, egy helikopter modellje aerodinamikai vizsgálatok céljára, a vízi erőmű modellje az ökológia hatás tanulmányozása céljából, továbbá olyan képletek, egyenletek és függvények, amelyeket felhasználhatunk az objektumok tanulmányozására;
- **tudományosak** (atomreaktor modellje, a villámot imitáló eszköz, PC egységeket tesztelő berendezések, hadronütköztető stb.);
- **játékmodellek** (katonai, gazdasági, sport és egyéb játékok, amelyeket viselkedésmódok kialakítása és ellenőrzése céljából hoznak létre).

Időbeliség szerint a modellek lehetnek:

- **statikusak** – olyan modellek, amelyek állapota egy adott pillanatban rögzítésre kerültek és a későbbiek során nem változtathatók meg. Például a Kheopsz piramis gipsz másolata;
- **dinamikusak** – olyan modellek, amelyeket az objektumok időben változó tulajdonságainak vizsgálata céljából hoztak létre. Például a hegyi folyók hóolvadás során bekövetkező vízhozamának változását tanulmányozandó modell, az időjárás modellje valamely régióban, a Naprendszer modellje, amely megváltozik az újabb égitestek felfedezése során.



2.2. ábra. A modellek típusai

Ellenőrző kérdések

1°. Mit értünk a vizsgálat tárgya alatt? Hozzunk fel példákat!
2°. Mi a modell? Hozzunk fel példákat!
3°. Mi célból hozzák létve a modelleket? Hozzunk fel példákat!
4°. Miben áll a modellezés folyamata?
5°. Miben különbözik a modell a valóságos objektumtól?
6°. Miért lehetnek egy objektumnak különböző modelljei? Hozzunk fel példákat!
7°. Miért alkalmazható ugyanaz a modell különböző objektumokra? Hozzunk fel példákat!
8°. Hogyan csoportosíthatók a modellek az előállítás módja szerint? Jellemezzük ezeket!
9°. Mik a anyagi modellek? Hozzunk fel példákat!
10°. Mik az információs modellek? Hozzunk fel példákat!
11°. Milyenek lehetnek az eszmei modellek? Jellemezzük ezeket!
12°. Hogyan csoportosíthatjuk a modelleket alkalmazási terület szerint? Hozzunk fel példákat!
13°. Hogyan csoportosíthatjuk a modelleket időbeliségük szerint? Hozzunk fel példákat!
14°. Mi a számítógépes modellezés?

Végezzük el a feladatokat!

1°. Jellemezzétek a következő modelleket, és írjátok be ezeket a táblázat megfelelő helyére:
a) egy piramisról készült rajz;
b) az iskola alaprajza;
c) a téglalap területének képlete;
d) egy zenemű kottája;
e) egy terület beépítésének makettje;
f) egy cukrászati recept;
g) egy kémiai kísérlet;
h) egy repülőgépmodell;
i) egy gépkocsi számítógépes modellje;
j) egy atommodell;
k) egy sakkparti lejegyzése;
l) egy fizikai kísérlet elvégzése;
m) egy kémiai reakció egyenlete;
n) egy kitömött madár;
o) egy ünnep lebonyolításának terve;
p) egy helyszín fényképe!
r) egy földgömb.

Anyagi modellek	*Eszmei modellek*

2°. Melyik csoportba sorolhatók az alábbi modellek az előállítás módja szerint:
a) az osztálynévsor;
b) a játékvasút;
c) az iskolaépület makettje;
d) az újévi ünnep lebonyolításának terve;
e) beszámoló egy kirándulásról;
f) a bevásárlólista?

3°. Melyik csoportba sorolhatók az alábbi modellek megjelenítésük módja szerint:
a) az ügyeleti rend;
b) egy színdarab forgatókönyve;
c) az Ohm-törvény képlete;
d) a hangjegyek a kottában;
e) egy reklám a televízióban;
f) a Pithagorasz-tétel megfogalmazása;
g) egy kémiai reakció egyenlete;
h) az ukrán kosárlabda-bajnokság tabellája;
i) egy önéletrajz?

4°. Képzeljük el a virág és a dinnye modelljét a művész, az orvos, a botanikus, a kereskedő és a beszállító szempontjából!.

5°. Hozzuk létre a hétvégi ház modelljét a művész, az építész és a vásárló szemszögéből!

6°. Az *A* és *B* közötti távolság 50 km. Ezekből a helységekből egyidejűleg egymás felé elindult két kerékpáros, akik sebessége 12 km/h és 15 km/h. Hozzunk létre matematikai modellt, amelynek segítségével meghatározható a két kerékpáros egymástól mért távolsága bármely, tetszőlegesen választott időpontban!

7*. Egy gépírónő 20, a másik 30 óra alatt képes legépelni egy kéziratot. Hozzunk létre matematikai modellt annak meghatározására, hogy hány oldalt képesek legépelni ketten x óra alatt, amennyiben mind a ketten dolgoznak!

8*. Nevezzünk meg olyan folyamatokat, melyeknek matematikai modellje: lineáris függvény; másodfokú függvény; fordított arányosság!

2.2. Az algoritmusok és azok tulajdonságai. Az algoritmusok megadásának módjai

1. Milyen mondatokat nevezünk felszólítónak?
2. Készítettél-e valamilyen ételt recept alapján? Hogy végezted ezt?
3. Milyen órákon követtetek utasításokat? Milyenek voltak ezek?
4. Hozz fel nyelvtani példákat, amelyekben ha..., akkor nyelvi szerkezet szerepel!

Az algoritmus fogalma

Az emberek gyakran használnak különféle szabályokat, használati utasításokat, recepteket, amelyek **parancsok** sorozatából állnak. Ezek némelyike annyira részévé vált az életünknek, hogy azok végrehajtását nem is vesszük észre, gyakorlatilag automatikusan végezzük.

A rántottakészítés során az alábbi utasításokat hajtjuk végre:

1. a serpenyőt feltesszük a tűzhelyre;
2. a serpenyőbe olajat öntünk;
3. bekapcsoljuk a tűzhelyet;
4. megvárjuk, amíg az olaj felmelegszik;
5. felütünk két tojást, és a serpenyőbe öntjük;
6. megsózzuk;
7. megvárjuk, amíg a fehérje megsül;
8. kikapcsoljuk a tűzhelyet.

Amikor az ismert oldalak alapján a háromszög fajtáját szögek szerint szeretnénk megállapítani, a következőképpen járunk el:

1. meghatározzuk a háromszög legkisebb oldalát;
2. meghatározzuk a legkisebb oldallal szemben fekvő szög koszinuszát;
3. ha a koszinusz értéke negatív, akkor a háromszög tompaszögű, ha nulla, akkor derékszögű, ha pozitív, akkor hegyesszögű.

Az ilyen és hasonló utasítássorozatot **algoritmusük** nevessak.

Az algoritmus kidolgozása során első lépésként meg kell határozni, mi lesz az algoritmus végrehajtója, és meg kell ismerni annak utasításkészletét.

2.3. ábra. Al Khwarizmi

Az algoritmus elnevezés a középkori kelet neves matematikusának, **Muhammad ibn Musza al Khwarizminek** (783–850) nevéből származik (2.3. ábra), aki matematikai, földrajzi és csillagászati munkáiban az indiai helyértékes (pozicionális) számrendszert használta és megfogalmazta a négy alapművelet – összeadás, kivonás, szorzás, osztás – végrehajtásának szabályait. Műveit a latin fordításnak köszönhetően az európai tudósok is megismerték. A fordító a szerző nevét Algorithmus-nak írta. Innen származik az algoritmus szó. A matematikai műveletek elvégzésére általa kidolgozott eljárásokat tekintjük az első algoritmusoknak.

Az algoritmusok tulajdonsága

Az algoritmus a következő tulajdonságokkal bírhat: **diszkrétség, determináltság (egyértelműség), végrehajthatóság, végesség, hatékonyság, általános érvényű.**

Az algoritmus **diszkrét** (a *discretus* latin szóból ered, ami annyit tesz, mint tagolt) azt jelenti, hogy az algoritmus végrehajtása elkülöníthető lépésekből áll. Az utasítások véges idő alatt végrehajthatók kell legyenek.

Az algoritmus **determinált (egyértelműen meghatározott)** (a latin *determinans – döntő, meghatározó* szóból), vagyis ugyanazokra a kezdőállapotokra és ugyanazokhoz a bemeneti adatokhoz mindig ugyanolyan utasítássorozat tartozik, amely mindig ugyanolyan eredményhez vezet. Az algoritmus nem tartalmazhat a végrehajtó számára nem egyértelmű utasításokat (például vegyünk két–három kanál cukrot, kissé melegítsük fel a tejet, kapcsoljuk ki a világítást néhány perc elteltével, osszuk el x-et a-val vagy b-vel). Az algoritmus végrehajtása során nem fordulhat elő olyan helyzet, amikor nem dönthető el egyértelműen, melyik a következő végrehajtandó utasítás.

A **végrehajthatóság** azt jelenti, hogy az algoritmus csak a végrehajtó utasításkészletéhez tartozó utasításokat tartalmazhat. Egy első osztályos számára készített algoritmus nem tartalmazhat olyan utasításokat, hogy szerkeszd meg a szög felezőjét, de a nyolcadik osztályos számára készített algoritmus már tartalmazhat ilyet.

Megjegyezzük, hogy a végrehajtónak csak **végre kell tudni hajtani** az utasításokat, értelmeznie nem kell. Az algoritmus végrehajtása **formális jellegű:** a végrehajtó nem kell értse az utasításokat és nem kell ismerje az algoritmus célját, de mindenképpen el kell érje ezt a célt. A számítógépes vezérlésű szerszámgép az általa végrehajtott utasítások egyikét sem érti, de mindenképpen elkészíti az alkatrészt.

Az algoritmus véges, vagyis véges (de tetszőlegesen sok) számú lépésből kell álljon és véges idő alatt be kell fejeződjön bármilyen bemenő adat esetében. A fenti algoritmusok mindegyike véges volt, a következő utasítássorozat végtelen:

1. vegyük a 2-es számot;
2. szorozzuk meg tízzel;
3. adjunk az eredményhez 5-öt;
4. ha a kapott szám pozitív, akkor térjünk vissza a 3. utasításhoz, egyébként leállítjuk az algoritmus végrehajtását.

Az algoritmus **hatékonysága azt jelenti,** hogy végrehajtásának eredményeképpen mindig elérjük a kitűzött célt. Az algoritmus akkor is hatékonynak tekintendő, ha képes meghatározni, ha a feladat megoldhatatlan, vagy a bemeneti adatok adott értékei mellett nem létezik megoldása.

Az algoritmus **általános érvényű,** vagyis nemcsak egy adott feladat megoldására alkalmas, hanem a hasonló, csak bemeneti adatokban eltérő feladatokra is.

A másodfokú egyenlet megoldásának algoritmusa segítségével meghatározhatjuk a gyököket minden valós együttható esetében, vagy kimondhatjuk, hogy az egyenletnek nincsenek valós gyökei. A szögfelező körzővel és vonalzóval történő megszerkesztésének algoritmusa is általános érvényű.

Nem minden algoritmus általános érvényű. Például a $2x^2+5x+2=0$ másodfokú egyenlet megoldásának algoritmusa, vagy a görög saláta receptje 4 személyre konkrét, tehát nem általános érvényű.

Az algoritmus leírásának eszközei

Vizsgáljuk meg a következő algoritmust.

1. feladat. Van egy 8 literes edény folyadékkal megtöltve, valamint két üres, egy 3 és egy 5 literes edény. Ezek segítségével ki kell mérnünk 1 liter folyadékot.

Az algoritmus végrehajtója a következő utasításkészlettel rendelkezik:

1. Kiválasztani a meghatározott edényt.
2. Átönteni a kiválasztott edény tartalmát egy másikba.
3. Megtölteni egy edényt egy másik edényből.

E végrehajtó részére a feladat megoldásának algoritmusa a következő:

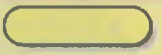
1. Vedd a 8 literes edényt.
2. Töltsd meg ebből a 3 literes edényt.
3. Öntsd át a 3-literes edény tartalmát az 5 literesbe.
4. Töltsd meg a 3 literes edényt a 8 literesből.
5. Töltsd meg az 5 literes edényt a 3 literesből.
6. Vedd a 3 literes edényt.

A fenti algoritmus, mint minden eddigi, egymás után leírt mondatokból áll, amelyek meg vannak számozva és magyar nyelven vannak leírva. Az algoritmus leírásának ezt a módját **mondatszerű leírásnak** nevezzük.

A mondatszerű leírás nem mindig kényelmes. Ezért a mondatszerű leírás mellett **grafikus** módszert is alkalmaznak, például **folyamatábrát.**

A folyamatábrában minden utasításhoz egy mértani alakzat tartozik. Az alakzatokat nyilak kötik össze, amelyek az utasítások sorrendjét mutatják. A folyamatábra elemeit a 2.1. táblázat mutatja.

2.1. táblázat. **A folyamatábra elemei**

Megnevezés	*Jel*	*Jelentés*
Terminátor		Az algoritmus eleje és vége
Utasítás		Egy vagy több utasítás végrehajtása
Adatok		**Adatbevitel- vagy kivitel**
Döntés		Döntéshozatal a vizsgált feltétel szerint

Az 1. feladat megoldásának algoritmusát szemléltető folyamatábra (2.4. ábra)

A vizsgált algoritmus sajátossága, hogy az utasításokat a leírt sorrendben kell végrehajtani, minden utasítást végre kell hajtani és mindet egyszer kell elvégezni. Az ilyen algoritmusokat *szekvenciálisnak (lineárisnak)* nevezzük.

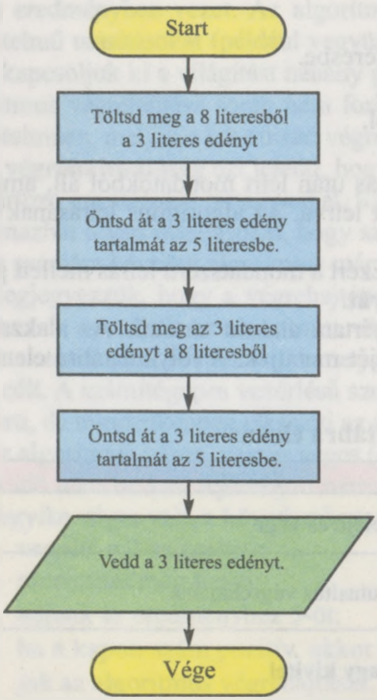
Vizsgáljunk most meg egy olyan feladatot, ahol a kifejezés értékének kiszámítását kell elvégezni, és össze kell állítani a megoldás algoritmusát!

2. feladat. Számítsuk ki az $(a - b)*(c - d)$ értékét, ahol a, b, c és d valós számok!

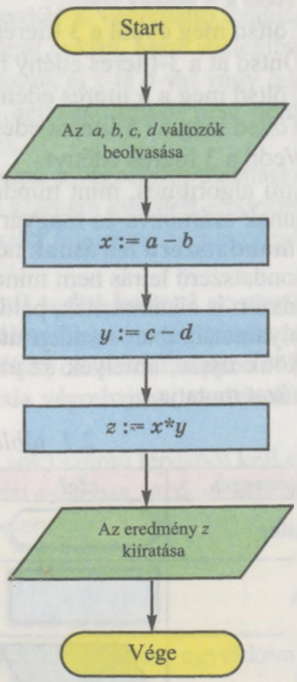
A végrehajtó tud matematikai műveleteket végezni, képes megjegyezni azok eredményeit és közölni az eredményt. Azt a folyamatot, amelynek során adatokat közlünk a végrehajtóval, **adatbevitelnek**, amelynek során pedig a végrehajtó közöl adatokat velünk, **adatkivitelnek** nevezzük.

Fogalmazzuk meg az algoritmust mondatszerűen.

1. Vigyük be az a, b, c és d értékeit (ennek eredményeképpen a végrehajtó megjegyzi ezek értékeit)!
2. Számítsuk ki az $a - b$ értékét, az eredményt mentsük el x-be!
3. Számítsuk ki a $c - d$ értékét, az eredményt mentsük el y-ba!
4. Számítsuk ki $x*y$ értékét, az eredményt mentsük z-be!
5. Vigyük ki a z értékét!



2.4. ábra. Az 1. feladat algoritmusának folyamatábrája



2.5. ábra. A 2. feladat algoritmusának folyamatábrája

A 2, 3 és 4 utasításokban a kiszámított értéket valamely változónak adjuk át. Ezek értékadó **utasítások,** amelyeket a következőképpen írunk le:

2. $x:=a-b$ (x legyen egyenlő $a-b$ -vel)

3. $y:=c-d$ (y legyen egyenlő $c-d$ -vel).

4. $z:=x*y$ (z legyen egyenlő $x*y$-vel).

A := jel az **értékadás jele,** *kettőspont* és *egyenlő* jelekből áll, amelyek közé nem teszünk szóközt.

Lássuk az algoritmus folyamatábráját (2.4. ábra).

Ez az algoritmus szintén szekvenciális, az utasítások az a, b, c és d bármely értékénél egyszer hajtandók végre.

Vizsgáljuk meg a 2. feladat algoritmusának végrehajtását, ha $a = 3$; $b = 4$; $c = -2$; $d = -5$.

Utasítás	*A végrehajtás eredménye*
a, b, c, d beolvasása	$a = 3, b = 4, c = -2, d = -5$
$x := a - b$	$x = 3 - 4 = -1$
$y := c - d$	$y = -2 - (-5) = 3$
$z := x*y$	$z = -1*3 = -3$
Z kiírása	$z = -3$

Hasonlóképpen járunk el a, b, c, és d tetszőleges értékeinél.

Ellenőrző kérdések

1°. Hozzunk fel példákat olyan mondatokra, amelyek utasítások és olyanokra, amelyek nem!

2°. Mi az *algoritmus*? Hozzunk fel példákat!

3°. Mi az *utasításkészlet*? Hozzunk fel példákat!

4°. Milyen tulajdonságai vannak az algoritmusoknak? Magyarázzuk el valamennyit!

5*. Hozzunk fel példát olyan utasássorra, amely nem hajtható végre!

6*. Hozzunk fel példát olyan utasássorozatra, amely nem egyértelmű!

7°. Mi különbség van az algoritmus mondatszerű és grafikus előállítása között?

8°. Soroljuk fel a folyamatábra elemeit! Mi ezek rendeltetése?

9°. Milyen algoritmust nevezünk szekvenciálisnak?

10*. Hozzunk fel példákat az algoritmus végrehajtójára a mindennapi életből! Milyen algoritmusokat hajtanak ezek végre?

11°. Mi az értékadó utasítás? Mi a jelölése?

Végezzük el a feladatokat!

1°. Válasszuk ki az utasításokat az alábbi mondatok közül:

a) Zárd be az ablakot!

b) Hány óra van?

c) 3+2=5.

d) Ne zavarj az olvasásban!

e) Ha esik, vigyél esernyőt!

f) Kijevben élek.

2*. Hozzunk fel példákat olyan szekvenciális algoritmusokra, amelyeket

a) magyar nyelv;

b) matematika;

c) más tantárgyak

tanulása során alkalmaznak! Írjuk fel az algoritmusokat folyamatábraként!

3°. Hajtsuk végre az algoritmust!
a) Adjuk össze az 1 és 3 számokat.
b) Adjunk a kapott számhoz 5-öt.
c) Adjunk a kapott számhoz 7-et.
d) Adjunk a kapott számhoz 9-et.
e) Adjunk a kapott számhoz 11-et.
f) Mondjuk meg az eredményt.

4°. Hajtsuk végre az algoritmust!
a) Rajzoljunk egy 5 cm-es AB szakaszt.
b) Szúrjuk a körzőt az A pontba.
c) Rajzoljunk AB sugarú kört.
d) Szúrjuk a körzőt a B pontba.
e) Rajzoljunk AB sugarú kört.
f) Kössük össze a körök metszéspontjait.

Hogy nevezhetjük el ezt az algoritmust? Milyen mértani feladat megoldásának algoritmusa ez? Állítsuk össze a folyamatábrát!

5°. Állítsuk össze a sajtos szendvics elkészítésének mondatszerű algoritmusát!

6°. Állítsuk össze a kedvenc ételünk elkészítésének mondatszerű algoritmusát!

7°. Adva van egy számegyenes egészértékű koordinátapontokkal. Ezen a számegyenesen mozog **Lovacska** végrehajtó, aki a következő utasításkészlettel bír: 1) ugrás 3 lépéssel jobbra; 2) ugrás 2 lépéssel balra. A **Lovacska** kezdetben a 0 pontban helyezkedik el. Állítsunk össze algoritmusokat, melyeket végrehajtva **Lovacska** a legkevesebb lépéssel jut: a) a 24; b) a 7; c) a –3 koordinátájú pontba.

8°. Van egy 8 literes edény folyadékkal megtöltve, valamint két üres, 3 és 5 literes edény. Ezek segítségével ki kell mérnünk 2 liter folyadékot. Fogalmazzuk meg az algoritmust mondatszerűen, és állítsuk össze a megoldás algoritmusának folyamatábráját!

9*. Levest kell készítenünk leveskockából. Rendelkezésünkre áll két, egy 3 és egy 8 perces homokóra. Állítsuk össze az időmérés algoritmusát, ha a levest: a) 5 percig; b) 7 percig; c) 10 percig kell főzni!

10°. A révésznek át kell szállítani a folyón egy farkast, egy kecskét és egy fej káposztát. A csónakban a révészen kívül csak a farkas, vagy a kecske, vagy pedig a káposzta fér el. A parton nem maradhat együtt a farkas a kecskével, illetve a kecske a káposztával. Állítsunk össze szállítási algoritmust a révész számára. (A feladat egy VIII. századi matematikai kéziratból származik). Fogalmazzuk meg az algoritmust mondatszerűen!

11°. Két katonának át kell jutni a folyó túlsó partjára. Megláttak egy ladikot, amelyikben két fiú ült. A csónak olyan kicsi, hogy abban csak két fiú, vagy egy katona fér el. Az átkelés után a csónaknak a fiúknál kell maradni. Állítsuk össze az átkelés algoritmusát mondatszerű leírás formájában!

12°. Állítsuk össze a körző és vonalzó segítségével történő szögfelezés algoritmusának folyamatábráját!

13. Állítsuk össze a háromszög megszerkesztésének algoritmusát három oldal alapján!

14*. 3 szelet kenyeret kell megsütni egy serpenyőben, amelyben egyszerre csak két szelet fér el. A kenyér egyik oldalának megsütése két percig tart. Állítsunk össze utasításkészletet egy olyan végrehajtó számára, ami ezt a feladatot el tudja végezni! Állítsuk össze a megoldás mondatszerű algoritmusát! Hány percig tart a feladat megoldása? Állítsunk össze egy hatperces algoritmust!

15*. Találjunk ki egy algoritmus-végrehajtót és annak az utasításkészletét! Fogalmazzunk meg számára egy feladatot, és állítsuk össze a megoldás algoritmusát!

16°. Állítsuk össze az $52 + (69 - 35) \cdot 5$ kifejezés kiszámításának algoritmusát!

17°. Állítsuk össze a $(81-12) \cdot (58+84)$ kifejezés kiszámításának algoritmusát!

18°. Állítsuk össze a $(a + b) - c \cdot a$ kifejezés kiszámításának algoritmusát! Hajtsuk végre az algoritmust különböző a, b és c értékek mellett!

19°. Állítsuk össze a $2x + a = s$ egyenlet megoldásának algoritmusát! Hajtsuk végre azt, ha: 1) $a = 5$; $c = 7$; 2) $a = -15$; $c = 105$; 3) $a = 5$; $c = 5$!

2.3. Számítógépes programok és programozási nyelvek. A számítógépes feladatmegoldás lépései

1. Mi a számítógépes szoftver rendeltetése, és milyen szoftvertípusokat ismerünk?
2. Mi az algoritmus? Mi az utasításkészlet? Miért nevezzük az algoritmus végrehajtását formálisnak?
3. Mit értünk a kódolás folyamata alatt? Mire alkalmazzák? Mi a bináris kódolás?
4. Mi a modell? Milyen modelltípusokat ismerünk?
5. Nevezzük meg a számítógépek alkalmazásának körét!

Számítógépes programok

Tudjuk, hogy a számítógépet programok vezérlik, amelyek között különböző rendeltetésű programokat találunk. Ilyenek a szövegszerkesztő, képszerkesztő, tömörítőprogram, vírusirtó, táblázatkezelő, prezentációkészítő, az oktatási szoftverek, játékprogramok és sok más.

A *számítógépes program* nem más, mint a számítógép számára készített utasítássorozat, amelyet annak a megfelelő sorrendben kell végrehajtani.

A programok tehát a számítógép számára értelmezhető algoritmusok. Ezek az algoritmusok alkotják a **program logikáját.**

Működése során a program adatokat dolgoz fel. Az adatokat a programmal közölheti a felhasználó, valamely külső eszköz (hőmérséklet-érzékelő, szöveges állomány). Ezek a bemeneti adatok. Vannak olyan programok, amelyek **bemeneti adatok** nélkül dolgoznak. A program maga is adatokat állít elő. Ezek a **kimeneti adatok.** A feladat megoldása során a program egyéb adatokkal is dolgozhat, ezek a **köztes adatok.**

A modern programok döntő többsége eszközöket tartalmaz a felhasználóval történő kommunikációhoz. Ezek lehetnek vezérlőeszközök (gombok, menük stb.), adatbeviteli eszközök (beviteli mezők, számlálók) és adatkiviteli eszközök (feliratok, mezők). Ezeknek az eszközöknek az összessége alkotja a **felhasználói felületet.**

A világ első programozójának **Ada Lovelace**-t (1815–1852), az ismert költő, George Gordon Byron lányát tekintik, aki leírást készített a **Charles Babbage** (1791–1871) (2.7. ábra) által tervezett Analytical Engine-hez, amelyben leírta, hogy lehet programot készíteni e gép számára. Az Analitikai Gép sajnos soha nem épült meg. A leírás elkészítőjéről nevezték el az **Ada** programozási nyelvet.

2.6. ábra. Ada Lovelace

2.7. ábra. Charles Babbage

Programozási nyelvek

Az emberek számára készült algoritmusokat valamely, emberek által beszélt (ukrán, magyar, angol, német) nyelven fogalmazzuk meg.

Ha az algoritmust egy automata (például számítógép) számára készítjük, az emberi nyelv már nem felel meg, mivel nagyon bonyolult és nem egyértelmű konstrukciókat is tartalmaz, például azonos alakú, de eltérő értelmű szavakat, szinonimákat. Ezért szükséges, hogy az automaták számára speciális nyelveket – programozási nyelveket – dolgozzanak ki.

***Programozási nyelv* a számítástechnikában használt utasítások sorozata, amelyen a számítógép számára értelmezhető alakban írhatjuk le az algoritmusokat.**

A programozási nyelvek a következő alapelemekből állnak:

1. **karakterkészlet** – rögzíti, milyen jeleket tartalmazhatnak az adott nyelv utasításai;
2. **foglalt szavak (vagy alapszavak)** – a nyelv kulcsszavait (utasításait) tartalmazza;
3. **szintaxis** – az nyelvi szerkezetek kialakításának és leírásának szabályai;
4. **szemantika** –a nyelvi szerkezetek egyértelműségét biztosító szabályok.

A meg nem engedett szimbólumok használata, az utasítások helytelen leírása, a szintaktikai szabályok be nem tartása ahhoz vezet, hogy a számítógép képtelen végrehajtani a program helytelen részét. Az ilyen hibákat **szintaktikai hibáknak** nevezzük.

A programozási nyelvek a **mesterséges nyelvek** közé tartoznak. A kémiai reakciók nyelve, a matematikai egyenletek és egyenlőtlenségek nyelve, a postai irányítószámok nyelve, a sakkjátszmák lejegyzésének nyelve szintén ilyen mesterséges nyelv.

Az utóbbi 70 évben nagyjából 3000 különböző programozási nyelvet hoztak létre. Ezek közül már sokat nem használnak. Vannak közöttük elterjedtek, amelyeknek egyre újabb, a programozás szempontjából kényelmesebb verzióit hozzák létre, de új nyelvek is megjelennek.

Vannak univerzális nyelvek, amelyeket a tudomány és a technika számos különböző ágában felmerülő problémák megoldására alkalmaznak, de vannak specializált, korlátozott körből származó feladatok megoldására alkalmazhatók is.

A számítógép processzora csak a **gépi kódú** utasítások végrehajtására alkalmas. A gépi kód olyan kódolási forma, amelyben az utasítok binárisan kódoltak. A gépi kód egy adott processzortípusra jellemző, vagyis a különböző processzortípusok gépi kódjai eltérnek egymástól.

A nem gépi kódú programokat először le kell fordítani gépi kódra, csak ezután lehet azokat végrehajtani. A fordítást speciális programok – **compilerek (fordítóprogramok)** – végzik. Ezek ellenőrzik a program szövegét, kiszűrik a szintaktikai hibákat, majd lefordítják azt **gépi kódú** programra. Ez a kód a compiler üzemmódjának függvényében vagy a számítógép memóriájában maradnak, vagy a merevlemezre íródnak (például egy **exe** végrehajtható fájl formájában). A lefordított fájlt már végre tudja hajtani a processzor. Ezért a végrehajtható fájlt olyan számítógépen is futtathatjuk, ahol nincs fordítóprogram.

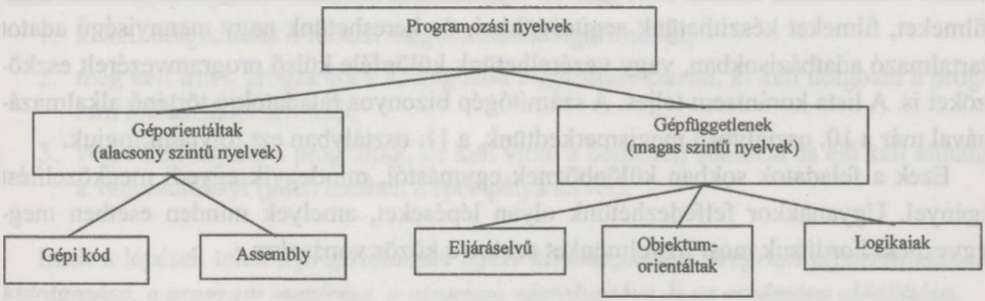
A nem gépi kódban írt programok futtatásának fent vázolt módja nem az egyetlen alkalmazott módszer.

Egyes programozási rendszerek más típusú speciális programokat **interpretereket (értelmező program)** tartalmaznak . Ezek a programok nem hoznak létre végrehajtható fájlokat, hanem utasításonként (soronként) elemzik és hajtják végre a fordítandó programokat. Ezért az értelmező programokat csak azokon a számítógépeken futtathatjuk, amelyeken telepítve van az interpreter.

A modern programozási rendszerek némelyike interpretert és compilert egyaránt tartalmaz. A végrehajtandó program előbb egy köztes (nem gépi) kódra compilálódik, majd ez a köztes kód interpretálódik.

A programozási nyelvek csoportosítása

A programozási nyelveknek különböző csoportosítása létezik. Az egyiket a 2.8. ábra szemlélteti.



2.8. ábra. A programozási nyelvek csoportosítása

Az **assembly** nyelvek a gépi kódhoz hasonlóan egy adott processzortípushoz tartoznak. A gépi kódtól eltérően, amely egy számhalmaz – az assembly nyelv utasításait speciális szavakkal, gépi utasítások neveivel írhatjuk le. Ez lényegesen leegyszerűsíti a program írását a gépi kódhoz képest.

A magas szintű – gépfüggetlen – nyelvek alkalmazása nem köthető egyetlen processzortípushoz.

Az **eljáráselvű programozási nyelvekben** a programokat részekre (eljárásokra) bontják, amelyek egy adott részfeladat megoldására szolgáló utasítás-sorozatok. Az eljárás meghívható más eljárásokból vagy a főprogramból. Az eljárásokból könyvtárakat hozhatunk létre, majd felhasználhatjuk azokat más programok írása során. Ilyen nyelv a **Fortran** (az angol *formula* translator – képletfordító szavakból), a **Basic** (Beginner's All-purpose Symbolic Instruction Code – általános célú programozási nyelv kezdőknek), a **Pascal** (az ismert XVIII. századi matematikus, Blaise Pascal tiszteletére nevezték el).

Az **objektum-orientált programozási nyelvek** alapja az objektum, amelynek tulajdonságai vannak, ezek a tulajdonságok meghatározott értékeket vehetnek fel, amelyeket a program futása során az objektumokhoz tartozó metódusok változtathatnak meg. Ilyen nyelvek a **Delphi**, a **C++**, a **C#**, a **Java.**

A **logikai programozási nyelveket** logikai feladatok megoldása során használják, továbbá nagy mennyiségű adat feldolgozása során, illetve szakértői rendszerekben. Logikai nyelv a **Prolog** (Programming in logic– logika programozása).

A **funkcionális nyelvekkel** a programozási feladatok megoldása függvények értékeinek kiszámítása által történik. Ilyen nyelv a **Lisp** (angolul a *List Processing* listafeldolgozást jelent). Ez a program alkalmas nagyméretű adathalmazok biztonságos és gyors kezelésére.

A modern programozási nyelvek nem mindig sorolhatók a fenti csoportokba, mivel gyakran több csoport jellemzőivel bírnak. Az objektum-orientált nyelvekben eljárásokat alkalmaznak, a funkcionális nyelvek objektum-orientált elemeket tartalmaznak.

A számítógépes feladatmegoldás lépései

Tudjuk, hogy a számítógépeket nagy mennyiségű számítás gyors és pontos elvégzése céljából alkották meg. A modern számítógépek nemcsak számítási feladatok elvégzésére alkalmasak, hanem szöveget és képeket szerkeszthetünk, prezentációkat, rajzfilmeket, filmeket készíthetünk segítségükkel, de kereshetünk nagy mennyiségű adatot tartalmazó adatbázisokban, vagy vezérelhetünk különféle külső programvezérelt eszközöket is. A lista korántsem teljes. A számítógép bizonyos feladatokra történő alkalmazásával már a 10. osztályban megismerkedtünk, a 11. osztályban ezt folytatni fogjuk.

Ezek a feladatok sokban különböznek egymástól, mindegyik egyedi megközelítést igényel. Ugyanakkor felfedezhetünk olyan lépéseket, amelyek minden esetben megegyeznek. Fordítsuk most figyelmünket ezekre a közös vonásokra.

1. feladat. Számítógép segítségével alkossuk meg egy sportpálya tervét.

A feladat megoldását azzal kell kezdenünk, hogy meghatározzuk a pálya méreteit, valamint azt, hogy milyen objektumokat kell azon elhelyezni (kosárlabda-, röplabda-, tenisz- és mini futballpályát, tornaszereket stb.). Ebbe a feladatkörbe tartozik még, hogy milyen formában kell ezt a projektet átadni, például műszaki rajz alakjában. Ez a lépés a ***feladat feltételeinek elemzése, a bemeneti és kimeneti adatok meghatározása.*** Röviden a feladat kitűzésének is szokás nevezni.

Ezután meg kell határozni, hogy az objektumok milyen tulajdonságait kívánjuk feltüntetni a terven, és ezek milyen értékeket vesznek fel. Ezek között az értékek között vannak szabványosak és vannak, amelyeket mi határozhatunk meg. A kosárlabdapalánk méretei, a gyűrű földszinttől mért magassága egy kosárlabdapálya esetében szabványos méret, ugyanakkor a bordásfal rúdjainak hosszát és számát tetszés szerint választhatjuk meg. Meg kell határozni az objektumok távolságát, meg kell határozni azokat a képleteket, amelyek segítségével a szükséges értékeket ki lehet számítani. Ez a lépés az ***információs modell felállítása.***

A következő lépésben ki kell választani a feladat megoldására alkalmas szoftvert. Jelen esetben ez lehet egy vektor- vagy egy pixelgrafikus program, vagy egy speciálisan műszaki rajzok készítésére kifejlesztett program. De akár saját magunk is fejleszthe-

tünk egy programot, amely az információs modell alapján előállítja a sportpálya néhány lehetséges tervét. Ez a lépés tehát a ***feladat megoldására alkalmas szoftvereszköz kiválasztása.***

Ha egy alkalmazói szoftvert választottunk, akkor azután:

1. meg kell határozni a szoftver azon eszközeit, amelyeket alkalmazni kívánunk;
2. a kiválasztott eszközök figyelembevételével ki kell dolgoznunk a feladat megoldásának algoritmusát;
3. végre kell hajtani a kidolgozott algoritmust, és elő kell állítanunk a sportpálya tervét.

Ezeket a lépéseket a következőképpen nevezik: az ***eszközök kiválasztása az alkalmazói szoftverben, az algoritmus kidolgozása,*** az ***algoritmus végrehajtása és az objektum számítógépes modelljének előállítása.***

Amennyiben saját program fejlesztése mellett döntöttünk, akkor először is ki kell választanunk egy programozási nyelvet. Ezután:

1. ki kell dolgoznunk a feladat megoldásának algoritmusát;
2. meg kell írni a programot a kidolgozott algoritmus szerint, ki kell dolgozni a program felhasználói felületét;
3. végre kell hajtani a programot, be kell vinni a bemeneti adatokat és elő kell állítani a végeredményt (jelen esetben a sportpálya tervét).

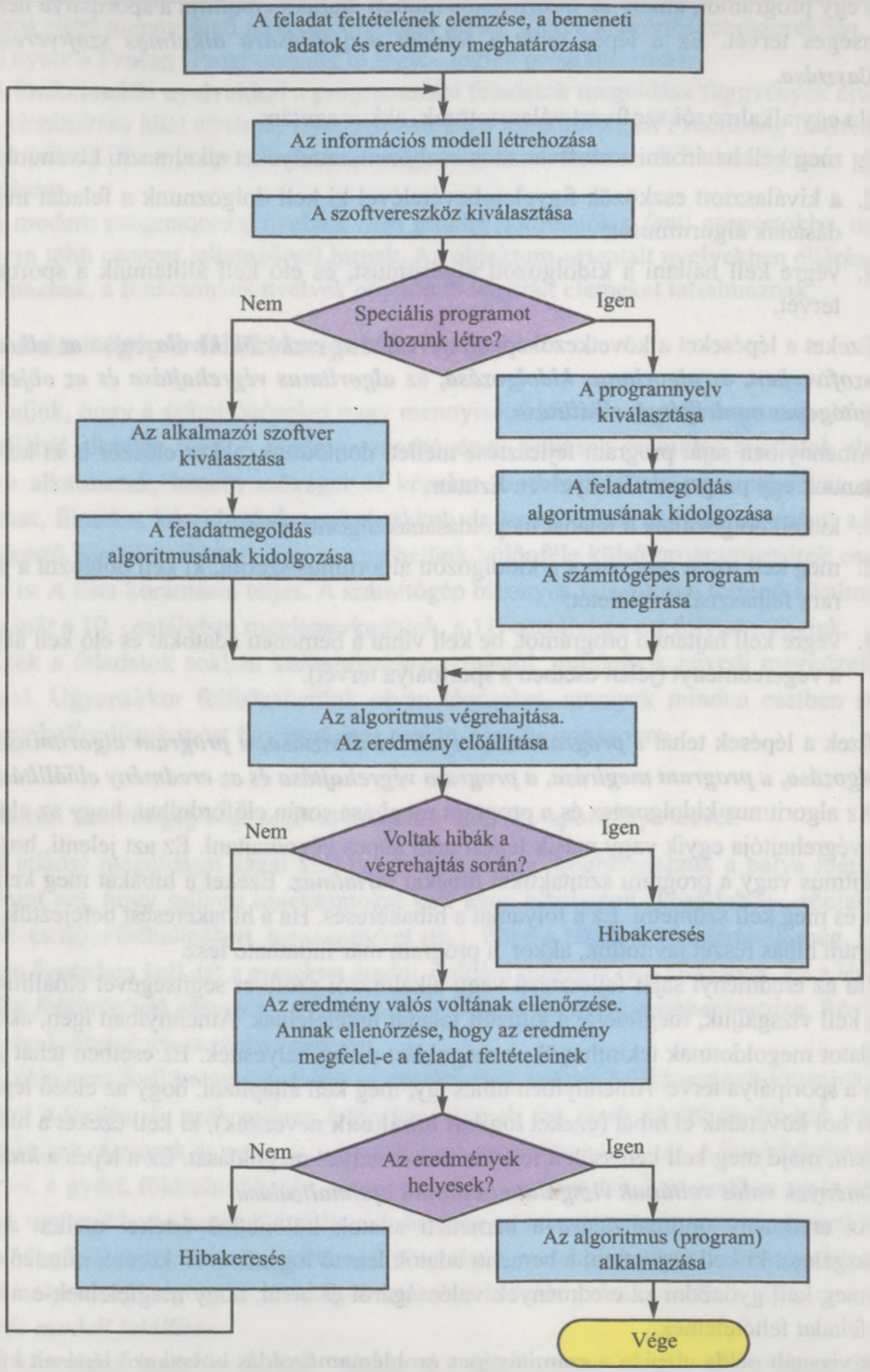
Ezek a lépések tehát a ***programozási nyelv kiválasztása, a program algoritmusának kidolgozása, a program megírása, a program végrehajtása és az eredmény előállítása.***

Az algoritmus kidolgozása és a program megírása során előfordulhat, hogy az algoritmus végrehajtója egyik vagy másik lépést nem képes végrehajtani. Ez azt jelenti, hogy az algoritmus vagy a program szintaktikai hibákat ***tartalmaz.*** Ezeket a hibákat meg kell keresni és meg kell szüntetni. Ez a folyamat a hibakeresés. Ha a hibakeresést befejeztük, és a program hibás részét javítottuk, akkor a program már futtatható lesz.

Ha az eredményt saját fejlesztésű vagy alkalmazói szoftver segítségével előállítottuk, meg kell vizsgáljuk, megfelel-e a kitűzött feladat feltételeinek. Amennyiben igen, akkor a feladatot megoldottnak tekinthetjük, a megoldást pedig helyesnek. Ez esetben tehát elkészült a sportpálya terve. Amennyiben nincs így, meg kell állapítani, hogy az előző lépések során hol követtünk el hibát (ezeket **logikai hibáknak** nevezzük), ki kell ezeket a hibákat javítani, majd meg kell keressük a feladat immár helyes megoldását. Ez a lépés a ***kutatási eredmények valós voltának vizsgálata és annak optimalizálása.***

Az eredmény optimalizálása a bemeneti adatok különböző értékei mellett zajlik. A vizsgálatot ki kell terjeszteni a bementi adatok lehető legszélesebb körére, minden esetben meg kell győződni az eredmények valósságáról és arról, hogy megfelelnek-e a kitűzött feladat feltételeinek.

A vizsgált példa alapján a számítógépes problémamegoldás következő lépéseit különböztethetjük meg (2.9. ábra):



2.9. ábra. A számítógépes feladatmegoldás lépései

Kövessük végig még egyszer a fent vázolt lépéseket egy másik feladat példáján.

2 feladat. A kereskedelmi lerakaton Ukrajnában gyártott és külföldre szállítandó termékeket tárolnak. Ismertek a termékek árai hrivnyában. Meg kell határozni a termékek árát dollárban, euróban és angol fontban!

I. lépés. A feladat elemzése, a bemeneti adatok és eredmények kiválasztása

Bemeneti adatok:

- a termékek megnevezése;
- a termékek ára hrivnyában;
- a dollár, az euró és az angol font árfolyama a hrivnyához képest.

Eredmények:

- a termékek ára dollárban, euróban és angol fontban.

II. lépés. Az információs modell létrehozása

Az adott feladat információs modellje nyilvánvalóan az a matematikai modell, amelynek segítségével a hrivnyában megadott árat dollárra, euróra és angol fontra átszámíthatjuk. A képletek a következők:

$$D = \frac{G}{KD},\ E = \frac{G}{KE},\ F = \frac{G}{KF},$$

ahol *D*, *E* és *F* a termékek ára dollárban, euróban és angol fontban, a *KD*, a *KE* és a *KF* pedig a megfelelő valuta árfolyama.

Az eredményeket századokra kerekítve kell meghatározni.

III. lépés. A szoftvereszköz kiválasztása

A feladat megoldására alkalmas lehet a számológép-program vagy a táblázatkezelő alkalmazása. A harmadik lehetőség saját program fejlesztése valamely programozási nyelven.

IV. lépés. Az algoritmus kidolgozása

Amennyiben táblázatkezelő alkalmazást, például az **Excel 2007**-et választjuk, az algoritmus a következő lehet:

1. Írjuk be az **A3** cellába a Dollárárfolyam szót!
2. Írjuk be az **A4** cellába az Euróárfolyam szót!
3. Írjuk be az **A5** cellába a Fontárfolyam szót!
4. Írjuk be a **B3** cellába a dollár árfolyamát!
5. Írjuk be a **B4** cellába az euro árfolyamát!
6. Írjuk be a **B5** cellába a font árfolyamát!
7. Írjuk be a **D2** cellába az Ár hrivnyában kifejezést!
8. Írjuk be a **E2** cellába az Ár dollárban kifejezést!
9. Írjuk be a **F2** cellába az Ár euróban kifejezést!
10. Írjuk be a **G2** cellába az Ár fontban kifejezést!
11. Írjuk be a **C** oszlop celláiba a **C3**-tól kezdődően a termékek nevét!
12. Írjuk be a **D** oszlop celláiba a **D3**-tól kezdődően a termékek árait hrivnyában!
13. Írjuk be az **E3** cellába az **=KEREKÍTÉS(D3/$B$3;2)** képletet!
14. Írjuk be az **F3** cellába az **=KEREKÍTÉS(D3/$B$3;2)** képletet!
15. Írjuk be a **G3** cellába az **=KEREKÍTÉS(D3/$B$5;2)** képletet!
16. Jelöljük ki az **E3:G3** tartományt!
17. Másoljuk be a kijelölt tartomány tartalmát az **E:G** oszlop celláiba!

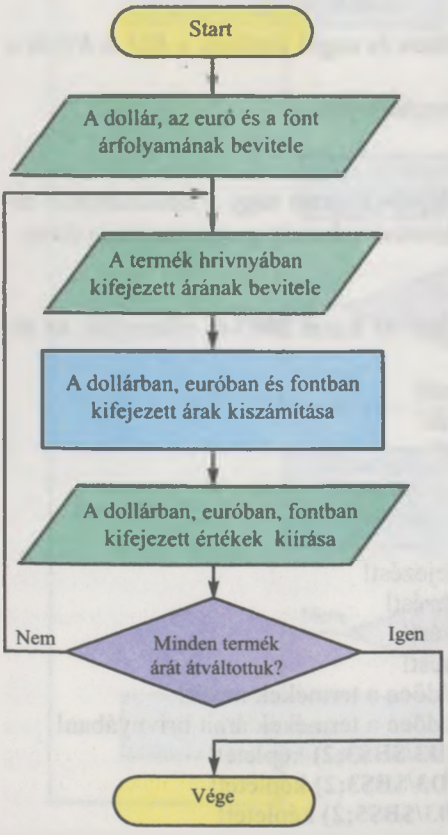
V. lépés. A számítógépes modell előállítása

2.2. táblázat. **A feladat megoldása Excel 2007 táblázatkezelővel**

	A	B	C	D	E	F	G
1							
2				Ár, hrivnyában	Ár dollárban	Ár euróban	Ár fontban
3	Dollárárfolyam	5,12		154,34	30,14	23,93	16,78
4	Euróárfolyam	6,45		346,3	67,64	53,69	37,64
5	Fontárfolyam	9,2		2234,6	436,45	346,45	242,89
6				638	124,61	98,91	69,35

VI. lépés. Az eredmények valós voltának és annak vizsgálata, megfelelnek-e azok a kitűzött feladatnak

Amikor megkaptuk az eredményeket, meg kell vizsgáljuk, hogy azok mennyire valósak. Ha például a dollár árfolyama 5,12, a dollárban kifejezett áraknak nagyjából 5-ször kisebbnek kell lenniük a hrivnyában megadottaknál. Leellenőrizhetünk néhány eredményt más számolóeszköz segítségével. Beírhatunk a táblázatba olyan ellenőrző értékeket, amelyek esetében ismert az eredmény, például 512 hrivnya = 100$. Hasonlóképpen járhatunk el a többi képlet esetében is.



2.10. ábra. A 2. feladat megoldásának folyamatábrája

Ha az ellenőrzés során hibákat találunk, meg kell állapítanunk azok okát, majd pedig ki kell javítani, végül újra végre kell hajtani a hibás lépést követő lépéseket.

Ha a III. lépésben a saját program mellett döntünk, akkor az utána következő lépések így módosulnának:

IV. lépés. A megoldás algoritmusának kidolgozása

A feladat más módon is megoldható. Például megtehetjük, hogy előbb bevisszük az összes árat hrivnyában, majd ezután végezzük el az átváltást.

V. lépés. A program megírása

Ha az algoritmussal elkészültünk, ki kell választani egy programozási nyelvet, meg kell határozni a felhasználói felületet, és meg kell írni a programot. Ezzel a lépéssel részleteiben is meg fogunk ismerkedni a későbbiekben.

VII. lépés. Hibakeresés. Az eredmények értékelése

Ha a program is elkészült, azt tesztelni kell a bemeneti adatok különböző értékei mellett. Ha nem várt eredményt kaptunk, az azt jelenti, hogy a programban hibák vannak. Ezeket meg kell keresni és ki kell javítani.

A tesztadatok kiválasztása nem mindig egyszerű, és eleget kell tegyen az alábbi követelményeknek:

1. A tesztadatoknak olyannak kell lenni, hogy segítségükkel könnyen ki lehessen számítani a végeredményeket, és össze lehessen hasonlítani a számítógépes program által kiszámított értékekkel.
2. A tesztadatoknak fel kell ölelniük az összes lehetőséget, amelyek a program használata során előfordulhatnak.

VIII. lépés. A program alkalmazása

Miután megbizonyosodtunk, hogy a program minden tesztadat esetén megfelelő eredményt ad, elkezdhetjük a program alkalmazását a feldolgozásra váró bemeneti adatokkal.

Ellenőrző kérdések

1°. Mi a számítógépes program?
2°. Mit értünk a program logikája és felhasználói felülete alatt?
3°. Melyik adatot tekintjük bemeneti adatnak? Mik a kimenti és a köztes adatok?
4°. Mi a programozási nyelv?
5°. Milyen részekből áll a programozási nyelv?
6°. Fogalmazzuk meg a nyelv komponenseinek értelmezését!
7°. Mi a szintaktikai hiba?
8°. Mi a gépi nyelv? Milyen alakban írhatók le a gépi nyelv utasításai?
9°. Mi a compiler? Ismertessük a compiler használatát!
10*. Mi az interpreter? Ismertessük az interpreter használatát!
11°. Ismertessük a programozási nyelvek csoportjait! Jellemezzük ezeket a csoportokat! Hozzunk fel példákat!
12°. Ismertessük a számítógépes feladatmegoldás lépéseit abban az esetben, ha felhasználói szoftver használata mellett döntöttünk!
13°. Ismertessük a számítógépes feladatmegoldás lépéseit abban az esetben, ha saját fejlesztésű program használata mellett döntöttünk!
14°. Jellemezzük a számítógépes feladatmegoldás lépéseit abban az esetben, ha felhasználói szoftver használata mellett döntöttünk!
15°. Jellemezzük a számítógépes feladatmegoldás lépéseit abban az esetben, ha saját program használata mellett döntöttünk!

Végezzük el a feladatokat!

1°. Derítsük ki, melyik évben hozták létre a **Basic, PL/1, Cobol, ALGOL, Fortran, Prolog, Lisp, Java, C, C++, C#, Smalltalk** programozási nyelveket! Jellemezzük röviden ezeket a nyelveket!
2*. Készítsünk rövid beszámolót a XX. század 50–60-as éveiben alkalmazott programozási nyelvekről.
3. Készítsünk rövid beszámolót a deklaratív programozásról!

4. Részletezzük a feladatmegoldás egyes lépéseit az alábbi feladatok esetében:
   a. ° Határozzuk meg az osztályterem padlójának lefestéséhez szükséges festék árát!
   b. • Határozzuk meg az informatikaterem kitapétázásához szükséges tapéta árát!
   c. • Az *A* és *B* pontokból két vonat indult el egymás felé. Ismerjük a pontok távolságát és a vonatok sebességét. Határozzuk meg a vonatok távolságát két óra elteltével; 3 óra elteltével! Mennyi idő múlva találkoznak a vonatok?
   d. • Az *A* és *B* pontokból két vonat indult el ellentétes irányokban. Ismerjük a pontok távolságát és a vonatok sebességét. Határozzuk meg a vonatok távolságát 1 óra elteltével; 3 óra elteltével!
   e. • Ismerjük egy háromszög két oldalát és a közbezárt szöget. Határozzuk meg a harmadik oldalt és a háromszög területét!
   f. • Ismerjük a paralelogramma két oldalát és a közbezárt szöget. Határozzuk meg az átlókat és a paralelogramma területét!

2.4. A Delphi programozási nyelv és a Turbo Delphi 2006 fejlesztőkörnyezet

1. Mi a számítógépes program?
2. Mi a programlogika és a felhasználói felület?
3. Mi a programozási nyelv? Mik egy programozási nyelv legfontosabb elemei? Mi ezeknek a rendeltetése?
4. Milyen lépésekből áll a számítógépes feladatmegoldás? Mi ezeknek a lényege?

A Delphi programozási nyelv és a Turbo Delphi 2006 integrált fejlesztői környezet

Hozzákezdünk a **Delphi** objektumorientált programozási nyelv tanulmányozásához. Nagyon sok tanuló és diák ezen a nyelven sajátítja el az objektumorientált programozás alapjait, de a programozók is gyakran választják összetett rendszerek létrehozása során.

A **Delphi** nyelven írt programok létrehozását, szerkesztését, tesztelését **Turbo Delphi 2006 integrált fejlesztői környezetben** fogjuk végezni. A rendszer szövegszerkesztőt, **Delphi compilert,** nyomkövetési, tájékoztatási és egyéb eszközöket tartalmaz. Az ilyen rendszereket integrált programozási rendszereknek **(integrált fejlesztőeszközöknek)** nevezik.

A rendszer számos beépített vizuális komponenst (VCL – Visual Componet Library) tartalmaz. Ezek a már ismert vezérlőelemek: gombok, mezők, számlálók, listák, választókapcsolók, görgetősávok stb. Amikor egy vizuális komponenst elhelyezünk a program ablakában, a programhoz automatikusan hozzáadódik a megfelelő programrészlet, így azt nem kell kézzel beírni, ezáltal egyszerűbbé válik a programozás folyamata. Ezt a rendszert **vizuálisnak** nevezzük.

Az integrált fejlesztési környezetben létrehozott programokat **projekteknek** nevezik. A projektek általában néhány fájlból állnak.

A **Delphi** programozási nyelv a **Pascal** továbbfejlesztéseként jött létre. A Pascal nyelv 1970-ben a Zürichi Műszaki Egyetem tanára, Nicolaus Wirth (szül. 1934-ben) vezetése alatt készült el.

A Pascal nyelvet oktatási céllal hozták létre. Később kiderült, hogy a nyelv használata annyira egyszerű, hogy gyakorlati célú programok fejlesztésére is használni kezdték. Így megírták az első compilert, majd létrehozták a **TurboPascal** fejlesztői környezetet, amely a későbbiek során nagy népszerűségre tett szert.

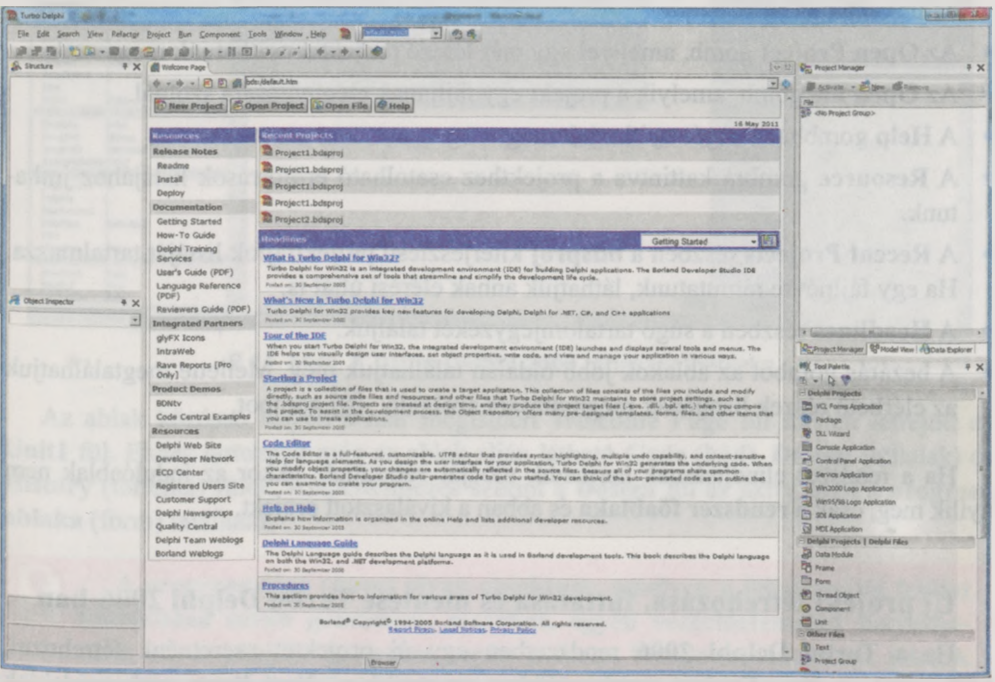
A későbbiekben a Pascal nyelvet továbbfejlesztették, tökéletesítették. A nyelv a 80-as években megjelent verziói objektumorientált eszközökkel gazdagodott. Így jelent meg az **Object Pascal.** Az Object Pascal 2000-es évek elején megjelent változata a **Delphi.**

2.11. ábra. Nicolaus Wirth

A Turbo Delphi 2006 indítóablaka

A **Turbo Delphi 2006** rendszert a szokott módokon indíthatjuk el:

- a ***Start*** ⇒ ***Programok*** ⇒ ***Borland Developer Studio 2006*** ⇒***Turbo Delpi*** paranccsal;
- a munkaasztalon a parancsikonra kattintva;
- duplán kattintva egy **bdsproj** kiterjesztésű fájlra.



2.12. ábra. A **Turbo Delphi 2006** kezdőablaka

Az első két esetben megnyílik a **Turbo Delphi 2006** kezdőablaka (2.12. ábra). Az ablak felső részében találjuk a menüsort és az eszköztárakat. Az ablak bal és jobb oldali részén a következő gyerekablakokat találjuk:

- **Structure** (szerkezet) ablakban láthatjuk a projekt részeit.
- **Object Inspector** (objektumtallózó vagy objektumfelügyelő) a kiválasztott objektum tulajdonságait tartalmazza.
- **Project Manager** (projektirányító) a projekt összetevő fájljait foglalja össze gráfba, azaz megjeleníti a projekt fájljainak szerkezetét.
- **Tool Palette** (eszköztár) az elérhető objektumok listáját tartalmazza.

A felsorolt ablakok mindegyike tartalmaz egy bezárás és egy minimalizálás (🗕) gombot. Minimalizálás gombra kattintva az ablak mérete minimális lesz, a gomb külalakja pedig megváltozik (🗗). Ha ezeket az ablakokat bezárjuk, a **View** menü segítségével nyithatjuk meg újra.

Előfordulhat, hogy a **Turbo Delphi 2006**-ban más gyerekablakok lesznek megnyitva. Ez az adott ablak beállításától függ. A további munkánkat megkönnyítendő, ez a mód **Default Leyout** (alapértelmezett elrendezés) vagy **Classic Undocked** (klasszikus dokkolatlan) kell legyen.

Az ablak középső részét a **Welcome Page** (Üdvözlő oldal) foglalja el. Ezen a következő elemeket találjuk:

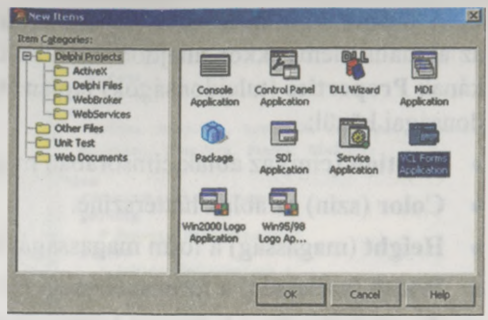
- A **New Project** gombot, melynek segítségével értelemszerűen egy új projektet hozhatunk létre.
- Az **Open Project** gomb, amellyel egy már létező projektet nyithatunk meg.
- Az **Open file** gomb, amelyik a projekt egy fájljának megnyitására szolgál.
- A **Help** gombbal a súgót nyithatjuk meg.
- A **Resource** gombra kattintva a projekthez csatolható erőforrások listájához juthatunk.
- A **Recent Projects** részben a **bdsproj** kiterjesztésű projektfájlok listáját tartalmazza. Ha egy fájlnévre rámutatunk, láthatjuk annak elérési útját is.
- A **Headlines** részben a súgó tartalomjegyzékét találjuk.
- A bezárás gombot az ablakok jobb oldalán találhatjuk meg. Mellette megtalálhatjuk az elérhető részek listájának megnyitására szolgáló ⊡ gombot.

Ha a rendszer elindításnak harmadik módját választjuk, akkor az indulóablak nem nyílik meg, csak a **rendszer főablaka** és abban a kiválasztott projekt.

Új projekt létrehozása, futtatása és mentése Turbo Delphi 2006-ban

Ha a **Turbo Delphi 2006** rendszerben egy új projektet szeretnénk létrehozni, kattintsunk az **New Project** gombra. Ekkor megnyílik az **New Items** párbeszédablak (2.13. ábra).

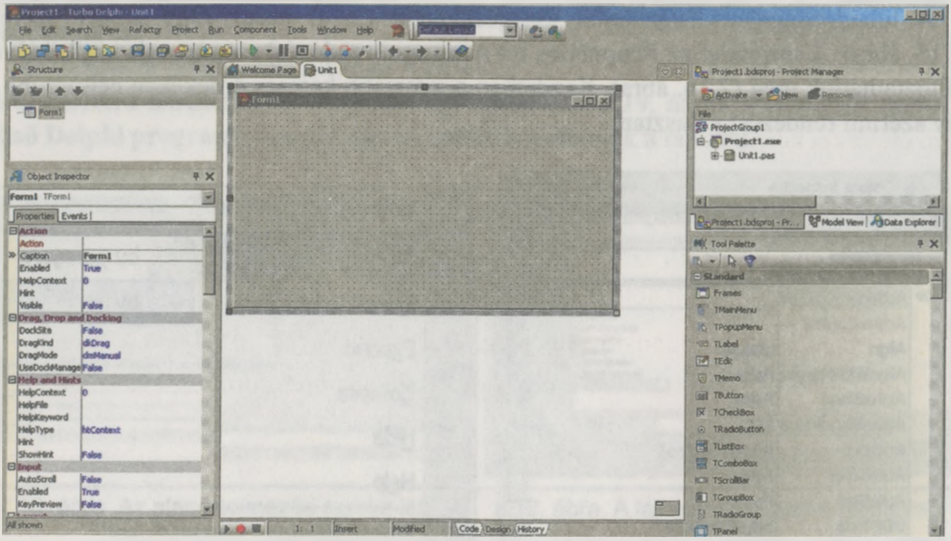
A **Turbo Delphi 2006** rendszerben különböző projekt-típusokat hozhatunk létre. Mi olyan projekteket szeretnénk létrehozni, amely ablakokat, gombokat, beviteli mezőket (vezérlőelemeket) tartalmaz. Az ilyen projekteket a **Turbio Delphi 2006**-ban **VCL Forms Application** (Űrlapos (ablakos) alkalmazás). Ahhoz, hogy ilyen projektet hozzunk létre a **New Items** ablakban a következőképpen kell eljárnunk:



2.13. ábra. A **New Items** párbeszédablak

1. Az ablak bal oldalán válasszuk ki a **Delphi Projects** elemet.
2. Az ablak jobb oldalában kattintsunk a **VCL Forms Application** ikonra.
3. Kattintsunk az **OK** gombra.

Ezután megnyílik a **Turbo Delphi 2006** főablaka (2.14. ábra).



2.14. ábra. A **Turbo Delphi 2006** rendszer főablaka

Az ablak középső részén a már megismert **Welcome Page** fül mellett létrejött az **Unit1** fül. Ehhez tartoznak még az ablak alján látható **Code** (kód), **Design** (külalak) és **History** (történet) fülek. Alapértelmezés szerint a **Design** fül az aktív, ahol a **program ablaka** (form) található.

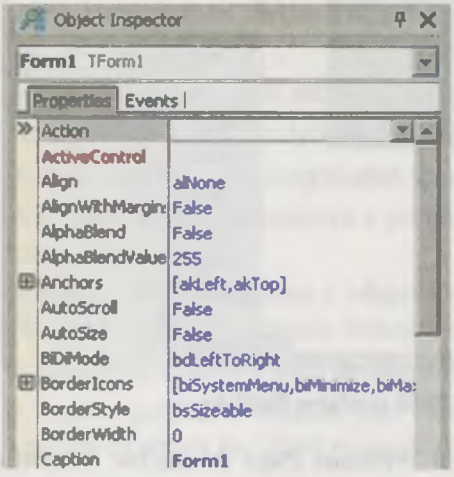
A *programablak (form)* olyan objektum, amelyen a felhasználói felület kialakítása során gombokat, mezőket, egyéb vezérlőelemeket helyezhetünk el. A projekt futtatása során a form Windows ablakként jelenik majd meg a rajta elhelyezett elemekkel, hacsak utóbbiak nem rejtettek vagy láthatatlanok.

A form-nak, mint minden objektumnak, vannak tulajdonságai. Amennyiben a form az aktuális elem, akkor tulajdonságait az **Object inspector (objektumfelügyelő)** ablakának **Properties** (tulajdonságok) lapján láthatjuk. Nézzünk meg néhányat a form tulajdonságai közül:

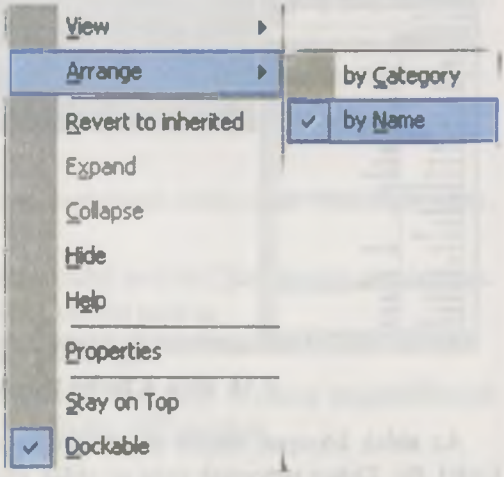
- **Caption** (cím) az ablak címsorában megjelenő felirat.
- **Color** (szín) az ablak háttérszíne.
- **Height** (magasság) a form magasságát állíthatjuk be pixelekben.
- **Width** (szélesség) a form szélességi méretét lehet beállítani pixelekben.
- **Top** (felső) a form helyzetét állíthatjuk be a képernyő tetejétől (pixelekben mérve).
- **Left** (bal) a form helyzetét állíthatjuk be a képernyő bal oldalától (pixelekben mérve).
- **Name** (név) ez a form-nak, mint objektumnak a neve.

A tulajdonságok döntő többségének van alapértelmezett értéke. A **Name** és a **Caption** értéke például *Form1*, a **Height**-é *338*, a **Width**-é *651*. A **Properties** fülön e tulajdonságok értékeket megváltoztathatjuk.

A tulajdonságokat listázhatjuk kategóriánként (2.14. ábra) vagy a név szerint (2.15. ábra). A sorrendet az **Properties** fül **Arrange** (rendezés) menüpontja segítségével változtathatjuk meg (2.16. ábra). Kezdetben a gyorsabb keresés érdekében célszerűbb a név szerinti rendezést választani.



2.15. ábra. Az **Object Inspector** ablak **Properties** lapja



2.16. ábra. A **Properties** lap helyi menüje

A **Code fülre** kattintva megnézhetjük a program szövegének azt a részét, amely a **Turbo Delphi 2006**-ban az új projekt létrejöttekor automatikusan keletkezik (2. 17. ábra).

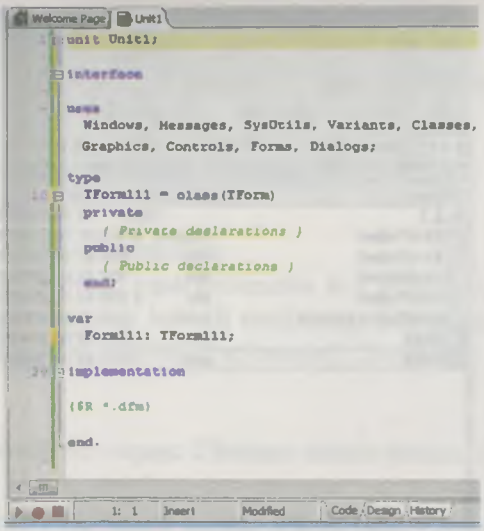
Ez a programrészlet elégséges ahhoz, hogy a projektünket lefuttassuk. Ezt **Run** (végrehatás, futtatás) paranccsal, vagy az eszköztár **Run** gombjára kattintva tehetjük meg.

Ha az alapértelmezés szerint létrehozott projektet futtatjuk, a képernyőn egy üres ablak jelenik majd meg, amelynek elhelyezése, méretei, színei, a címsor szövege megfelelnek az alapértelmezés szerinti beállításoknak (2.18. ábra).

A projekt futását a bezárás gombbal szakíthatjuk meg

A **Turbo Delphi 2006 Structure** ablakában a projekthez tarozó egyetlen objektumot láthatjuk, a Form1 ablakot. A **Tool Palette** ablakban az elérhető objektumokat, a **Project Manager** (2.14. ábra) ablakban pedig a projekt fájljainak listáját találjuk.

A projekt összes fájlját a **File** menü **Save All** (mindent ment) pontjára, vagy az eszköztár **Save All** gombjára kattintva menthetjük el.



2.17. ábra. A **Code** lap az alapértelmezés szerinti programkóddal

A megnyíló párbeszédablakban (2.19. ábra) célszerű létrehozni a projektnek egy mappát (a 2.19. ábrán a **Перший проект** (**Első Delphi program**) mappa), ahová aztán elmenthetjük a fájlokat.

2.18. ábra. Az alapértelmezés szerint létrehozott projekt futtatása

2.19. ábra. A fájl mentése párbeszédablak

A **Turbo Delphi 2006** két fájl mentését ajánlja fel:

- Az **Unit1** nevű **pas** kiterjesztésű fájlt.
- A **Project1** nevű **bdsproj** kiterjesztésű fájlt.

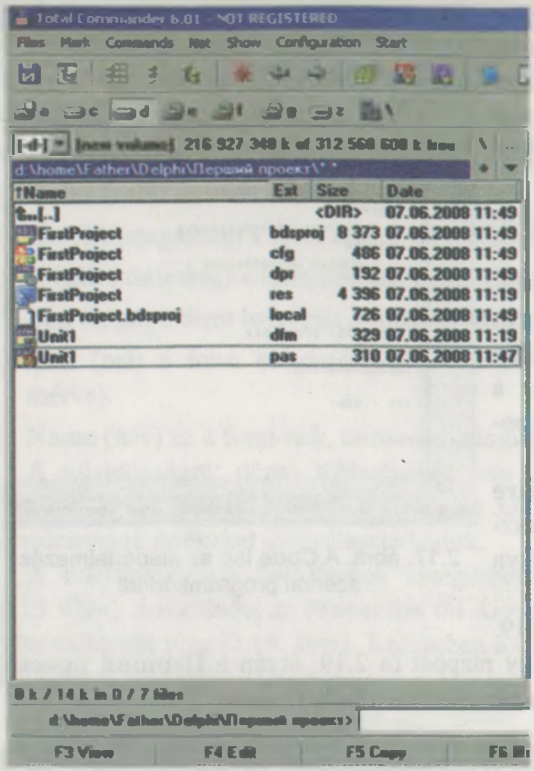
A fájlok neveit megváltoztathatjuk, a kiterjesztést viszont nem.

A fájlok neve csak latin betűket, számokat és alulvonás jelet tartalmazhat és betűvel kell kezdődjön.

A **Turbo Delphi 2006** rendszerben egy projekthez a fenti két fájlon kívül még néhány fájl tartozik, amelyeket a rendszer automatikusan elment (2.20. ábra).

Ha be szeretnénk fejezni a munkát, kattintsunk a **File** menü **Close All** (mindent bezár) gombjára.

A rendszerben egyidejűleg több projekttel is dolgozhatunk.



2.20.рес. Проект файлларнен Бириндзи проект

A projekt szerkesztése

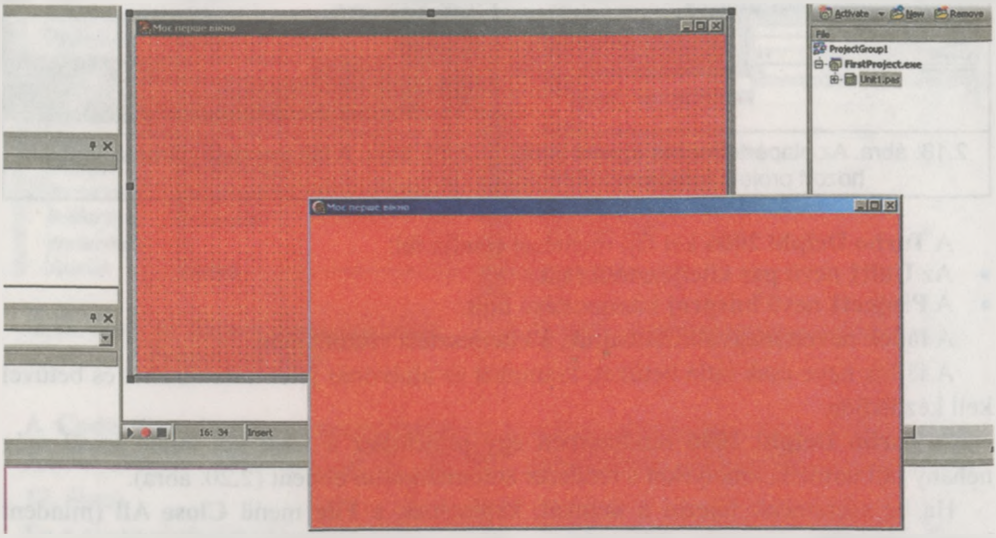
Az elkezdett projektet természetesen meg lehet változtatni. Megváltoztathatjuk például a form tulajdonságait az **Object Inspector** ablak **Properties** fülén.

Ha a tulajdonságok értékeit az alábbiakra változtatjuk:

- **Caption** = *Моє перше вікно* (Első ablakom)
- **Color** = *clRed* (***colour*** – szín, ***red*** – piros)
- **Height** = *450*
- **Width** = *700*
- **Left** = *600*
- **Top** = *350*

majd ha újra lefuttatjuk a projektet, akkor egy megváltozott méretű, helyzetű, színű és feliratú ablakot kapunk (2.21. ábra).

A **Height** és **Width** értékeit az ablak keretének elhúzásával is megváltoztathatjuk.



2.21. ábra. A megváltoztatott projekt ablaka

Vezérlőelemek elhelyezése a form-on

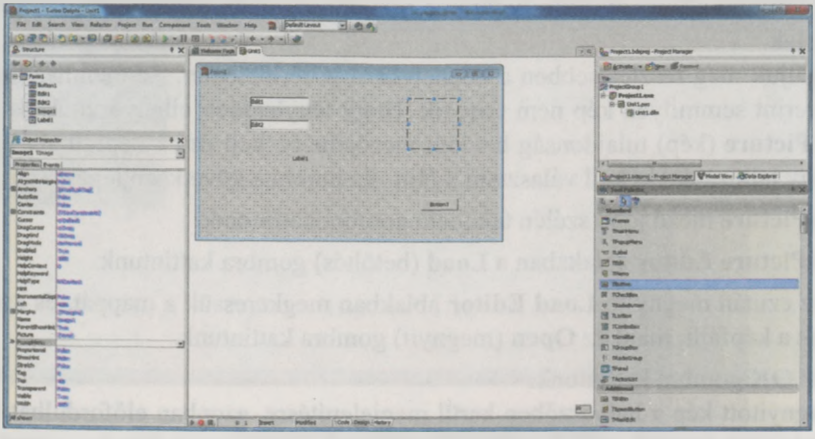
Az eddigiek során megismert programokat alkalmazva különböző vezérlőelemeket használtunk:

- gombokat – bizonyos műveletek elvégzésére;
- beviteli mezőket – adatok bevitelére;
- feliratokat – szöveg és számok kiírására;
- jelölőket és kapcsolókat – az üzemmódok megváltoztatására.

Hogy kialakítsuk a program felhasználói felületét, vezérlőelemeket kell elhelyezni a form-on. Helyezzünk el az ablakban egy gombot, egy beviteli mezőt, egy feliratot és egy képhelyőrzőt (ez utóbbi segítségével helyezhetünk el képet a form-on) . E célból:

1. Tegyük a form-ot aktívvá.
2. Keressük meg a **Tool Palette**-en:
   - a gomb elhelyezése céljából a **Standard** csoport **TButton** elemét (button = gomb);
   - a beviteli mező elhelyezése céljából a **Standard** csoport **TEdit** (edit = szerkesztés) elemét;
   - a felirat elhelyezése céljából a **Standard** csoport **TLabel** (label = felirat) elemét;
   - a képhelyőrző elhelyezése céljából az **Additional** (kiegészítő) csoport **TImage** (image = kép) elemét.
3. Majd kattintsunk duplán a megfelelő elemre.

E művelet eredményeképpen a kiválasztott elemek a form közepén jelennek meg . A vezérlőket áthúzással helyezzük el a form megfelelő helyére (2.22. ábra). Amikor új elemet helyezünk a form-ra, a **Structure** ablakban megjelenik a megfelelő vezérlő jele és neve, az **Object Inspector Properties** fülén megjelennek az elemek tulajdonságai és azok alapértelmezett értékei, az **Object Inspector** ablakban a fülek fölött pedig az elemek nevei. Ez a lista a form-on elhelyezett összes elem nevét tartalmazza. A lista egyik elemének kiválasztásával aktívvá tehetjük azt.



2.22. ábra. A form a rajta elhelyezett vezérlőelemekkel

A form-hoz hasonlóan a gomb, a mező, a felirat és a képhelyőrző szintén sok egyéb tulajdonság mellett rendelkezik a már megismert **Name, Height, Width, Left, Top** tulajdonságokkal. A form-on elhelyezett elemek tulajdonságainak ez az alapértelmezett értéke. Ezeket a már ismert módon, az **Object Inspector** ablak **Properties** fülén változtathatjuk meg. Egyes tulajdonságokat áthúzással is megváltoztathatunk.

Figyeljünk a következőkre:

1. A form-mal szemben a vezérlőelemek esetében a **Left** és **Top** tulajdonság értékei nem a képernyő felső és bal oldali széléhez képest értelmezendő, hanem ahhoz az ablakhoz képest, amelyeken elhelyezkednek.
2. A mező és felirat rendelkezik **Color** tulajdonsággal, a gomb és a képhelyőrző nem.
3. A gomb és a felirat rendelkezik **Caption** tulajdonsággal, a mező és a képhelyőrző nem.
4. A mezőnek **Caption** (felirat) helyett **Text** (szöveg) tulajdonsága van, amelynek értéke a beviteli mezőbe beírt szöveg.

Az imént a form-on elhelyezett mindegyik vezérlőelem rendelkezik a következő két tulajdonsággal:

- **Enabled** (engedélyezett). Ennek értéke meghatározza, hogy az elem szerkeszthető-e futtatás közben. Két állása van: *True* (igaz), ekkor az elem szerkeszthető, illetve *False* (hamis), ekkor pedig nem.
- **Visible** (látható). Meghatározza, látható-e a vezérlőelem a program futása közben. A *True* beállítása azt jelenti, hogy az elem látható, ha *False*, akkor nem látható.

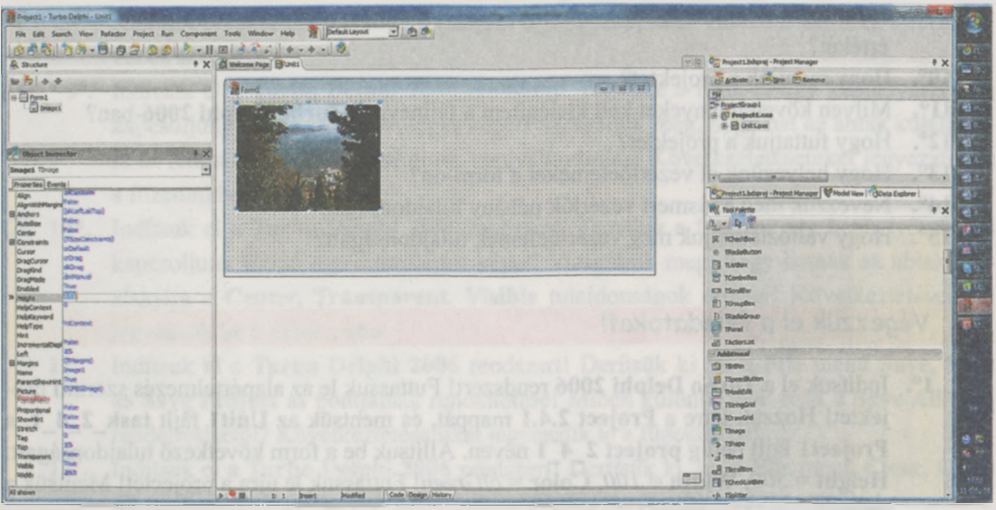
A képhelyőrző kivételével valamennyi vizsgált vezérlőelem rendelkezik a **Font** (betűtípus) tulajdonsággal, ami a vezérlő feliratának vagy tartalmának betűtípusa. Ez egy **komplex** tulajdonság, amely karaktertulajdonságok sokaságát öleli fel, mint például **Color, Height, Name, Size.** Ezeket a tulajdonságokat úgy érhetjük el, hogy a **Font** tulajdonság bal oldalán található ⊞ jelre kattintunk. Ha ezután a hármaspont gombra kattintunk, megnyílik a betűtípus tulajdonságainak beállítására szolgáló párbeszédablak.

Vizsgáljuk meg részletesebben a képhelyőrző vezérlőelemet. Az elemhez alapértelmezés szerint semmilyen kép nem tartozik. Hogy ténylegesen elhelyezzünk benne egy képet, a **Picture** (kép) tulajdonság beviteli mezőjébe be kell írni a képfájl elérési útvonalát, vagy tallózással ki kell választani a fájlt. Ez utóbbi a következő lépésekből áll:

1. A **Picture** mező jobb szélén található gombra kattintunk.
2. A **Picture Editor** ablakában a **Load** (betöltés) gombra kattintunk.
3. Az ezután megnyíló **Load Editor** ablakban megkeressük a mappát, és kiválasztjuk a képfájlt, majd az **Open** (megnyit) gombra kattintunk.
4. Az **OK** gombra kattintunk.

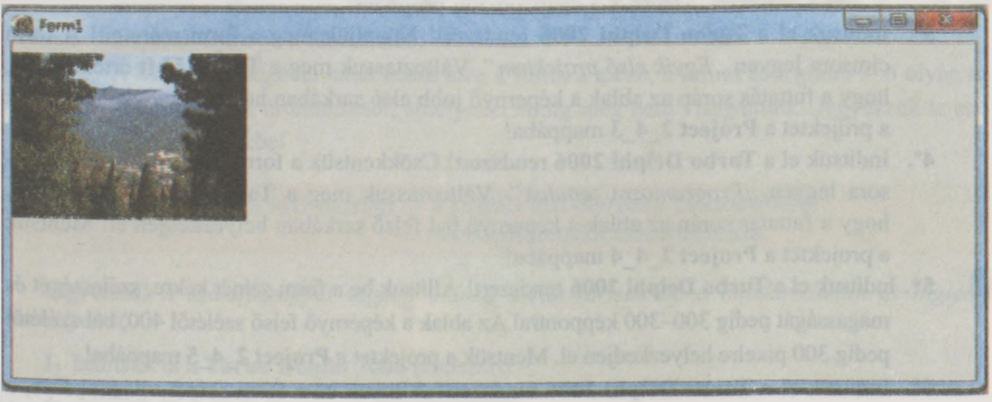
A megnyitott kép a helyőrzőben kerül megjelenítésre, azonban előfordulhat, hogy a kép mérete nem egyezik meg a helyőrző méretével, ezért csak a kép egy részét láthatjuk. Ezért célszerű lesz a helyőrző következő tulajdonságait beállítani (2.23. ábra):

- A **Stretch** (kinyújtás, kiterjesztés) értékét állítsuk *True*-ra, hogy a kép a helyörző teljes területét elfoglalja.
- A **Proportional** (arányos) tulajdonság értékét szintén állítsuk *True*-ra, hogy a kép a nyújtás során ne torzuljon.
- Az **Align** (igazítás) értékét állítsuk *alCustom*-ra (custom = szokásos), ez lehetővé teszi a kép igazítását (ha ezt az értéket *alClient*-re (kliens) állítjuk, a kép a form teljes felületét takarja majd).
- Állítsuk be a **Top, Left, Width, Height** értékeit, hogy a kép a form megfelelő helyén és helyes méretben jelenjen meg.



2.23. ábra. A form-on elhelyezett kép és annak néhány megváltoztatott tulajdonsága

A 2.24. ábrán a projektünk ablakát láthatjuk futás közben.



2.24. ábra. Képet tartalmazó ablak

Ellenőrző kérdések

1°. Nevezzük meg a **Turbo Delphi 2006** integrált fejlesztőkörnyezet alkotóelemeit!
2°. Mi a vizuális komponenskönyvtár? Mi a rendeltetése?
3°. Hogy indíthatjuk a **Turbo Delphi 2006**-ot?
4°. Milyen gyerekablakai vannak a **Turbo Delphi 2006** indítóablakának?
5°. Mi a gyerekablakok rendeltetése?
6°. Hogy hozunk létre új projektet **Turbo Delphi 2006**-ban?
7°. Mi a form?
8°. Nevezzük meg a form néhány tulajdonságát!
9°. Mi a form tulajdonságainak értelme? Hogyan változtathatjuk a form tulajdonságainak értékét?
10°. Hogy mentjük a projektet?
11°. Milyen követelményeket kell kielégíteni a fájlnévnek **Turbo Delphi 2006**-ban?
12°. Hogy futtatjuk a projektet?
13°. Hogy helyezünk el vezérlőelemeket a form-on?
14°. Nevezzük meg az ismert vezérlők néhány tulajdonságát!
15°. Hogy változtathatjuk meg vezérlőelemek tulajdonságait?

Végezzük el a feladatokat!

1°. Indítsuk el a **Turbo Delphi 2006** rendszert! Futtassuk le az alapértelmezés szerinti projektet! Hozzuk létre a **Project 2.4.1** mappát, és mentsük az **Unit1** fájlt **task_2_4_1**, a **Project1** fájlt pedig **project 2_4_1** néven. Állítsuk be a form következő tulajdonságait: **Height** = *500*, **Width** = *100*, **Color** = *clGreen*! Futtassuk le újra a projectet! Mentsük a megváltozott projektet!

2°. Indítsuk el a **Turbo Delphi 2006** rendszert! Futtassuk le az alapértelmezés szerinti projektet! Hozzuk létre a **Projekt 2.4.2** mappát, és mentsük az **Unit1** fájlt **task_2_4_2**, a **Project1** fájlt pedig **project 2_4_2** néven! Állítsuk be a form következő tulajdonságait: **Top** = 50 és **Left** = 100, és a címsorát **Caption** = *„Első saját projektem”*. Futtassuk le újra a projektet! Mentsük a megváltozott projektet!

3°. Indítsuk el a **Turbo Delphi 2006** rendszert! Növeljük meg a form méreteit! A form címsora legyen *„Egyik első projektem”*. Változtassuk meg a **Top** és **Left** értékeit úgy, hogy a futtatás során az ablak a képernyő jobb alsó sarkában helyezkedjen el! Mentsük a projektet a **Project 2_4_3** mappába!

4°. Indítsuk el a **Turbo Delphi 2006** rendszert! Csökkentsük a form méreteit! A form címsora legyen *„Programozni tanulok”*. Változtassuk meg a **Top** és **Left** értékeit úgy, hogy a futtatás során az ablak a képernyő bal felső sarkában helyezkedjen el! Mentsük a projektet a **Project 2_4_4** mappába!

5*. Indítsuk el a **Turbo Delphi 2006** rendszert! Állítsuk be a form színét kékre, szélességét és magasságát pedig 300–300 képpontra! Az ablak a képernyő felső szélétől 400, bal szélétől pedig 300 pixelre helyezkedjen el. Mentsük a projektet a **Project 2_4_5** mappába!

6*. Indítsuk el a **Turbo Delphi 2006** rendszert! Állítsuk be a form színét világoskékre, a szélességét és magasságát pedig 300, illetve 200 képpontra! Az ablak a képernyő felső szélétől 100, bal szélétől pedig 150 pixelre helyezkedjen el. Mentsük a projektet a **Project 2_4_6** mappába!

7*. Indítsuk el a **Turbo Delphi 2006** rendszert!Tegyünk a form-ra egy 250 pont magas 200 pont széles képhelyőrzőt, amely a form felső szélétől 20, a bal szélétől 30 képpontnyira helyezkedik el! Kapcsoljuk a helyőrzőhöz a **pic2.4.7** képfájlt! Módosítsuk a kép beállításait úgy, hogy az a helyőrzőt teljes egészében kitöltse! Futtassuk le a projektet, és mentsük el a **Project 2_4_7** mappába!

8*. Indítsuk el a **Turbo Delphi 2006** rendszert! Tegyünk a form-ra egy 50 pont magas 100 pont széles képhelyőrzőt, amely a form felső szélétől 50, a bal szélétől 80 képpontnyira helyezkedik el! Kapcsoljuk a helyőrzőhöz a saját fényképünket tartalmazó képfájlt! Módosítsuk a kép beállításait úgy, hogy az a helyőrzőt teljes egészében kitöltse! Futtassuk le a projektet! Változtassuk meg a kép tulajdonságait úgy, hogy az a form teljes felületét betöltse! Mentsük el a **Project 2_4_8** mappába! Változtassuk meg a form nevét és színét!

9*. Indítsuk el a **Turbo Delphi 2006** rendszert! Tegyünk a form-ra egy képhelyőrzőt, és kapcsoljunk hozzá egy tetszőleges képet! Vizsgáljuk meg, hogy hat az ablak külalakjára az **Align** tulajdonság értékeinek megváltoztatása! Következtetéseinket jegyezzük be a füzetünkbe!

10*. Indítsuk el a **Turbo Delphi 2006** rendszert! Tegyünk a form-ra egy képhelyőrzőt, és kapcsoljunk hozzá egy tetszőleges képet! Vizsgáljuk meg, hogy hatnak az ablak külalakjára a **Center, Transparent, Visible** tulajdonságok értékei! Következtetéseinket jegyezzük be a füzetünkbe!

11*. Indítsuk el a **Turbo Delphi 2006** rendszert! Derítsük ki mi a **File** menü **Save, Save as Save Project** as pontjainak rendeltetése! Miben különböznek ezek a **Save All** pont rendeltetésétől? Következtetéseinket jegyezzük le a füzetünkbe!

12*. Indítsuk el a **Turbo Delphi 2006** rendszert! Derítsük ki, mi a **File** menü **Close, Close all** pontjainak rendeltetése! Következtetéseinket jegyezzük be a füzetünkbe!

13*. Nyissunk meg egy projektet, például a **Project 2_4_13**-at! Határozzuk meg a form 5, valamint a rajta található gomb 4 tulajdonságának értékeit! Jegyezzük le ezeket a füzetünkbe!

14*. Nyissunk meg egy projektet, például a **Project 2_4_14**-at! Határozzuk meg a beviteli mező és a felirat tulajdonságait! A tulajdonságok értékeit összehasonlító táblázat formájában írjuk be a füzetünkbe!

15*. A **Súgó** segítségével határozzuk meg a form, a mező, a felirat és a gomb 5–6 olyan tulajdonságának értelmezését, amelyeket eddig még nem vizsgáltunk! Jegyezzük le ezeket a füzetünkbe!

5. számú gyakorlati munka — Projekt létrehozása és futtatása. Vezérlőelemek alkalmazása

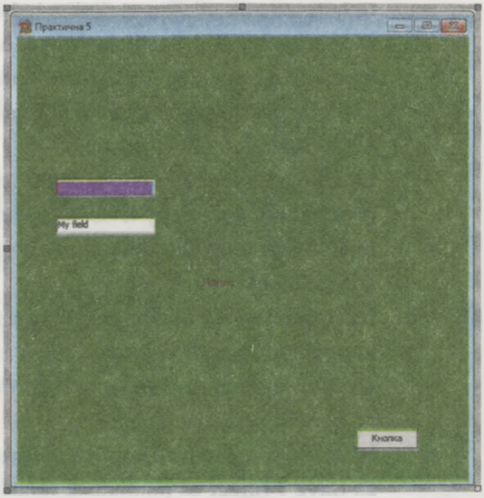
Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert!
2. Állítsuk be a form következő tulajdonságait:
   - **Caption** = *Практична 5*
   - **Color** = *clGreen (zöld)*

- **Height** = *580*
- **Width** = *580*
- **Left** = *500*
- **Top** = *150*

3. Helyezzünk el a form-on két beviteli mezőt, egy feliratot és egy gombot (2.25. ábra)!
4. Helyezzünk el a form-on egy képhelyőrzőt, és állítsuk be a következő tulajdonságokat:
   - **Stretch** = *True*
   - **Align** = *alCustom*
   - **Left** = *10*
   - **Top** = *30*
   - **Height** = *150*
   - **Width** = *130*

   Kapcsoljunk a helyőrzőhöz egy képet, például a **5.4.gyakorlati.jpg**-t! Állítsuk be a kép tulajdonságait úgy, hogy az a helyőrző teljes felületét elfoglalja!
5. Helyezzünk el a formon még egy képhelyőrzőt! Állítsuk be a tulajdonságait úgy, hogy mindkét mérete 60 pixel legyen, és a form jobb felső sarkában helyezkedjen el! Kapcsoljuk hozzá az **5.5.gyakorlati.jpg** képet!
6. Hozzuk létre **5.gyakorlati** nevű mappát, és mentsük el bele a projektet!
7. Futtassuk a projektet!



2.25. ábra

2.5. Események és azok feldolgozása

1. Mi a **TurboDelphi 2006** ablakainak rendeltetése?
2. Mi a form? Nevezzük meg néhány tulajdonságát! Hogy lehet ezeket a tulajdonságokat megváltoztatni?
3. Milyen közös tulajdonságokkal rendelkeznek a gomb, a beviteli mező, a felirat és a képhelyőrző? Milyen eltérő tulajdonságai vannak ezeknek az objektumoknak?
4. Mik a képlet beírásának szabályai az **Excel**-ben?
5. Mi az algoritmus? Mi a szekvenciális algoritmus? Milyen a szekvenciális algoritmus folyamatábrájának külalakja?

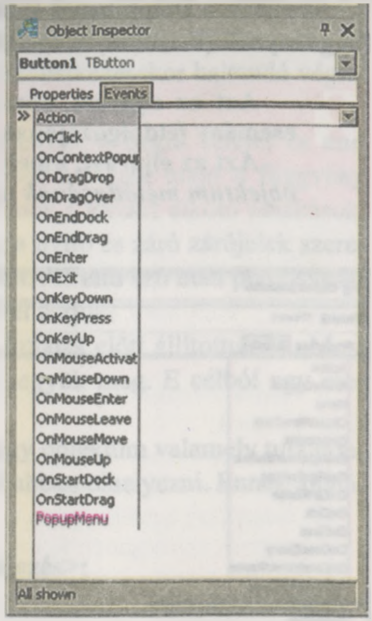
Események és azok feldolgozása

A felhasználói szoftverek alkalmazása során megszokhattuk, hogy egy gomb lenyomása feltétlenül kivált valamilyen eseményt: megnyílik vagy bezáródik egy ablak,

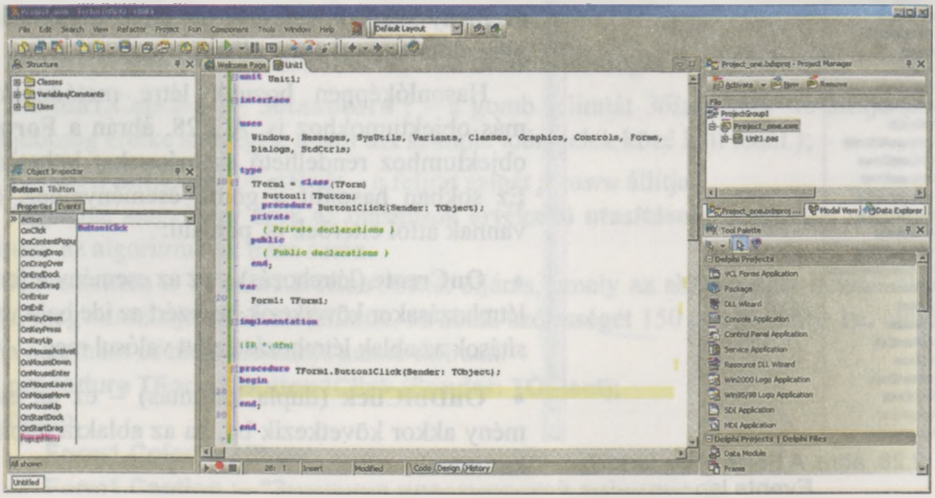
megváltoznak egyes objektumok tulajdonságai, az objektumok elmozdulnak a képernyőn stb. Ha azonban az eddig létrehozott projektekben rákattintunk egy gombra, semmi sem történik.

Ennek oka az, hogy a gombok elhelyezése nem elég ahhoz, hogy azokra rákattintva bármi is történjen. A gomb lenyomása egy esemény és a programmal „közölni" kell, hogy ekkor minek kell történnie, vagyis hogy milyen újabb eseményt kell kiváltson a kattintás.

Ha kijelölünk egy gombot, az **Object Inspector** ablak **Events** (események) fülén megadhatjuk, milyen eseményt kell, hogy kiváltson, ha a gombra kattintunk. Ehhez kétszer kell kattintsunk az **OnClick** felirat melletti mezőre. Ennek eredményeképpen az **OnClick** melletti mezőben megjelenik a **Button1Click** (a Button1 a gomb neve) szöveg, megnyílik a **Code** fül és a program szövegébe beíródik a **TForm1.Button1Click** eljárás sablonja, ahová majd be kell írni, mi is történjen a gombra kattintás következtében (2.27. ábra).



2.26. ábra. A **Button1** gombhoz tartozó **Events** lap



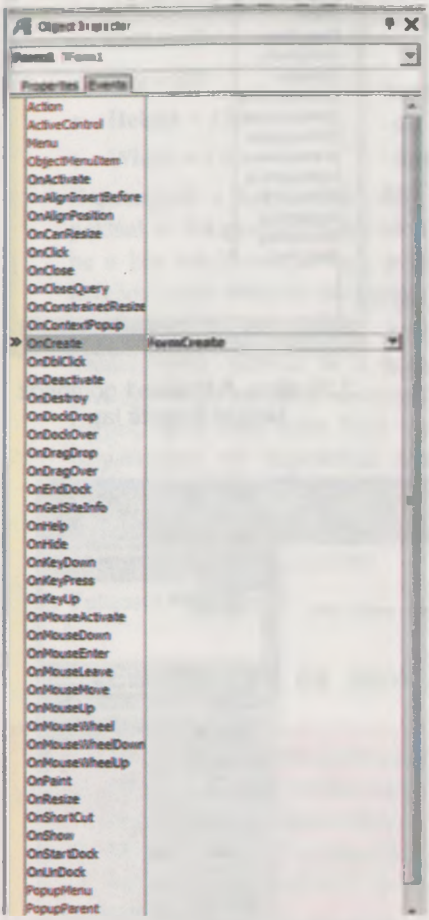
2.27. ábra. A **Code** ablak a **TForm1.Button1Click** eljárás sablonjával

Az *eljárás* (Procedure) a latin *procedere* (előrehaladás) szóból származik, a programnak az a része, amelyet a nevére történő hivatkozással a program tetszőleges részéből meghívható.

Ha tehát a **Button1** gombra kattintunk, bekövetkezik az **OnClick** esemény, és végrehajtásra kerül a **TForm1.Button1Click** eljárás.

Az eljárás a **Delphiben a subroutine-ok (alprogramok)** egyik típusa. A másik típust **függvénynek (function)** nevezzük. Ezekre a későbbiek során térünk majd ki.

Azt az eljárást, amely valamely esemény hatására következik be, az esemény feldolgozójának *(eseménykezelőnek)* nevezzük.
Azt az eljárást, amely egy meghatározott objektumhoz kapcsolódik, az *objektum metódusának* nevezzük.



2.28. ábra. A **Form1**-hez tartozó **Events** lap

A **TForm1.Button1Click** az **OnClick** esemény feldolgozója (eseménykezelője), ugyanakkor a **Button1** gomb az objektum metódusa is.

A fenti módon hozzuk létre bármely esemény feldolgozóját, például:

- **OnMouseMove** (egérmozgás) – ez az esemény akkor következik be, ha rámutatunk az egérmutatóval a gombra.
- **OnKeyPress** (billentyű leütése) – ez az esemény akkor következik be, ha egy billentyűt lenyomunk a billentyűzeten.
- **OnStartDrag** (elhúzás megkezdése) – az esemény akkor következik be, ha az objektumot elkezdjük egérrel elmozdítani.

Hasonlóképpen hozunk létre metódusokat más objektumokhoz is. A 2.28. ábrán a **Form1** objektumhoz rendelhető eseményeket láthatjuk. Ez sokban hasonlít a gomb eseményeihez, de vannak attól eltérőek is, például:

- **OnCreate** (létrehozás) – ez az esemény a form létrehozásakor következik be, ezért az ide beírt utasítások az ablak létrehozása előtt valósul meg.
- **OnDblClick** (dupla kattintás) – ez az esemény akkor következik be, ha az ablakra duplán kattintunk.

Az esemény-feldolgozók létrehozása

Vizsgáljuk meg az **OnClick** esemény feldolgozóját, az **TForm1.Button1Click** eljárást:

```
procedure TForm1.Button1Click (Sender: TObject);
begin

end;
```

Az első sor az eljárás **fejléce.** A **procedure** kulcsszóból és az eljárás nevéből áll. Az eljárás neve után a zárójelben azt láthatjuk, hogy az eljárás valamely objektumon, jelen esetben a **Button1** objektumon létrejövő esemény bekövetkezésekor hajtandó végre. A fejléc sora **pontosvesszővel** (;) zárul.

Az **OnClick** esemény hatására végrehajtandó események a **begin** (eleje) és **end** (vége) kulcsszavak között helyezkednek el. Az utasítások alkotják majd a **függvény törzsét.** A **begin** és **end** szavak jelölik, hogy hol kezdődnek a törzset alkotó utasítások, illetve hol érnek véget. Mondhatjuk, hogy gyakorlatilag a nyitó és záró zárójelek szerepét játsszák, azért **utasítás-zárójeleknek** nevezzük ezeket. Az **end** szó után pontosvesszó következik. A törzs utasításait szintén pontosvesszővel zárjuk.

Az előző részben a vezérlőelemek tulajdonságait a futtatás előtt állítottuk be. Most már lehetőségünk lesz arra, hogy ezeket futási időben tegyük meg. E célból egy eseménykezelőt kell a programban elhelyezni.

Ha azt szeretnénk , hogy az eljárás változtassa meg egy objektum valamely tulajdonságának értékét, egy **tulajdonság-beállító utasítást** kell abban elhelyezni. Ennek általános alakja a következő:

<objektumnév>.<tulajdonság>:= <érték vagy kifejezés>;

Például:

Form1.Color := clGreen; – az ablak háttérszínét állítja *zöldre*;

Form1.Left := 300; – az ablak bal oldal és a képernyő bal oldala közötti távolságot *300 pontra* állítja;

Form1.Width := Form1.Width*2; – az ablak szélességét megkétszerezi;

Button1.Caption := 'Збільшити'; – a gomb feliratát *Збільшити* -re állítja (ha a tulajdonság értéke szöveges, akkor azt szimpla idézőjelek közé kell tenni.);

Label1.Font.Color := clRed; – a felirat színét *pirosra* állítja.

Figyeljük meg, hogy ezek az utasítások **értékadó utasítások,** amelyeket már használtunk az algoritmusok írása során.

Következzen a **TForm1.Button1Click** eljárás, amely az ablak színét kékre változtatja, megváltoztatja az ablak feliratát, az ablak szélességét 150 pontra állítja be, elmozdítja a gombot és megváltoztatja annak feliratát:

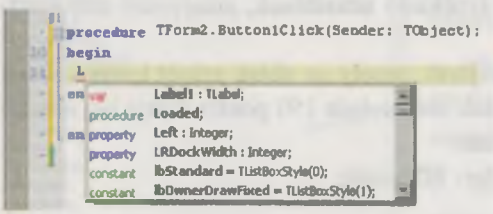
```
procedure TForm1.Button1Click (Sender: TObject);
begin
   Form1.Color := clBlue;
   Form1.Caption := 'Значення властивостей змінилися';
   Form1.Width := Form1.Width – 150;
   Button1.Left := 60;
   Button1.Top := 100;
   Button1.Caption := 'Змінити!';
end;
```

A 2.29. ábrán az alkalmazás ablakát láthatjuk a **Button1** gombra kattintás előtt, illetve után.

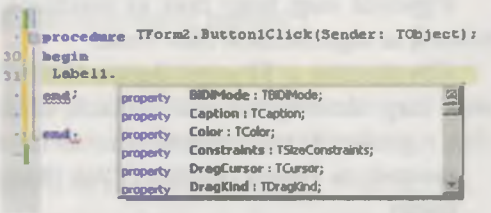
2.29. ábra. Az alkalmazás ablaka a **Button1**-re kattintás előtt és után

A **Turbo Delphi 2006** tartalmaz olyan eszközöket, amelyek megkönnyítik az utasítások beírását:

- amikor beírtuk az utasítás első betűjét, a **Ctrl+Szóköz** leütés után megnyílik az adott betűvel kezdődő objektumok, tulajdonságok és eljárások listája (2.30. ábra), amiből kiválasztható a szükséges elem és az bekerül a program szövegébe;
- a program szövegének beírása során a rendszer ügyel a szintaktikai szabályok betartására és amennyiben hibát észlel, hullámos vonallal aláhúzza azt (például a 2.30. ábrán az **L** betű, a 2.31. ábrán pedig a **Label1** után van egy pont, de nem következik tulajdonság); a rendszer jelzéseit figyelmesen kell követni, és a kimutatott hibákat javítani kell;
- a **Turbo Delphi 2006** nem különbözteti meg a kis- és nagybetűket; ha azonban egy név több szóból áll, célszerű a szavak kezdőbetűit nagybetűvel írni, például a **TForm1.Button1Click**;
- amikor az objektum neve után leírjuk a pontot, automatikusan megnyílik az objektum tulajdonságainak listája (2.31. ábra); a tulajdonság nevét tehát nem kell beírni, csak ki kell választani a listából, majd **Entert** ütni. Ez a módszer is csökkenti a szintaktikai hibák elkövetésének valószínűségét, különösen azoknál, akik nem beszélnek angolul.

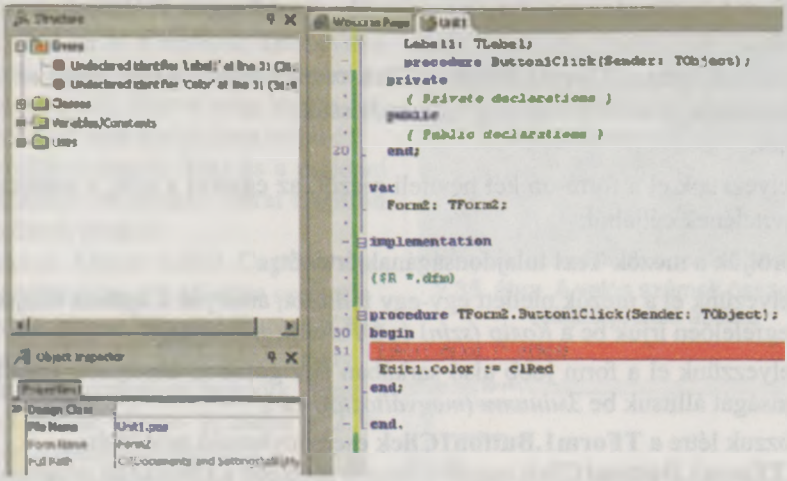


2.30. ábra. A **Ctrl+Szóköz** leütése után megnyíló lista

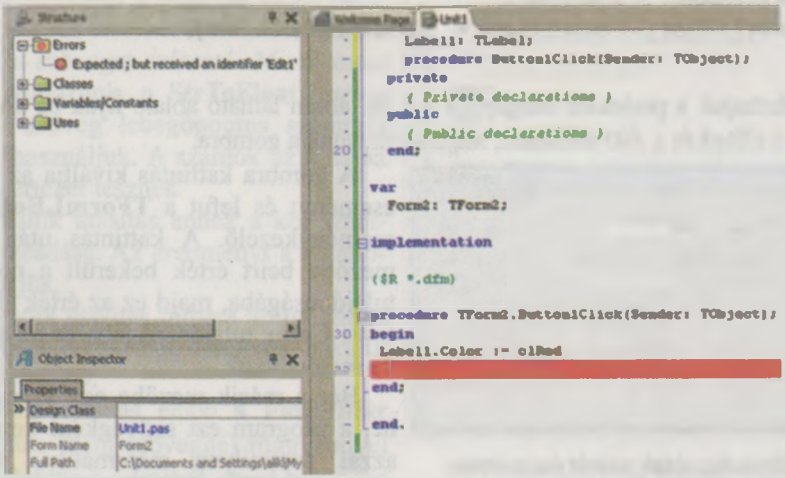


2.31. ábra. A **Label1** objektum tulajdonságainak listája

A projekt futtatása előtt a compiler megvizsgálja, tartalmaz-e a program szövege szintaktikai hibákat. Amennyiben talál, a fordítás folyamatát megszakítja és piros háttérrel kiemeli a programnak az első olyan sorát, amelyikben hibát talált (a 2.32. ábrán hibásan van leírva a **Label1** szó; a 2.33. ábrán pedig hiányzik a pontosvessző a sor végéről, ilyen esetben a következő sor lesz pirossal kiemelve). A többi hibát piros hullámvonallal jelöli a compiler. Ezen kívül a **Structure** ablakban is megjelennek a hibajelzések, amelyek a hiba típusát is megjelenítik. A hibák kijavítása után a projektet újra futtathatjuk.



2.32. ábra. A **Label** szó beírása során elkövetett szintaktikai hiba jelölése



2.33. ábra. A pontosvessző kihagyása miatt keletkezett szintaktikai hiba jelölése

Megvizsgáltuk, hogyan hoztuk létre és használtunk **TForm1.Button1Click** eljárást az **OnClick** eseménykezelő segítségével. Hasonlóképpen hozhatók létre más esemény-feldolgozók.

Futásidejű adatbevitel és kivitel

Megvizsgáltuk, hogyan változtathatjuk meg a vezérlőelemek tulajdonságait az **OnClick** eseménykezelő segítségével. A vizsgált esetben a gombra kattintva mindig ugyanazok a tulajdonságok változnak majd meg, mivel ugyanazok az utasítások kerülnek végrehajtásra. Ha azt szeretnénk, hogy a tulajdonságok értékei más-más értékeket vegyenek fel, ha a gombra kattintunk, a programba futási időben be kell vinni ezeket az értékeket.

Az adatbevitel egyik lehetséges módja a beviteli mezők alkalmazása. Tudjuk, hogy a mező **Text** tulajdonsága tartalmazza a mezőbe írt szöveget. Próbáljuk most meg ezt felhasználni.

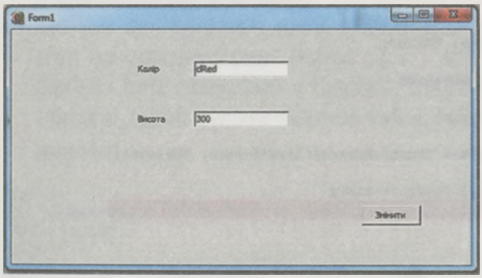
Változtassuk meg a **Tform1.Button1Click** eseménykezelőt úgy, hogy az ablak színét és magasságát tetszőlegesen megváltoztathassuk.

E célból:

1. Helyezzünk el a form-on két beviteli mezőt, az egyiket a szín, a másikat a méret bevitelének céljából;
2. Töröljük a mezők **Text** tulajdonságának értékét;
3. Helyezünk el a mezők mellett egy-egy feliratot, amelyek **Caption** tulajdonságába megfelelően írjuk be a *Колір (szín)* és a *Розмір(méret)* szót!
4. Helyezzünk el a form jobb alsó sarkában egy gombot, amelynek **Caption** tulajdonságát állítsuk be *Змінити (megváltoztat)*-ra !
5. Hozzuk létre a **TForm1.Button1Click** eseménykezelő sablonját;
6. A **TForm1.Button1Click** eseménykezelő törzsébe a következő programrészt:

```
procedure TForm1.Button1Click (Sender: TObject);
begin
   Form1.Color := Edit1.Text;
   Form1.Hight := Form1.Hight +  StrToInt (Edit2.Text);
end;
```

Ha lefuttatjuk a projektet, megnyílik a 2.34. ábrán látható ablak. Írjuk be a megfelelő mezőkbe a clRed és a *300* értékeket, majd kattintsunk a gombra.



2.34. ábra. Az ablak színét és méretét megváltoztató projekt futása

A gombra kattintás kiváltja az **OnClick** eseményt és lefut a **TForm1.Button1Clik** eseménykezelő. A kattintás után az első mezőbe beírt érték bekerült a mező **Text** tulajdonságába, majd ez az érték átadódik a form **Color** tulajdonságába és az ablak színe megváltozik.

Bár a másik mezőbe egy számot írtunk be, a program ezt szövegként kezeli, ezért azzal semmilyen matematikai műveletet nem végezhetünk. Ezért ezt a szöveget először számmá kell alakítani. Erre szolgál a **StrToInt** (*string to integer* – szöveg számmá) függvény. Ezután a szám hozzáadódik az ablak aktuális magasságához.

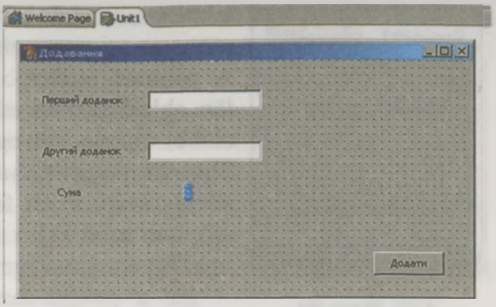
A projekt futtatása vagy a gombra történő minden kattintás előtt megváltoztathatjuk a mezőkbe beírt értékeket, tehát az ablak színe és magassága a megadott értékeket veszi fel.

A feliratokat nemcsak az objektumok feliratozására, hanem adatkivitelre is használhatjuk.

Hozzunk létre egy egyszerű számológépet, amely két valós számot ad össze. A program futása során a felhasználó a két beviteli mezőbe beírja az összeadandókat, majd a *Додати* gombra kattint. Ekkor bekövetkezik az **OnClick** esemény, és lefut a **TForm1. Button1Click** eseménykezelő, amely elvégzi az összeadást.

A projekt (2.35. ábra) létrehozásának lépései:

1. Helyezzünk el a form-on két beviteli mezőt, egy gombot és egy feliratot az eredménynek, illetve még három feliratot a külalak kialakítása céljából!
2. Töröljük a mezők **Text** és a majdani eredményt tartalmazó felirat **Caption** mezőinek értékét!
3. A másik három felirat **Caption** mezőit állítsuk be a 2.35. ábra szerint!
4. Hozzuk létre a **TForm1.Button1Click** eseménykezelő sablonját, amely a gomb lenyomásakor hajtódik végre!
5. Írjuk be az eljárás törzsébe a 2.36. ábrán látható kódrészletet!



2.35. ábra. A valós számok összeadását végző projekt ablaka

Tekintsük át részletesebben az egyes utasítások rendeltetését.

Ahogy azt már megtanultuk, a beviteli mezőbe írt értékeket a program szövegként kezeli. Az eljárás első két értékadó utasítása alakítja át a szöveget számmá. Mivel most törtekkel dolgozunk, a **StrToFloat** (*string to float* – szöveg lebegőpontos számmá) függvényt használjuk. A számok az *x* és az *y* változók értékei lesznek.

A harmadik utasítás ennek a két számnak az összeadása. Az eredményt a *z* változóban mentjük.

A negyedik utasítás pirosra állítja annak a feliratnak a betűszínét, amelybe bekerül majd az eredmény.

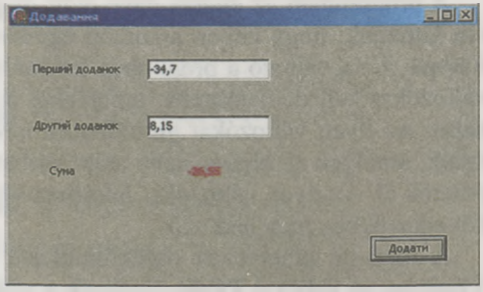
Az ötödik utasítás előbb a **FloatToStr** (lebegőpontos szám szöveggé) függvénnyel szöveggé alakítja a *z* értékét, majd beírja azt a **Label1** felirat **Caption** tulajdonságába.

2.36. ábra. A project programjának szövege a **Code** lapon

A projekt futtatása során a felhasználó beírja a beviteli mezőkbe a két összeadandót (például – 34,7 és 8,15), majd megnyomja a *Додати* gombot, és megkapja a (–26,55) eredményt (lásd a 2.37. ábrát).

A felhasználó a projekt bezárása előtt tetszőleges mennyiségű számot adhat össze. Ez az algoritmus teljességét mutatja.

Megeshet, hogy az eredményt nem a form vezérlőibe, hanem egy speciális ab-



2.37. ábra. A két valós szám összeadását végző projekt futás közben

2.37. ábra. Az eredményt megjelenítő párbeszédablak

lakba szeretnénk kiíratni. Ezt a **ShowMessage(<szöveg>)** (*show message* – megmutatni az üzenetet) paranccsal tehetjük meg. A parancs értelmében a paraméterébe beírt szöveget egy külön ablakban kell megjeleníteni. Ha ezt az utasítást írjuk be az előbbi eljárás szövegébe, akkor az a következőképpen nézne ki: **ShowMessage('Сума '+FloatToStr(z));**. A + jel a zárójelben azt jelenti, hogy a két felirat egyesítését kell megjeleníteni. A **Сума** szó után azért kell szóközt tenni, hogy a két szó ne legyen egybeírva. Az utasítás végrehajtásának eredményét a 2.37. ábra mutatja.

Ha a programban párbeszédablakos megjelenítést alkalmazunk, akkor előbb be kell zárni ezt a párbeszédablakot, csak azután lehet bezárni a program ablakát.

Változók és azok tulajdonságai. A matematikai kifejezések megírásának szabályai Delphi-ben

Az utolsó eljárásunkban három változó szerepelt: *x*, *y* és *z*. A **Delphi**-ben egy változónak neve, jelentése és **típusa** van.

A *változó típusa* meghatározza, milyen értékeket vehet fel a változó, milyen műveleteket végezhetünk vele, továbbá azt, mekkora tárterületet foglal el.

Az *x*, *y* és *z* a feladat feltétele szerint tetszőleges valós értéket vehet fel. Az ilyen változó **Real** (valós) típusú. A valós változó 8 **byte** helyet foglal a memóriában, értékkészlete $-5 \cdot 10^{-324}$ és $1{,}7 \cdot 10^{308}$ közötti.

A változó típusát a **var** (*variable* – változó) kulcsszó után adjuk meg, mely az eljárásban a fejléc és a **begin** foglalt szó között kell, hogy elhelyezkedjen (lásd a 2.36. ábrát). Ezt a sort a **deklaráció sorának** nevezzük. A sor általános alakja a következő:

Var <az azonos típusú változók vesszővel elválasztott felsorolása>: <típus neve>;<az azonos típusú változók vesszővel elválasztott felsorolása>: <típus neve>; és így tovább.

Ha a változó csak egész értékeket vehet fel, akkor azt célszerű **Integer**-nek (egész) deklarálni. Az egész változók **4 byte** helyet foglalnak el, értékeik –2 147 483 648 (-2^{32}) és 2 147 483 647 ($2^{32}-1$) közöttiek.

A programban az eljárás neve előtt még egy **var** kezdetű sort találunk. A 2.36. ábrán láthatjuk, hogy ebben a sorban egyetlen, **TForm** típusú változó van deklarálva, a **Form1**. Ez a változó a projektben használt form-nak felel meg. Ebbe a sorba azokat a változókat kell deklarálnunk, amelyeket a program tetszőleges helyén kívánunk használni. Az ilyen változókat **globálisnak** nevezzük. A globálisaktól eltérően azok a változók, amelyek az eljárás neve utáni sorban vannak deklarálva, csak az eljáráson belül érhetők el. Az ilyen változókat **lokálisnak** nevezzük. A lokális és globális változókról a következőkben még lesz szó.

A form-on elhelyezett objektumoknak is van neve, illetve típusa: a **Button1**-é a **TButton**, az **Edit1**-é **TEdit**. Ezeknek a változóknak a neveit és típusait a rendszer automatikusan helyezi el a **type** (típus) részben (2.36. ábra).

A **Delphi**-ben az értékadó utasítás általános alakja a következő:

<változónév>:=<kifejezés>. A kifejezés tartalmazhat **állandókat** (számokat, szimpla idézőjelben vett szöveget), változóneveket, illetve matematikai műveleteket, függvényeket, zárójeleket.

A matematikai műveletek jelei és megadásának szabálya a **Delphi**-ben sokban megegyeznek az **Excel**-ben megszokottal. Vannak azonban eltérések is:

1) a számok egész részét és törtrészét tizedesponttal választjuk el;
2) ha egy beviteli mezőbe írunk számot, a tizedes rész elválasztója a **Windows** beállításaitól függ (a magyar változatban ez általában vessző);
3) se a program szövegében, sem pedig a beviteli mezőkben nem használhatunk közönséges törtet;
4) A **Delphi**-ben nincs hatványraemelés-jel; amennyiben hatványozásra van szükségünk, akkor a későbbiek során tárgyalandó függvényt kell használni. Egyelőre úgy járunk el, hogy a természetes kitevőjű hatványt szorzattal helyettesítjük;
5) a műveletek prioritási szabályai megegyeznek a matematikában megismert szabályokkal.

Ellenőrző kérdések

1°. Mi az eseménykezelő? Hogy kapcsolódik az esemény és a kezelő egymáshoz?
2°. Mi az eljárás? Milyen az eseménykezelő eljárás szerkezete?
3°. Milyen annak az utasításnak az általános alakja, amellyel egy vezérlő tulajdonságát meg lehet változtatni? Hozzunk fel példákat!
4°. Hogy jelöli a fejlesztőrendszer a szintaktikai hibákat? Hozzunk fel példákat!
5°. Milyen vezérlőelemeket használhatunk adatok bevitelére és kivitelére?
6°. Hogy írhatunk ki üzenetet egy külön ablakba?
7°. Mi a **StrToFloat, FloatToStr, StrToInt, IntToStr** függvények rendeltetése?
8°. Mit határoz meg a változó típusa? Milyen változótípusokat ismerünk?
9°. Milyen értékeket vehetnek fel az **Integer** és **Real** típusú változók? Mekkora helyet foglalnak el ezek a számítógép memóriájában?
10°. Hogy deklaráljuk az eljárásban használt változókat? És az egész programban használtakat?
11°. Mik a globális és lokális változók?
12°. Milyen az értékadó utasítás általános alakja a **Delpi**-ben?

Végezzük el a feladatokat!

1°. Írjuk fel az értékadó utasításokat az alábbi esetekben:

a) $72-(15\cdot(-5)+23):7$;

b) $-55+12^3+(-4)^4:22$;

c) $\dfrac{23{,}5+(-6{,}2)^3+45\cdot(-3)}{34+(7{,}2)^2-44\cdot 2{,}2}$;

d) $\dfrac{(-4{,}17)^3\cdot(-12{,}1)+5{,}76}{-12{,}33-4{,}24^3\cdot(-32{,}55)+1{,}1}$;

e) $(-3{,}15)^3\cdot 5{,}15+3{,}12:(-18)$;

f) $-31{,}3:2{,}15+9{,}15^2\cdot(-3{,}76)$.

2°. Írjuk fel az értékadó utasításokat az alábbi esetekben:

a) $3x^2-5y^3+7z$;

b) $(x+y+z):3$;

c) $\frac{7x-12y^3}{45-2z^2}$;

d) $\frac{34-5x^3+yz}{-24-4{,}76y^2}$;

e) $\frac{2x^2-5}{3-4y^2}-7zy$;

f) $7{,}23xyz-\frac{7x^2-15y}{4z-8{,}23y^3}-x$.

3°. Nyissunk egy új projektet! Változtassuk meg a form címsorát *„Я змінюю значення властивостей форми"*- ra. Helyezzünk el a form-on egy gombot, és változtassuk meg annak feliratát *„Змінити"*-re! Változtassuk meg a form nevét **MyForm1**-re! Hozzuk létre az **OnClick** eseményt feldolgozó eljárás sablonját! Írjunk bele olyan utasítást, amely **clPurple**-re változtatja az ablak színét, 100 pixellel csökkenti az ablak szélességét és 200 pixellel növeli annak magasságát! Futtassuk a projektet! Győződjünk meg arról, hogy az eseménykezelő helyesen működik! Hozzuk létre a **Project_2_5_3** mappát, és mentsük a projektet!

4°. Nyissunk egy új projektet! Változtassuk meg a form címsorát *„Я вмію змінювати значення властивостей кнопки"*- ra! Helyezzünk el a form jobb alsó sarkában egy gombot, és változtassuk meg annak feliratát *„Плаваюча кнопка"*-ra! Hozzuk létre az **OnClick** eseményt feldolgozó eljárás sablonját! Írjunk bele utasításokat, amelyek a gombot 30 ponttal felfelé és 50 ponttal jobbra helyezi át! Futtassuk a projektet! Figyeljük meg a működését! Hozzuk létre a **Project_2_5_4** mappát, és mentsük a projektet!

5°. Hozzunk létre két valós szám szorzására szolgáló projektet! Az ablak hátterét állítsuk sárgára, az eredmény színét pedig kékre. Az eredményt írassuk ki különálló ablakban is! Futtassuk le a projektet néhány különböző számpárra, és ellenőrizzük az eredmény helyességét! Hozzuk létre a **Project_2_5_5** mappát, és mentsük a projektet!

6°. Hozzunk létre projektet, amely az egyik gombra kattintva kiszámítja két egész szám négyzeteinek összegét, egy másik gombra kattintva pedig a négyzetek különbségét! Az ablak háttere legyen világoskék, az eredménye pedig piros. Az eredményeket írassuk ki különálló ablakba is! Futtassuk le a projektet néhány különböző számpárra, és ellenőrizzük az eredmény helyességét! Hozzuk létre a **Project_2_5_6** mappát, és mentsük a projektet.

7°. Nyissunk meg egy új projektet! Az ablak címsorát állítsuk be **Disabled**-re. Tegyünk a form-re két gombot, és állítsuk be a következő tulajdonságokat: **Caption** – *Кнопка 1*, **Enabled** – *True* az egyik, **Caption** – *Кнопка 2*, **Enabled** – *False* a másik esetében. Az első gomb **OnClick** eseménykezelőjébe írjuk be az alábbi utasításokat:

```
Button2.Enabled := True;
Button1.Enabled := False;
```

A másik gomb esetében pedig:

```
Button1.Enabled := True;
Button2.Enabled := False;
```

Elemezzük az eljárásokat, és állapítsuk meg, minek kell történnie a project futtatása során! Futtassuk a programot, és ellenőrizzük feltételezéseink helyességét! Hozzuk létre a **Project_2_5_7** mappát, és mentsük a projektet!

8°. Nyissunk meg egy új projektet! Az ablak címsorát állítsuk be **Disabled**-re! Tegyünk a form-re két gombot, és állítsuk be a következő tulajdonságokat: **Caption** – *Кнопка 1*, **Visible** – *True* az egyik, **Caption** – *Кнопка 2*, **Visible** – *False* a másik esetében. Az első gomb **OnClick** eseménykezelőjébe írjuk be az alábbi utasításokat:

```
Button2.Visible := True;
Button1.Visible := False;
```

A másik gomb esetében pedig:

```
Button1.Visible := True;
Button2.Visible := False;
```

Elemezzük az eljárásokat, és állapítsuk meg, minek kell történnie a projekt futtatása során! Futtassuk a programot, és egymás után megnyomva a gombokat, figyeljük meg mi történik! Hozzuk létre a **Project_2_5_8** mappát, és mentsük a projektet!

9*. Hozzunk létre új projektet, amelyben az **OnClick** esemény megváltoztatja az ablak színét, méreteit, elhelyezését és címsorát! Hozzuk létre a **Project_2_5_9** mappát, és mentsük a projektet!

10*. Hozzunk létre új projektet, amelyben az **OnMouseMove** esemény megváltoztatja az ablak színét, méreteit, elhelyezését és címsorát! Hozzuk létre a **Project_2_5_10** mappát, és mentsük a projektet!

11*. Hozzunk létre új projektet, amelyben az **OnClick** esemény megváltoztatja az ablak színét, méreteit, majd egy mezőben piros betűkkel megjeleníti a nevünket! Hozzuk létre a **Project_2_5_11** mappát, és mentsük a projektet!

12*. Hozzunk létre új projektet, amelyben az **OnClick** esemény megváltoztatja az felirat színét, méreteit, elhelyezését és megjeleníti a **Вітаю** szöveget kék színben! Hozzuk létre a **Project_2_5_12** mappát, és mentsük a projektet!

13*. Hozzunk létre új projektet, amely a szobátok szélessége, hossza és magassága alapján meghatározza a padló és a falak felszínét! Hozzuk létre a **Project_2_5_13** mappát, és mentsük a projektet!

14*. Hozzunk létre projektet, amely kiszámítja két valós szám összegét, különbségét és szorzatát! Hozzuk létre a **Project_2_5_14** mappát, és mentsük a projektet!

15. Hozzunk létre egy új projektet, és helyezzünk el a form-on három gombot! Állítsuk be a tulajdonságaikat úgy, hogy a projekt futásának kezdetén az első gomb legyen látható és engedélyezett, a második látható és tiltott, a harmadik pedig láthatatlan és tiltott! Írjuk meg az **OnClick** események kezelőit a következőképpen:

a) az első gombra kattintva az változzon láthatatlanná és tiltottá, a második láthatóvá és engedélyezetté, a harmadik láthatóvá és tiltottá;

b) a második gombra kattintva az változzon láthatatlanná és tiltottá, a harmadik láthatóvá és engedélyezetté, az első láthatóvá és tiltottá;

c) a harmadik gombra kattintva az váljon láthatatlanná és tiltottá, az első láthatóvá és engedélyezetté, a harmadik pedig láthatóvá és tiltottá.

Hozzuk létre a **Project_2_5_12** mappát, és mentsük a projektet!

6. számú gyakorlati munka **Adatbevitel és kivitel. Változókezelés**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert, és hozzunk létre egy új projektet!

2. Állítsuk be az ablak héttérszínét sárgára, méreteit *1000* és *600* pixelre, a címsora legyen *Практична 6!*

3. Helyezzünk el a form-on két gombot, három beviteli mezőt és két feliratot!

4. Legyen az első gomb felirata *Переміщення*, a másodiké *Обчислення*, a mezőkben és a feliraton ne legyen szöveg!

5. Helyezzünk el a mezők és a felirat mellett olyan kiegészítő feliratokat, amelyek az előbbi vezérlők rendeltetéséről informálnak!

6. Hozzuk létre az **OnClick** eseménykezelőket! Az első gomb mozdítsa el az ablakot *50 ponttal fel és 30 ponttal jobbra*, minden mezőt *20 ponttal balra* és *30-cal felfelé* és változtassa azok háttérszínét pirosra!

7. Futtassuk a projektet, és ellenőrizzük az első gomb működését!

8. A második gomb eseménykezelője határozza meg a téglatest térfogatát és felszínét, ha a beviteli mezők tartalmazzák az éleket!

9. Futtassuk a projektet, és ellenőrizzük a második gomb működését!

10. Adjunk a második gomb eseménykezelőjéhez egy utasítást, amely a téglatest kiszámított értékét kiírja egy ablakba! Futtassuk a projektet, és győződjünk meg az eredmények helyességéről!

11. Mentsük a munkát a **Gyakorlati_6** mappába!

2.6. Hibakeresés

1. Milyen részei vannak az integrált fejlesztői környezetnek?
2. Mi a compiler rendeltetése? Mik a működésének sajátosságai?
3. Milyen programhibák derülnek ki a fordítás (kompilálás) során? Hogyan jelennek meg ezek a **Turbo Delphi 2006** rendszerben? Hol jelenítődnek meg a hibák leírásai?
4. Mik a logikai hibák? Hogy lehet a logikai hibákat megtalálni?
5. Hogy nyithatunk meg egy régebben mentett projektet?

A program tesztelése. Kommentárok

Már tudjuk, hogy a projekt létrehozása során szintaktikai és logikai hibákat követhetünk el. A programírás során észre nem vett szintaktikai hibákat a compiler a fordítás során kimutatja. Ezután ki kell javítani a hibákat, majd újra kell fordíttatni a projektet.

A logikai hibákat nehezebb megtalálni, mint a szintaktikaiakat. Ilyenkor a projekt lefut, de a kapott eredmények nem felelnek meg a kitűzött feladat feltételeinek.

A számítógépes feladatmegoldás lépéseivel való ismerkedésünk során megtudtuk, hogy a logikai hibák kimutatásának módja, hogy a projektet előre eltervezett tesztadatokkal futtatjuk. Ezután összevetjük a programunk által szolgáltatott eredményeket az elvárt teszteredményekkel, és meghatározzuk azokat az utasításokat, amelyek az esetleges eltéréseket okozzák. A logikai hibák keresésének tesztadatokkal történő módját **tesztelésnek** nevezzük.

Vizsgáljuk meg a tesztelés folyamatát a következő példán. Tegyük fel, hogy az $y = \frac{2x-5}{x^2+1}$ kifejezés értékét kell meghatározni. A tanulók gyakran írják le az $y := (2*x-5)/(x*x+1)$ kifejezést anélkül, hogy kitennék a zárójeleket. Ekkor az $y := (2*x-5)/x*x+1$, vagy $y := 2*x-5/x*x+1$ kifejezést írják be. Ezt a logikai hibát alkalmasan megválasztott tesztadatokkal könnyen kiszűrhetjük. A projekt $x = 0$ esetében nem az elvárt –5 eredményt adja, hanem **division by zero** (nullával osztás) hibaüzenettel leáll. Most már csak meg kell találni azt a műveletet, amely a nullával történő osztást okozta.

Az olyan egyszerű programok esetében, amilyeneket eddig létrehoztunk, viszonylag könnyű tesztadatokat választani. Lényegesen nehezebb alkalmas tesztadatokat találni bonyolultabb logikai szerkezetű projektek és profi felhasználói programok esetében, amelyek több ezer utasításból állnak.

Jelentősen megkönnyíthetjük a logikai hibák keresését, ha a program szövegébe **kommentárokat** helyezünk el.

A *kommentár* olyan szöveg, amelyet a compiler a fordítás során figyelmen kívül hagy és a program szerkezetének könnyebb megértése céljából írjuk a programba.

Ahhoz, hogy a compiler meg tudja különböztetni a kommentárokat a program többi részétől, kapcsos zárójelek között kell elhelyezni: **{Ez itt egy kommentár}**. Amennyiben a kommentár nem foglal el egy sornál többet, akkor nincs szükség zárójelekre, elég, ha elé két dőlt vonalat teszünk: // **Ez egy egysoros kommentár.** A kapcsos zárójel helyett használhatunk zárójel-csillag párt is: **(*A hányados kiszámítása*).**

Ha a kommentár rövid, beírható közvetlenül az utasítás után, de akár új sorba is, de nem helyezkedhet el a sor közepén.

A 2.38. ábrán a kommentárok alkalmazásának különböző példáit láthatjuk.

Ha a program nagyon nagy, a logikai hibák keresését célszerű részenként elvégezni. Ha meggyőződtünk a program egy részének helyes működéséről, akkor azt „kommenteljük", azaz kapcsos zárójelbe tesszük. A compiler a következő fordításnál kihagyja a programnak ezt a részét. Ilyen módon a fejlesztő minden figyelmét a programnak azon részeire összpontosíthatja, ahol még lehetnek logikai hibák és időt takarít meg.

2.38. ábra. Kommentárok a program szövegében

A program lépésenkénti végrehajtása

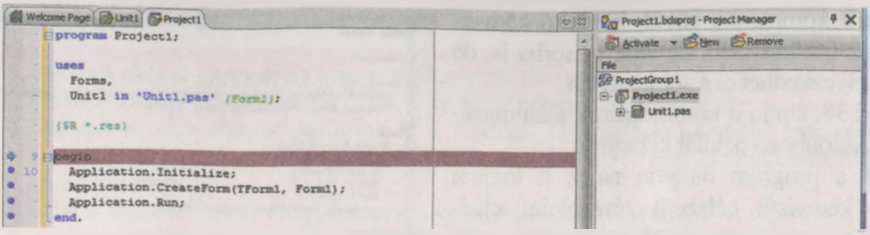
A logikai hibák keresésének másik elterjedt módja a program lépésenkénti végrehajtása. Egy program végrehajtása **soronként történik.** A program egy sora egy vagy több utasítást tartalmazhat. Az eljárások és függvények a lépésenkénti végrehajtás módjától függően végrehajthatók egy lépésként, illetve lépésenként.

Egy program lépésenkénti végrehajtását az **F7** billentyű lenyomásával vagy a ***Run*** ⇒ ***Trace Into*** (nyomkövetés indítása) kezdeményezhetjük. Amennyiben az eljárást vagy függvényt egy sorként kívánjuk végrehajtani, akkor azt az **F8** billentyűre kattintva vagy a ***Run*** ⇒ ***Step Over*** (átlépés) paranccsal tehetjük meg. Még egyszer hangsúlyozzuk, hogy ha az eljárás vagy függvény működésének helyességéről meg vagyunk győződve, célszerű azt egy lépésként végrehajtani.

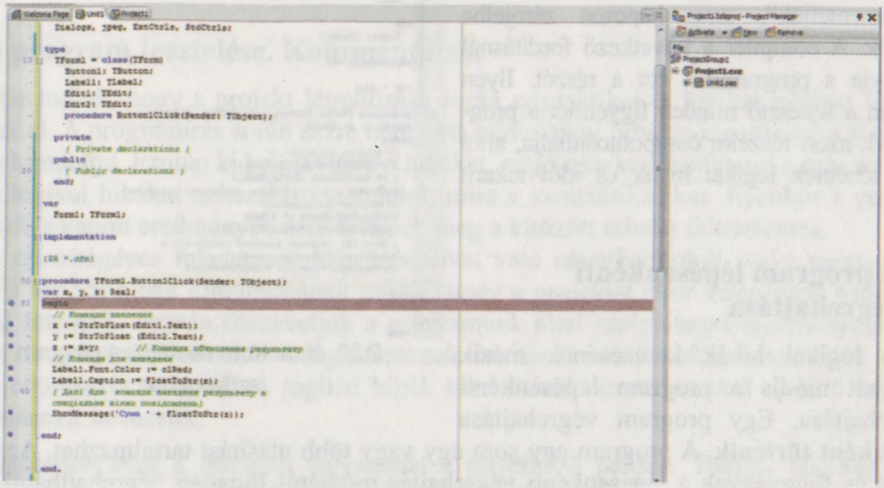
Vizsgáljuk meg a lépésenkénti végrehajtás működését a valós számok összeadására szolgáló projekt példáján (2.38. ábra).

Az **F7** első leütése után megtörténik a program fordítása, és ha nem volt benne hiba, akkor ezután elkezdődik a végrehajtása is. Ezután megnyílik a **dpr** (*delphi project*) fájlt tartalmazó lap. A mi estünkben ez a **Project1.dpr,** ahogy azt a 2.39. ábrán láthatjuk. A fájl első sora színnel lesz kiemelve. Az **F7** minden következő leütése a következő sor végrehajtását eredményezi. A 2.39. ábrán látható utasítások végrehajtása azt eredményezi, hogy megnyílik a projekt ablaka a vezérlőelemekkel (ha mégsem, akkor a tálcán megtaláljuk, csak rá kell kattintani a megfelelő **tálcaikonra**).

Ezután be kell írni a számokat a beviteli mezőkbe, majd rá kell kattintani a *Додати* gombra. Ennek eredményeképpen megnyílik egy **pas** kiterjesztésű fájl, esetünkben az **Unit1.pas.** A 2.40. ábrán láthatjuk ezt a fájlt, amelynek első sora színnel van kiemelve. Az **F7** minden leütése a következő sor utasításának végrehajtásához vezet. Az **F7** sorozatos leütéseivel végigjárhatjuk a programot, végül kiíródik a végeredmény az eredményt megjelenítő párbeszédablakban. A kommentárok értelemszerűen figyelmen kívül lesznek hagyva.



2.39. ábra. A **Project1.dpr** fájl tartalmának tesztelése

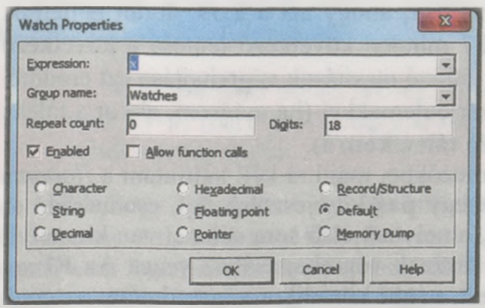


2.40. ábra. Az **Unit1.pas** fájl tartalma

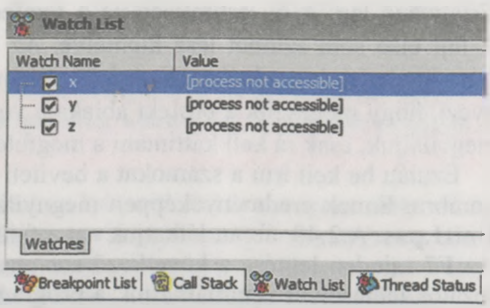
Ha ki szeretnénk lépni a lépésenkénti végrehajtásból, üssük le a **Ctrl+F2** billentyűkombinációt.

A fentiek során lényegében csak szemléltettük a lépésenkénti végrehajtás folyamatát, de semmilyen logikai hibát nem fedezhettünk fel, mivel nem láthattuk, hogy változnak az egyes változók értékei a program végrehajtása során. Márpedig pontosan ez segíthet a logikai hibák megtalálásában.

Ahhoz, hogy a program végrehajtása során követhessük a változók értékeinek alakulását, meg kell nyitnunk a **Watch List** (megfigyelési lista) ablakot, és el kell helyezni ebben az ablakban azokat a változókat, amelyek viselkedésére kíváncsiak vagyunk a program lépésenkénti végrehajtása során.



2.41. ábra. A **Watch Properties** ablak

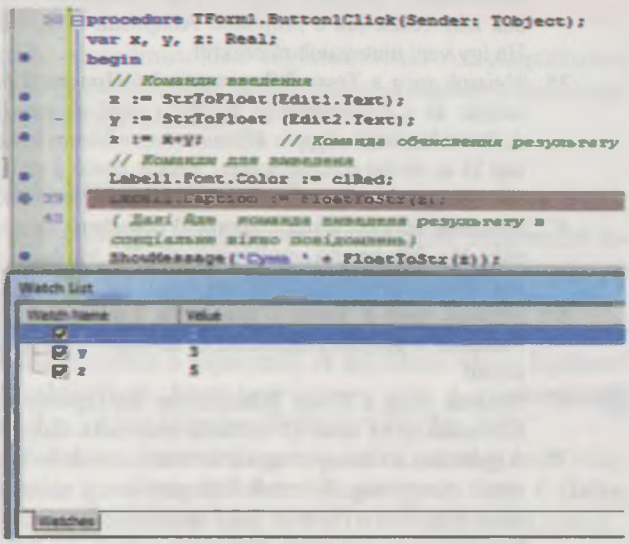


2.42. ábra. A **Watch List** ablak

Egy változót a **Ctrl+F5** billentyűkombináció leütésével, vagy a ***Run*** ⇒ ***Add Watch*** (a megfigyelési listához ad) parancs végrehajtásával adhatunk hozzá listához. Ekkor megnyílik a **Watch Properties** (*tulajdonságok bemutatása*) párbeszédablak (2.41. ábra) Az ablak legfelső mezőjébe be kell írni a változó nevét, majd meg kell nyomni az **OK** gombot.

Ezután megnyílik a **Watch List** (figyelőlista) ablak (2.42. ábra), amelyben szerepel a felvett változó neve. Ilyen módon a figyelési listához hozzáadhatjuk az összes olyan változót, amelynek értékét célszerűnek tartjuk megfigyelni a program futása során.

A **Watch List** ablakban (2.43. ábra) a változók aktuális értékeit láthatjuk majd a program futása során. Ha figyelmesen vizsgáljuk majd a változók értékeit, és folyamatosan összevetjük azokat az általunk elvárt értékekkel, észrevesszük azokat a helyzeteket, ha egy változó váratlan értéket vesz fel. Ez lesz a logikai hiba helye.



2.43. ábra. A **Watch List** ablak a program futásakor

Ellenőrző kérdések

1°. Mit értünk a program tesztelése alatt?
2°. Mi célból végezzük a program tesztelését?
3°. Mi a kommentár? Hogy helyezhetünk el kommentárokat a program szövegében?
4°. Miért célszerű kommentárokat elhelyezni a programban?
5°. Mi célból alkalmazzuk a lépésenkénti végrehajtást?
6°. Mi a lépésenkénti végrehajtás menete?
7°. Mondjatok példát egy program lépésenkénti végrehajtására!
8°. Miben különböznek az **F7** és **F8** billentyűvel kezdeményezett lépések a lépésenkénti végrehajtás során?
9°. Hogy vizsgálhatjuk meg egy változó értékeinek alakulását a lépésenkénti végrehajtás során?

Végezzük el a feladatokat!

1°. Nyissunk meg egy projektet, például a **Тема 2\Завдання 2.6\Проект 2.6.1**-t! A projektfeladat a következő: *Két vonat egyidejűleg indult el egymással szemben. Meg kell határozni, mikor találkoznak, ha ismert a kiindulási pontok távolsága és a vonatok sebessége.* Az **OnClick** eseménykezelő biztosítja a kért eredmény megjelenítését. Az adatokat beviteli mezőkön keresztül adjuk meg. Futtassuk a projektet a következő adatokkal:
a) távolság – 1000, sebességek 120 és 130;
b) távolság – 600, sebességek 80 és 120;
c) távolság – 1200, sebességek 115 és 85!

Vizsgáljuk meg, megfelelnek-e az eredmények a feladat feltételének? Magyarázzuk meg, hol lehet hiba!

2°. Hozzunk létre három készlet tesztadatot a következő projekthez: *Két vonat egyidejűleg indul el ugyanarról az állomásról ellentétes irányba. Meg kell határozni a közöttük levő távolságot az adott időpontban a sebességek ismeretében.* Számítsuk ki a megoldásokat a tesztesetekre! Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.2.** projektet és futtassuk azt! Teszteljük a projektet! Állapítsuk meg, vannak-e a programban logikai hibák! Ha így van, mutassunk rá ezekre!

3*. Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.3.**projektet! Olvassuk el a program kódját, és a kommentárok alapján próbáljuk meg kideríteni, mi volt a projektfeladat. A feltételezésünk alapján állítsunk össze három készlet tesztadatot a projekthez! Számítsuk ki az elvárt eredményeket! Vessük össze a projekt futásának eredményeit az elvárttal! Állapítsuk meg, helyes volt-e a következtetésünk!

4°. Hozzunk létre projektet a körvonal hosszának és a kör kerületének kiszámítására! A kör sugarát a felhasználó egy beviteli mezőbe írja be. Kommentáljuk a programunk azon sorait, ahol a számítások történnek majd! Mentsük el a projektet a megfelelő mappába!

5°. Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.5** mappában található projektet! Futtassuk a projektet lépésenkénti módban! Figyeljük meg, hogy változik a kép a futtatás során!

6°. Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.6** mappában található projektet! Kövessük az **S1** és az **S2** változók értékeinek alakulását a program futása során!

7°. A gyerekek az ünnepre csokit és sütit vásároltak. Ismert a csoki és a süti ára és a megvásárolt mennyiség. Hozzunk létre projektet a vásárlás költségeinek kiszámítására! Mentsük a projektet a **Проект 2.6.7** mappába! Futtassuk a projektet lépésenkénti módban, és figyeljük meg a változók értékeinek alakulását a futtatás során!

8°. Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.8** mappában található projektet! Futtassuk a projektet lépésenként, és figyeljük az *x*, *a* és *S* változók értékeinek alakulását! Állapítsuk meg, milyen értékeket vesz fel az *a* változó és hogy függ ez az érték *x*-től. Változtassuk meg a projektet úgy, hogy az *S* az $x + x^2 + x^4$ értéket tartalmazza! Bővítsük a projektet kommentárral, amely tartalmazza, milyen változtatásokat ejtettünk a kódban! Mentsünk a megváltozott projektet!

9*. Nyissuk meg a **Тема 2\Завдання 2.6\Проект 2.6.9** mappában található projektet! A projektfeladat a következő volt: *Határozzuk meg, mennyivel több időre van szüksége az adott távolság megtételére az egyik vonatnak, mint a másiknak, ha ismerjük a távolságot és a sebességeket!* Állapítsuk meg, vannak-e a programban logikai hibák! Ha vannak, derítsük ki, mely utasítások a hibásak, és javítsuk ki azokat! Jegyezzük fel azokat a tesztadatokat, amelyek segítségével fel tudtuk deríteni a hibákat! Mentsük a javított projektet!

7. számú gyakorlati munka **Hibakeresés**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert!
2. Hozzunk létre projektet, amely meghatározza, *hányszor több két szám szorzata azok összegénél!*
3. Hozzunk létre tesztadatokat a projekt ellenőrzésére! Számítsuk ki az elvárt eredményeket! Vessük össze a futtatás eredményét az elvárttal!
4. Írjunk kommentárokat a számításokat tartalmazó sorokhoz!
5. Futtassuk a projektet lépésenként, amelynek során figyeljük az összeg és a szorzat értékét! Írjuk le, milyen változásokat észleltünk a **Value** oszlopban a program futtatása során!
6. Mentsük el a projektet a **Gyakorlati 7** mappába!

2.7. A matematikai logika alapjai

1. Mi a logikai kifejezés? Milyen értékeket vehet fel?
2. Írjuk fel a táblázatkezelő **And, Or és Not** függvényeinek általános alakját, argumentumainak számát és a függvények értékeit!
3. A változók mely tulajdonságait ismerjük?
4. Mit határoz meg a változó típusa? Milyen típusokat ismerünk?

Az eddigiek során szekvenciális algoritmusokkal foglaekoztunk. Az elágazásokat tartalmazó algoritmusok logikai kifejezéseket alkalmaznak. Most a logikai kifejezésekkel és az azokon végzett műveletekkel foglalkozunk.

Kifejezés. Logikai állandók. Logikai műveletek

A **logika** (a görög λόγος – szó, gondolat, értelem) az intellektuális megismerő tevékenység formáinak és módszereinek, az érvelés helyes módszereinek a tudománya. A logika tudománya a helyes következtetések megalkotásának módjait tanulmányozza.

A logika egyik része – a **matematikai logika** – a matematikai gondolkodás tudománya. A matematikai logika egyik alapfogalma a kijelentés. A kijelentés olyan kijelentő mondat, amelyikről egyértelműen eldönthető, hogy igaz-e vagy sem. A kijelentéseket latin nagybetűvel jelöljük, például *A* = *„Kijev Ukrajna fővárosa”*, *B*=*”2+2=5”*.

A kijelentés egyetlen lényeges tulajdonsága annak igazságértéke, minden más tulajdonságát lényegtelennek tekintjük. Az **igazságérték** lehet 1 (**true,** igaz), vagy 0 (**false,** hamis). Az *A* = *„Kijev Ukrajna fővárosa”* kijelentés **igaz,** a *C*=*”3 > 5”* kijelentés hamis.

A true és false értékeket *logikaállandónak* nevezzük.

A **tagadás, konjunkció** (latin, *conjunctio* – egyesítés) és **diszjunkció** (latin, *disjunctio* – elválasztás) **logikai műveletek** segítségével adott kijelentésekből összetett logikai kifejezéseket képezhetünk.

A *tagadás* az a logikai művelet, amely az *A* kijelentést olyan *A* (nem *A*) kijelentéssé alakítja, amely akkor igaz, ha *A* hamis, és akkor hamis, ha *A* igaz.

A tagadás műveletét a magyar nyelvben a nem tagadószóval fejezhetjük ki. Az *A*=*”Létezik legnagyobb prímszám”* kijelentés tagadása az $\overline{A}$ = *”Nem létezik legnagyobb prímszám”* állítás.

A kijelentés és annak tagadása nem lehetnek egyidejűleg igazak és hamisak.

Ahhoz hasonlatosan, ahogy a matematikában az összeadási és szorzási táblázatokat alkalmazzák, a logikában **igazságtáblázatokat** alkalmazunk. Az igazságtáblázatok azt mutatják, hogy milyen lesz egy logikai művelettel előállított összetett kijelentés **igazságértéke** az elemi kijelentések igazságértékétől függően.

A tagadás műveletét egyetlen kijelentésen végezzük, amely igaz vagy hamis lehet. Ennek megfelelően a tagadás igazságtáblázata a következő lesz:

A	$\overline{A}$
0	1
1	0

Két, *A* és *B* kijelentés konjunkciója egy olyan kijelentés, amely csak akkor igaz, ha mind az *A*, mind pedig a *B* kijelentés igaz. Jelölése $A \wedge B$ (*A* és *B*).

A konjunkció műveletét a magyar nyelvben az és kötőszóval fejezzük ki. A *B* = *„27 osztható 3-mal* és *27 osztható 9-cel"* kijelentés a = *B „27 osztható 3-mal"* és a *„27 osztható 9-cel"* kijelentések konjunkciója.

Definíció szerint a konjunkciót két kijelentésen végezzük. Ezek mindegyike lehet hamis és igaz is. Ennek megfelelően a művelet igazságtáblázata az alábbi:

A	*B*	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

A konjunkciót **logikai szorzásnak** nevezzük.

Két, *A* és *B* kijelentés diszjunkciója egy olyan kijelentés, amely csak akkor igaz, ha az *A* és a *B* kijelentések közül legalább az egyik igaz. Jelölése $A \vee B$ (*A* vagy *B*).

A diszjunkció műveletét a magyar nyelvben a *vagy* kötőszóval fejezzük ki. A *C* = *«21 ≤ 21»* kijelentés a *21=21* és a *21<21* kijelentések diszjunkciója.

Definíció szerint a konjunkciót két kijelentésen végezzük. Ezek mindegyike lehet hamis és igaz is. Ennek megfelelően a művelet igazságtáblázata az alábbi:

A	*B*	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A diszjunkciót **logikai összeadásnak** is nevezik.

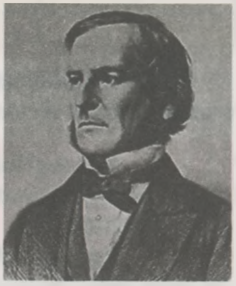
A programok írása során sokszor találkozunk változók közötti relációt – > (nagyobb), < (kisebb), = (egyenlő), <> (nem egyenlő), > (nagyobb, vagy egyenlő), <= (kisebb vagy egyenlő) – kifejező állításokkal, mint például $x > 5$, $y < z$. Az ilyen állítások a változók értékeinek függvényében lehetnek igazak vagy hamisak.

Az $x < 0$ állítás elemi, a $3 < x < 5$ pedig összetett. Ez utóbbit előállíthatjuk két elemi állítás konjunkciójaként: $(x > 3) \wedge (x < 5)$.

Ha például azt kell eldönteni, hogy az $(x; y)$ koordinátájú pont az első negyedhez tartozik-e, azt jelenti, hogy meg kell határozni a z $(x > 0)$ és $(y > 0)$ kijelentés igazságértékét. A (3; –5) pont esetében $(3 > 0)$ és $(-5 > 0)$. Mivel az első kijelentés igaz, a második pedig hamis, a konjunkció eredményeképpen előálló kijelentés hamis. Vagyis ez a pont nem tartozik az első negyedhez.

Logikai változók. Logikai kifejezések és azok igazságértéke. Logikai képletek

A logikai algebra (Boole-algebra, kijelentések algebrája) a matematikai logikának az a része, amely a logikai műveletek segítségével előállított kijelentések általános tulajdonságait tanulmányozza.



2.44. ábra George Boole

A logikai algebra alaptételeit **George Boole** (1815–1864) angol matematikus és filozófus fektette le (2.44. ábra). 1954-ben jelent meg Boole legfontosabb, *A gondolkodás törvényeinek vizsgálata, amelyeken a logika és a valószínűség matematikai teóriái alapulnak* című munkája. A könyv mindmáig a matematika klasszikus alkotásai közé tartozik. Ebben Boole a „kijelentések algebráját" vizsgálja, amelyet ma már boole-algebrának neveznek. A boole-algebra a számítógépek megalkotásának egyik legfontosabb matematika eszköze.

A logikai algebrában alkalmazzák a **logikai változó** fogalmát.

A *logikai változó* olyan változó, amely igaz (true), vagy hamis (false) értékeket vehet fel.

A logikai algebra feladata a logikai állandókat, logikai változókat, logikai műveleteket és zárójeleket – például $(A \vee B) \wedge (A \vee B)$ – logikai kifejezések igazságértékének meghatározása.

Egy logikai kifejezés igazságértéke nyilvánvalóan csak igaz vagy hamis lehet.

A fentebb már vizsgált logikai **műveletek prioritása** a következő: tagadás, konjunkció, diszjunkció. A logikai műveletek végrehajtásának sorrendjét zárójelek segítségével változtathatjuk meg.

A logikai kifejezések igazságértékeinek meghatározása során alkalmazhatunk igazságtáblázatokat is.

Az igazságtáblázatok összeállítása s következőképpen történik:

1. Meg kell határozni a logikai kifejezésben szereplő változók értékeinek összes lehetséges variációját. Ha a kifejezés n változót tartalmaz, akkor a variációk lehetséges száma 2^n lesz. Ez a szám mutatja meg, hány sort tartalmaz majd az igazságtáblázat.
2. Meg kell határozni a kifejezésben szereplő műveletek számát. Ha ezt hozzáadjuk a változók számához, megkapjuk a táblázat oszlopainak számát.
3. Az első n oszlopot fel kell tölteni a változók lehetséges értékeivel.
4. Minden következő oszlop esetében meg kell határozni a soron következő kifejezés igazságértékét és be kell azt írni a megfelelő sorba. A műveletek sorrendjét a prioritási szabályok figyelembevételével döntjük el.

A táblázat utolsó sora fogja tartalmazni a kifejezés összes lehetséges értékét.

Az $A \vee B \wedge A$ kifejezés két logikai változót tartalmaz, vagyis $n = 2$. Ez azt jelenti, hogy a változók értékei $2^2 = 4$ lehetséges variációt alkotnak. A kifejezés 3 logikai műveletet tartalmaz: tagadást, konjunkciót és diszjunkciót. Az igazságtáblázat tehát négy sorból és öt oszlopból áll. A prioritási szabályoknak megfelelően először a tagadást végezzük, majd a konjunkciót, végül a diszjunkciót, amelyet az első két művelet eredményén kell elvégeznünk.

Az igazságtáblázat a következő lesz:

A	B	1	2	3
		$\overline{A}$	$B \wedge A$	$\overline{A} \vee (B \wedge A)$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	0
1	1	0	1	1

Az igazságtáblázatot elemezve az alábbi következtetést vonhatjuk le: A kifejezés csak A=*igaz* és B=*hamis* estében lesz hamis, minden más esetben igaz lesz.

Két logikai kifejezést akkor tekintünk *egyenlőnek*, ha a benne szereplő változók tetszőleges egymással megegyező értékei mellett azonos igazságértékük van.

Két egyenlő logikai kifejezés *logikai szabályt* alkot.

Néhány érdekes és hasznos logikai szabály:

1. $\overline{\overline{A}} = A$ – a tagadás tagadásának szabálya;
2. $A \wedge (A \vee B) = A$ – az elnyelés szabálya;
3. $\overline{A \vee B} = \overline{A} \wedge \overline{B}$ – a diszjunkció tagadásának szabálya (De Morgan azonosság).

Amennyiben le kívánjuk ellenőrizni ezeket a szabályokat, létre kell hoznunk a bal és jobb oldalak igazságtáblázatait, és meg kell győződjünk azok egyenlőségéről. Próbáljuk meg elvégezni ezt önállóan.

2.45. ábra. Augustus De Morgan

Augustus De Morgan (1806–1871) skót matematikus (2.45. ábra) a Londoni University College professzora a londoni matematika társaság első elnöke. A logika területén elért eredményeit Boole-tól függetlenül publikálta 1847-ben.

Ellenőrző kérdések

1°. Mit tanulmányoz a logika? A matematikai logika?
2°. Mi a kijelentés? Hozzunk fel példákat igaz és hamis kijelentésekre
3*. Milyen mondat lehet kijelentés?
4°. Mi a logikai állandó?
5°. Hogyan képezzük az összetett kijelentéseket? Hozzunk fel példákat!
6°. Mi a tagadás? Hozzunk fel példákat!
7°. Mire használjuk az igazságtáblázatokat?
8°. Adjuk meg a tagadás igazságtáblázatát!
9°. Mi a konjunkció? Hozzunk fel példákat!
10°. Adjuk meg a konjunkció igazságtáblázatát!
11°. Mi a diszjunkció? Hozzunk fel példákat!
12*. Adjuk meg a diszjunkció igazságtáblázatát!
13°. Mi a logikai változó?
14°. Miben különbözik a logikai változó a kijelentéstől?

15°. Miből állnak a logikai kifejezések? Milyen értékeket vehetnek fel?

16°. Adjuk meg a logikai műveletek precedencia-szabályait!

17°. Ismertessük a logikai kifejezés igazságtáblázata kitöltésének algoritmusát!

18°. Mikor tekintünk két logikai kifejezést egyenlőnek? Mik a logikai szabályok?

Végezzük el a feladatokat!

1°. Állítsuk elő a kifejezések tagadását, és állapítsuk meg azok igazságértékét:

a) 3 osztója 545-nek.
b) A gépkocsi nem haladhat piros jelzésnél.
c) Vannak derékszögű paralelogrammák.
d) A $2x^2 - 3x + 1 = 0$ egyenletnek van egész gyöke.
e) Nem létezik 2-vel osztható természetes szám.
f) Van olyan egész szám, amelyik minden egész számmal osztható!

2°. Az alábbi kijelentések közül keressük ki a konjunkciókat és a diszjunkciókat, majd határozzuk meg igazságértéküket!

a) 27oszthető 3-mal és 9-cel.
b) $17 < 42 < 18$.
c) A 2-es szám prím és páros.
d) Az $ABC\Delta$ hegyes- derék- vagy tompaszögű.
e) Bármely paralelogramma átlói merőlegesek és felezik egymást.
f) $7^2 = 49$ és $(-7)^2 = 49$.
g) Ha a háromszög egyenlő szárú, akkor egyenlő oldalú is.
h) $21 \leqslant 21$
i) $21 \leqslant 18$.

Konjunkció	
Diszjunkció	
Egyik sem	

3°. Határozzuk meg a következő az $x > 0$ és $y > 0$ vagy $x < 0$ és $y < 0$ és $x > 0$ és nem $y < 0$ vagy $x < 0$ és $y > 0$ kifejezések igazságértékét, ha

a) $x = 5, y = 8$; b) $x = 5, y = -8$;
c) $x = -5, y = 8$; d) $x = -5, y = -8$;

4°. Adva vannak a következő kijelentések A=*"Miklós angolt tanul"* és B =*„Miklósnak 8-asa van informatikából"* Fogalmazzuk meg a következő kijelentéseket:

a) $A \wedge \overline{B}$; b) $\overline{A} \vee B$; c) $A \vee \overline{A} \wedge B$; d) $\overline{A} \vee \overline{B} \wedge A$.

5°. Nevezzük meg a műveletek sorrendjét:

a) $A \vee \overline{A} \wedge B$; c) $(A \vee \overline{B}) \wedge (\overline{A} \vee \overline{B})$;
b) $\overline{A} \vee \overline{B} \wedge A$; d) $\overline{A} \wedge \overline{B} \vee C \wedge D$.

6°. Állítsuk össze a következő kifejezések igazságtáblázatát:

a) $A \vee A \wedge B$; d) $(A \vee \overline{B}) \wedge (\overline{A} \vee \overline{B})$;
b) $A \vee \overline{A} \wedge B$; e) $\overline{A} \vee \overline{B}$;
c) $A \wedge B \vee \overline{B}$; f) $A \wedge \overline{B} \vee \overline{A} \wedge B$!

7°. Bizonyítsuk a következő szabályokat:

a) $A \wedge A = A$; d) $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$;
b) $\overline{A \wedge B} = \overline{A} \vee \overline{B}$; e) $A \wedge (A \vee B) = A$!
c) $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$;

2.8. Elágazások az algoritmusokban és a programokban

1. Mi az algoritmus? Milyen jelekből áll egy algoritmus folyamatábrája, és mi ezeknek a rendeltetése?
2. Milyen programokat (programrészeket) nevezünk szekvenciálisnak? Mi ezek sajátossága?
3. Mik a logikai kifejezések? Milyen értékeket vehetnek fel?
4. Milyen logikai műveleteket ismerünk? Adjuk meg ezek definícióit!
5. Milyen logikai függvények vannak az **Excel 2007** táblázatkezelőben? Hogy függenek ezeknek a függvényeknek az értékei az argumentumok értékeitől?

Elágazó algoritmusok

Az eddigiek során szekvenciális algoritmusokat vizsgáltunk, főképp aritmetikai számításokat tartalmazókat. Vizsgáljunk most meg néhány olyan feladatot, amelyek algoritmusa nem szekvenciális, azaz nem egymás után végrehajtandó utasításokból áll.

1. feladat. Számítsuk ki a $(a - b) / (c - d)$ kifejezés értékét, ahol a, b, c és d valós számok!

Figyeljük meg, hogy a kifejezés értékét nem lehet kiszámítani az a, b, c, d bármily értéke mellett. A kifejezés ugyanis osztást tartalmaz, amit nem lehet elvégezni, amennyiben az osztó értéke nulla. Az $(a - b) / (c - d)$ kifejezés értéke tehát csak akkor számítható ki, ha a c – d értéke nem nulla, egyébként pedig nem.

Ez azt jelenti, hogy az algoritmusnak két számot összehasonlító „*s=t?*”. lépést kell tartalmaznia, ahol s és t számok vagy számkifejezések, amelyeknek értéke ismert. Az ilyen **utasítást egy feltétel ellenőrzésének** nevezzük, melynek eredménye igaz (a feltétel teljesül), vagy hamis (a feltétel nem teljesül) lehet.

Másfelől az „*s=t?*” ellenőrzést kijelentésnek is tekinthetjük, amely az s és t értékétől függően **igaz** vagy **hamis** értéket vehet fel. Ez esetben a feltétel ellenőrzését egy **logikai kifejezés értéke kiszámításának** is tekinthetjük.

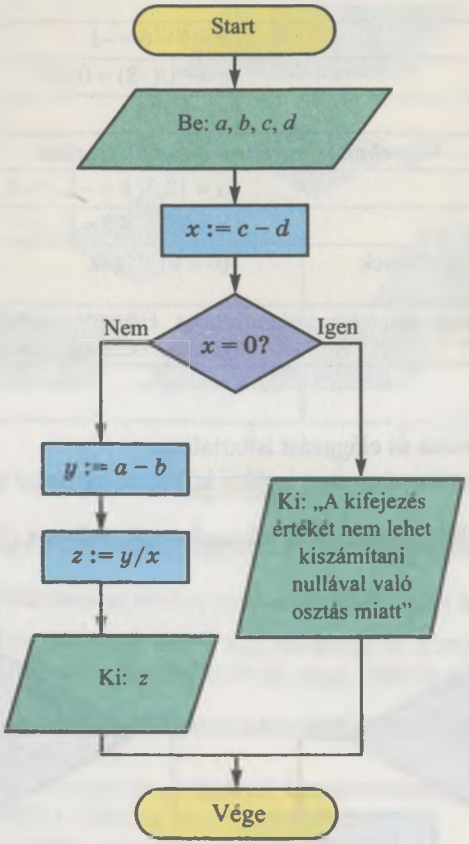
A feladat megoldásának algoritmusa tehát a következő:

1. Az a, b, c és d értékeinek bevitele.
2. $x := c - d$.
3. Az $x = 0$ logikai kifejezés értékének meghatározása.
4. Amennyiben x **igaz**, akkor irassuk ki az „A kifejezés értékét nem lehet kiszámitani nullával való osztás miatt” üzenetet, ha **hamis**, folyassuk az 5. lépéssel.
5. $y := a - b$.
6. $z := y/x$.
7. Kiíratjuk a z értékét.
8. Vége.

Az algoritmus 1–3. lépéseit a bemeneti adatok minden értéke mellett végre kell hajtani. A folytatásban az algoritmus végrehajtása a 3. lépésben kiszámított logikai változó értékétől függ. Amennyiben ez az érték hamis, végre kell hajtani az 5–8. lépéseket, ha igaz, ki kell írni az „A kifejezés értékét nem lehet kiszámítani nullával való osztás miatt” üzenetet.

Az algoritmus egy logikai kifejezés vizsgálatát tartalmazza, amit ◇-szal jelölünk. Mivel a logikai kifejezésnek két értéke lehet, ebből a vizsgálatból kétféle kivezető út lehetséges, az egyik a kifejezés igaz, a másik pedig hamis voltának esetére.

Az algoritmus folyamatábráját a 2.46. ábrán láthatjuk.



2.46. ábra A $(a - b) / (c - d)$ kifejezés kiszámítása algoritmusának folyamatábrája

Vizsgáljuk meg a 2.46. ábrán látható folyamatábrát az elágazástól a végéig.

Ennek a résznek az a különlegessége, hogy a futtatás során mindig lesznek olyan utasítások, amelyek végrehajtódnak és lesznek olyanok, amelyek nem. Ez a feltétel vizsgálatának eredményétől (a logikai kifejezés eredményétől) függ.

Az ilyen algoritmusrészletet **elágazásnak** nevezzük.

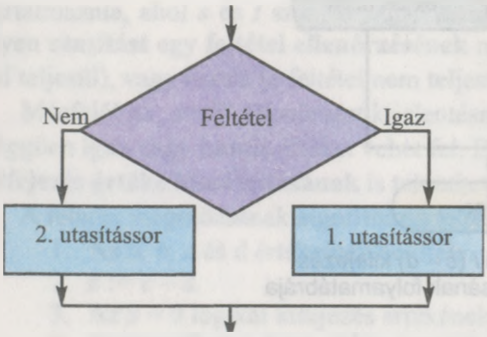
Vizsgáljuk meg az algoritmus végrehajtását két esetben. Ezeket az eseteket úgy választjuk meg, hogy láthassuk az algoritmus végrehajtását mind a két lehetséges esetben.

Utasítás	*A végrehajtás eredménye*
Végrehajtás menete az első esetben	
Be: *a, b, c, d*	$a = 5, b = 6, c = -3, d = 5$
$x := c - d$	$x = -3 - 5 = -8$
Az $x = 0$ logikai kifejezés értékének meghatározása	$(-8 = 0)$ = **hamis**
$y := a - b$	$x = 5 - 6 = -1$
$z := y / x$	$z = -1/(-8) = 0{,}125$
Ki: *z*	$z = 0{,}125$
Végrehajtás menete a második esetben	
Be: *a, b, c, d*	$a = 12{,}3, b = -1, c = 8{,}2, d = 8{,}2$
$x := c - d$	$x = 8{,}2 - 8{,}2 = 0$
Az $x = 0$ logikai kifejezés értékének meghatározása	$(0 = 0)$ = **igaz**
Ki: *„A kifejezés értékét nem lehet kiszámítani nullával való osztás miatt"*	*Az A kifejezés értékét nem lehet kiszámítani nullával való osztás miatt" üzenet.*

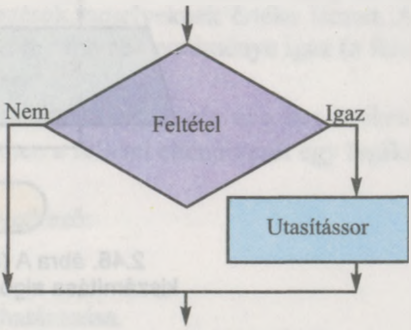
Figyeljük meg, hogy:

- az algoritmus szekvenciát és elágazást is tartalmaz;
- az elágazásban egyszerű és összetett logikai kifejezés egyaránt alkalmazható.

Az algoritmusok kétféle elágazást tartalmazhatnak: **teljeset** (2.47. ábra) és **nem teljeset** (2.48. ábra).



2.47. ábra. Teljes elágazás



2.48. ábra Nem teljes elágazás

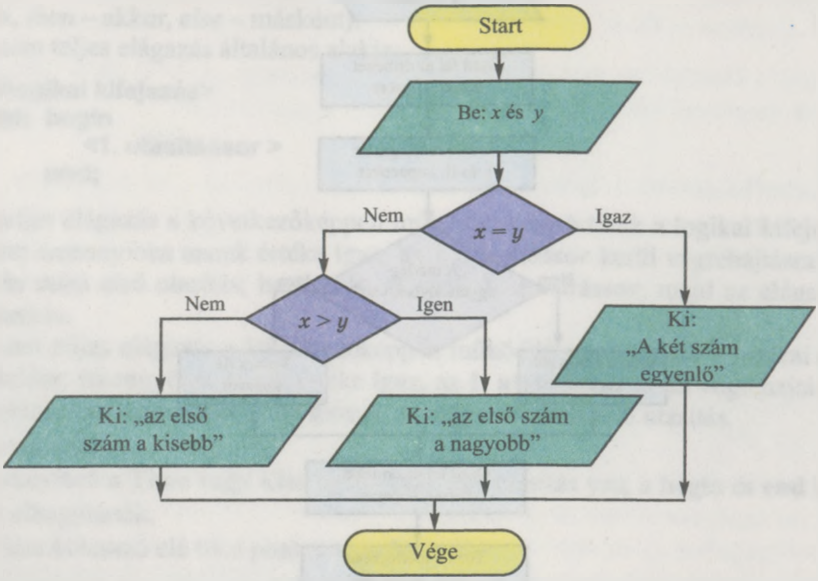
A teljes elágazás a következőképpen működik: az algoritmus végrehajtója **elvégzi a feltétel ellenőrzését (a logikai kifejezés értékelését)**; amennyiben a kifejezés értéke **igaz,** végrehajtja az **első utasítássort,** majd átlép az elágazást követő első utasításra; ha a kifejezés értéke **hamis,** végrehajtja az **második utasítássort** és átlép az elágazás utáni első utasításra.

A nem teljes elágazás végrehajtása ettől abban tér el, hogy ha a kifejezés értéke hamis, akkor azonnal az elágazás utáni első utasítás következik.

Az elágazás szekvenciákat és más elágazásokat is tartalmazhat. Nézzünk egy példát arra, amikor az elágazás egy másik elágazást tartalmaz.

2. feladat. Adva van két szám. Állapítsuk meg, hogy ezek egyenlők-e és ha nem, melyik a nagyobb.

Az algoritmus folyamatábrája a 2.42. ábrán látható:



2.49 ábra. A 2. feladat algoritmusának folyamatábrája

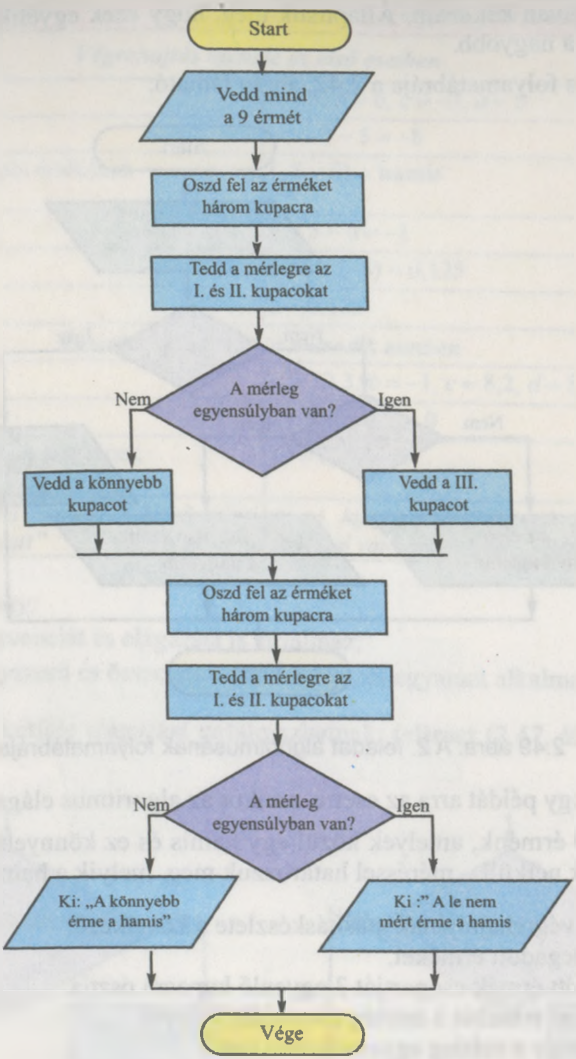
Lássunk még egy példát arra az esetre, amikor az algoritmus elágazást tartalmaz.

3. feladat. Van 9 érménk, amelyek közül egy hamis és ez könnyebb a többinél. Két – súlyok nélküli – méréssel határozzuk meg, melyik a hamis érme.

Az algoritmus végrehajtójának utasításkészlete a következő:
1. Elveszi a megadott érméket.
2. A kiválasztott érmék csoportját 3 egyenlő kupacra osztja.
3. A kiválasztott érméket a mérleg tányérjára helyezi.
4. Ellenőrzi, hogy a mérleg egyensúlyban van-e.
5. Meghatározza, melyik érmecsoport a könnyebb.
6. Közli az eredményt.

A fenti végrehajtó számára a 3. feladat megoldásának algoritmusa a következő:
1. Vedd el mind a 9 érmét.
2. Oszd fel az érmék csoportját 3 egyenlő kupacra.
3. Tedd a mérlegre az első két csoportot.
4. Ellenőrizd, egyensúlyban van-e a mérleg.
5. Ha **igen**, vedd a harmadik csoportot, ha **nem**, vedd a könnyebbet.
6. Oszd fel a kiválasztott csoportot 3 egyenlő kupacra.
7. Tedd a mérlegre az első két csoportot.
8. Ellenőrizd, egyensúlyban van-e a mérleg.
9. Ha **igen,** közöld: *A hamis az az érme, amelyiket nem tettünk a mérlegre*; ha **nem,** közöld: *A könnyebb érme a hamis.*

A 2.50. ábrán az algoritmus folyamatábráját láthatjuk.



2.50. ábra A 3. feladat algoritmusának folyamatábrája

Figyeljük meg, hogy a 3. feladat algoritmusában melyik utasítások kerülnek végrehajtásra, az attól függ, melyik kupacba kerül a hamis érme. Ha az első mérésnél a mérleg nem lesz egyensúlyban, akkor a „Vedd a harmadik kupacot" utasítás nem kerül végrehajtásra. Ha azonban az érmék eloszlása más és a mérleg egyensúlyba kerül, akkor viszont a „Vedd a harmadik kupacot" nem kerül végrehajtásra.

Elágazások Delphi-ben

A teljes elágazás általános alakja a **Delphi**-ben a következő:

```
If <logikai kifejezés>
Then  begin
          < 1. utasítássor>
```

```
        end
Else    begin
            < 2. utasítássor>
        end;
```

(*if* – ha, *then* – akkor, *else* – másként).

A nem teljes elágazás általános alakja:

```
If <logikai kifejezés>
Then  begin
            <1. utasítássor >
        end;
```

A teljes elágazás a következőképpen működik: megtörténik a logikai kifejezés kiértékelése; amennyiben ennek értéke **igaz,** az **1. utasítássor** kerül végrehajtásra, majd az elágazás utáni első utasítás; ha **hamis,** akkor a **2. utasítássor**, majd az elágazás utáni első utasítás.

A nem teljes elágazás a következőképpen működik: megtörténik a logikai kifejezés kiértékelése; amennyiben ennek értéke **igaz,** az **1. utasítássor** kerül végrehajtásra, majd az elágazás utáni első utasítás; ha **hamis**, az elágazás utáni első utasítás.

Fontos tudni:

- amennyiben a **Then** vagy **Else** ágban csak egy utasítás van, a **begin** és **end** kulcsszavak elhagyhatók.
- az **Else** kulcsszó elé tilos pontosvesszőt tenni.

Az 1. feladat algoritmusának egy részlete **Delphi**-ben (2.46. ábra) a következő lesz:

```
a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
c := StrToFloat (Edit3.Text);
d := StrToFloat (Edit4.Text);
x := c–d;
If x = 0
Then Label1.Caption := 'A kifejezés értékét nem lehet kiszámítani nullával való
osztás miatt'
Else  begin
          y := a–b;
          z := y/x;
          Label1.Caption := FloatToStr (z);
      end;
```

Figyeljünk oda a következőre: a *z* értékének kiszámítását két utasítás helyett eggyel is leírhatjuk: $z := (a–b)/x$.

A logikai kifejezés az elágazás feltételében lehet egyszerű vagy összetett is. Az összetett kifejezéseket egyszerű kifejezésekből képezzük az **and** (és), **or** (vagy) és **not** (nem) logikai műveletek segítségével. A logikai művelet precedenciája megegyezik a matematikai logikában elfogadottal.

Nézzünk néhány példát a logikai kifejezések alkalmazására.

4. feladat. Számítsuk ki a $y = \begin{cases} 2x - 12, & \text{ha} \quad -3 < x < 5, \\ 7 - 8x & \text{minden más esetben } x. \end{cases}$ kifejezés értékét!

A feladat megoldására szolgáló program részlete az alábbi:

```
x := StrToFloat (Edit1.Text);
If (x > –3) and (x < 5)
Then y := 2*x – 12
Else y := 7 – 8*x;
Label1.Caption := FloatToStr (y);
```

Az elágazás **Delphi**-ben más elágazásokat is tartalmazhat. A 2.49. ábrán látható algoritmusnak megfelelő kódrészlet:

```
If x = y
Then Label1.Caption := 'Számok egyenlők'
Else If x > y
    Then Label1.Caption := 'Az első szám nagyobb'
    Else Label1.Caption := 'Az első szám kisebb';
```

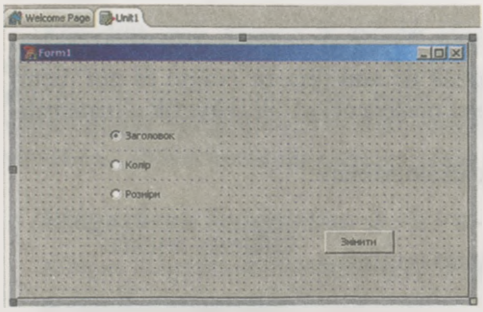
Választógombok és jelölőnégyzetek

Projektjeinkben az elágazás megvalósítására **jelölőnégyzeteket** és **választógombokat** is használhatunk. Az eddig megismert alkalmazói szoftverek (szövegszerkesztő, grafikus szerkesztők stb.) alkalmazása során ezeket a vezérlőket: a választógombot és a jelölőnégyzetet már megismerhettük.

A választógombok közül egy csoportban pontosan egy lehet bekapcsolva, a jelölőnégyzetek közül akár több is vagy egyetlen egy se. A választógombok arra szolgálnak, hogy a program futása közben kiválaszthassunk egyet a lehetőségek közül. Ilyen lehet egy objektum valamely tulajdonságának beállítása, ha ez a tulajdonság meghatározott értékeket vehet fel. A jelölőnégyzetek arra szolgálnak, hogy a lehetőségek közül többet is választhassunk, ha például az objektum tulajdonsága egyidejűleg több értéket is felvehet.

Ha választógombot szeretnénk elhelyezni a form-on, akkor kattintsunk a **ToolPalette** lap **TRadioButton** (kerek gomb) elemére, ha pedig jelölőnégyzetet, akkor ugyanezen lap **TCheckBox** (választógomb) elemére.

A választógombok és a jelölőnégyzetek is rendelkeznek azoknak a tulajdonságoknak a többségével, amelyekkel a form-ok, mezők, gombok és feliratok, ezeken kívül rendelkeznek a **Checked** (kiválasztva) tulajdonsággal, amely a *True* (igaz) és *False* (hamis) értékeket veheti fel attól függően, hogy az objektum ki volt választva vagy sem. Ez okból ezeket a vezérlőket az elágazások megvalósítása során is felhasználhatjuk.



2.51. ábra. A 3 választógombot és egy gombot tartalmazó form

Tegyük fel, hogy a program futása közben az ablak címsora, színe és mérete közül csak az egyiket szeretnénk megváltoztatni.

Helyezzünk el a form-on három választógombot és egy gombot (2.51. ábra). Ha választógombokat helyezünk a form-ra, azok **Cheked** tulajdonsága alapértlmezés szerint *False* lesz. Állítsuk a gombok egyikének ezt a tulajdonságát a futtatás előtt *True*-ra. Ekkor ez a form-on ki lesz jelölve (2.51. ábra). Ilyen marad a projekt futtatásának megkezdése után is.

Hozzuk létre a gomb **OnClick** eseményét feldolgozó eljárást:

```
If RadioButton1.Checked
Then Form1.Caption := 'A címsor megváltozott'
Else  If RadioButton2.Checked
      Then Form1.Color := clRed
      Else  begin
                Form1.Width := 600;
                Form1.Height := 300;
            end;
```

A programrészlet a **RadioButton1.Checked** értékének ellenőrzésével kezdődik.

Amennyiben az **OnClick** esemény bekövetkezésekor az első választógomb volt bekapcsolva, akkor **RadioButton1.Checked=true.** Ez esetben az ablak címsora meg fog változni, a programrészlet többi utasítását kihagyjuk.

Amennyiben az **OnClick** esemény bekövetkezésekor a második választógomb volt bekapcsolva, akkor **RadioButton1.Checked= false.** Ekkor az **Else** utáni utasítás kerül végrehajtásra, ami egy újabb elágazás. A második elágazás végrehajtása a **RadioButton2.Checked** ellenőrzésével kezdődik. Mivel ez a választógomb volt bekapcsolva, ez az érték true. Ekkor tehát az ablak színe fog megváltozni, a többi utasítás végrehajtása elmarad.

Amennyiben az **OnClick** esemény bekövetkezésekor a harmadik gomb volt kiválasztva, a **RadioButton1.Checked** állapota = **false**. Ezért az első elágazásban az **Else** utáni részre kerül a vezérlés. Mivel most a **RadioButton2.Checked** értéke is **false,** a második elágazásban is az **Else** utáni rész kerül végrehajtásra, ami pedig az ablak méretének megváltozását eredményezi.

Ha a projekt futtatása előtt egyik választógomb sem volt bekapcsolva, a **Checked** az alapértelmezés szerinti értéken marad. Ha a futtatás során se kapcsoljuk be egyik választógombot se, akkor az ablak mérete változik meg (próbáljuk meg önállóan meghatározni, miért). Hogy ez ne így történjen, vagy kötelezően be kell kapcsolni a választógombok egyikét, vagy az **Else** ágba be kell írni még egy elágazást:

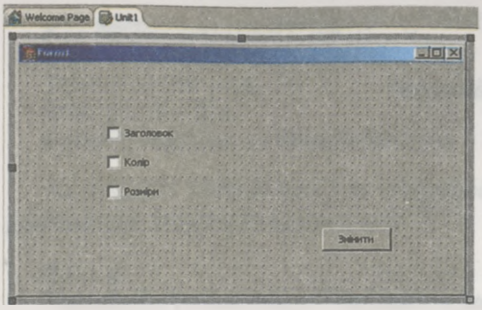
```
If RadioButton1.Checked
Then Form1.Caption := 'A címsor megváltozott'
Else  If RadioButton2.Checked
      Then Form1.Color := clRed
      Else  If RadioButton2.Checked
            Then
               begin
                  Form1.Width := 600;
                  Form1.Height := 300;
               end;
```

Magyarázzuk meg a programunk viselkedését abban az esetben, ha egyik választógomb sincs bekapcsolva.

Próbáljuk most a jelölőnégyzeteket használni az elágazás megvalósítására.

Tegyük fel, hogy nemcsak egy ablaktulajdonságot szeretnénk megváltoztatni a program futtatása során, hanem néhányat közülük, esetleg egyiket sem.

Helyezzünk a form-ra egy gombot és három jelölőnégyzetet (2.52. ábra). A jelölőnégyzetek **Checked** tulajdonságának alapértelmezett értéke **False**. Mivel a program futása során



2.52. ábra. Ablak három jelölőnégyzettel és egy gombbal

előfordulhat, hogy egyik jelölő sincs bekapcsolva, célszerű ezeket az értékeket változatlanul hagyni.

Hozzuk létre a gomb **OnClick** eseménykezelőjét, és írjunk bele három elágazást (az elágazások nincsenek egymásba ágyazva):

```
If CheckBox.Checked
Then Form1.Caption := 'A címsor megváltozott';
If CheckBox.Checked
Then Form1.Color := clRed;
If CheckBox.Checked
Then  begin
        Form1.Width := 500;
        Form1.Height := 200;
      end;
```

Az elágazások mindegyike a többitől függetlenül ellenőrzi a jelölőnégyzetek állapotának ellenőrzését és ezektől függően megváltoztatja vagy változatlanul hagyja az ablak megfelelő tulajdonságát.

A vizsgált példákból az alábbi következtetéseket vonhatjuk le:

- a választógombokat akkor alkalmazzuk, ha a lehetőségek közül csak egyiket lehet kiválasztani;
- a választógombok állapotát feldolgozó eljárásban egymásba ágyazott elágazásokat használunk;
- a jelölőnégyzeteket akkor alkalmazzuk, ha a lehetőségek közül többet, vagy akár egyet sem szeretnénk kiválasztani;
- A jelölőnégyzetek állapotát feldolgozó programrészlet olyan elágazásokat tartalmaz, amelyek egymástól függetlenek.

Ellenőrző kérdések

1°. Milyen eredménye lehet egy feltétel ellenőrzésének (vagy a logikai kifejezés kiértékelésének)?
2°. Hogyan jelöljük egy feltétel ellenőrzését a folyamatábrán?
3°. Ismertessük az elágazás jellegzetességeit!
4*. Írjuk fel a teljes és a nem teljes elágazás folyamatábráját! Ismertessük azok végrehajtásának jellegzetességeit!
5*. Magyarázzuk meg a teljes és a nem teljes elágazás működését!
6*. Hozzunk fel olyan nyelvi vagy matematikai példákat, amelyek elágazást tartalmaznak!
7*. Hozzunk fel életszerű példákat, ahol elágazásos algoritmusokat alkalmazunk!
8°. Ismertessük a **Delphi**-ben megírt elágazás általános alakját!
9*. Magyarázzuk el a teljes és a nem teljes elágazás működését **Delphi**-ben!
10°. Milyen műveleteket alkalmazunk összetett logikai kifejezések létrehozására **Delphi**-ben?
11*. Mire használhatjuk a választógombokat és a jelölőnégyszögeket?
12°. Mit jelent a választógombok és a jelölőnégyzetek esetében a **Checked** tulajdonság?
13*. Miben különböznek a választógombokat és a jelölőnégyzeteket tartalmazó projektekben használt elágazások?
14*. Magyarázzuk meg a következő programrészlet működését:

a)
```
If <1. logikai kifejezés>
Then begin
        <1. utasítássor>
     end
Else If <2. logikai kifejezés>
     Then begin
             <2. utasítássor>
          end
     Else begin
             <3. utasítássor>
          end;
```

b)
```
If <1. logikai kifejezés>
Then begin
        <1. utasítássor>
     end;
If < 2. logikai kifejezés>
Then begin
        <2. utasítássor>
     end;
```

Végezzük el a feladatokat!

1°. Hajtsuk végre a következő algoritmust!

1. Gondoljunk két számot!
2. Számítsuk ki ezek összegét!
3. Döntsük el, hogy az összeg meghaladja-e a 30-at.
4. Ha igen, küldjünk „Nagyobb" jelentést, és lépjünk a 8. lépésre.
5. Döntsük el, hogy az összeg kisebb-e, mint 30.
6. Ha igen, küldjünk „Kisebb" jelentést, és lépjünk a 8. lépésre. Ha nem, lépjünk a 7. lépésre.
7. Küldjünk „egyenlő" jelentést.
8. Vége.

Milyen számokat gondoltunk és mi lett az eredmény? Hogy nevezhetnénk el a feladatot?

2°. Hozzuk létre a 1. feladat folyamatábráját! Hajtsuk végre az algoritmust három különböző esetre, amelyek az algoritmus három lehetséges kimenetének felelnek meg!

3°. Hozzuk létre az $(a + b) - c : a$. kifejezés kiszámításának algoritmusát! Hajtsuk végre az algoritmust a, b és c különböző értékeinél! Válasszuk meg ezeket az értékeket úgy, hogy az algoritmus viselkedése minden érték mellett más legyen!

4*. Hozzuk létre az $(a + b) - c : (a-2b)$. kifejezés kiszámításának algoritmusát! Hajtsuk végre az algoritmust a, b és c különböző értékeinél! Válasszuk meg ezeket az értékeket úgy, hogy az algoritmus viselkedése minden érték mellett más legyen!

5*. Hozzuk létre az $ax = b$ egyenlet megoldásának algoritmusát! Hajtsuk végre az algoritmust, ha: 1) $a = 2$; $b = -8$; 2) $a = 20$; $b = 0$; 3) $a = 0$; $b = 12$; 4) $a = 0$; $b = 0$!

6°. Hozzuk létre az $ax + b = c$ egyenlet megoldásának algoritmusát! Hajtsuk végre az algoritmust, ha: 1) $a = 2$; $b = -8$; $c = 18$; 2) $a = 20$; $b = 5$; $c = 5$; 3) $a = 0$; $b = 12$; $c = 3$; 4) $a = 0$; $b = 10$; $c = 10$.

7*. Hozzuk létre a másodfokú egyenlet megoldásának algoritmusát! Hajtsuk végre az algoritmust az együtthatók különböző értékei mellett!

8*. Van néhány egyenlő nagyságú érménk, melyek közül az egyik hamis és ez nehezebb a többinél. Hozzunk létre algoritmust, amelynek segítségével a lehető legkevesebb méréssel ki lehet választani a hamis érmét! Hajtsuk végre az algoritmust, ha: 1) 3; 2) 9; 3) 27; 4) 30 érménk van!

9*. Hozzunk létre projektet, amely két nem egyenlő valós szám közül kiválasztja a nagyobbat. Mentsük a projektet a megfelelő mappába.

10°. Hozzunk létre projektet, amely két valós szám közül kiválasztja a kisebbet! A projekt jelezze, ha a két szám egyenlő. Mentsük a projektet a megfelelő mappába!

11°. Hozzunk létre projektet, amely meghatározza egy szám abszolút értékét! Mentsük a projektet a megfelelő mappába!

12°. Hozzunk létre projektet, amely három valós szám közül kiválasztja a legkisebbet! Mentsük a projektet a megfelelő mappába!

13°. Hozzunk létre projektet, amely a háromszög két szögének alapján meghatározza a harmadik szög fokmértékét és jelzi, hogy a háromszög hegyes- derék- vagy tompaszögű! Mentsük a projektet a megfelelő mappába!

14. Hozzunk létre projektet, amelyben egy gomb 20 pixellel jobbra mozdul, amikor az egérrel rámutatunk. Ha a gomb eltűnik az ablak jobb szélén, akkor jelenjen meg annak bal szélén. Mentsük a projektet a megfelelő mappába!

15. Hozzunk létre projektet, amely meghatározza, hogy 4 adott hosszúságú szakaszból szerkeszthető-e paralelogramma! Mentsük a projektet a megfelelő mappába.

16. Hozzunk létre projektet, amely kiszámítja az $y = \begin{cases} 15 - 3x, & \text{ha } -1 < x < 4, \\ 6 + 4x & \text{miden más esetben } x \end{cases}$ kifejezés értékét! Mentsük a projektet a megfelelő mappába!

17. Hozzunk létre projektet, amely a választógombok állapotától függően kiszámítja két szám összegét, különbségét, szorzatát vagy hányadosát! Mentsük a projektet a megfelelő mappába!

18. Hozzunk létre projektet, amely a jelölőnégyzet állapotától függően kiszámítja két szám összege, különbsége, szorzata és hányadosa közül néhányat! Mentsük a projektet a megfelelő mappába!

19. Egy munkás alkatrészeket készít. Ismert, hogy mennyit kell elkészítenie egy hónap alatt (tervezett mennyiség), mennyit készített el ténylegesen, továbbá az alkatrész ára. Amennyiben a munkás teljesíti a tervet 10%, ha túlteljesíti, akkor 30% jutalomban részesül. Amennyiben nem teljesíti a tervet, 20% megvonást kap. Hozzunk létre projektet, amely kiszámítja a munkás keresetét a teljesítménye függvényében! Mentsük a projektet a megfelelő mappába!

8. számú gyakorlati munka Elágazásokat tartalmazó programok létrehozása

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert és hozzunk létre benne egy új projektet!

2. Helyezzünk a form-ra vezérlőelemeket a következő feladat megoldásához: *Ismerjük két téglalap oldalait. Ki kell számítani és össze kell hasonlítani azok területét. Meg kell határozni, van-e a téglalapok között négyzet. Meg kell határozni, hogy elfér-e az egyik téglalap a másikban. Az eredményeket ki kell íratni a képernyőre.* A projektben:
   a) az első gomb **OnClick** eseménykezelője számítsa ki a területeket és írja ki, hogy melyik terület a nagyobb;
   b) a második gomb **OnClick** eseménykezelője egy jelölőnégyzet állapotától függően döntse el, hogy a kiválasztott téglalap négyzet-e;
   c) a harmadik gomb **OnClick** eseménykezelője a választógomb állapotától függően döntse el, hogy befér-e az első téglalap a másodikba, vagy a második az elsőbe;
   d) futtassuk a megfelelő eljárásokat és győződjünk meg azok helyességéről.

3. Mentsük a projektet a **8_gyakorlati** mappába!

2.9. Ciklikus algoritmusok és programok

1. Milyen folyamatokat nevezünk ciklikusnak? Hozzunk fel példákat!
2. Melyek a szekvenciális algoritmusok sajátosságai? Az elágazásoké?
3. Milyen a teljes elágazás általános alakja **Delphi**-ben? Hogy működik ez az elágazás?
4. Milyen a nemteljes elágazás általános alakja **Delphi**-ben? Hogy működik ez az elágazás?
5. Milyen célra és hogyan használjuk a választógombokat és a jelölőnégyzeteket?

Ciklikus algoritmusok

A szekvenciális és elágazásos algoritmusok minden utasítása sem kerül egynél többször végrehajtásra.

Sok olyan algoritmus is létezik azonban, amelyekben az utasításokat egynél többször kell végrehajtani. Most ezek közül vizsgálunk meg néhányat.

1. feladat. Van egy dézsánk és egy vedrünk. A vödör segítségével töltsük meg a dézsát vízzel. (Feltételezzük, hogy létezik egy forrás, ahol a vödröt megtölthetjük, és a forrás vízhozama elegendő a dézsa feltöltéséhez).

Tegyük fel továbbá, hogy az algoritmus végrehajtója az alábbi utasításkészletet ismeri:

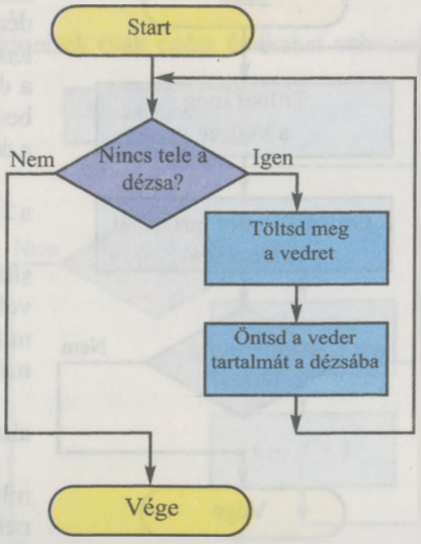
1. Töltsd meg a vedret vízzel.
2. Öntsd a vödör tartalmát a dézsába.
3. Ellenőrizd a „Nincs-e tele a dézsa?" feltételt.

Mivel előre nem ismert, nincs-e tele a dézsa, ellenőrizni kell a „Nincs-e tele a dézsa?" feltételt. Ha ennek értéke **igaz**, akkor meg kell tölteni a vedret, át kell önteni annak tartalmát a dézsába, majd újból ellenőrizni kell a „Nincs-e tele a dézsa?" feltételt. Ezt addig kell folytatni, amíg az ellenőrzés **hamis** eredményre nem vezet. Azután az algoritmus működése véget ér.

Az algoritmus mondatszerű megfogalmazása a következő (folyamatábráját a 2.53. ábrán láthatjuk).

1. Ellenőrizd a „Nincs-e tele a dézsa?" feltételt.
2. Ha az ellenőrzés eredménye **igaz**, lépj a 3. pontra, ha **hamis**, vége.
3. Töltsd meg a vedret vízzel.
4. Öntsd a vödör tartalmát a dézsába.
5. Lépj az 1. pontra.

Előfordulhat, hogy az algoritmus végrehajtása során a 3–5. utasításokat többször is végre kell hajtani. Az, hogy ezeket az utasításokat újra végre kell-e hajtani, az 1. ellenőrzés eredményétől függ. Ha ez **igaz**, akkor az utasításokat még legalább egyszer végre kell hajtani, ha **hamis**, akkor nem.



2.53. ábra. Az 1. feladat megoldásának folyamatábrája

Fontos, hogy a 3–5. utasításokat *esetlegesen többször is végre kell hajtani, de ez nem feltétlenül van így.* Előfordulhat, hogy az első veder víz betöltése után a dézsa tele lesz, akkor az algoritmus véget ér, sőt, akár az is, hogy a dézsa eleve tele volt, ekkor ezeket az utasításokat egyetlen egyszer sem kell végrehajtani.

Az algoritmusnak azt a részét, amelynek végrehajtása egynél többször ismétlődhet az algoritmus végrehajtása során *ciklusnak* (iterációnak) nevezzük. Az olyan algoritmusokat, amelyekben ciklusok fordulnak elő, *ciklusokat tartalmazó algoritmusoknak* nevezzük.

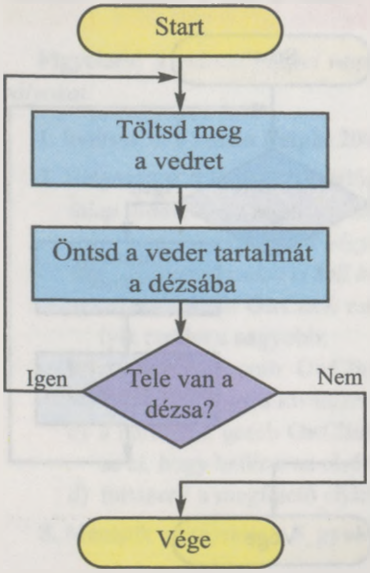
A vizsgált algoritmusban az iteráció három utasítást tartalmaz: a kilépési feltétel ellenőrzését és a ***ciklusmagot*** alkotó két utasítást.

2.54. ábra. Az elölteszető ciklus folyamatábrája

A vizsgált ciklust **elöltesztelő ciklusnak** nevezzük, mivel a kilépési feltétel ellenőrzése a ciklusmag végrehajtása előtt történik.

A 2.54. ábrán az elöltesztelő ciklus általános alakját láthatjuk. A ciklus végrehajtása a következőképpen történik: először is **ellenőrzésre kerül a kilépési feltétel (kiértékelődik a logikai kifejezés).** Amennyiben a feltétel teljesül (a kifejezés értéke igaz), végrehajtásra kerülnek a **ciklusmag utasításai,** amelyek után újra **megtörténik a kilépési feltétel ellenőrzése.** Amennyiben a **feltétel nem teljesül (a kifejezés értéke hamis),** átlépünk a ciklust követő első utasításra.

Amennyiben ismert lenne, hogy kezdetben a dézsa üres, az algoritmus végrehajtójának felesleges lenne kezdetben ellenőrizni a „Nincs-e tele a dézsa?" feltételt. Ez esetben legalább egyszer meg kellene tölteni a vedret és át kellene önteni annak tartalmát a dézsába. A „Nincs-e tele a dézsa?" (vagy, amennyiben az algoritmus végrehajtója ezt ismeri, a „Tele van-e a dézsa?")feltételt elég lenne ezután ellenőrizni.



2.55. ábra A módosított 1. feladat megoldásának folyamatábrája

E feladat megoldásának a folyamatábráját láthatjuk a 2.55. ábrán.

A fenti ciklus esetében a ciklusmagot alkotó utasítások legalább egyszer végrehajtásra kerülnek, mivel a kilépési feltétel ellenőrzése ezek végrehajtása után történik. Az ilyen ciklust **hátultesztelő ciklusnak** nevezzük.

A 2.54. ábrán a hátultesztelő ciklus általános alakjának folyamatábráját láthatjuk.

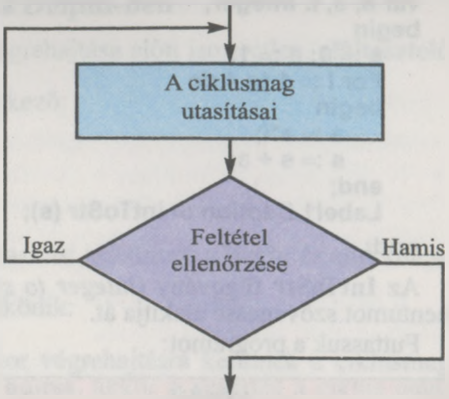
A ciklus végrehajtása a következőképpen történik: először a **ciklusmagot alkotó utasítások** kerülnek végrehajtásra, majd pedig megtörténik a **kilépési feltétel ellenőrzése.** Amennyiben a feltétel nem teljesül (a logikai kifejezés értéke **hamis**), akkor újra

végre kell hajtani a ciklusmagot; ha a kifejezés értéke igaz, a vezérlés a ciklus utáni első kifejezésre kerül.

Figyeljük meg, hogy amennyiben a 2.55. folyamatábrán a „Nincs-e tele a dézsa?" feltételt használnánk, akkor a ciklusnak a kifejezés **igaz** értékénél kellene folytatódnia és a hamis értéknél pedig ki kellene lépni a ciklusból.

Megvizsgáltuk mind a **három alapvető algoritmikus szerkezetet: a szekvenciát, az elágazást és a ciklust (iterációt).** Bizonyított, hogy amennyiben egy feladat megoldható, akkor annak algoritmusa e három alapvető szerkezet segítségével előállítható.

Az algoritmusok többsége mindhárom alapvető szerkezetet tartalmazza.



2.56. ábra. A hátultesztelt ciklus folyamatábrája

Számlálós ciklus Delphi-ben

A **Delphi** programozási nyelvben többféle eszköz áll rendelkezésünkre a ciklus megvalósítására. Ezek egyike a **számlálós ciklus.** Ennek alkalmazása akkor célszerű, ha már a ciklus végrehajtása előtt pontosan ismerjük, hogy hányszor kell azt végrehajtani. A ciklus általános alakja a következő:

```
for <ciklusváltozó> := <kifejezés1> to <kifejezés2> do
begin
    <a ciklusmag utasításai>
end;
```

A **for...to...do** sort (*for* – valami részére, *to* – valameddig, *do* – végezni) a **ciklus fejének** nevezzük. A változó, amelyet ebben a sorban az értékadás jele előtt áll, **ciklusváltozó.**

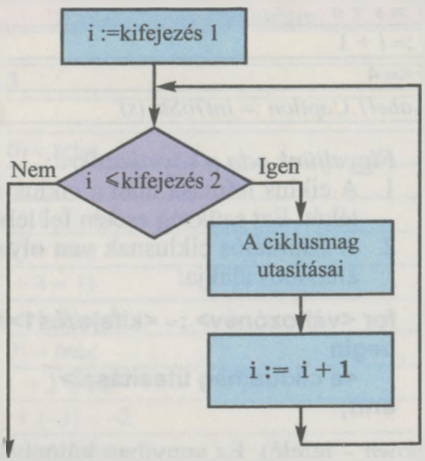
A **ciklusváltozó, a kifejezés1 és kifejezés2** kifejezések csak egész értékeket vehetnek fel. Ha a ciklusmag csak egy utasításból áll, a **begin** és **end** kulcsszavak elhagyhatók.

A számlálós ciklus a következőképpen működik:

1. A ciklusváltozó felveszi a **kifejezés1** értékét.
2. Kiértékelésre kerül az **i ⩽ kifejezés2** feltétel.
3. Ha a kifejezés értéke **igaz,** a vezérlés a ciklusmag utasításaira kerül, amennyiben **hamis,** a ciklus utáni első utasításra.
4. A cilusváltozó értékét eggyel növeljük.
5. A 2. pontra lépünk.

A számlálós ciklus folyamatábráját a 2.57. ábrán láthatjuk.

Vizsgáljuk meg példaként az alábbi, 1! + 2! + 3! +4! (emlékeztetőül $n! = 1 \cdot 2 \cdot 3 \cdot ... \cdot n$) kifejezés értékét kiszámító programrészt:



2.57. ábra A számlálós ciklus folyamatábrája

```
var a, s, i: Integer;
begin
    s := 0; a := 1;
    For i := 1 to 4 do
    begin
        a := a*i;
        s := s + a;
    end;
    Label1.Caption := IntToStr (s);
end;
```

Az **IntToStr** függvény (*integer to string* – egész szöveggé) az egész értékű argumentumot szövegessé alakítja át.

Futtassuk a programot:

Utasítás	*A végrehajtás eredménye*
s := 0	*s* = 0
a := 1	*a* = 1
i := 1	*i* = 1
i <= 4	(1 <= 4) = ***true***
a := *a***i*	*a* = 1 *1= 1
s := *s* + *a*	*s* = 0 + 1 = 1
i := *i* + 1	*i* = 2
i <= 4	(2 <= 4) = ***true***
a := *a***i*	*a* = 1 *2= 2
s := *s* + *a*	*s* = 1 + 2 = 3
i := *i* + 1	*i* = 3
i <= 4	(3 <= 4)= ***true***
a := *a***i*	*a* = 2 *3= 6
s := *s* + *a*	*s* = 3 + 6 = 9
i := *i* + 1	*i* = 4
i <= 4	(4 <= 4) = ***true***
a := *a***i*	*a* = 6 *4= 24
s := *s* + *a*	*s* = 9 + 24 = 33
i := *i* + 1	*i* = 5
i <= 4	(5 <= 4) = ***false***
Label1.Caption := *IntToStr (s)*	*Label1.Caption* = 33

Figyeljünk oda a következőkre:

1. A ciklus lefutása után a ciklusváltozó értéke 1-gyel haladja meg a **kifejezés2** értékét. Ezt szükség esetén fel lehet használni.
2. A számlálós ciklusnak van olyan változata is, amelyik visszafelé számol. Ennek általános alakja:

```
for <változónév> := <kifejezés1> downto <kifejezés2> do
begin
    <a ciklusmag utasításai>
end;
```

(*down* – lefelé). Ez annyiban különbözik a már megismert számlálós ciklustól, hogy a ciklusszámláló értéke minden lépésben eggyel csökken.

Elöltesztelő és hátultesztelő ciklusok a Delphi-ben

Amennyiben az ismétlések száma a ciklus végrehajtása előtt ismeretlen, **elöltesztelő** vagy **hátultesztelő** ciklusokat alkalmazunk.

Az elöltesztelő ciklus általános alakja a következő:

```
while <logikai kifejezés> do
begin
   <a ciklusmag utasításai>
end;
```

(*while* – amíg). Ha a ciklusmag csak egyetlen utasítást tartalmaz, a **begin** és **end** kulcsszavak elhagyhatók.

Az elöltesztelő ciklus a következőképpen működik:

1. Kiértékelődik a logikai kifejezés.
2. Amennyiben a kifejezés értéke **igaz**, akkor végrehajtásra kerülnek a ciklusmag utasításai, majd az 1. pontra lépünk; ha **hamis**, akkor a vezérlés a ciklus utáni első utasításra kerül.

Vizsgáljuk meg a ciklus működését a következő példán.

2. feladat. Számítsuk ki a számtani sorozat pozitív tagjainak összegét, ha ismerjük annak pozitív első tagját és negatív különbségét!

A sorozat első tagját az **Edit1**, a különbségét az **Edit2** beviteli mezőn keresztül adjuk meg.

```
var a, d, s: Real;
begin
   a := StrToFloat (Edit1.Text);
   d := StrToFloat (Edit2.Text);
   s := 0;
   while a > 0 do
      begin
         s := s + a;
         a := a + d;
      end;
   Label1.Caption := FloatToStr (s);
end;
```

Futtassuk le a programot az *a* és a *d* valamely értékei mellett.

Utasítás	*A végrehajtás eredménye*
a := StrToFloat (Edit1.Text)	$a = 7$
d := StrToFloat (Edit2.Text)	$d = -3$
$s := 0$	$s = 0$
$a > 0$	$(7 > 0) =$ ***true***
$s := s + a$	$s = 0 + 7 = 7$
$a := a + d$	$a = 7 + (-3) = 4$
$a > 0$	$(4 > 0) =$ ***true***
$s := s + a$	$s = 7 + 4 = 11$
$a := a + d$	$a = 4 + (-3) = 1$
$a > 0$	$(1 > 0) =$ ***true***
$s := s + a$	$s = 11 + 1 = 12$
$a := a + d$	$a = 1 + (-3) = -2$
$a > 0$	$(-2 > 0) =$ ***false***
Label1.Caption := FloatToStr (s)	*Label1.Caption* $= 12$

A **hátultesztelő ciklus** általános alakja a következő:

```
repeat
    <a ciklusmag utasításai>
until <logikai kifejezés>;
```

A ciklus a következőképpen működik:

1. Végrehajtásra kerülnek a ciklusmag utasításai.
2. Kiértékelődik a **logikai kifejezés.**
3. Ha a kifejezés értéke **hamis,** visszatérünk az 1. pontra, ha **igaz**, akkor a ciklus utáni első utasításra lépünk.

A hátultesztelő ciklust akkor alkalmazzuk, ha a ciklusmagot alkotó utasításokat legalább egyszer végre kell hajtani.

Nézzük meg, hogy lehet megoldani az előző feladatot hátultesztelő ciklus alkalmazásával:

```
var a, d, s: Real;
begin
    a := StrToFloat (Edit1.Text);
    d := StrToFloat (Edit2.Text);
    s := 0;
    repeat
        s := s + a;
        a := a + d;
    until a <= 0;
    Label1.Caption := FloatToStr (s);
end;
```

Futtassuk le a programot az *a* és *d* valamely értékei mellett.

Utasítás	*A végrehajtás eredménye*
a := StrToFloat (Edit1.Text)	*a* = 7
d := StrToFloat (Edit2.Text)	*d* = –3
s := 0	*s* = 0
s := *s* + *a*	*s* = 0 + 7 = 7
a := *a* + *d*	*a* = 7 + (–3) = 4
a <= 0	(4 <= 0) = ***false***
s := *s* + *a*	*s* = 7 + 4 = 11
a := *a* + *d*	*a* = 4 + (–3) = 1
a <= 0	(1 <= 0) = ***false***
s := *s* + *a*	*s* = 11 + 1 = 12
a := *a* + *d*	*a* = 1 + (–3) = –2
a <= 0	(–2 <= 0) = ***true***
Label1.Caption := *FloatToStr (s)*	*Label1.Caption* = 12

Figyeljük meg:

1. A hátultesztelő ciklus nem tartalmazza a **begin** és **end** kulcsszavakat.
2. Ha az előltesztelő ciklust hátultesztelővé írjuk át (és viszont), a kilépési feltételt az ellenkezőjére kell változtatni.

Példák különböző ciklusok alkalmazására

Vizsgáljunk most meg néhány érdekes, a következő témakörök tanulmányozása során hasznos feladatot, amelyek megoldása során ciklusokat kell alkalmazni.

A feladatok megoldása során sokszor szükség van a következő két műveletre: egy egész szám természetes számmal való osztásakor keletkező hányados, illetve a maradék kiszámítására. Bármely egész m és természetes n számpár esetében egy és csakis egy olyan q és r ($0 \leq r < n$) számpár létezik, amely kielégíti a $m = nq + r$ egyenlőséget. A q számot nem teljes hányadosnak, az r-t pedig az osztás maradékának nevezzük. A **Delphi**-ben a hányadost a **div** (*divide* – osztani), a maradékot a **mod** (*modulo* – az osztás maradéka) műveletekkel számítjuk ki. Például:

23 div 5 = 4, 23 mod 5 = 3,
28 div 4 = 7, 28 mod 4 = 0,
2 div 3 = 0, 2 mod 3 = 2.

3. feladat. Ismert az egynél nagyobb n természetes szám. Határozzuk meg, hogy prímszám-e ez a szám!

Prímszámnak azt a természetes számot nevezzük, amelynek pontosan két osztója van. Vegyük tehát sorra a számokat 1-től az adott számig, és számoljuk meg, hogy ezek közül hánnyal volt osztható a vizsgált szám. Amennyiben ez a szám kettővel lesz egyenlő, akkor a szám prímszám, ha több, akkor nem. A megfelelő programrészlet:

```
var i, k, n: Integer;
begin
   n := StrToInt (Edit1.Text);
   k := 0; // Az n osztóinak száma
   for i := 1 to n do
      if n mod i = 0 //Ellenőrizzük, hogy i osztója-e n-nek
      then k := k + 1; //Amennyiben i osztja n-t, az osztók számát eggyel növeljük.
         if k = 2
         then Label1.Caption := 'prím'
         else Label1.Caption := 'nem prím';
end;
```

A **StrToInt** függvény (*string to integer* – szöveget egésszé) a szöveges adatok egésszé alakítására szolgál.

A program futását lényegesen meggyorsíthatjuk, ha figyelembe vesszük a következőket:

1. Bármely egynél nagyobb természetes szám osztható 1-gyel és saját magával. Ezért prímszám minden olyan szám, amelynek nincs ettől eltérő osztója.
2. A prímszámok között egyetlen páros szám van, a 2. Minden más prím páratlan.
3. Ha magát a számot nem vesszük figyelembe, akkor amennyiben a számnak nincs a saját számtani gyökénél kisebb osztója, akkor annál nagyobb osztója sincs.

Figyelembe véve a mondottakat a programot a következőképpen írhatjuk át:

```
var i, k, n: Integer; f: Boolean;
begin
   n := StrToInt (Edit1.Text);
   f := true; n egyelőre prímnek tekintjük, mivel eddig nem találtunk osztót
   if (n mod 2 = 0) and (n <> 2)
```

```
        then f := false // // Ha n páros és nem 2, akkor nem prím
        else
            begin
                k := 3; Ha n páratlan, akkor a 3-mal kezdve keressük az osztóit.
            while (k <= sqrt (n)) and f do Az osztókat az n számtani gyökénél kisebb szá-
mok között keressük és egyelőre egyet sem találtunk
                    if n mod k = 0 // Ellenőrizzük, hogy k osztója-e n-nek
                    then f := false
                    else k := k + 2; Ha k nem osztója n-nek, a következő lehetséges osztó
a következő páratlan szám
            end;
        if f
            then Label1.Caption := 'prím'
            else Label1.Caption := 'nem prím';
    end;
```

A programban az **f** logikai változót használtuk annak nyilvántartására, hogy az **n**-t prímszámnak tekintjük (a **f** értéke **igaz**), vagy nem (**f** értéke **hamis**). A logikai változót a **Delphi**-ben **Boolean**-ként deklaráljuk. A szám számtani gyökét az **sqrt** (*square root* – négyzetgyök) beépített függvény segítségével számítjuk ki.

4. feladat. Határozzuk meg két, *a* és *b (a >b)* természetes szám legnagyobb közös osztóját!

A matematikaórákon két szám legnagyobb közös osztóját prímtényezőkre bontás segítségével határoztuk meg. Lehet olyan programot írni, amely ezt a módszert valósítja meg.

Van azonban ennél egyszerűbb megoldás is, amelynek alapja a következő állítás: **ha *a* > *b*, akkor LNKO(*a, b*)=LNKO(*b, r*), ahol *r a b*-vel való osztásának maradéka.** Ennek alapján a nagyobb számot mindig a kisebbre, a kisebbet pedig a nagyobb szám kisebbel való osztásának maradékára cseréljük. Az utolsó nullától eltérő maradék lesz a két szám legnagyobb közös osztója.

Például:

LNKO(80, 12)=LNKO(12, 8)=LNKO(8, 4)=LNKO(4, 0) = 4, vagy

LNKO(125, 54)=LNKO(54, 17)=LNKO(17, 3)=LNKO(3, 2)=LNKO(2, 1)= =LNKO (1, 0)=1.

Ez a módszer **Eukleidész-algoritmusa** néven ismert. Próbáljuk ki a módszert néhány különböző számpár esetére.

Eukleidész-algoritmusa szerint működik a következő program:

```
var a, b, r: Integer;
begin
    a := StrToInt (Edit1.Text);
    b := StrToInt (Edit2.Text);
    r := a mod b;
    while r <> 0 do
        begin
            a := b;
            b := r;
            r := a mod b;
        end;
    Label1.Caption := IntToStr (b);
end;
```

Figyeljük meg, hogy a módszer $a \leq b$ esetében is helyesen működik. Derítsük ki, miért!

Eukleidész (i.e. 300 körül – i.e. 265 körül) (2.58. ábra) neves ógörög matematikus. Ismertségét sokban annak köszönheti, hogy *Elemek* című 13 kötetes művében összegyűjtötte és rendszerezte korának geometriai és aritmetikai ismereteit.

A matematikatörténet kutatói bizonyítékokat találtak arra, hogy Eukleidész-algoritmusát már Eukleidész előtt ismerték. Az algoritmust azért nevezték el mégis Eukleidészről, mert először az *Elemek*ben volt részletesen ismertetve.

2.58. ábra. Eukleidész

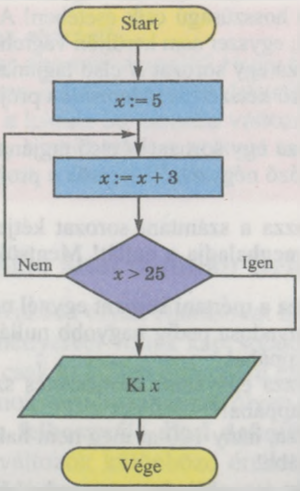
Ellenőrző kérdések

1°. Mi a ciklus az algoritmusokban?
2•. Írjuk fel az elöltesztelő ciklus folyamatábráját! Ismertessük a ciklus működését és hozzunk fel példákat!
3•. Írjuk fel az hátultesztelő ciklus folyamatábráját! Ismertessük a ciklus működését és hozzunk fel példákat!
4•. Előfordulhat-e, hogy a ciklusmag utasításai egyszer sem kerülnek végrehajtásra? Válaszunkat indokoljuk meg, és hozzunk fel példákat!
5*. Előfordulhat-e, hogy egy ciklusnak soha nem lesz vége? Válaszunkat indokoljuk meg, és hozzunk fel példákat!
6•. Magyarázzuk meg a három alapvető algoritmus-szerkezet – szekvencia, elágazás, ciklus – működése közötti különbségeket!
7°. Írjuk fel a számlálós ciklus általános alakját **Delphi**-ben!.
8•. Magyarázzuk meg a számlálós ciklus működését!
9°. Írjuk fel az elöltesztelő ciklus általános alakját **Delphi**-ben!
10•. Magyarázzuk meg az elöltesztelő ciklus működését!
11*. Írjuk fel a hátultesztelő ciklus általános alakját **Delphi**-ben!
12*. Magyarázzuk meg a hátultesztelő ciklus működését!

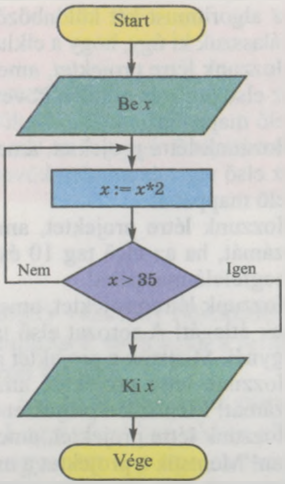
Végezzük el a feladatokat!

1. Hajtsuk végre az algoritmusokat!

a°)



b•)



2°. Hajtsuk végre az alábbi programrészleteket, és határozzuk meg a változók értékeit a végrehajtás után:

a)
```
p := 1; a := 2;
for i := 1 to 5 do
begin
    a := 3*a + 1;
    p := p*a;
end;
```

b)
```
k := 1; a := 12;
while a < 100 do
begin
    a := 2*a – 4;
    k := k + 1;
end;
```

c)
```
s := 0; a := 1; k := 1;
repeat
    s := s + a;
    k := k + 1;
    a := k*k;
until a > 50;
```

3°. Hajtsuk végre az alábbi programrészleteket, és határozzuk meg a változók értékeit a végrehajtás után!

a)
```
p := 1; a := 8;
for i := 1 to 4 do
begin
    a := 2*a – 5;
    p := p*a;
end;
```

b)
```
k := 1; a := 100;
while a > 10 do
begin
    a := a div 2;
    k := k + 1;
end;
```

c)
```
s := 0; a := 5428;
repeat
    c := a mod 10;
    s := s + c;
    a := a div 10;
until a < 1;
```

4°. Hajtsuk végre a paragrafus 3. feladatához tartozó programrészletet!

5°. Hajtsuk végre a paragrafus 4. feladatához tartozó programrészletet!

6°. Állítsuk össze annak az algoritmusnak a folyamatábráját, amely összeadja a számtani sorozat első öt tagját, ha az első tag $x = 7$, és minden következő tagot az $x:=2*x+3$ képlettel számíthatunk ki!

7°. Állítsuk össze annak az algoritmusnak a folyamatábráját, amely összeszorozza a számtani sorozat első nyolc tagját, ha az első tag 9 és a különbség 4!

8°. Állítsuk össze annak az algoritmusnak a folyamatábráját, amely meghatározza a számtani sorozat pozitív tagjainak számát, ha ismert annak első tagja, különbsége pedig –5! Hajtsuk végre az algoritmust különböző első tagok esetében! A tesztértékeket válasszuk ki úgy, hogy a ciklusmag többször; egyszer; egyszer sem kerüljön végrehajtásra!

9°. Állítsuk össze annak az algoritmusnak a folyamatábráját, amely meghatározza a számtani sorozat kétjegyű tagjainak összegét, ha annak első tagja 25, és ismert a különbsége! Hajtsuk végre az algoritmust a sorozat három különböző differenciája esetében! A tesztértékeket válasszuk ki úgy, hogy a ciklusmag többször; egyszer; egyszer sem kerüljön végrehajtásra!

10°. Állítsuk össze az alábbi feladat megoldásának folyamatábráját: határozzuk meg, hány négyzetet lehet kivágni egy ismert szélességű és hosszúságú papírcsíkból, ha abból minden lépésben a kisebbik oldallal azonos hosszúságú négyzetet vágunk le. Hajtsuk végre az algoritmust két különböző szélességű és hosszúságú csík esetében! A tesztértékeket válasszuk ki úgy, hogy a ciklusmag többször; egyszer sem kerüljön végrehajtásra!

11*. Hozzunk létre projektet, amely meghatározza egy sorozat N első tagjának szorzatát, ha az első tag x és minden követező tag az előző kétszerese. Mentsük a projektet a megfelelő mappába!

12°. Hozzunk létre projektet, amely meghatározza egy sorozat N első tagjának összegét, ha az első tag x és minden következő tag az előző négyzete! Mentsük a projektet a megfelelő mappába!

13°. Hozzunk létre projektet, amely meghatározza a számtani sorozat kétjegyű tagjainak számát, ha az első tag 10 és a különbség meghaladja a nullát! Mentsük a projektet a megfelelő mappába!

14°. Hozzunk létre projektet, amely meghatározza a mértani sorozat egynél nagyobb tagjainak átlagát! A sorozat első tagja 100, a hányadosa pedig nagyobb nullánál, de kisebb egynél. Mentsük a projektet a megfelelő mappába!

15°. Hozzunk létre projektet, amely meghatározza egy adott természetes szám osztóinak számát! Mentsük a projektet a megfelelő mappába!

16*. Hozzunk létre projektet, amely meghatározza, hány 100-at meg nem haladó prímszám van! Mentsük a projektet a megfelelő mappába!

17°. Hozzunk létre projektet a következő feladat megoldására: egy ember pénzt helyezett el a bankban. A bank meghatározott kamatot fizet az előző év végén a betétek után.

Mennyi lesz az emberünk betétjének értéke *N* év eltelte után? Mentsük a projektet a megfelelő mappába!

18*. Hozzunk létre projektet a következő feladat megoldására: egy ember pénzt helyezett el a bankba. A bank meghatározott kamatot fizet az előző év végén a betétek után. Mennyi idő elteltével lesz az emberünk betétjének értéke legalább S hrivnya? Mentsük a projektet a megfelelő mappába!

19*. Hozzunk létre projektet egy természetes szám számjegyei összegének kiszámítására! A szám maximálisan kilencjegyű lehet. Mentsük a projektet a megfelelő mappába!

9. számú gyakorlati munka Ciklusokat tartalmazó programok létrehozása

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert!
2. Hozzunk létre projektet a következő feladat megoldására: *Áradás előtt a folyó szintje H méter volt. Az áradás során a vízszint P%-kal emelkedett óránként. Mennyi lesz a vízszint N órával az áradás kezdete után? Hány óra elteltével éri el a vízszint a K méteres magasságot?*
   1. Helyezzük el a form-on az adatbevitelhez és kivitelhez szükséges vezérlőket, a vezérlők rendeltetését magyarázó feliratokat, valamint három gombot!
   2. Az egyes gombok feliratai megfelelően a következők legyenek: *„1. kérdés", „2. kérdés", „Újra"*. A mezők maradjanak üresen.
   3. Az első gomb **OnClick** eseménykezelőjébe írjuk azt a programrészt, amely megadja a választ a feladat első kérdésére!
   4. Hajtsuk végre az eljárást, és győződjünk meg működésének helyességéről!
   5. A második gomb **OnClick** eseménykezelőjébe írjuk azt a programrészt, amely megadja a választ a feladat első kérdésére, és egy felirat szövegében megjeleníti az eredményt!
   6. Hajtsuk végre az eljárást, és győződjünk meg működésének helyességéről!
   7. A harmadik gomb **OnClick** eseménykezelője törölje a beviteli mezőkbe, és feliratokba beírt értékeket!
3. Mentsük a projektet **Gyakorlati9** néven!

2.10. Felhasználó által definiált és beépített eljárások és függvények Delphi-ben

1. Mi az eljárás?
2. Mi az eseménykezelő? Hogy hozzuk létre?
3. Milyen a **Delhi** eseménykezelőjének általános alakja?
4. Mik a lokális és globális változók? Mi közöttük a különbség?
5. Hogy alakíthatjuk a számot tartalmazó változót szöveggé **Delphi**-ben? Hogy történik a visszaalakítás?

Felhasználó által definiált Delphi-ben

Az előző részben olyan eljárásokat hoztunk létre, amely valamely objektum metódusával vagy eseményeivel voltak kapcsolatban. Az ilyen eljárások sablonjait a rendszer hozta létre, nekünk csak ki kellett egészíteni ezeket a **var** sorral kezdve.

A programok tartalmazhatnak olyan eljárásokat is, amelyek nincsenek eseményekhez kötve. Ezek a **felhasználó által definiált eljárások.** Leggyakrabban akkor alkalmazzuk ezeket, ha a változók különböző értékei mellett kell ugyanazt a programrészt lefuttatni. Ilyen lehet például a mértani alakzatok területének kiszámítása, két szám közül a nagyobb kiválasztása, annak ellenőrzése, hogy egy szám prímszám-e. Előfordul, hogy egy felada-

tot több független részre sikerül bontani, vagyis ha a lépésenkénti részletezés módszerével élünk.

A felhasználó által definiált eljárás – mint minden eljárás – a program névvel illetett része, amelyet a program más részeiből meghívhatunk.

Az eljárások lehetnek:

- paraméter és visszatérési érték nélküliek;
- paraméteresek és visszatérési érték nélküliek;
- paraméter nélküliek, de visszatérési értékkel bírók;
- paraméterrel és visszatérési értékkel bírók.

Az *eljárás paraméterei* – azok a változók, amelyek az eljárás hívásakor kapják meg az értéküket.

Az *eljárás visszatérési* értékei – azok a változók, amelyek az eljárás futásának eredményeképpen kapják meg az értéküket.

A leggyakrabban alkalmazott eljárástípus a negyedik. Az eljárás általános alakja **Delphi**-ben:

```
procedure <eljárás_neve>(<eljárás paraméterei>;
var<az eljárás visszatérési érékei>);
begin
   <utasítások>
end;
```

Az eljárás hívásának általános alakja:

<eljárás neve>(<azoknak a változóknak a nevei, amelyek értékeket adnak az eljárásnak>,<azoknak a változóknak a nevei, amelyek az eljárásból kapnak értékeket>);

Fontos, hogy az eljárás hívásában a változók sorrendje és típusa pontosan meg kell hogy egyezzen a változóknak az eljárás fejében alkalmazott sorrendjével és típusával.

Alakítsuk át eljárássá a két szám legnagyobb közös osztóját meghatározó programot. Az eljárás paramétere a két egész szám lesz, a visszatérési értéke pedig a legnagyobb közös osztó. Az eljárás a következő lesz:

```
procedure MCD (a, b: Integer; var c: Integer);
var r: Integer;
begin
   r := a mod b;
   while r <> 0 do
      begin
         a := b;
         b := r;
         r := a mod b;
      end;
   c := b;
end;
```

Az eljárást például az **MCD(35,77,x)** utasítással hívhatjuk meg. Az utasítás a következőképpen működik:

1. Az *a* és *a b* (az eljárás paraméterei) megkapják a 35 és 77 értékeket.
2. Lefut az eljárás.
3. A *c* változó értéke átkerül az *x* változóba.

Az *x* értéke tehát az eljárás futásának eredményeként 7-tel lesz egyenlő.

Az eljárás hívásakor nemcsak konkrét számokat használhatunk paraméterként, hanem változókat is. Az eljárás hívásának pillanatában ezeknek a változóknak konkrét ér-

tékkel kell rendelkezniük és egész típusúnak kell lenniük. Az **MCD(k, p, x)** híváskor tehát az *a* és a *b* változók a *k* és *p* értékeit kapják meg. A változók helyén matematikai kifejezések is állhatnak, amelyek értékeit a paraméterátadás előtt ki kell számítani. Az **MCD(2*k+3, 3*p–1,x)** híváskor tehát előbb a kifejezések kerülnek kiszámításra, majd azok értékeit kapják meg az *a* és *b* változók.

Figyeljük meg!

1. Az eljárás paraméterei és eredményei lehetnek ugyanazok a változók, tehát helyes a **Procedure One(var a, b:Integer)** eljárásfej. Ekkor az eljárás hívásakor az *a* és *b* megkapják a megfelelő értékeket és ezek a változók tartalmazzák az eredményt is. Az eljárás hívása: **One(x, y).**
2. Az eljárás más eljárások hívását is tartalmazhatják.

Az eljárásfejben alkalmazott változókat *formális paramétereknek* nevezzük.

Az eljárás hívásakor alkalmazott változókat az eljárás *valóságos paramétereinek* nevezzük.

Vizsgáljuk meg az eljárások alkalmazását a következő példán.

1. feladat. Határozzuk meg két közönséges tört összegét!

A fenti projekt (2.59. ábra) futása során az eljárás kétszer kerül meghívásra. Először meghatározzuk a törtek nevezőinek legkisebb közös osztóját. Ezt használjuk fel a törtek közös nevezőjének meghatározása során, mivel **LKKT(*x*, *y*) = *x**y*/LNKO(*x*, *y*).** Az eljárást másodszor a tört számlálójának és nevezőjének legnagyobb közös osztójának meghatározására használjuk a tört egyszerűsítése céljából. Ha nem használnánk a programban eljárást, lényegesen hosszabb lenne a program és rosszabb lenne annak átláthatósága.

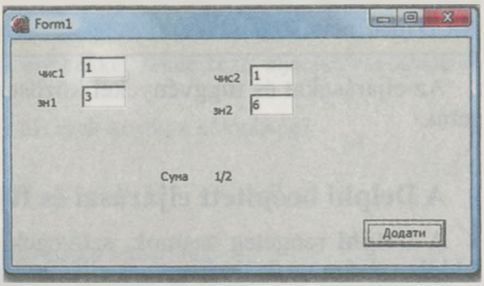
```
{$R *.dfm}

Procedure MCD (a, b: Integer; Var c: Integer);
Var r: Integer;
Begin
r := a mod b;
While r <> 0 do
Begin
a := b;
b := r;
r := a mod b;
End;
c := b;
End;

procedure TForm1.Button1Click(Sender: TObject);
var nom1, nom2, den1, den2, den, nom, k, r: Integer;
begin
  nom1 := StrToInt (Edit1.Text);
  den1 := StrToInt (Edit2.Text);
  nom2 := StrToInt (Edit3.Text);
  den2 := StrToInt (Edit4.Text);
  MCD(den1, den2, r);
  den := den1*den2 div r;
  nom := nom1*den div den1 + nom2*den div den2;
  MCD(den,nom, k);
  If k <> 1
  Then
  Begin
    den := den div k;
  nom := nom div k;
  End;
  Label5.Caption := IntToStr(nom) + '/' + IntToStr(den);
end;

end.
```

2.59. ábra. Két közönséges törtet összeadó program, amely használja a legnagyobb közös osztót meghatározó eljárást



2.60. ábra. A két közönséges tört összeadására szolgáló program ablaka

Felhasználó által definiált függvények Delphi-ben

```
{$R *.dfm}

Function MCD (a, b: Integer): Integer;
Var r: Integer;
Begin
 r := a mod b;
 While r <> 0 do
 Begin
  a := b;
  b := r;
  r := a mod b;
 End;
 MCD := b;
End;

procedure TForm1.Button1Click(Sender: TObject);
Var num1, num2, den1, den2, den, num, k: Integer;
Begin
 num1 := StrToInt (Edit1.Text);
 den1 := StrToInt (Edit2.Text);
 num2 := StrToInt (Edit3.Text);
 den2 := StrToInt (Edit4.Text);
 den := den1*den2 div MCD(den1, den2);
 num := num1*den div den1 + num2*den div den2;
 k := MCD(den,num);
 If k <> 1
 Then
 Begin
  den := den div k;
  num := num div k;
 End;
 Label5.Caption := IntToStr(num) + '/' + IntToStr(den);
End;

end.
```

2.61. ábra. A két közönséges törtet összeadó program, amelyben a legnagyobb közös osztó függvényeként határozzuk meg

Ahogy azt már megtárgyaltuk, az eljárásnak egy vagy több visszatérési értéke lehet, de akár az is előfordulhat, hogy egyetlen egy sincs. Amennyiben az eljárásnak egy visszatérési értéke van, célszerű azt **függvényként** definiálni.

A felhasználó által definiált függvény az eljáráshoz hasonlóan a program névvel ellátott része, amelyet a program tetszőleges részéből meghívhatunk.

A függvény az eljárástól abban különbözik, hogy:

1. Más a fejléce: **function <a függvény neve>(<paraméterek és azok típusai>): <eredmény típusa>;**

2. A függvény visszatérési értékét a függvény törzsében a függvény nevével megegyező nevű változónak kell értékül adni.

3. A függvényt egy értékadó utasítás jobb oldalában vagy egy kiviteli utasításban hívhatjuk meg.

Ha a legkisebb közös többszöröst függvénnyel számítjuk ki, a program a következőképpen módosul (2.61. ábra).

Figyeljük meg az eljárás (2.59. ábra) és a függvény (2.61. ábra) közötti megírása és meghívása közötti különbségekre. Ezeket a 2.3. táblázatban foglaltuk össze.

2.3. táblázat. **Eltérések az eljáráshívás és a függvényhívás között**

Az MCD eljárás	*Az MCD függvény*
MCD (den1, den2, z); den := den1*den2/z;	den := den1*den2/MCD (den1, den2);
MCD (den, nom, k);	k := MCD (den, nom);

Az eljárásokat és függvényeket közösen alprogramokként (subroutine) szokás emlegetni.

A Delphi beépített eljárásai és függvényei

A **Delphi** rengeteg számok, szövegek, vezérlőelemek, grafikai objektumok, fájlok feldolgozására szolgáló beépített eljárást tartalmaz.

Ezek közül néhánnyal már megismerkedtünk. Ilyenek voltak az adattípusok közötti átalakításokat megvalósító **StrToInt** és **IntToStr** függvények, a négyzetgyökvonást megvalósító **sqrt** függvény.

A 2.4. táblázat a **Delphi** beépített matematikai függvényei közül mutat be néhányat.

2.4. táblázat. **Néhány a Delphi beépített matematikai függvényei közül**

Függvény	*Paraméterek száma és típusa*	*Eredmény*	*Eredmény típusa*
Abs	1, egész vagy valós	A szám abszolút értéke	A paraméterrel megegyező
Arctan	1, valós	A szám arctangense (radiánokban)	Valós
Cos	1, valós	A szám koszinusza	Valós
Exp	1, valós	A szám e^x alapú hatványa	Valós
Frac	1, valós	A szám törtrésze	Valós
Int	1, valós	A szám egész része	Valós
Ln	1, valós	A szám természetes logaritmusa	Valós
Pi	0	A π szám	Valós
Power	2, valós	**A II paraméter I alapú hatványa**	Valós
Random	0	0 és 1 közötti véletlen szám	Valós
Random	1, egész	**0 és a paraméter közötti véletlen szám**	Egész
Sin	1, valós	A szám szinusza	Valós
Sqr	1, valós	A szám négyzete	Valós

Figyeljünk oda a következőkre:

1. A paraméter nélküli függvény hívása a függvény nevével történik, például **x:=Pi**.
2. A **Random** függvény első hívása előtt meg kell hívni a **Randomize** paraméter és visszatérési érték nélküli függvényt, amely elindítja a véletlenszám-generátort.
3. A **Power** függvény alkalmazása céljából az **Uses** sort ki kell egészíteni a **Math** szöveggel. Ez azt eredményezi, hogy a projekthez hozzákapcsolódik egy újabb függvényeket és eljárásokat tartalmazó könyvtár. A **Súgóból** részletesen tájékozódhatunk az ilyen módon elérhető eljárásokról és függvényekről.

Ellenőrző kérdések

1°. Mik a felhasználó által definiált eljárások? Milyenek lehetnek ezek?
2°. Milyen esetekben célszerű eljárásokat alkalmazni? Mik a módszer előnyei?
3°. Milyen a paraméterekkel és visszatérési értékekkel rendelkező eljárás általános alakja?
4°. Mik az eljárás paraméterei? Mik a visszatérés értékei?
5°. Milyen a paraméterekkel és visszatérési értékekkel rendelkező eljáráshívás általános alakja?
6°. Mik a formális és valóságos paraméterek? Mi ezek között a különbség?
7°. Mi a felhasználó által definiált függvény?
8°. Miben különbözik az eljárás és a függvény?
9*. Milyen a függvény általános alakja?
10°. A **Delphi** milyen beépített eljárásait és függvényeit ismerjük?
11*. Keressünk a **Súgóban** információkat olyan eljárásról, amelynek nincs paramétere és visszatérési értéke; amelynek mind paramétere, mind pedig visszatérési értéke van!

Végezzük el a feladatokat!

1. Írjuk fel a ***stepin*** eljárás fejlécét, amelynek paraméterei az *a* valós és a *b* egész, visszatérési értéke pedig a *c* valós. Írjuk fel az eljárás hívását is!

2*. Írjuk fel a ***chastka*** eljárás fejlécét, amelynek paraméterei az a és b valós változók, visszatérési értéke pedig a c valós és az f logikai változók! Írjuk fel az eljárás hívását is!

3*. Írjunk eljárást ***simp*** néven, amely ellenőrzi, hogy az adott természetes szám prímszám-e! A visszatérési értékek: egy k logikai változó, amely azt mutatja, prím-e a vizsgált szám és a *k* egész, a szám osztóinak száma.

4. Írjunk eljárást **square** néven, amely kiszámítja a háromszög területét az oldalak alapján. Az oldalak az eljárás paraméterei. Használjuk Hérón képletét.

5. Hozzunk létre projektet, amely a háromszög csúcsainak koordinátái alapján meghatározza a háromszög területét! A projektben alkalmazzunk eljárást, amely meghatározza a szakasz hosszát a végpontok koordinátái alapján! Mentsük a projektet a megfelelő mappába!

6. Hozzunk létre projektet, amely két pont koordinátái alapján eldönti, azonos koordináta-negyedhez tartoznak-e azok! Alkalmazzunk eljárást annak eldöntésére, melyik negyedben fekszik az adott pont! Mentsük a projektet a megfelelő mappába!

7. Írjuk fel a ***length*** függvényt fejlécét, amelynek négy egész típusú paramétere van, eredménye pedig valós!

8. Írjuk fel a **step** függvény fejlécét, amelynek paraméterei az *a* valós és a *b* egész, eredménye valós. Írjuk fel a függvény hívását is!

9. Írjunk függvényt, amely kiszámítja a szakasz hosszát a végpontok koordinátái alapján! Írjuk fel a függvény hívását a (0, 0) és (20,30) pontok esetére!

10. Írjunk függvényt a *n!* értékének kiszámítására (n! = 1*2*3*...*n)! Írjuk fel a függvény hívását öt különböző természetes szám esetére!

11. Hozzunk létre projektet, amely meghatározza, hogy négy egész közül melyik a legnagyobb! Alkalmazzunk függvényt, amely meghatározza, hogy két egész közül melyik a nagyobb. Mentsük a projektet a megfelelő mappába!

12. Hozzunk létre projektet, amely két adott egész számról eldönti, hogy melyik számjegyeinek összege a nagyobb. A számjegyek összegének meghatározására használjunk függvényt! Mentsük a projektet a megfelelő mappába!

13. Hozzunk létre projektet, amely két adott egész számról eldönti, hogy melyiknek nagyobb az osztóinak összege! Az osztók összegének meghatározására használjunk függvényt! Mentsük a projektet a megfelelő mappába!

14. Hozzunk létre projektet, amely meghatározza, hogy prímszám-e az adott szám, illetve a szám számjegyeinek fordított sorrendű felírásával keletkező szám! Használjunk függvényt annak eldöntésére, hogy egy szám prím-e! Mentsük a projektet a megfelelő mappába!

15. Írjunk utasítást a **Delphi** beépített függvényeinek alkalmazásával az *a, b, c* együtthatójú másodfokú egyenlet diszkriminánsának és gyökeinek kiszámítására!

16. Írjunk projektet az $1^1 + 2^2 + 3^3 + 4^4 + 5^5 + \ldots + nn$ kifejezés értékének kiszámítására! Használjuk a négyzetre emelés beépített függvényt. Mentsük a projektet a megfelelő mappába!

17. Írjunk projektet a $\sqrt{1} + \sqrt{2} + \sqrt{3} + \sqrt{4} + \sqrt{5} + \ldots + \sqrt{n}$ kifejezés értékének kiszámítására! Mentsük a projektet a megfelelő mappába!

18. Bővítsük a két közönséges törtet összeadó projektet egy eljárással, amely az elkülöníti az áltört egész részét!

10. számú gyakorlati munka Alprogramok alkalmazása

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert!
2. Hozzunk létre projektet, amely kétféle módon határozza meg a konvex négyszög területét!
   a) Az *AB, BC, CD* és *DA* szakaszok és az *AC* átló hosszai alapján. A háromszög területének meghatározására írjunk függvényt, amely Heron képletét használja! Az első gomb **OnClick** eseményének kezelője írja ki az eredményt egy külön ablakba.
   b) A négyszög adott irányú bejárása szerint megadott csúcsainak koordinátái alapján. Írjunk függvényt, amely a szakasz végpontjainak koordinátái alapján meghatározza a szakasz hosszát! A második gomb **OnClick** eseménykezelője jelenítse meg az eredményt egy feliraton!
3. Mentsük a projektet **Gyakorlat10** néven!

2.11. Egydimenziós tömbök

1. Mit határoz meg a változó típusa? Milyen típusokat ismerünk?
2. Hogy néz ki az elágazás általános alakja **Delphi**-ben? Hogyan működik?
3. Hogy néz ki a számlálós ciklus általános alakja **Delphi**-ben? Hogyan működik?
4. Hogy néz ki az elöltesztelő ciklus általános alakja **Delphi**-ben? Hogyan működik?
5. Hol találkozhatunk rendezett adathalmazokkal? Miért kényelmes ezek használata?

Az egydimenziós tömb fogalma

Az eddigiek során olyan feladatokkal foglalkoztunk, amelyekben viszonylag kevés adat került feldolgozásra. Sokkal több azonban az olyan feladat, amely nagy mennyiségű adat feldolgozását igényli.

Ha például 100 tizenegyedikes átlagtömegét kell meghatározni, akkor 100 változóra lenne szükség az egyes emberek tömegének nyilvántartására. Ezeket aztán össze kellene adni, majd el kellene osztani 100-zal.

Ha a változóknak az elektronikus táblázat oszlopaihoz hasonlóan *a, b, c, ..., z, aa, ab, ac, ..., az, ba, bb, bc, ..., bz, ca, cb, cc, ..., cv* neveket adnánk, akkor az összegzés 100 összeadandót tartalmazna, vagy a programunknak kellene 100 összeadási műveletet tartalmaznia, valahogy így:

```
summa := a;
summa := summa + b;
summa := summa + c;
...
summa := summa + cv;
```

Bármelyik módszert választanánk, programunk áttekinthetetlen és mértéktelenül nagy lenne, különösen ha figyelembe vesszük, hogy a tanulók száma akár több is lehet.

Amikor nagy mennyiségű adat feldolgozására van szükség, célszerű ezeket **egy sorszámozott adathalmaz elemeinek** tekinteni. Ennek a halmaznak lesz egy neve, az egyes változók neve pedig a halmaz (vagyis a tömb) nevéből és az elem sorszámából áll majd. A tömb neve tehát a tömb összes elemére vonatkozik, a sorszám pedig egy elemet jelöl a tömbön belül.

A tömb azonos típusú változók sorszámozott *halmaza*. Minden, a tömbhöz tartozó változó a *tömb eleme*.

Az olyan tömb, amelyben minden elemnek egyetlen sorszáma van, *egydimenziós*.

Az egydimenziós tömb elemének neve a tömb nevéből áll, amelyet a szögletes zárójelbe írt sorszám követ.

A fenti 100 változó helyett célszerű egy 100-elemű tömböt alkalmazni. Ha a tömb neve *a*, akkor annak elemei a[1], a[2], a[3], ..., a[100]. A tömb minden elemének típusa **Real.** Ebben az esetben a 100 elem összegzése a következőképpen oldható meg:

```
summa := 0;
for i := 1 to 100 do
summa := summa + a[i];
```

Figyeljük meg, hogy a tömb elemének sorszáma lehet konkrét szám, vagy egész típusú változó.

A programrészlet végrehajtása a következőképpen zajlik:

Utasítás	*A végrehajtás eredménye*
summa := 0	*summa* = 0
i := 1	*i* = 1
i ⩽ 100	(1 ⩽ 100) = ***true***
summa := *summa* + *a*[*i*]	*summa* = 0 + *a*[1] = *a*[1]
i := *i* +1	*i* = 2
i ⩽ 100	(2 ⩽ 100) = ***true***
summa := *summa* + *a*[*i*]	*summa* = *a*[1] + *a*[2]
i := *i* +1	*i* = 3
i ⩽ 100	(3 ⩽ 100) = ***true***
summa := *summa* + *a*[*i*]	*summa* = *a*[1] + *a*[2] + *a*[3]
...	...
i := *i* +1	*i* = 100

Utasítás	*A végrehajtás eredménye*
$i \leq 100$	$(100 \leq 100) = true$
summa := *summa* + *a*[*i*]	*summa* = *a*[1] + *a*[2] + *a*[3] + *a*[4] + ... + *a*[100]
i := *i* +1	*i* = 101
$i \leq 100$	$(101 \leq 100) = false$

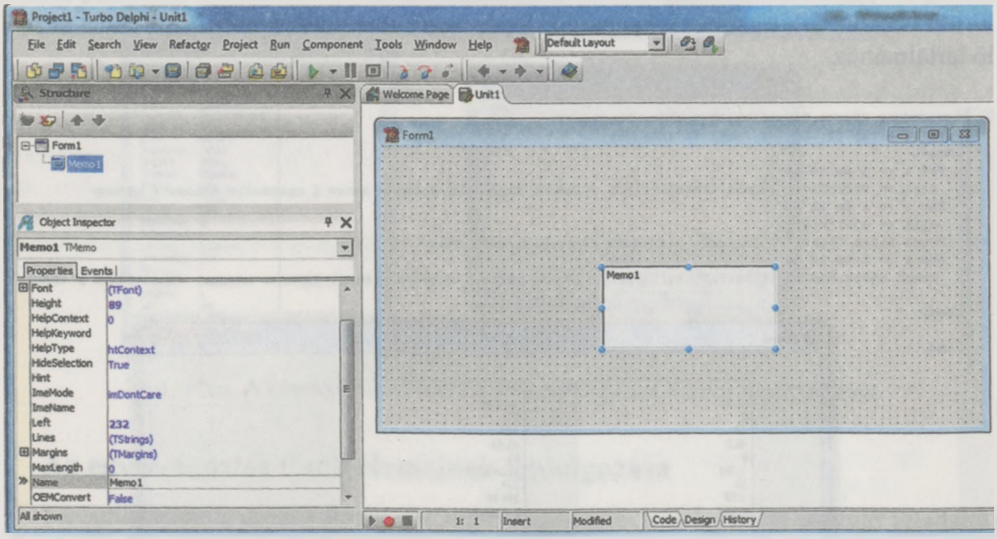
A programrészlet végrehajtásának eredményeképpen a ***summa*** változó, a tömb 100 elemének összegét tartalmazza majd, azaz a 100 tanuló tömegének összegét.

A **Delphi**-ben az egydimenziós tömböt a **var** sorban **var a: array [1..100] of Real** (*array* – tömb) -ként deklaráljuk. A szögletes zárójel az indexek lehetséges értékeit tartalmazza: 1-től 100-ig. Az utasítás végrehajtásakor a számítógép 100 változó számára foglal helyet a memóriában. Az egyes változók számára lefoglalt hely mérete a tömb típusától függ.

Előfordulhat, hogy a változókat nem 1-től kezdődően célszerű sorszámozni. Ha például Ukrajna lakosságának számát tároljuk egy tömbben az 1990 és 2011 közötti években, akkor az **a: array [1990..2011] of Integer** tömb alkalmazása célszerűbb. A tömb elemeinek sorszámait tetszőleges sorszámozott típusból választhatjuk.

Az egydimenziós tömb elemeinek bevitele és kivitele

Az egydimenziós tömb elemeinek be- és kivitelét megoldhatjuk **többsoros beviteli mező** (**TMemo** vezérlő) segítségével (2.62. ábra).



2.62. ábra. **A TMemo** vezérlő és annak tulajdonságai

A **TMemo** vezérlő rendelkezik az olyan, már jól ismert tulajdonságokkal, mint **Name, Color, Font, Enabled, Visible, Height, Width, Top, Left.** A régebben megismert vezérlők nem rendelkeztek olyan tulajdonságokkal, mint:

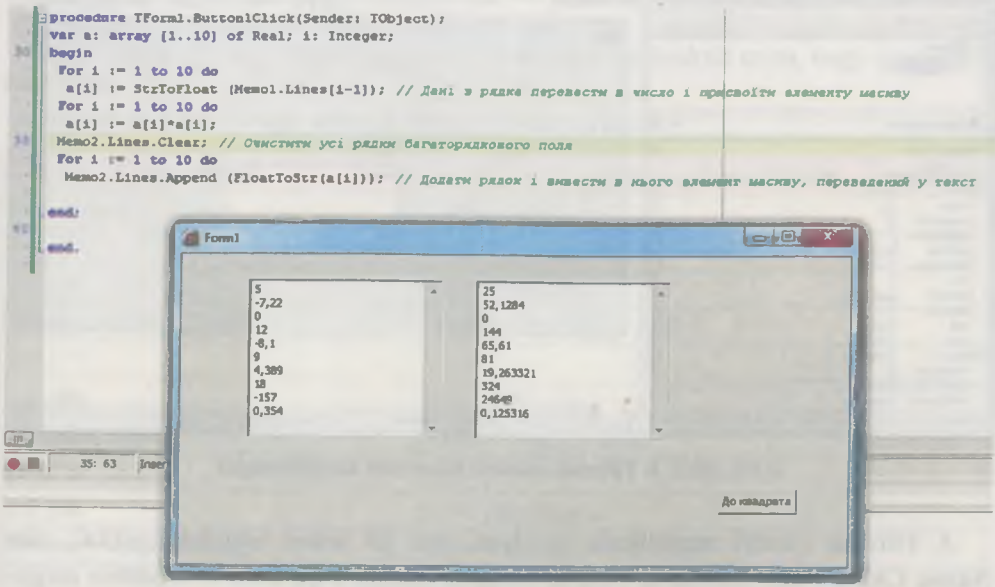
- **ScrollBars** (görgetősáv) – a tulajdonságban tárolt érték mutatja, hogy a vezérlő tartalmaz-e görgetősávokat és ha igen, milyet. Lehetséges értékei: vízszintes – *ssHorizontal*, függőleges – *ssVertical*, mindkettő – *ssBoth*).
- **ReadOnly** (csak olvasható) – a tulajdonság azt mutatja, hogy a vezérlő tartalma futási időben megváltoztatható (*False*), vagy pedig nem (*True*).
- **Lines** (sorok) – e tulajdonság értékei a vezérlőbe írt sorok. A sorszámozás 0-val kezdődik.
- **Lines.Count** (sorok száma) – a kitöltött sorok száma. Ez a tulajdonság nem látható az **Object Inspector** ablakban.

Lássunk egy olyan feladatot, amelynek segítségével közelebbről megismerkedhetünk az egydimenziós tömb és a többsoros bevitele mező által nyújtott lehetőségekkel.

1. feladat. Hozzunk létre 10 valós számot tartalmazó egydimenziós tömböt. A tömböt egy többsoros beviteli mező segítségével töltsük fel. Az elemek négyzeteit írjuk ki egy másik többsoros mezőbe.

Helyezzünk a form-ra két, *Memo1* és *Memo2* mezőt. A tulajdonságot állítsuk be a következőképpen: **Height** = *150*, **ScrollBars** = *ssVertical*, **ReadOnly** = *False*. Helyezzünk el rajta egy gombot, amelynek felirata legyen **Caption** = *„Négyzetre emelés”* és hozzuk létre a gomb **OnClick** eseménykezelőjét.

A 2.63. ábrán láthatjuk az eljárást és annak ablakát futtatás közben. Az eseménykezelő a többsoros mező két metódusát alkalmazza, a **Clear** (tisztítani) metódust, amely törli a vezérlő tartalmát és az **Append** (kiegészíteni) metódust, amely új sort ad a vezérlő tartalmához.



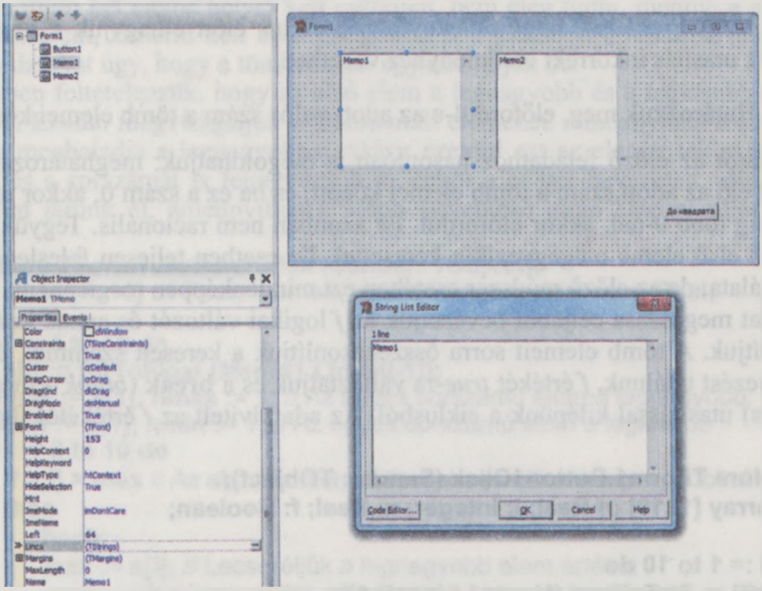
2.63. ábra. A tömb feltöltése és elemeinek kiíratása

A 2.63. ábrán látható eseménykezelő csak a jobb érthetőség kedvéért tartalmaz három ciklust: az adatbevitel, a négyzetre emelés és az adatkivitel ciklusait. A három műveletet egyetlen ciklusba is lehet szervezni.

A projekt indítása után a *Memo1* többsoros mezőnek törölni kell az alapértelmezés szerinti tartalmát, majd be kell írni soronként a tömb elemeit. Ezután a gombra kell kattintani, hogy az **OnClick** eseménykezelő elvégezze a négyzetreemelést és az eredmények kiírását a másik többsoros mezőbe.

A többsoros beviteli mezőt még a projekt futása előtt feltölthetjük adatokkal, ha megváltoztatjuk a mező **Lines** tulajdonságát. E célból:

1. Jelöljük ki a mezőt.
2. Kattintsunk az **Object Inspector** ablakban a **Lines** mező melletti gombra (2.64. ábra).
3. A **String List Editor** (*string* – szöveg, *list* – lista, *editor* – szerkesztő) ablakba írjuk be a 10 számot, mindegyik után **Enter**-t ütve.
4. Kattintsunk az **OK** gombra.



2.64. ábra. A *Memo1* mező feltöltése a **String List Editor** segítségével

Az egydimenziós tömb elemeinek feldolgozása

Lássunk néhány, az egydimenziós tömb elemeinek feldolgozását igénylő feladatot. A tömböt minden esetben többsoros mező segítségével töltjük fel, és az eredményeket is egy ilyen mezőbe írjuk majd ki, amennyiben az egy tömb lesz. Más esetben egy feliratot használunk majd. Szükség esetén egyéb objektumokat és metódusokat is használhatunk.

Általában 10-elemes tömbökkel fogunk dolgozni. Ahol mégsem, ki kell majd javítani az eljárások szövegét.

2. feladat. Határozzuk meg, hányszor fordul elő egy adott szám a tömb elemei között!

A feladat megoldása céljából nyilvánvalóan sorra kell venni a tömb elemeit, és össze kell azokat hasonlítani a vizsgált számmal. Ha ezek megegyeznek, a megfelelő számláló értékét 1-gyel növelni kell.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i, k: Integer; x: Real;
begin
   x := StrToFloat (Edit1.Text); // a keresett szám bevitele
   k := 0; // az adott számmal megegyező elemek számának nullázása
   for i := 1 to 10 do
      begin
      a[i] := StrToFloat (Memo1.Lines[i-1]); // a tömb elemének bevitele
      if a[i] = x then k := k + 1; ; // Amennyiben az elem megegyezik a keresettel, a számlálót eggyel növeljük
      end;
   Label1.Caption := IntToStr (k) + 'paзив' // Az eredmény és a magyarázó szöveg kiírása
end;
```

Fontos megjegyeznünk, hogy amennyiben a ciklus előtt elhagyjuk a **k:=0** sort, az első **k:=k+1** utasítás inkorrekt eredményhez vezethet.

3. feladat. Határozzuk meg, előfordul-e az adott valós szám a tömb elemei között!

A feladatot az előző feladathoz hasonlóan is megoldhatjuk: meghatározzuk, hányszor fordul elő az adott szám a tömb elemei között, és ha ez a szám 0, akkor nem fordul elő, ha pedig több 0-nál, akkor előfordul. Ez azonban nem racionális. Tegyük fel, hogy már a tömb első eleme megegyezik a keresettel. Ez esetben teljesen felesleges a többi elem vizsgálata, de az előző módszer esetében ezt mindenképpen megtennénk.

A feladat megoldása céljából bevezetjük az *f* logikai változót és annak kezdőértékét *false*-ra állítjuk. A tömb elemeit sorra összehasonlítjuk a keresett számmal, és amennyiben egyezést találunk, *f* értékét *true*-ra változtatjuk és a **break** (*break* – megszakítás, ugrás, törés) utasítással kilépünk a ciklusból. Az adatkivitelt az *f* értékétől függően végezzük el.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i: Integer; x: Real; f: Boolean;
begin
   for i := 1 to 10 do
      a[i] := StrToFloat (Memo1.Lines[i-1]);
      x := StrToFloat (Edit1.Text);
      f := false; // A keresett szám még nem fordult elő a tömbben
   for i := 1 to 10 do
      if a[i] = x
      then  begin
               f := true; // Megtaláltuk a keresett számot
               break; // A ciklus megszakítása
            end;
   if f
      then Label1.Caption :=  // találtunk
      else Label1.Caption :=  // nem találtunk;
end;
```

4. feladat. Határozzuk meg a tömb legnagyobb elemét!

Kezdetben fogadjuk el, hogy az első elem a legnagyobb. Ezután a tömb minden megmaradt elemét összehasonlítjuk az aktuálisan legnagyobbnak elfogadottal. Amennyiben a soron következő elem a nagyobb, ezentúl ezt tekintjük majd a legnagyobbnak.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i: Integer; max: Real;
begin
   for i := 1 to 10 do
      a[i] := StrToFloat (Memo1.Lines[i-1]);
      max := a[1]; // Az első elemet fogadjuk el legnagyobbnak
   for i := 2 to 10 do
      if a[i] > max then max := a[i]; // aktualizáljuk a legnagyobb elem értékét
   Label1.Caption := FloatToStr (max) + ' – legnagyobb elem értéke';
end;
```

5. feladat. Keressük meg a 10-elemű tömb legnagyobb és legkisebb elemét és cseréljük meg azokat.

Mivel a tömb két eleme helyet kell cseréljen, nem elég tudni, mennyi a legnagyobb és legkisebb elem, ismerni kell ezek sorszámát is a tömbben. Próbáljuk meghatározni ezeket az adatokat úgy, hogy a tömböt csak egyszer járjuk be.

Kezdetben feltételezzük, hogy az első elem a legnagyobb és a legkisebb is, és ezek sorszáma 1. Ezután megvizsgáljuk a fennmaradt elemeket. Amennyiben a soron következő elem meghaladja a legnagyobbat, akkor ezentúl ezt az elemet tekintjük a legnagyobbnak és a sorszámát is feljegyezzük az ebből a célból bevezetett változóba. Hasonlóképpen járunk el, amennyiben a soron következő elem kisebb, mint az eddigi legkisebb.

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i, nmax, nmin: Integer; max, min: Real;
begin
   for i := 1 to 10 do
      a[i] := StrToFloat (Memo1.Lines[i-1]);
      max := a[1]; nmax := 1; // Az egyes sorszámú elem a legnagyobb
      min := a[1]; nmin := 1; // Az egyes sorszámú elem a legkisebb
   for i := 2 to 10 do
      if a[i] > max // Az aktuális elemet összehasonlítjuk a legnagyobbal
      then
         begin
         max := a[i]; // Lecseréljük a legnagyobb elem értékét
         nmax := i; // Lecseréljük a legnagyobb elem sorszámát
         end
      else
         if a[i] < min // Az aktuális elemet összehasonlítjuk a legkisebbel
      then
         begin
         min := a[i]; // Lecseréljük a legkisebb elem értékét
         nmin := i; // Lecseréljük a legkisebb elem sorszámát
               end;
   a[nmax] := min; a[nmin] := max; // A legnagyobb és legkisebb elem cseréje
   for i := 1 to 10 do
      Memo1.Lines.Append (FloatToStr(a[i]); // A megváltozott tömb elemeinek
kiírása
end;
```

6. feladat. Válasszuk ki az adott 10-elemű tömb elemei közül azokat, amelyek meghaladják az adott valós értéket!

Nyilvánvaló, hogy a 2. feladathoz hasonlóan meg kell vizsgálni a tömb összes elemét, mivel ezek nagyobbak vagy kisebbek lehetnek az adott számnál. A következő kérdés az, hol tároljuk a vizsgált feltételt kielégítő elemeket, mivel ezek száma 0 és 10 között bármennyi lehet.

Célszerű lenne ezeket egy új tömbbe írni, de sajnos előre nem ismerjük a tömb elemszámát.

A **Delphi**-ben használhatunk olyan tömböket is, amelyek elemszáma előre nem ismert. Ezeket **dinamikus tömböknek** nevezzük. Az olyan tömböket, amelyek elemszáma ismert és a program futása során nem változhat, **statikus tömböknek** nevezzük.

Megjegyezzük, hogy a tömb első elemének sorszáma alapértelmezés szerint 0.

A valós elemekből álló **b** dinamikus tömböt a **var** sorban a következőképpen deklaráljuk: **var b: array of Real;**

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; b: array of Real; i, k: Integer; x: Real;
begin
   for i := 1 to 10 do
      a[i] := StrToFloat (Memo1.Lines[i-1]);
      x := StrToFloat (Edit1.Text);
      k := 0; // A b tömb elemeinek k számlálója (kezdetben a b nem tartalmaz elemeket)
   for i := 1 to 10 do
      if a[i] > x
      then  begin
         k := k+1; // A b tömb elemszámát eggyel növeljük
               SetLength (b, k); // Beállítjuk a b dinamikus tömb elemszámát
               b[k-1] := a[i]; // a megfelelő elemek átírása a b tömbbe
            end;
Memo2.Lines.Clear;
if k = 0
then Memo2.Lines[0] := Nincsenek ilyen elemek '//
else
   for i := 1 to k do
   Memo2.Lines.Append (FloatToStr(b[i-1]));
end;
```

Setlength (*set* – beállítani, *length* – a hosszúságot).

A dinamikus tömbök alkalmazásával tetszőleges számú elemet olvashatunk be a többsoros mező soraiból. Ez esetben a mező sorait dinamikus tömbbe olvassuk be, valahogy így:

k := Memo1.Lines.Count; // Meghatározzuk, hány sort tartalmaz a mező

SetLength (a, k); // Beállítjuk a dinamikus tömb méretét, és lefoglaljuk a szükséges memóriaterületet

For i := 0 to k-1 do a[i] := StrToFloat (Memo1.Lines[i]); // Adatok beolvasása a tömbbe.

Ezután a dinamikus tömb elemeit a megismert módszerekkel dolgozhatjuk fel.

A tömb rendezése és keresés a rendezett tömbben

A tömb elemeinek feldolgozása sok esetben lényegesen gyorsabb, ha a tömb elemei növekvő vagy csökkenő sorrendbe vannak állítva. Ez elsősorban a tömb elemei közötti keresést gyorsítja meg.

A tömb elemeinek sorba rendezésére több mint 10 módszer létezik. Ezek között vannak gyorsabbak és lassabbak, logikailag egyszerűek és összetettek. Mi ezek közül eggyel fogunk foglakozni, a **kiválasztás módszerével.**

Vizsgáljuk meg a módszer működését egy példán. Rendezzük a 2.5. táblázat 0. sorában megadott tömböt!

2.5. táblázat. **Egydimenziós tömb rendezése a kiválasztás módszerével**

№	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
0	23	15	7	2	6	4
1	2	15	7	23	6	4
2	2	4	7	23	6	15
3	2	4	6	23	7	15
4	2	4	6	7	23	15
5	2	4	6	7	15	23

Első lépésként megkeressük a tömb legkisebb elemét (a[4] = 2), és megcseréljük azt az első elemmel, ahogy azt a táblázat első sorából láthatjuk. Második lépésként megkeressük a tömb fennmaradó részében a legkisebb elemet (a[6] = 4), és megcseréljük azt a másodikkal. Ilyen módon, ahogy a második sorban látjuk, az első két elem a helyére kerül. Harmadik lépésként ismét megkeressük a fennmaradó rész legkisebb elemét (a[5] = 6), és megcseréljük azt a harmadikkal. Megkapjuk a táblázat harmadik sorát. Megismételve az eljárást még kétszer megkapjuk a rendezett tömböt.

A módszer működését elemezve megérthetjük, miért kapta a módszer a kiválasztás módszere nevet. Minden lépésben kiválasztjuk a tömb rendezetlen részében a legkisebb elemet, és azt a rendezetlen rész elejére tesszük.

A módszer lehetséges megvalósítása így néz ki:

```
procedure TForm1.Button2Click (Sender: TObject);
var a: array [1..10] of integer; i, j, min, nmin: integer;
begin
   for i := 1 to 10 do
   a[i] := StrToInt (Memo1.Lines[i-1]);
   for i := 1 to 9 do
   begin
      min := a[i]; nmin := i;
      for j := i+1 to 10 do
      if a[j] < min then
      begin
         min := a[j]; nmin := j;
      end;
      a[nmin] := a[i];
      a[i] := min;
   end;
   Memo2.Lines.Clear;
   for i := 1 to 10 do
   Memo2.Lines.Append (IntToStr (a[i]))
end;
```

Győződjünk meg róla, hogy a keresés (3. feladat) rendezett tömbben lényegesen gyorsabban megoldható, mint rendezetlenben.

Rendezetlen tömbben csak a már megismert módon kereshetünk egy elemet. Ez esetben a keresett elemet össze kell hasonlítani a tömb **valamennyi elemével** még akkor is, ha az egyszer sem fordul elő a tömbben. Egy nagy elemszámú (1000 vagy annál is több) tömbben ilyen nagy lesz az összehasonlítások száma is.

Rendezett tömbben a keresés sokkal gyorsabban megvalósítható. Lássunk egy példát. Legyen egy rendezett 10 elemű tömbünk: 2, 5, 8, 12, 13, 16, 17, 20, 22, 30, amelyben meg kell találnunk egy x számot. Hasonlítsuk össze ezt a számot a tömb középső elemével, a 13-mal. Amennyiben $x = 13$, akkor megtaláltuk, ha nem, megvizsgáljuk, hogy x meghaladja-e a 13-at. Ha igen, akkor azt a tömb jobb oldali felében kell tovább keresni, ha nem, akkor a bal oldaliban. A vizsgálatot tehát most már csak egy fele akkora tömbön kell elvégezni. A folytatást hasonló módon végezzük. Megkeressük a fennmaradó rész középső elemét, és összehasonlítjuk azt a keresendő számmal. Ha ezek egyenlők, akkor megtaláltuk a keresett számot, ha nem, amennyiben kisebb, a fennmaradt rész bal, ha nagyobb, akkor a jobb oldali felében folytatjuk a keresést.

A módszer sokkal hatékonyabb az előzőnél, különösen nagy N elemszám esetében. A maximális lépésszám $[\log_2 N] + 1$, ahol N a tömb elemeinek száma. A szögletes zárójel a képletben a szám egész részét jelenti. A módszer **bináris vagy logaritmikus keresés** néven ismert.

Lássuk a módszert megvalósító kódrészletet:

```
procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Integer; i, x, left, right, m: Integer; f: Boolean;
begin
   for i := 1 to 10 do
      a[i] := StrToInt (Memo1.Lines[i-1]);
      x := StrToInt (Edit1.Text);
      left := 1; // A tömb vizsgálandó része bal oldali elemének sorszáma
      right:= 10; // A tömb vizsgálandó része jobb oldali elemének sorszáma
      f := false; // A keresett számot még nem találtuk meg
      while (left <= right) and not f do
         begin
            m := (left + right) div 2; // A középső elem sorszáma
            if x > a[m]
            then left := m+1 // A tömb vizsgálandó részét balról szűkítjük
            else   if x < a[m]
                   then right := m-1 // A tömb vizsgálandó részét jobbról szűkítjük
                   else f := true; // Megtaláltuk a keresett számot
         end;
   if f
   then Edit1.Text := 'A keresett szám előfordul a tömbben'
   else Edit1.Text := 'A keresett szám nem fordul elő a tömbben';
end;
```

Érdekes történelmi tények

A keresési intervallum felezésének módszerét az ókori görögök is ismerték. Ők dichotómiának (δῐχῆ – felére, τομή – osztani) nevezték. A módszert a „Találd ki a gondolt számot" játékban is jól lehet használni.

Ellenőrző kérdések

1°. Mi az egydimenziós tömb?
2°. Milyen részekből áll a tömb egy elemének neve?
3•. Mit alkalmazhatunk a tömb elemeinek sorszámozására?
4*. Milyen feladatok megoldása során célszerű a tömbök alkalmazása?
5•. Hogyan deklaráljuk az egydimenziós tömböt?
6°. Milyen számköz lehet a tömb elemeinek sorszáma?
7•. Ismertessük a többsoros mező tulajdonságait!
8•. Ismertessük a többsoros mező metódusait!
9•. Hogyan tölthetjük fel a többsoros mezőt a projekt futtatása előtt?
10•. Ismertessük, milyen módszerekkel állapíthatjuk meg, tartalmazza-e az adott tömb az általunk keresett elemet!
11•. Ismertessük a tömb legnagyobb eleme megkeresésének módját!
12°. Mi a dinamikus tömb, és hogyan deklaráljuk?
13•. Mi különbség van a statikus és dinamikus tömbök között?
14•. Ismertessük a tömb rendezésének a legkisebb elem kiválasztásán alapuló módszerét!
15•. Ismertessük a bináris keresést!
16*. Magyarázzuk el, miért hatékonyabb a bináris keresés a rendezetlen tömbben történő keresésnél! Hozzunk fel példákat arra, hol alkalmazzuk a két keresési módszert a mindennapokban!

Végezzük el a feladatokat!

1°. Végezzük el az alábbi programrészlet futtatását! Határozzuk meg a **dob** változó eredményét a 2; –3,5; 1,2; 0,25; –4 tömb esetében:

```
dob := 1;
for i := 1 to 5 do
dob := dob * a[i];
```

2°. Végezzük el az alábbi programrészlet futtatását! Határozzuk meg a **sum** változó eredményét a 2; –3,5; 1,2; 0,25; –4 tömb esetében:

```
sum := 0;
for i := 1 to 4 do
if a[i] mod 2 =0 then sum := sum + a[i]*a[i];
```

3•. Deklaráljuk a következő tömböket:
a) *x* nevű egész tömb, amelynek indexei 1 és 50 közöttiek;
b) *mas* nevű valós tömb, amelynek indexei 10 és 35 közöttiek;
c) *tab* nevű egész tömböt, amelynek indexei –7 és 12 közöttiek.

4°. Írjuk meg azt a programrészt, amely feltölt egy nyolc elemű tömböt egy többsoros beviteli mező soraiból!

5•. Írjuk meg azt a programrészt, amely kiír egy 11 elemű tömböt egy többsoros beviteli mező soraiba!

6•. Hozzunk létre projektet, amely feltölt egy 10-elemű tömböt egy többsoros mező soraiból, kiszámítja azok átlagát és kiírja azt egy feliratba. Mentsük a projektet.

7•. Hozzunk létre projektet, amely feltölt egy 12-elemű tömböt egy többsoros mező soraiból és kiírja az elemek háromszorosát egy másik többsoros mezőbe! Mentsük a projektet!

8•. Hozzunk létre projektet, amely feltölt egy 9-elemű tömböt egy többsoros mező soraiból, és kiírja az elemek 5-tel csökkentett értékét egy másik többsoros mezőbe! Mentsük a projektet!

9•. A 11-edik osztályos tanulók az év végén oklevelet kaphatnak azokból a tárgyakból, amelyekből az éves jegyük 12-es. Az osztály éves eredményeit egy többsoros mező tartalmazza. Hozunk létre projektet, amely meghatározza, hány oklevélre lesz szükség! Mentsük a projektet!

10*. Hozzunk létre projektet, amely feltölt egy 7-elemű egész tömböt egy többsoros mezőből, és kiszámítja a páros számok összegét és kiírja azt egy feliratba! Mentsük a projektet!

11*. Hozzunk létre projektet, amely feltölt egy 9-elemű egész tömböt egy többsoros mezőből és meghatározza, tartalmaz-e a tömb 3-mal osztható számokat, és kiírja az eredményt egy feliratba! Mentsük a projektet!

12*. Hozzunk létre projektet, amely feltölt egy 10-elemű valós tömböt egy többsoros mezőből és meghatározza annak legkisebb elemét és felcseréli azt az elsővel! A módosított tömböt írjuk ki egy másik többsoros mezőbe! Mentsük a projektet!

13*. Hozzunk létre projektet, amely feltölt egy 10-elemű valós tömböt egy többsoros mezőből és meghatározza annak legnagyobb elemét valamint azt, hányszor fordul elő ez az elem a tömbben! Az eredményt írjuk ki egy feliratba! Mentsük a projektet!

14*. Hozzunk létre projektet, amely feltölt egy 9-elemű valós tömböt egy többsoros mezőből és meghatározza annak legnagyobb elemét és felcseréli azt a középsővel! A módosított tömböt írjuk ki egy másik többsoros mezőbe! Mentsük a projektet!

15*. Hozzunk létre projektet, amelyben létre kell hozni egy dinamikus tömböt a többsoros mező sorainak beolvasására. Határozzuk meg a tömb elemeinek átlagát. A tömb azon elemeit, amelyek meghaladják az átlagot, írjuk át egy másik dinamikus tömbbe. Mentsük a projektet.

16*. Hozzunk létre projektet, amelyben létre kell hozni egy tömböt a többsoros mező sorainak beolvasására! A tömböt rendezzük és határozzuk meg, hány különböző elemet tartalmaz a tömb! Mentsük a tömböt!

17*. Hozzunk létre projektet, amelyben létre kell hozni két tömböt két többsoros mező sorainak beolvasására! A program döntse el, hogy a két tömb megfelelő elemei egyenlők-e! Mentsük a projektet!

11. számú gyakorlati munka Egydimenziós tömbök kezelése

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Indítsuk el a **Turbo Delphi 2006** rendszert!
2. Hozzunk létre projektet a következő feladat megoldására: *Ismerjük a hét minden napjának átlaghőmérsékletét. Határozzuk meg, hány napon volt az átlaghőmérséklet pozitív! Rendezzük az adatokat, és állapítsuk meg, voltak-e a hét folyamán olyan napok, amelyeken az átlaghőmérséklet ugyanolyan volt!* A projektben:
   a) helyezzük el a form-on a szükséges vezérlőket: többsoros mezőket, feliratokat, gombokat;
   b) az egyik gomb **OnClick** eseménykezelője határozza meg az átlaghőmérsékletet, és írja ki azt egy feliratba;
   c) a másik gomb **OnClick** eseménykezelője határozza meg, hányszor volt pozitív a hőmérséklet, és írja ki azt egy feliratba;
   d) a harmadik gomb **OnClick** eseménykezelője rendezze a tömböt, határozza meg, vannak-e azonos elmei, írja ki a rendezett tömböt egy többsoros mezőbe, azt pedig, hogy vannak-e azonos elemek, egy feliratba;
   e) végezzük el a fenti programrészleteket, és győződjünk meg arról, hogy helyes eredményt kaptunk.
3. Mentsük a projektet **Gyakorlati11** néven!

3. fejezet

ADATBÁZISOK. ADATBÁZIS-KEZELŐ RENDSZEREK

Ebben a fejezetben megismerkedünk:

- az adatbázisokkal, a számítógépes adatbázisokkal, azok csoportosításával és rendeltetésével;
- az adatmodellekkel és azok csoportosításával;
- a relációs adatmodellel, annak objektumaival és sajátosságaival;
- az egyed-kapcsolat modellel, annak felépítésével és adatbázissá alakításával;
- az adatbázis-kezelő rendszerekkel és azok rendeltetésével;
- az **Access 2007** adatbázis-kezelővel és annak használatával;
- a keresés, rendezés és szűrés végrehajtásával **Access 2007**-ben.

3.1. Adatbázisok Az egyed-kapcsolat modell

1. Mi az információs rendszer? Mik ennek összetevői és mi azok rendeltetése?
2. Mi az objektum modellje? Milyen modelleket ismerünk?
3. Mi a tudományterület? Hozzunk fel példákat!

Az adatbázis fogalma

Az embernek a mindennapi tevékenysége során, szakmájától és érdeklődési területétől függően sokszor van szüksége bizonyos információk kikeresésére. Fontos lehet megtudni, melyik vágányról indul a vonat Hmelnickbe, hogyan kell elkészíteni a lekvároS derelyét, milyen a vízmolekula felépítése, hány napig tartott a második világháború, igaz-e a fekete macskáról szóló babona, milyen a hálózati áram frekvenciája, hogy képezzük a főnevek birtokos esetét stb. A feleleteket megtalálhatjuk a memóriánkban, érdeklődhetünk a vasút információs szolgálatánál, elolvashatunk egy szakácskönyvet, fizikai, kémia, vagy elektrotechnikai kézikönyvet vagy helyesírási szótárt. A keresést megkönnyítendő az emberek sokféle eszközt hoztak létre, például általános és szakmai enciklopédiákat, szótárakat, tájékoztató táblákat, tudakozókat.

Az emberek által felhalmozott tudás mértéke folyamatosan növekszik. Az ókori sumér Ur városának ásatásai során 20 ezer agyagtáblát találtak, amelyek tartalmazták az akkor élt emberek szokásait, legendáit és nagyjából 5000 év történelmi eseményeit. Az alexandriai könyvtárban az i. e. 3. században tárolt kéziratok száma 100 ezer és 700 ezer közötti volt. Jelenleg a világ legnagyobb könyvtárának tartott londoni könyvtárban 150 millió tárolási egységet tartanak számon. Kijevben, a Vernadszkij nevét viselő nemzeti könyvtárban 15 millió a tárolási egységek száma.

A tudósok állítása szerint az ilyen nagy mennyiségű információ tárolása csak abban az esetben indokolt, ha abban a keresést gyorsan el lehet végezni, és a keresés eredményét jól értelmezhető formában lehet előállítani. A mai számítógépes tárolási módszerek kielégítik ezeket a feltételeket. A technológia alapját a számítógépes **adatbázisok** képezik.

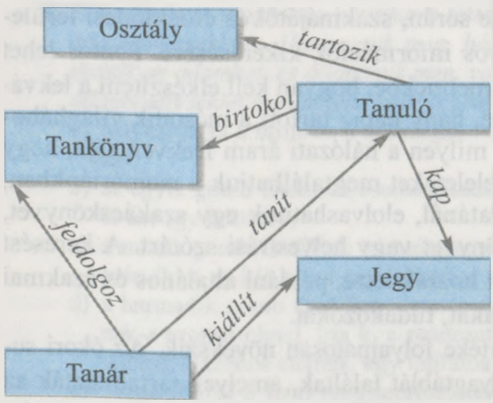
Az *adatbázisok* alatt összekapcsolt adatok rendezett összességét értjük.

Ukrajnában az első számítógépes adatbázist a Kijev elektronikus számítógép tervezése során hozták létre 1959-ben. A számítógépet L. N. Dasevszkij, K. L. Juscsenko, K. O. Skarabara, Sz. B. Pogrebinszkij tervezte az Ukrán Tudományos Akadémia számára B. V, Gnidenko és V. M. Gluskov akadémikusok vezetése alatt.

3.1. ábra. Viktor Mihajlovics Gluskov

Viktor Mihajlovics Gluskov (3.1. ábra) (1923–1982) több, olyan munkát is publikált, amely a számítógépnek egy vállalat vagy egy egész ország vezetésében történő alkalmazásával foglakozott. Az ő vezetésével hozták létre az **Országos Információgyűjtő és Feldolgozó Projektet**, amelynek célja a gazdasági döntéshozatal számítógépes segítése volt. A rendszer alapját egy adatbázis, vagy ahogyan Gluskov nevezte, adatbank képezte. 1975-ben a tudós azt jósolta, hogy a fejlett országokban már a következő évtizedekben nemzeti adatbankokat fognak létrehozni. A Gluskov által megjósolt adatbankok lényegesen megváltoztatták a mérnök, a tudós, az újságíró munkáját.

Az egyed fogalma. Az egyed-kapcsolat modell



3.2. ábra. Az **iskolai informatikaoktatás** modellje.

Az adatbázisok tervezésekor először létre kell hozni a leendő alkalmazási terület modelljét. Ez a modell tartalmazza, milyen objektumokra terjed ki az adatbázis, és milyen kapcsolatok állnak fenn ezen objektumok között. Vizsgáljuk meg példaként az **iskolai informatikaoktatás** 3.2. ábrán látható modelljét. Az érthetőség kedvéért a modellt grafikusan ábrázoltuk.

A téglalapok az objektumok típusait tartalmazzák, a nyilak az objektumok közötti kapcsolatokat reprezentálják, a nyilak felirata jellemzi magát a kapcsolatot. A **Tanár** előzetesen *feldolgozza* a **Tankönyvet** és megszervezi az oktatási folyamatot (*tanít*). Ugyanő értékeli a tanulók eredményeit és **Jegyeket** ad. A **Tanulók** a tanár vezetésével vagy önállóan tanulnak, *ren-*

delkeznek **Tankönyvvel**. Ugyanők valamelyik **Osztályhoz** tartoznak és **Jegyeket** kapnak.

Az objektumok típusát **egyedhalmaznak** nevezzük. A vizsgált modell egyedhalmazai az **Osztály, Tanuló, Tanár, Tankönyv** és a **Jegy.** Az egyedhalmaz tehát azonos típusú objektumok összessége. Az **Osztály** egyedei a **11. A, 11. B, 11. C** stb. Az egyes objektumok tehát az **egyedeknek** felelnek meg. Minden egyedhalmazhoz legalább egy egyed tartozik. A **Tankönyv** egyedhalmazhoz például csak egy egyed tartozik, amennyiben mindenki ugyanolyan tankönyvet használ.

Az *egyedhalmazok* – az adott alkalmazási terület valós vagy elképzelt objektumainak típusai.

A további formalizáció során az egyedhalmazok közötti kapcsolatok szóbeli leírását egyezményes jelekkel cseréljük fel, amelyen feltüntetjük a kapcsolat típusát. A kapcsolat típusa a következő lehet:

- **egy az egyhez (1:1)**, amikor az egyik egyedhalmaz egy egyedéhez a másik egyedhalmaz egy egyede tartozik. A vizsgált esetben ilyen kapcsolat áll fenn a **Tanár** és a **Tankönyv** között, amennyiben a tanár egyféle tankönyvet használ, mégpedig olyat, amilyet a többiek nem használnak;
- **egy a többhöz (1:M),** amikor az egyik egyedhalmaz egy egyedéhez a másik egyedhalmaz több egyede tartozhat. Ilyen kapcsolat áll fenn az **Osztály** és a **Tanuló** egyedhalmazok között. A **Tanuló** és az **Osztály** közötti kapcsolat típusa ezek szerint **több az egyhez (M:1)** lesz. Mivel a két kapcsolattípus ugyanazt jelöli, nem különböztetjük meg ezeket egymástól.
- **Több a többhöz (M:M),** amikor az egyik egyedhalmaz egy egyede a másik több egyedével áll kapcsolatban és fordítva. Ilyen kapcsolat áll fenn a **Tanulók** és a **Jegyek** között.

A kapcsolatot jellemző másik tulajdonság a ***kapcsolat teljessége.*** Teljesség szerint megkülönböztetünk:

- Olyan kapcsolatot, amikor az egyik egyedhalmaz minden egyede ***kötelezően kapcsolódik*** a másik egyedhalmaz egy vagy több egyedéhez (kötelező). Ilyen a kapcsolat az **Osztály** és a **Tanuló** egyedhalmazok között, amennyiben feltételezzük, hogy egy tanuló kötelezően tagja kell legyen valamelyik osztálynak.
- Olyan kapcsolatot, amikor az adott egyedhalmaz elemeire nézve ***nem kötelező kapcsolódnia*** a másik egyedhalmaz elemeihez (***opcionális***). Ilyen kapcsolat állhat fenn például a **Tanuló** és a **Számítógép** egyedhalmazok között, amennyiben nem jut minden tanulónak számítógép.

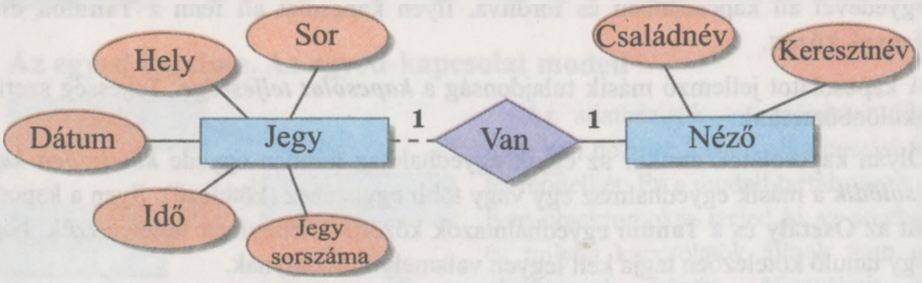
A vizsgált modellt **egyed-kapcsolat** modellnek, vagy **ER-modellnek** (*Entity* – egyed, *Relationship* – kapcsolat) nevezik. Az ER-modell ábrázolásakor speciális jeleket alkalmazunk az egyedhalmaznak, az egyedhalmaz tulajdonságainak és az egyedhalmazok között fennálló kapcsolatnak a jelölésére. Ezt az egyezményes jelkészletet **notációnak** (*notation* – jelölés, jelölési rendszer) nevezzük. A 3.1. táblázatban ezeket a jeleket foglaltuk össze.

3.1. táblázat. **Az egyed-kapcsolat modell megadásának grafikus elemei**

Egyezményes jel	*Alakzat*	*Mit jelöl*
Neve	Téglalap	Egyedhalmaz
Neve	Ovális	Az egyed egy tulajdonsága
Neve	Rombusz	Kapcsolat
———	Folytonos vonal	Kötelező kapcsolat
- - - - - - - - - - -	Szaggatott vonal	Nem kötelező kapcsolat

Gyakorlásképpen hozzuk létre a **Mozilátogatás** ER-modelljét. A modellben két egyedhalmaz lesz, a **Jegy** és a **Nézŏ**. A Nézŏ-nek a mozilátogatáshoz kötelező a jegy. Nem létezik két ugyanolyan nézŏ vagy két ugyanolyan jegy (3.3. ábra).

A **Nézŏ** egyedhalmaznak két tulajdonsága van: **Családneve, Keresztneve,** a **Jegy** egyedhalmaznak pedig öt: **Dátum, Idŏ, Sor, Hely, Sorszám.** A két egyedhalmaz között a **Van** kapcsolat áll fenn. Ez egy az egyhez kapcsolat, amit a kapcsolatot reprezentáló vonalon látható számok is mutatnak. A kapcsolat kötelező a **Nézŏ**-re nézve, mivel jegy nélkül nem mehet be a moziba, de nem köztelező a **Jegy**-re nézve, mivel nem mindig adnak el minden jegyet.



3.3. ábra. A mozilátogatás ER-diagramja

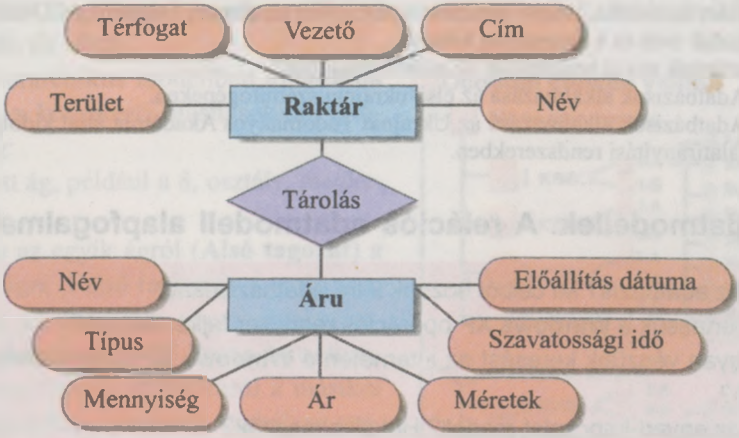
Ellenőrző kérdések

1°. Miért van szükség az adatok rendszerezett tárolására?
2°. Milyen követelményeket támaszt az információs társadalom az adattárolás elé?
3°. Mi az adatbázis? Hol alkalmazzák?
4°. Mi az egyedhalmaz?
5°. Mi az egyed? Hozzunk fel példákat!
6°. Milyen kapcsolatok állhatnak fel az egyedhalmazok között?
7°. Jellemezzük az egy az egyhez kapcsolatot!
8°. Jellemezzük az egy a többhöz kapcsolatot! Hozzunk fel példákat!

9°. Jellemezzük a több a többhöz kapcsolatot!
10°. Milyenek lehetnek a kapcsolatok teljesség szerint?
11°. Mi az egyed-kapcsolat modell? Mit ír le?
12°. Milyen egyezményes jeleket használunk az ER-diagramon?
13°. Mi az ER-diagram egyezményes jeleinek értelme?
14°. Milyen szerepe volt Gluskov akadémikusnak az adatbázisok alkalmazásának kidolgozásában?

Végezzük el a feladatokat!

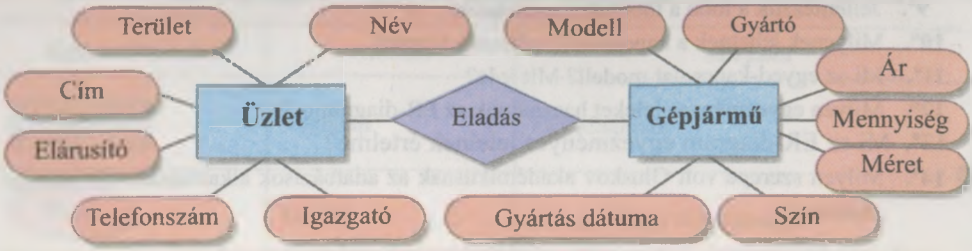
1°. Vizsgáljuk meg a 3.4. ábra ER-diagramját!
a) Milyen egyedhalmazok szerepelnek rajta?
b) Milyen tulajdonságokkal bírnak az egyes egyedhalmazok?



3.4. ábra

c) Határozzuk meg a kapcsolatok típusát a következő esetekben.
- Van néhány raktárunk és néhány árufajtánk, az egyes raktárakban minden árucikket tárolnak.
- Csak egyfajta árunk van, amit egy raktárban tárolnak.
- Több raktárunk van és több árucikkünk. Egy árucikket csak egy raktárban tárolnak és minden raktárban csak egyfajta árut tárolnak.
- Van néhány árufajtánk, amelyek mindegyikét ugyanabban az egy raktárban tárolják.
- Egy árufajtánk van, ezeket több raktárban tárolják.

2°. Vizsgáljuk meg a 3.5. ábra ER-diagramját!
a) Milyen egyedhalmazok szerepelnek rajta?
b) Milyen tulajdonságokkal bírnak az egyes egyedhalmazok?
c) Határozzuk meg a kapcsolatok típusát a következő esetekben:
- Egy üzlet van és az egy autómárkát forgalmaz.
- Több üzlet és autómárka van, minden üzletben csak egy márkát forgalmaznak és minden márkát csak egy üzletben forgalmaznak.
- Több üzlet van és ezek mindegyike ugyanazt az autómárkát forgalmazza.
- Több márka és üzlet van, egy üzlet minden márkát forgalmaz.
- Egy üzlet van, amely minden márkát forgalmaz.



3.5. ábra

3°. Hozzuk létre az ER-modellt a következő esetekre!
   a) Az osztály telefonkönyve, amely mind a mobilszámokat tartalmazza, valamint a tanuló nevét és címét.
   b) Házi audiotéka, amely sorszámozott CD-ket tartalmaz, valamint a CD címét, kiadásának évét és a hanganyag hosszát.

4°. Készítsünk rövid beszámolót az alábbi témákban!
   - Adatbázisok alkalmazása az első ukrajnai számítógépeken.
   - Adatbázisok alkalmazása az Ukrajnai Tudományos Akadémia által kidolgozott vállalatirányítási rendszerekben.

3.2. Adatmodellek. A relációs adatmodell alapfogalmai

1. Mi az adatbázis? Mi célból hoznak létre adatbázisokat?
2. Ismertessük a **Windows XP** operációs rendszer fájlrendszerét!
3. Hogyan végzünk keresést az interneten a **Windows XP** operációs rendszerben?
4. Mi az egyed-kapcsolat modell? Hol alkalmazzák?

Az adatmodell fogalma. Az adatmodellek típusai

Az adatbázisokat nagy mennyiségű adat rendszerezett tárolása és kezelése céljából hozzák létre. Kezelésüket az adatbázisok felhasználói vagy számítógépes programok végezhetik. Mivel az adatbázis szerkezetét tekintve egy összetett objektum, létrehozásuk előtt meg kell azokat tervezni – ***adatmodelleket*** kell létrehozni. Az adatbázisok tervezése során valamelyik adatmodellt alkalmazzák. Az ***adatmodellek*** három részből állnak.

1. ***Adatszerkezetek,*** melyek meghatározzák az adatok szervezését, az alkalmazott adattípusokat (egész számok, valós számok, szövegek, multimédiás adatok, tömbök stb.), valamint azokat a műveleteket, amelyeket az adott adattípuson el lehet végezni.
2. ***Adatkezelési eszközök*** azok az utasítások, amelyek az adatok szervezésétől függetlenül (az algoritmus végrehajtójának utasításkészletéhez hasonlóan) biztosítják azok kezelését.
3. ***Integritási kényszerek,*** amelyek biztosítják az adatok épségét és hibátlanságát az adatbázis használata során. A kényszerek között vannak olyanok, amelyet a kiválasztott adatmodell követel meg és olyanok, amelyet a felhasználó hoz létre.

A legelterjedtebb adatmodellek **a hierarchikus, a hálós, a relációs és az objektum-orientált modellek.**

A **hierarchikus** (ιεραρχία – szent hatalom) **adatmodellben** az adatok egy fához hasonló szerkezetbe vannak szervezve. Ez az adatmodell megköveteli, hogy minden objektumhoz egyetlen fölérendelt objektum (szülőobjektum) tartozzék, leszármazott objektuma (gyerekobjektum) viszont több is lehet. Ezek közül csak a hierarchia tetején álló objektum képez kivételt, mivel ennek nem lehet szülőobjektuma.

A hierarchikus adatmodell példája a **FAT32** fájlrendszer (amelyet a 9. osztályban a rendszerprogramok tanulmányozása során már megismertünk). Ebben a fájlrendszerben létezik egy gyökérkönyvtár, amely fájlokat és más könyvtárakat tartalmazhat. A mappák (könyvtárak) mindegyike tartalmazhat fájlokat és más mappákat.

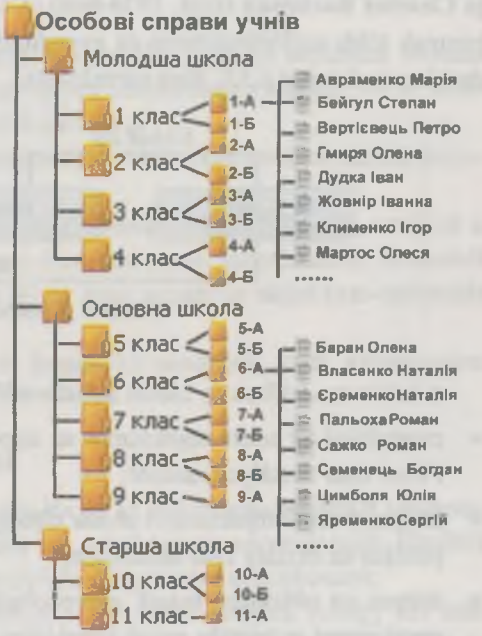
A 3.6. ábrán egy másik hierarchikusan szervezett adattárolási rendszert láthatunk.

Figyeljük meg, hogy bármely szülőobjektum az alárendelt objektumokkal egyetemben szintén fát alkot.

Az hierarchikus modellben az **adatok kezelését** a következő utasításokkal valósítják meg:

- az adott ág, például a 8. osztály, megkeresése;
- átlépés az egyik ágról (**Alsó tagozat**) a másik ágra (**Felső tagozat**).
- átlépés az objektum gyerekobjektumához, például a 6. **A**-tól **Boros Annához**;
- átlépés az egyik objektumról a másikra a hierarchikus szerkezet szabályai szerint, például a **7. B**-ről az 1. **A**-ra;
- új objektum elhelyezése az adott helyen;
- az aktuális objektum törlése;

és így tovább.



3.6. ábra. Egy hierarchikusan szervezett adattárolási rendszer

Az integritási ***kényszerek közé*** tartozik a hierarchikus modellben például az objektumok és szülőobjektumaik közötti kapcsolat feltétlen megőrzése. Alapszabály, hogy a gyökérelemen kívül egyetlen objektum sem létezhet szülőobjektum nélkül. Vagyis a **9. A** nem létezhet a **9. osztály** szülőobjektum nélkül, vagy amennyiben nincs kilencedik osztály az iskolában, akkor nem létezhet annak névsora sem.

A hierarchikus adatmodell alkalmazása akkor célszerű, ha az adatbázis alkalmazási területe olyan objektumokra terjed ki, amelyek között hierarchikus kapcsolat áll fenn.

A hierarchikus modell szerint megszervezett adatbázisokban nagy mennyiségű adat esetében a keresés meglehetősen nehézkes lehet. Próbáljunk például megkeresni a merevlemezen tárolt fájlok között azt, amely az adott szövegrészt tartalmazza. A keresés percekig, sőt, még tovább is eltarthat. A keresés során az információs ablakban megfi-

3.7. ábra. Charles Bachman

gyelhetjük, hogy a hierarchikus szerkezet vizsgálata áganként történik. Ugyanakkor az interneten a keresés néhány másodpercet vesz igénybe. Az operációs rendszer beépített keresési szolgáltatása a fájlszerkezetet ágról ágra bejárva vizsgálja meg. Az interneten a keresés a keresőszerverek által elemzett és szervezett adatbázisok alapján történik, amelyek más modell, például a relációs modell alapján vannak kialakítva. A **Windows 7** operációs rendszerben a keresést a gyakran használt fájlok indexelése segíti, aminek alapja az operációs rendszer által a fájlrendszer elemzése által létrehozott adatbázis.

A hierarchikus modellt nehéz alkalmazni olyan objektumok esetében, amelyek kapcsolatai nem hierarchikusak. Az ilyen esetekben a XX. század 60-as éveiben kifejlesztett ***hálós adatmodell*** lehet segítségünkre. A modell megalkotója **Charles Bachman** (szül. 1924-ben) (3.7. ábra). A modell feltételezi, hogy egy objektumnak több szülőobjektuma és gyerekobjektuma lehet. A hálós modellben megengedett kapcsolatokat a 3.8. ábra szemlélteti.

3.8.ábra. A hálós adatmodell példája

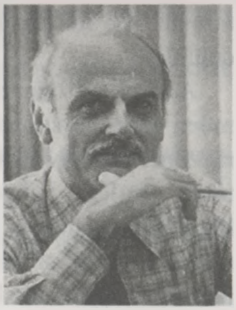
A hálós modellben az alábbi ***adatkezelő utasítások*** fordulhatnak elő:

- megtalálni az adott objektumot az adott azonos típusú objektumok között, például *Fehér* első osztályos tanulót;
- átlépni a szülőobjektumról annak első gyerekobjektumára az adott kapcsolat mentén, például az osztály első tanulójára;
- átlépni az objektum másik gyerekobjektumára az adott kapcsolat mentén, például megkeresni az osztály másik tanulójára;
- új objektum beszúrása az adott helyre;
- objektum törlése;
- objektum megváltoztatása;
- hozzákötni az objektumot az adott kapcsolathoz;

stb.

A hálós modell ***integritási kényszerei*** között lehetnek olyanok, amelyek megkövetelik bizonyos kapcsolatok feltétlen megtartását.

A hálós modell alkalmazása nagy adatmennyiség és összetett kapcsolatrendszer esetében nagyon bonyolult lehet. Nagyon nehézkes lehet az objektumok megváltoztatása és a kapcsolatok szerkesztése is, mivel ezek egymáshoz szorosan kapcsolódhatnak.



3.9. ábra. Edgar Frank Ted Codd

Az objektumok és a közöttük levő kapcsolatok könnyebb kezelhetősége céljából dolgozta ki **Edgar Frank Ted Codd** (1923–2003) amerikai tudós (3.9. ábra) a ***relációs adatmodellt.*** A matematikus végzettségű tudós az adatbázisok elméletét halmazelméleti alapokra helyezte.

A modell alapeleme a tábla (vagy távlázat). A tábla minden sora egy objektum tulajdonságait tartalmazza. A tábla egy oszlopa az objektumok azonos típusú adatát tartalmazza.

Egy tábla olyan oszlopok összessége, amelyek mindegyike bizonyos véges halmaz elemeit tartalmazhatja. A tábla matematikai szempontból egy relációnak tekinthető. Leírásukra ezért használta Codd a **reláció** (az angol *relation* –viszony) fogalmát és ezért kapta a modell a *relációs* nevet.

Mivel a tábla oszlopai bizonyos halmaz elemei, az ***adtok kezelésére*** halmazműveleteket használnak.

Az ***integritási kényszerek*** két elv megtartását követelik meg: az adatbázis minden objektumát egyértelműen lehessen azonosítani egyes tulajdonságai alapján, és az adatbázis táblái közötti kapcsolatok korrektek kell legyenek.

A relációs modell szerint felépített adatbázisokkal a következő részek tanulmányozása során ismerkedünk meg.

Az utóbbi években néhány tudós rámutatott a relációs modell korlátaira, amelyek a multimédiás adatok, illetve az összetett adatok kezelése során jelentkeznek. A modell lehetőségeit kiszélesítendő fejlesztették ki a 90-es évek végén az **objektum-orientált adatmodellt.**

Ezenkívül létezek még többdimenziós és kombinált modellek, ezek alkalmazása azonban nem terjedt el eléggé.

A relációs adatmodell alapfogalmai

A fentiekben a hierarchikus, a **hálós, a relációs és az objektum-orientált adatmodellekkel** foglakoztunk. E modellek képezik az adatbázisok tervezésének elméleti alapját. A folytatásban csak a relációs adatmodellek vizsgálatára szorítkozunk.

Ismerkedjünk meg a modell alapfogalmaival a 3.10. ábra alapján. Ahogy azt már említettük, a modell alapeleme a tábla. Mint minden kétdimenziós táblázat, a tábla is sorokból és oszlopokból áll. Az alkalmazási területtől függően az adattárolás több kapcsolt táblában is történhet.

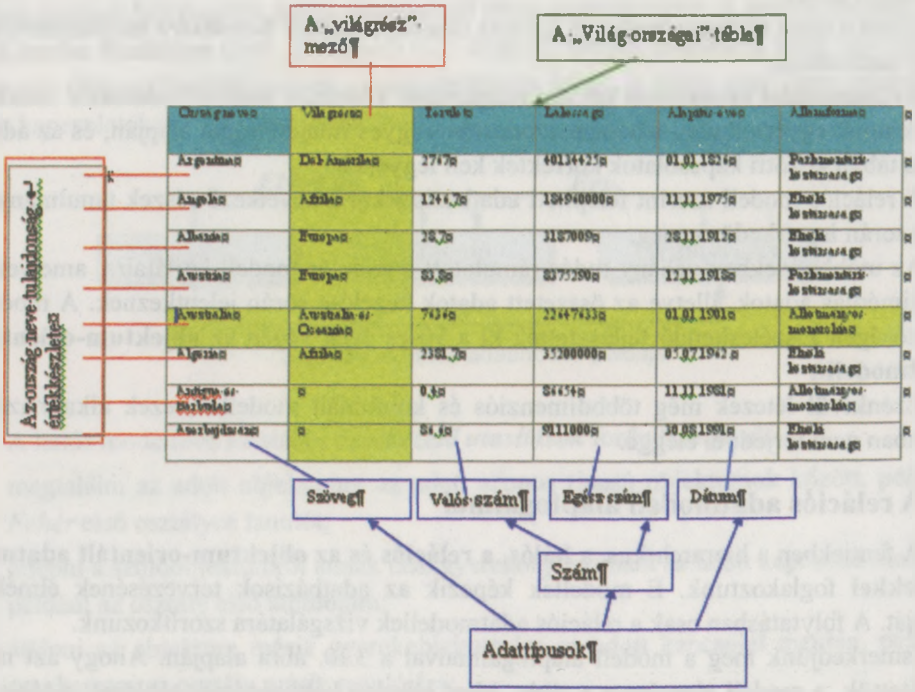
A tábla egy sorát **rekordnak** nevezzük. A rekord az egyed egy előfordulásának attribútumait (tulajdonságait) tartalmazza. A **világ országai** tábla (3.10. ábra) például a következő értékeket tartalmazza: *Ausztrália, Ausztrália és Óceánia, 7636, 19138, 01.01.1901, Alkotmányos Monarchia.* Ezek az egyed adott előfordulását jellemzik. **A világ országai** egyed attribútumai tehát az **ország neve, világrész, terület, lakosság, az alapítás éve** és az **államforma.** Bármelyik másik rekord ettől eltérő adatokat tartalmaz, mivel egy másik egyed-előfordulás egy másik országot jelent.

A tábla egy oszlopa a **mező** vagy **attribútum.** A mezőnek van neve és egy meghatározott tulajdonság értékeit tartalmazza. Egy adott mező azonos típusú adatokat tar-

talmaz. Az adatbázis tervezése során minden oszlophoz meghatározott adattípust rendelünk.

Ahogy azt már említettük, a relációs modellben a tábla minden sorának egyértelműen azonosíthatónak kell lennie. Ez a tábla esetében azt jelenti, hogy kell lennie olyan mezőnek, amely nem tartalmazhat két azonos értéket. Ezt a mezőt **kulcsnak** vagy a tábla **kulcs tulajdonságú mezőjének** nevezzük. A kulcsot szokták még **identifikátornak** is nevezni. A relációs modell egyes megvalósításai megengedik összetett kulcsok alkalmazását is, ilyen lehet például a *Családnév, Keresztnév, Apja neve* három mezőből álló kulcs.

A 3.10. ábrán látható táblában az **Ország neve** lehet a kulcs, mivel az egyedi minden ország esetében.



3.10. ábra. A relációs adatbázis egy táblája

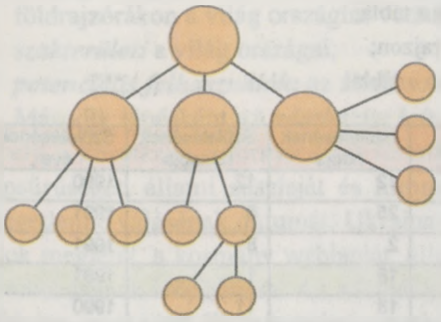
Ellenőrző kérdések

1°. Mi az adatmodell? Miért van szükség adatmodellekre?
2°. Milyen részekből áll az adatmodell?
3°. Mi határozza meg az adatmodell szerkezetét?
4°. Milyen adattípusok fordulnak elő az adatmodellekben?
5°. Mi az integritási kényszer?
6°. Melyek a legelterjedtebb adatmodellek?

7°. Jellemezzük a hierarchikus adatmodellt! Milyen követelményeket támaszt az adatszerkezettel, az adatkezeléssel és az integritási kényszerekkel szemben?
8°. Jellemezzük a hálós adatmodellt! Milyen követelményeket támaszt az adatszerkezettel, az adatkezeléssel és az integritási kényszerekkel szemben?
9°. Ki és mikor fejlesztette ki a relációs adatmodellt?
10°. Jellemezzük a relációs adatmodellt. Milyen követelményeket támaszt az adatszerkezettel, az adatkezeléssel és az integritási kényszerekkel szemben?
11°. Miben térnek el a relációs és az objektum-orientált adatmodellek?
12°. Mi határozza meg az adatbázisok típusát? Milyen adatbázis-típusokat ismerünk?
13°. Mi a relációs adatmodell alapeleme?
14°. Jellemezzük az adattáblát!
15°. Mi a kulcs? Milyen követelményeket támaszt a relációs modell a kulcs tulajdonságával szemben?

Végezzük el a feladatokat!

1°. Határozzuk meg, milyen adatmodellt mutatnak be a 3.11. ábrán látható rajzok!



a)

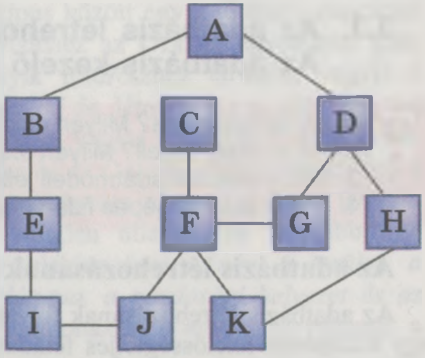
b)

3.11. ábra

2°. Határozzuk meg, milyen adatmodellt mutatnak be a 3.12. ábrán látható rajzok!

A Tulajdonság	B Tulajdonság	C Tulajdonság	D Tulajdonság
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
...	...	...	...
A_n	B_n	C_n	D_n

a)



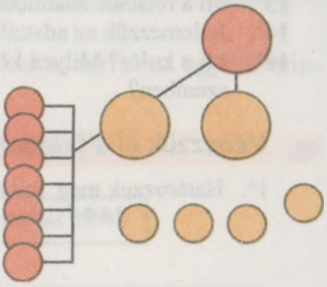
b)

3.12. ábra

3°. A Bit and Byte céget a vezérigazgató vezeti, akinek egy négytagú igazgatótanács van alárendelve: a marketing igazgató, a termelési igazgató, a személyzeti igazgató és a gazdasági igazgató. A marketing igazgatónak a marketing osztály, a termelési igazgatónak három szerelőműhely van alárendelve. A gazdasági igazgató a gazdasági előrejelzési osztályt és a könyvelést vezeti. Rajzoljuk meg a cégvezetés grafikai modelljét!

4°. Ki kell dolgozni egy olyan könyvtári adatbázis modelljét, amely a következő adatokat tartalmazza: *szerző* (családnév, keresztnév, apai név), a *kiadvány címe, típusa, a kiadás helye, ideje, a kiadó neve, oldalszám, a kötetek száma, a kötet sorszáma, nyilvántartási szám.* Milyen adatmodellt választanánk?

5°. Vizsgáljuk meg a 3.13. ábrát. Egészítsük ki a rajzot a kapcsolatokat ábrázoló vonalakkal, feltételezve, hogy:
a) a hierarchikus adatmodellt;
b) a hálós adatmodellt;
ábrázolja!



3.13. ábra

6°. Vizsgáljuk meg a relációs modell 3.14. ábrán látható tábláját. Határozzuk meg:
a) az 1-es és 2-es számokkal jelölt, színnel kiemelt objektumok neveit;
b) melyik mező lehet a táblázatban a kulcs;
c) milyen egyed tulajdonságait tartalmazza a tábla;
d) az egyed hány előfordulását láthatjuk a rajzon;
e) az egyed mely tulajdonságait tartalmazza a tábla!

Személyi akta száma	Családnév	Keresztnév	Apja neve	Születésének napja	[illegible]	Születésének éve
F-09	Nagy	Ferenc	János	12	12	1990
F-10	Kis	Miklós	Barnabás	25	5	1991
F-11	Fehér	Klára	Péter	2	8	1991
F-12	[illegible]	Piroska	Aladár	15	1	1991
F-13	Török	Gizella	Attila	18	11	1990
F-14	Vizi	Péter	Péter	7	3	11

3.14. ábra

3.3. Az adatbázis létrehozásának lépései. Az adatbázis-kezelő rendszer fogalma

1. Mi az adatbázis? Milyen célból hozzák létre?
2. Mi az adatmodell? Milyen részei vannak?
3. Mik a relációs adatmodell jellegzetességei?
4. Mik a számítógépes feladatmegoldás lépései?

Az adatbázis létrehozásának lépései

Az adatbázis létrehozásának folyamata a számítógépes feladatmegoldás példája. Ez egy különösen felelősségteljes feladat, különösen a tervezési szakasza az. A tervezés során elkövetett hibák néha csak nagy mennyiségű adat bevitele után derülnek ki. Ez nagy idő- és pénzveszteséggel járhat.

Az adatbázis létrehozása a következő lépésekből áll:

1. **A feladat megfogalmazása.** Ekkor határozzuk meg az adatbázis létrehozásának célját, az adatbázis alkalmazási területét és a lehetséges felhasználók körét.
2. **Az adatbázist leíró információs modell megalkotása,** amelynek során meghatározzuk a létrehozandó adatbázis egyedeit és azok kapcsolatait.
3. **Az egyed-kapcsolat modell létrehozása.**
4. **Az adatmodell kiválasztása.**
5. **Az adatbázis szerkezetének létrehozása valamely adatbázis-kezelő rendszerben.**
6. **Az adatok feltöltése.**
7. **Az adatbázis tesztelése és a szükséges változtatások elvégzése.**

A tesztelés és javítás még nem jelenti a munka tényleges befejezését. Az adatbázis használata során szükség lehet az adatok frissítésére, de az egyedek közötti kapcsolatok megváltoztatására is.

Vizsgáljuk meg az adatbázis létrehozásának folyamatát a **Világ országai** adatbázis példáján, amelyet az iskolai földrajzoktatásban lehet felhasználni.

Ahogy azt már tudjuk, az adatbázis létrehozásának első szakaszában a következő kérdéseket kell megválaszolni:

- a ***Világ országai adatbázis létrehozásának célja,*** hogy azt az általános iskolákban a földrajzórákon a világ országira vonatkozó ismeretek elmélyítésére használják;
- ***szakterület:*** a világ országai;
- ***potenciális felhasználók:*** az általános iskolák 9–10. osztályos tanulói.

Második lépésként az ***adatbázist leíró információs modell létrehozása*** következik, amely a következő információkat tartalmazza: az ország nevét, területét, népességét, népsűrűségét, állami zászlaját és himnuszát, nemzeti valutáját, létrehozásának vagy függetlenné válásának dátumát, Ukrajna és az adott ország közötti diplomáciai kapcsolatok meglétét, a kormány weblapját, államformáját és gazdasági helyzete az adatbázis létrehozásának időpontjára. Az adatbázis szerkezete lehetővé kell tegye, hogy egy adott földrész vagy adott államformára vonatkozó adatokat külön kielemezhessünk.

Az adatbázis létrehozásának harmadik szakaszában:

- létrehozzuk a **Világ országai** szakterület ***egyed-kapcsolat modelljét,*** amelyet a 3.15. ábrán láthatunk. Ebben három egyedtípus szerepel: az **Ország,** a **Földrész** és az **Államforma.** Az **Ország** és a **Földrész** egyedtípus között egy a többhöz kapcsolat áll fenn, azaz a **Földrész** egyedtípus egy egyedéhez az **Ország** egyedtípus több egyede kapcsolódik. Minden ország valamelyik földrészhez tartozik, vagyis a kapcsolat az ország egyedtípusra nézve kötelező, de létezik olyan földrész (az Antarktisz), amelyhez egyetlen ország sem tartozik, vagyis a földrész egyedtípusra nézve a kapcsolat nem kötelező. Az **Államforma** ás az **Ország** között szintén egy a többhöz kapcsolat áll fenn, amely mindkét egyedtípusra nézve kötelező: minden országnak van valamilyen államformája és minden államforma legalább egy országban előfordul. Az **Ország** egyedtípus tulajdonságai a *név, a terület, a népesség, a zászló, a himnusz, a létrehozás dátuma, a gazdasági helyzet és az kormányhonlap.* Az **Államforma** egyedtípus tulajdonságai a *név* és az *államfő*, a **Földrészé** pedig a *név*.
- A választott ***adatmodell*** relációs.

Az adatbázis létrehozásának további szakaszai speciális szoftver használatát igénylik, azért azokkal később foglalkozunk.

3.15. ábra. A **Világ országai** szakterület egyed-kapcsolat modellje

Az adatbázis-kezelő rendszer fogalma

Az adatbázisokat speciális szoftvereszközök – **adatbáziskezelő rendszerek** – segítségével fejleszthetjük és használhatjuk. A felhasználók ezek segítségével hozzák létre az adatbázisokat, hajtanak végre abban kereséseket, végezhetik az adatok karbantartását.

Az *adatbázis-kezelő* rendszerek – olyan számítógépes szoftverek, amelyek adatbázisok létrehozását, tárolását és kezelését teszik lehetővé.

Az alkalmazott adatmodelltől függően az adatbázis-kezelő rendszerek lehetnek **hierarchikusak, hálósak, relációsak vagy objektum-orientáltak.**

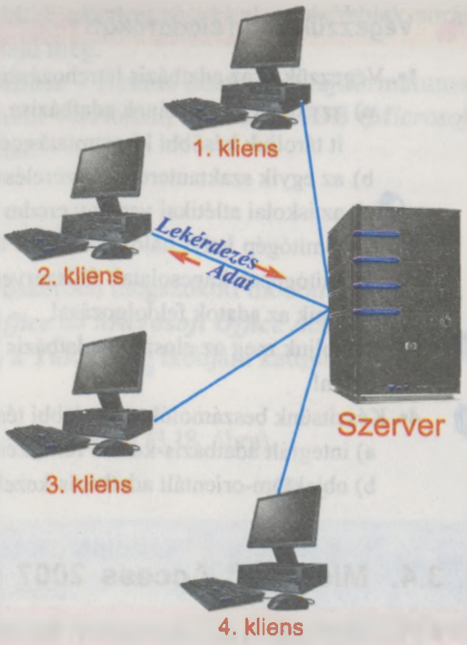
Attól függően, hogy az adatbázisok, illetve az azok kezelését végző szoftverek hol helyezkednek el, megkülönböztetünk **lokális és elosztott rendszereket.** A lokális rendszerek esetében az adatok és a szoftver is a helyi számítógépen vannak elhelyezve. Elosztott rendszer esetében az adatok tárolása és feldolgozása a helyi vagy a globális hálózat számítógépein történik.

A hozzáférés módjának függvényében az adatbázis-kezelő rendszerek (DBMS – Darta base management system) lehet **fájlszerver architektúrájú, kliens-szerver architektúrájú** és **integrált.** Fájlszerver architektúra esetében csak az adatok helyezkednek el a szerveren, a kliens-gépek mindegyikére telepítve van a teljes DBMS. Az ilyen rendszer hátrányai, hogy az általában drága szoftvert sok gépre kell telepíteni, nagy a hálózat leterheltsége és meglehetősen jó teljesítményű kliensgépekre van szükség. Előnye, hogy viszonylag kis teljesítményű fájlszervert igényel.

Az utóbbi években egyre nagyobb teret hódít a kliens-szerver technológia (3.16. ábra). Ebben az esetben az adatokon kívül a szerveren helyezkedik el a DBMS szerver része is. A kliens gépekre csak a DBMS nem túl nagy méretű és kisebb gépigényű kli-

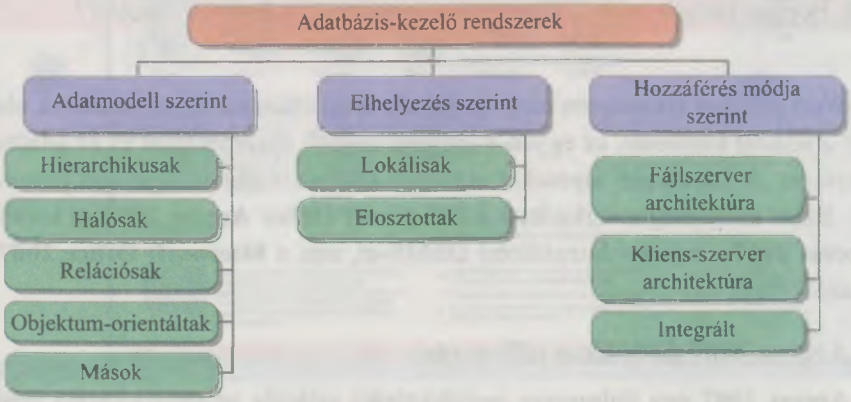
ens-részét kell feltelepíteni, melynek szerepe, hogy lekérdezéseket küldjön a szerverre, ahol a tényleges adatfeldolgozás zajlik, továbbá az, hogy megjelenítse az eredményeket. Ez esetekben nincs szükség nagy teljesítményű kliens-gépekre, nincs túl nagy hálózati forgalom, könnyebb az adatbázis integritásának biztosítása. Ugyanakkor megnőnek a szerverrel szemben támasztott követelmények. Ilyen például a vasúti jegybiztosítás és az ATM hálózatok architektúrája.

Az integrált DBMS-ek más szoftverek, például elektronikus enciklopédiák, szótárak, keresőgépek részeit képezik. Nem igényelnek külön installációt, általában korlátolt funkcionalitásúak (például nem biztosítják az adatok frissítését). Az adatok kezelését leginkább az a szoftver biztosítja, amelybe a DBMS be van építve.



3.16. A kliens-szerver architektúra

A DBMS-ek csoportosítását a 3.17. ábrán láthatjuk.



3.17. ábra. Az adatbázis-kezelő rendszerek csoportosítása

Ellenőrző kérdések

1•. Nevezzük meg az adatbázisok létrehozásának lépéseit!
2•. Jellemezzük a feladat megfogalmazásának folyamatát!
3°. Milyen feladatokat kell megoldani az adatbázis létrehozásának második lépésében?
4*. Milyen feladatokat oldunk meg az egyed-kapcsolat modell létrehozása során?
5°. Mik az adatbázis-kezelő rendszerek?
6°. Milyenek lehetnek a DBMS-ek az alkalmazott adatmodell szerint?
7•. Miben különböznek a lokális és elosztott DBMS-ek?
8*. Miben különböznek a fájlszerver és kliens-szerver architektúrák?
9•. Ismertessük a DBMS-ek osztályozását a 3.17. ábra alapján!

Végezzük el a feladatokat!

1•. Végezzük el az adatbázis létrehozásának első három lépését az alábbi esetekben:
a) az iskola végzőseinek adatbázisa, amelyben az egyes tanulók tantárgyankénti jegyeit tároljuk későbbi kinyomtatás céljából;
b) az egyik szaktanterem felszerelése nyilvántartásának adatbázisa;
c) az iskolai atlétikai verseny eredményeinek nyilvántartására szolgáló adatbázis.

2*. 6 számítógép helyi hálózatot alkot. Ezek egyike az adatbázisszerver. Rajzoljuk meg a számítógépek kapcsolatait fájlszerver architektúra esetére. A rajzon mutassuk meg, hol történik az adatok feldolgozása!

3*. Rajzoljuk meg az elosztott adatbázis vázlatát! Keressünk ehhez információkat az interneten!

4•. Készítsünk beszámolókat az alábbi témákban:
a) integrált adatbázis-kezelő rendszerek a mobil hálózatokban;
b) objektum-orientált adatbázis-kezelő rendszerek létrehozása.

3.4. Microsoft Access 2007 adatbázis-kezelő rendszer

1. Jellemezzük az adatbázisok létrehozásának lépéseit!
2. Ismertessük a relációs adatbázis alapelemeit!
3. Mi az adatbázis-kezelő rendszer? Mi a rendeltetése?
4. Hogyan hozunk létre új **Word** dokumentumot? Új **Excel** táblázatot?

Az előző pontban részletesen megvizsgáltuk az adatbázisok létrehozásának első két lépését: a feladat kitűzését, az egyed-kapcsolat modell összeállítását és az adatmodell kiválasztását. A következő lépéseket már egy adatbázis-kezelő rendszer használatát igényli. Ezért most megismerkedünk a **Microsoft Office Access 2007** (a továbbiakban **Access 2007,** *access* – hozzáférés) DBMS-el, ami a **Microsoft Office 2007** csomag része.

Az Access 2007 általános jellemzése

Az **Access 2007** egy fájlszerver architektúrájú relációs adatbázis-kezelő rendszer, amelyet személyes vagy kisvállalati használatra fejlesztettek ki, és viszonylag kis mennyiségű adat feldolgozására alkalmas.

Az **Access 2007** alapobjektumai a következők:

- **táblák,** amelyekben adatokat tárolunk, és amelyek segítségével leírjuk a táblák közötti kapcsolatokat;
- **űrlapok,** amelyek segítségével megkönnyíthetjük az adatbevitelt és az adatok áttekintését;
- **lekérdezések,** amelyek az adatok keresését, szűrését és feldolgozását teszik lehetővé egy speciális nyelv, az **SQL** (***Structured Query Language*** – strukturált adatlekérdező nyelv) segítségével;
- **jelentések,** amelyek az adathalmazok nyomtatását teszik lehetővé.

Ezekkel az objektumokkal, tulajdonságaikkal, alkalmazásokkal a későbbiek során elvégzendő gyakorlatok során ismerkedünk majd meg.

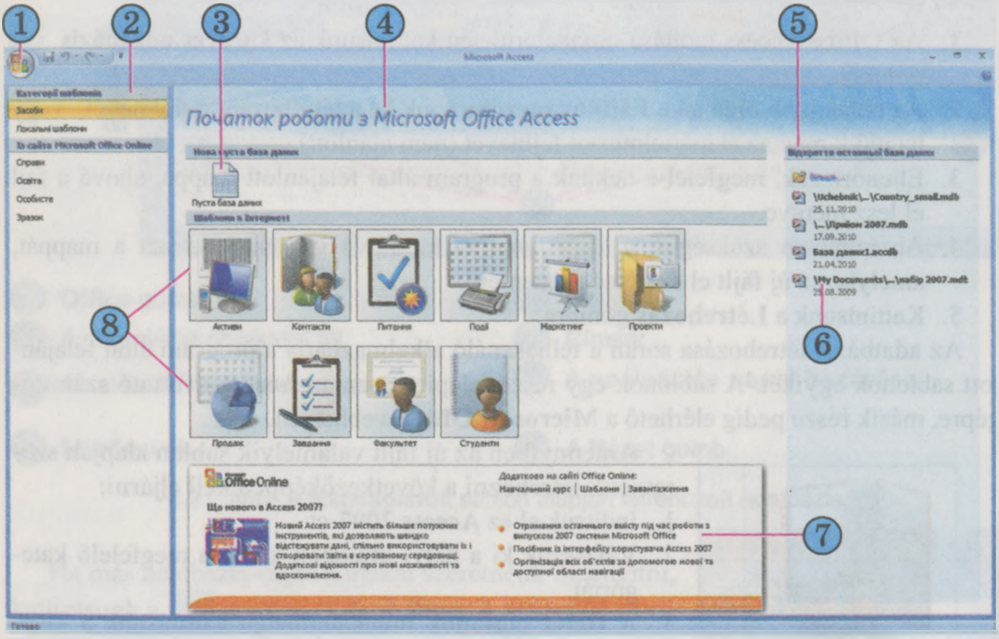
Az **Access 2007 az ACCDB** (*Access DataBase* – **Access** adatbázis) fájlformátumot használja, amely kompatibilis az Access korábbi verzióiban használt **MDB** (*Microsoft DataBase* – **Microsoft** adatbázis) formátummal.

Az Access 2007 indítása. A program ablaka

Az **Access 2007** a Windows operációs rendszerben megszokott módon indíthatjuk.

- ***Start* ⇒ *minden Program* ⇒ *Microsoft Office* ⇒ *Microsoft Office Access 2007;***
- a **Munkaasztal** parancsikonjára vagy a **Tálca** ikonjára kattintva;
- duplán kattintva egy adatbázisfájl nevén.

Az első két esetben megnyílik a program indítóablaka (3.18. ábra).



1. **Office gomb**
2. **A sablonok kategóriái**
3. **Új üres adatbázis** gomb
4. Az **Office Access** indítása terület
5. **Adatbázis megnyitása** munkaterület
6. A legutóbb megnyitott fájlok listája
7. **Microsoft Office Online** terület
8. Az elérhető sablonok gombjai

3.18. ábra. Az **Access 2007** indítóablaka

A címsorban az **Office** gomb mellett található **Gyorsgombok** ilyenkor inaktívak. Az ablak három részre van felosztva. Az ablak bal oldalán találjuk a **Sablonok kategóriáit,** amelyek egy része a **Microsoft Office Online** webhelyen érhető el.

Az ablak középső részén találjuk a kiválasztott kategóriába tartozó sablonokat. Alapértelmezés szerint ezek a 3.18. ábrán látható sablonok. A munkaterület három vízszintes sávra van felosztva. Legfelül az **Új üres adtabázis gombot** találjuk. A középső részben találjuk a sablonok gombjait. A legalsó részben a **Microsoft Office Online** webhelyre mutató linkek helyezkednek el, amelyek segítségével sok hasznos információhoz juthatunk a program működésével kapcsolatban.

Az ablak jobb oldali részében a **Legutóbb megnyitott adatbázisok** listáját találjuk. A fájlok listája fölött található linkre kattintva elérhetjük a merevlemezen vagy hálózati meghajtón található adatbázisfájlok listáját.

Új adatbázisok létrehozása

A legtöbb alkalmazói szoftvertől eltérően az **Access 2007**-ben az új adatbázissal csak annak létrehozása után kezdhetjük el a munkát.

Új adatbázist a következőképpen hozunk létre:

1. Az **Office Access** indítása munkaterületen kattintsunk az **Új üres adatbázis** gombra.
2. Az ablak jobb oldalán a **Fájlnév mezőbe** írjuk be a létrehozandó fájl nevét, vagy hagyjuk meg az alapértelmezett fájlnevet (nem ajánlott).
3. Ellenőrizzük, megfelel-e nekünk a program által felajánlott mappa, ahová a fájl el lesz mentve.
4. Amennyiben szükséges, a gombra kattintva válasszuk ki azt a mappát, amelybe az új fájlt el szeretnénk menteni.
5. Kattintsunk a **Létrehozás** gombra.

Az adatbázis létrehozása során a felhasználó alkalmazhatja a program által felajánlott sablonok egyikét. A sablonok egy része telepítve van az **Access**-t futtató számítógépre, másik része pedig elérhető a **Microsoft Office** webhelyén.

3.19. ábra. **Kapcsolatok** mező

Amennyiben az új fájlt valamelyik sablon alapján szeretnénk létrehozni a következőképpen kell eljárni:

1. Indítsuk el az **Access 2007**-et.
2. Válasszuk ki a helyi sablonok közül a megfelelő kategóriát.
3. A **Helyi sablonok** munkaterületen kattintsunk a kiválasztott sablonra.
4. Az ablak jobb oldalán a beviteli mezőbe írjuk be a létrehozandó fájl nevét (3.19. ábra), vagy hagyjuk meg az alapértelmezett fájlnevet (nem ajánlott).
5. Szükség esetén válasszuk ki a mentés helyét.
6. Kattintsunk a **Létrehozás** gombra.

A program létrehozza a megadott nevű fájlt a kiválasztott helyen és a megfelelő sablon szerint.

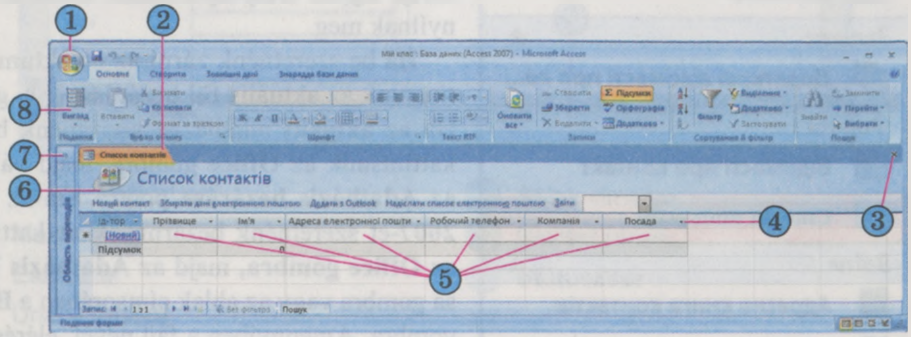
Amikor megnyitjuk a létrehozott fájlt, megjelenik egy információs sáv (3.20. ábra), amely arra figyelmeztet, hogy bizonyos tartalmak le vannak tiltva. Amennyiben szüksé-

ges, kattintsunk a **Beállítások gombra** és engedélyezzük ezeket. Az aktív tartalmak egyébként a makróvírusok működésének kizárása céljából vannak blokkolva.

3.20. ábra. A makrók letiltása információs sáv

A kapcsolatok sablon alapján létrehozott adatbázis segítségével ismerkedjünk meg a program felhasználói felületével és az alapvető műveleteivel.

Ha az adatbázist a **Kapcsolatok** sablon alapján volt létrehozva, megnyitása után a kapcsolatok listáját (3.21. ábra) láthatjuk a képernyőn, amelybe azokról a személyekről írhatunk be adatokat, akikkel baráti vagy hivatalos kapcsolatban vagyunk.

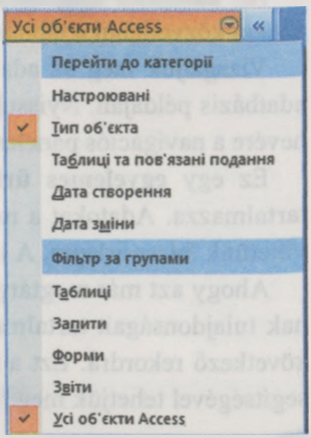


1. **Office** gomb
2. A megnyitott objektum fül
3. Az aktuális fül bezárása gomb
4. Mezőnevek
5. Mezők
6. Címsor
7. A navigációs panel bezárása
8. A **Nézet** gomb

3.21. ábra. A **Kapcsolatok** sablon alapján létrehozott adatbázis

Ha más adatbázis-objektumokat szeretnénk megnyitni, kattintsunk a » gombra (3.21. 7 ábra). A megnyíló navigációs panel címsorában kattintsunk a ▾ gombra. A megnyíló lista (3.22. ábra) két részből áll. Az egyikben kiválaszthatjuk a kategóriák egyikét, a másikban pedig a szűrőt, amelyek segítségével meghatározhatjuk, hogy a választott kategória mely objektumainak körét szűkíthetjük: *Minden Access objektum, Táblák, Űrlapok, Lekérdezések* és *Jelentések*.

Ha az **Objektum típusa** lehetőségre kattintunk, akkor láthatjuk az adatbázis objektumainak listáját, amelyek a **Táblák, Űrlapok, Lekérdezések** és **Jelentések** csoportokba vannak foglalva (3.23. ábra).



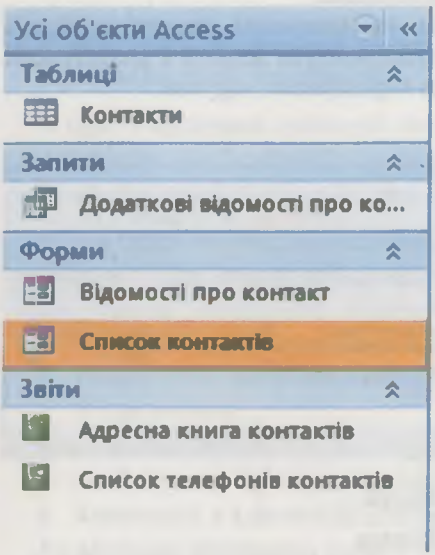
3.22. ábra. Az adatbázis objektumai

3.21. ábra. Az adatbázis objektumainak listája **Objektum típusa** párbeszédablakban.

Minden objektumnak saját ikonja van:

– táblák; – lekérdezések;

– űrlapok; – jelentések.



3.23. ábra. A navigációs panel

Ha meg szeretnénk nyitni egy objektumot, kattintsunk duplán a nevére. Az objektum megnyílik a képernyőn és az ablak középső részén megnyílik a hozzá tartozó fül (3.21. 2 ábra). Egyes űrlapobjektumok külön ablakban nyílnak meg.

Ha be szeretnénk zárni az objektumot, kattintsunk az aktuális fül bezárása gombra. Ha az aktuális adatbázist szeretnénk bezárni, kattintsunk az **Office** gombra, majd válasszuk az **Adatbázis bezárása** pontra. Ha az **Access 2007**-et szeretnénk bezárni, akkor kattintsunk az **Office gombra,** majd az **Adatbázis** bezárása gombra vagy az ablak címsorában a **Bezárás** gombra. Amennyiben a fájl nevét, elérési útvonalát vagy formátumát nem szeretnénk megváltoztatni, az adatbázist bezárás előtt nem kell menteni, mivel minden elvégzett művelet után automatikus mentés történik.

Adatok feltöltése Access 2007-ben

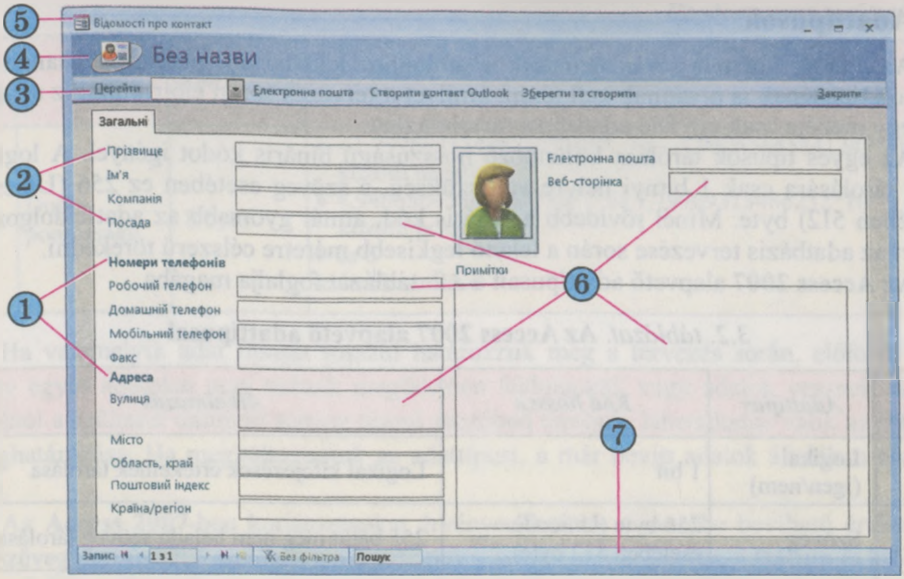
Az adatbázis-kezelőben az adatok feltöltését kétféleképpen végezzük: beírhatjuk azokat közvetlenül a táblába vagy alkalmazhatunk űrlapokat.

Vizsgáljuk meg az adatfeltöltés módjait a **Kapcsolatok** sablon alapján létrehozott adatbázis példáján. Nyissuk meg a **Kapcsolat részletei** űrlapot duplán kattintva annak nevére a navigációs panelen. Az űrlap külön ablakban nyílik meg (3.24. ábra).

Ez egy **egyelemes űrlap,** mivel csak egyetlen egyed-előfordulás információit tartalmazza. Adatokat a megfelelő mező kitöltésével, illetve a **Vágólapról** beillesztve vihetünk fel az űrlapra. A következő mezőre a **Tab** billentyű leütésével léphetünk.

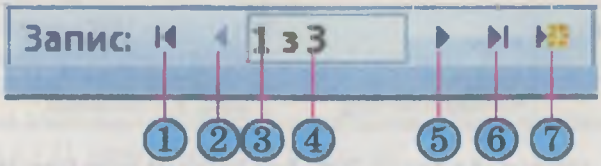
Ahogy azt már megtárgyaltuk, a tábla egy rekordja az egyed egyetlen előfordulásának tulajdonságait tartalmazza. Ha befejeztük egy rekord feltöltését, át kell lépjünk a következő rekordra. Ezt a **Státuszsor** bal oldalán található vezérlőelemek (3.25. ábra) segítségével tehetjük meg. A bevitt adatok automatikus el lesznek mentve az adatbázisba.

A bevitt adatok automatikus el lesznek mentve az adatbázisba.



1. Adatblokkok címei
2. Mezőnevek
3. Eszköztár
4. Űrlap címsora
5. Ablak címsora
6. Beviteli mező
7. **Státuszsor**

3.24. ábra. A **Kapcsolat részletei** űrlap



1. Első rekord
2. Előző rekord
3. Aktuális rekord sorszáma
4. Rekordok száma
5. Következő rekord
6. Utolsó rekord
7. Mentés

3.25. ábra. A rekordok közötti átmenet vezérlőelemei

Adattípusok

Az adatok bevitele során nemcsak a sablonhoz kell tartani magunkat, hanem az adatok típusának is olyannak kell lenni, amilyet a tervezés során előírtunk. Az adatbázis egy mezője csak egyféle adattípust tartalmazhat.

Az egyes típusok tárolása különböző hosszúságú bináris kódot igényel. A logikai adat tárolására csak 1 bitnyi helyre van szükség, a szöveg esetében ez 256 (Unicode esetében 512) byte. Minél rövidebb a bináris kód, annál gyorsabb az adatfeldolgozás. Ezért az adatbázis tervezése során a lehető legkisebb méretre célszerű törekedni.

Az **Access 2007** alapvető adattípusait a 3.2. táblázat foglalja magába.

3.2. táblázat. **Az Access 2007 alapvető adattípusai**

Sorszám	*Adattípus*	*Kód hossza*	*Alkalmazás*
1	Logikai (igen/nem)	1 bit	Logikai kifejezések értékeinek tárolása
2	Szöveg	256 byte (Unicode esetében 512 byte)	255 betűt meg nem haladó szöveg tárolása
3	Dátum és idő	8 byte	Az évszám 100 és 9999 közötti, az idő 0:00:00 és 23:59:59 közötti lehet
4	Megjegyzés	2,14 Gbyte-ig	A megadott méretet meg nem haladó szöveg tárolása
5	**OLE** objektum	2,14Gbyte-ig	Az adott méretet meg nem haladó tetszőleges típusú objektum, például képek, videók tárolása
6	Hiper-hivatkozás	1 Gbyte-ig	Weblapokra, helyi és hálózati fájlokra mutató hivatkozások tárolása
7	Automatikus sorszám	4 byte	A program által generált 0 és 4 294 967 295 közötti szám, általában elsődleges kulcs
8	Szám	Formátumtól függ	Lásd. 3.3. táblázatot

3.3. táblázat. **Az Access 2007-ben használt számformátumok**

Sorszám	*Méret*	*Kód hossza*	*Alkalmazás*
1	Byte	1 byte	0 és 255 közötti számok
2	Egész szám	2 byte	– 32 768 és 32 767 közötti számok
3	Hosszú egész szám	4 byte	–2 147 483 648 és 2 147 483 647 közötti számok
4	Valós szám	4 byte	• $-3{,}402823*10^{38}$ és $-1{,}401298*10^{-45}$ közötti negatív; • $1{,}401298*10^{-45}$ és $3{,}402823*10^{38}$ közötti pozitív; • és 0 számok tárolása

Продовження таблиці 3.3

Sor-szám	*Méret*	*Kód hossza*	*Alkalmazás*
5	Dupla pontosságú	8 byte	• - 1,79769313486232*10^{308} és – 4,94065645841247*10^{-324} közötti negatív; • 4,94065645841247*10^{-324} és 1,79769313486232*10^{308} közötti pozitív • és 0 számok tárolása

Ha valamelyik adat típusát rosszul határozzuk meg a tervezés során, előfordulhat, hogy egyes adatokat nem tudunk megfelelően feldolgozni, vagy adatok vesznek el. Ha például a születési dátumot szöveg típusú mezőben tároljuk, lehetetlenné válik az életkor meghatározása. Ha megváltoztatjuk az adattípust, a már bevitt adatok általában elvesznek.

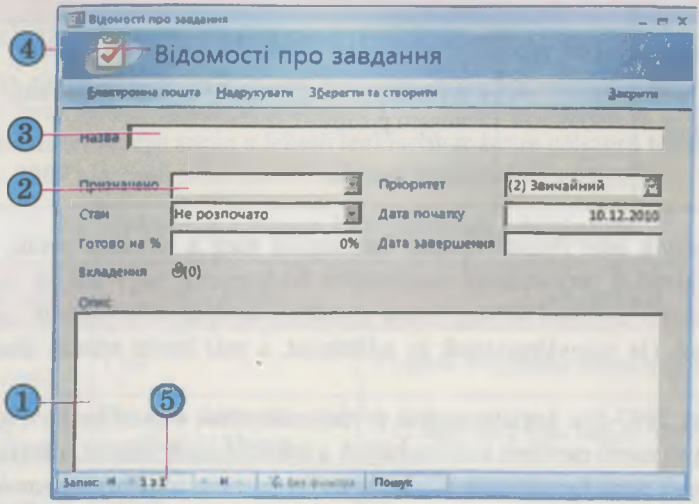
Az **Access 2007**-ben korlátozásokat érvényesíthetünk a mezőbe bevihető értékekre. A **Szöveg** típusú mező esetében korlátozhatjuk a szimbólumok számát, a **Dátum és idő** esetében előírhatjuk azok formátumát. Ezeket a korlátozásokat **adatformátumoknak** nevezik. Az adatformátumok sokban hasonlítanak az **Excel**-ben megismert adatformátumokra.

Ellenőrző kérdések

1°. Milyen típusú adatbázis-kezelő az **Access 2007**?
2•. Mi az **Access 2007** rendeltetése?
3°. Milyen fájlformátumot használ az **Access 2007**?
4°. Hogyan indíthatjuk el az **Access 2007**-et?
5•. Írjuk le az **Access 2007** indulóablakát!
6°. Ismertessük az üres adatbázis létrehozásának folyamatát!
7•. Hogyan hozhatunk létre új adatbázist internetes sablon alapján?
8°. Miért vannak blokkolva az adatbázisok egyes elemei? Hogy engedélyezhetjük ezeket?
9•. Ismertessük a 3.21. ábrán látható ablak szerkezetét!
10•. Mire szolgál a **Navigációs panel**?
11°. Hogyan nyithatjuk meg az adatbázist **Navigációs** panel segítségével?
12°. Hogyan zárjuk be az adatbázist? Az **Access 2007**-et?
13•. Hogyan töltjük fel az adatbázist adatokkal? Mire kell a feltöltés során feltétlenül odafigyelni?
14•. Milyen adattípusokat használ az **Access 2007**?
15*. Miben különbözik a **Szöveg** adattípus a **Megjegyzéstől**? Melyiknek mi a rendeltetése?
16*. Miben különbözik az egész **Szám** adattípus a **Valóstól**? Mi ezeknek az adattípusoknak a rendeltetése?
17•. Mire használják az **OLE** adattípust? Milyen adatokat tárolhatunk az egész típusú mezőben?
18*. Miért van szükség a különböző adattípusok alkalmazására? Lehetséges volna-e valamilyen univerzális adattípus alkalmazása?
19•. Mekkora helyet foglalnak el az **Access**-ben alkalmazott számtípusok?
20•. Mi az adatformátum? Milyen célból alkalmazzák?

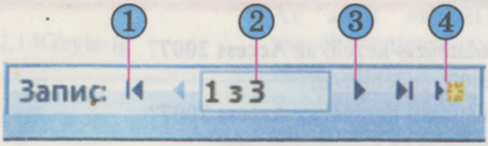
Végezzük el a feladatokat!

1°. Nevezzük meg a 3.26. ábrán látható objektumokat!



3.26. ábra

2°. Nevezzük meg a 3.27. ábrán látható objektumokat!



3.27. ábra

3°. Nyissuk meg a **зразок 3.4.1.accdb fájlt!**

1. Engedélyezzük a tiltott tartalmakat.
2. A **Navigációs panelen** végezzük el a következő beállításokat: **Kategória:** *változtatás dátuma,* **Szűrés:** *Minden dátum.*
3. Határozzuk meg, melyik objektumot szerkesztették legrégebben és melyiket legutóbb.
4. Változtassuk meg a **Navigációs panel** beállításait a következőre: **Kategória** – *Objektum típusa*; **Szűrés** – *Minden Access objektum.*
5. Határozzuk meg, hány objektumot tartalmaz az adatbázis!
6. Zárjuk be a **Navigációs panelt** és az adatbázist.

4°. Nyissuk meg a **зразок 3.4.4.accdb fájlt.**

1. Vizsgáljuk meg az adatbázis kezdőoldalát.
2. Engedélyezzük a tiltott tartalmakat.
3. Zárjuk be a kezdőoldalt a × gombbal.
4. Nyissunk meg egy tetszőleges megrendelő, például *Sergienko Mariya* adatait és kattintsunk a **Вхід** gombra.
5. Változtassuk meg a **Navigációs panel** beállításait a következőre: **Kategória** – *objektum típusa,* **Szűrés** – *minden Access objektum.*

6. Nyissuk meg a **Táblákat,** és határozzuk meg, hány táblából áll az adatbázis.
7. Nyissuk meg az egyik táblát, és nevezzük meg az ott található objektumokat.
8. Zárjuk be a táblát, majd az adatbázist.

5°. Nyissuk meg a **зразок 3.4.5.accdb fájlt!**
1. Változtassuk meg a **Navigációs panel** beállításait a következőre: **Kategória** – *objektum típusa;* **Szűrés** – *minden Access objektum.* Nyissuk meg az objektumokat és győződjünk meg arról, hogy azok nem tartalmaznak adatokat. Zárjuk be az objektumokat.
2. Nyissuk meg a **Kapcsolat részletei** űrlapot, és töltsük azt fel néhány osztálytársunk adatával. Ha valamelyik adatot nem ismerjük, a megfelelő mezőt hagyjuk kitöltetlenül. A **Компанія** mezőbe írjuk be az iskolánk nevét, a **Посада** mezőbe pedig hogy tanuló.
3. Zárjuk be az űrlapot, majd nyissuk meg az adatbázis más objektumait. Győződjünk meg arról, hogy azok tartalmazzák a bevitt adatokat.
4. Mentsük az adatbázist a **вправа 3.4.5. accdb.** mappánkba.

6°. A 3.26. ábra űrlapja alapján határozzuk meg a **Назва, Дата початку, Опис** mezők adattípusait!

3.5. Az egyed-kapcsolat modell leképezése az adatbázisra. Táblakezelés Access 2007-ben

1. Ismertessük az egyed-kapcsolat modellt az Iskolai könyvtár példáján!
2. Milyen lépésekből áll egy új adatbázis létrehozása?
3. Milyen adattípusokat használ az **Access 2007**? Miért van szükség különböző adattípusokra?
4. Hogyan hozunk létre üres adatbázist **Access 2007**-ben?
5. Ismertessük az űrlapok kitöltésének mechanizmusát!

Az egyed-kapcsolat modell leképezése az adatbázisra

A 3.3. pontban megismerkedtünk az adatbázisok létrehozásának lépéseivel, és megvizsgáltuk az első lépéseket a **Világ országai** adatbázis példáján. Meghatároztuk az adatbázis létrehozásának célját és a felhasználók körét, megalkottuk az adatbázis leíró modelljét és az egyed-kapcsolat modellt, eldöntöttük, hogy relációs adatmodellt fogunk alkalmazni.

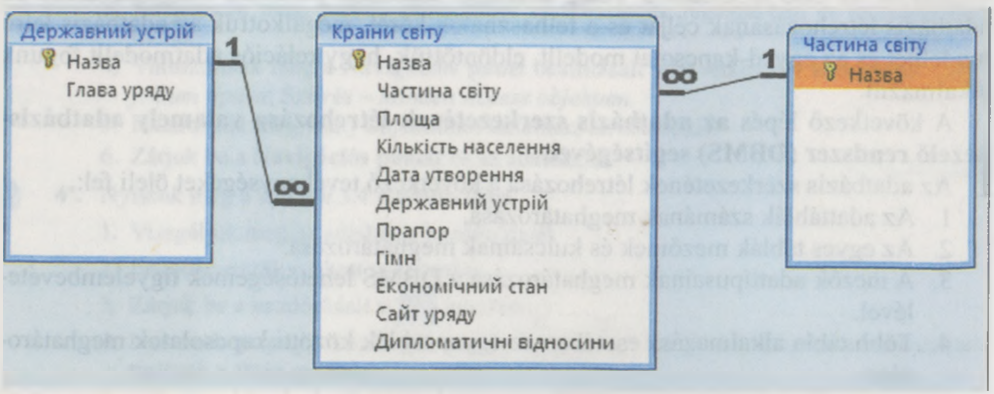
A következő lépés **az adatbázis szerkezetének létrehozása valamely adatbáziskezelő rendszer (DBMS) segítségével.**

Az adatbázis szerkezetének létrehozása a következő tevékenységeket öleli fel:
1. Az adattáblák számának meghatározása.
2. Az egyes táblák mezőinek és kulcsainak meghatározása.
3. A mezők adattípusainak meghatározása a DBMS lehetőségeinek figyelembevételével.
4. Több tábla alkalmazása esetében az egyes táblák közötti kapcsolatok meghatározása.
5. Az egyéb adatbázis-objektumok – űrlapok lekérdezések, jelentések – mennyiségének és típusainak meghatározása.

Az egyed-kapcsolat *modell leképezése alatt* az adatbázis szerkezetének meghatározását értjük az egyed-kapcsolat alapján.

Vizsgáljuk meg az egyed-kapcsolat modell egy lehetséges leképezését. A **Világ országai** adatbázis alapján, amit **Access 2007**-ben fogunk végrehajtani. Az adatbázis szerkezete a következő lesz:

1. A táblák száma megegyezik az egyed-kapcsolat modell (3.15. ábra) egyedtípusainak számával, tehát három: a **Világ országai,** az **Államforma** és a **Földrész.**
2. Az egyes táblák mezői megegyeznek az egyedtípusok tulajdonságainak számával. A világ országai tábla tartalmaz egy *Államforma* és egy *Földrész* mezőt a kapcsolatok biztosítása céljából. A táblák és mezők tehát a következők lesznek:
   - A **Világ országai** tábla: *Név, Világrész, Terület, Népesség, Létrehozás dátuma, Államforma, Zászló, Gazdasági helyzet, Országhonlap, Diplomáciai kapcsolatok;*
   - Az **Államforma** tábla: *Név, Kormányfő;*
   - A **Földrész** tábla: *Név.*
3. Az egyes táblák mezőinek adattípusai:
   - A **Világ országai** tábla:
     - **Név** – *szöveg, 70 szimbólum,* **kulcs;**
     - **Világrész,** *keresőmező* (más tábla adatainak felhasználását teszi lehetővé);
     - **Terület, szám** – *valós szám;*
     - **Lakosság, szám** – *hosszú egész;*
     - **Létrehozás dátuma** – *dátum és idő, rövid dátum;*
     - **Államforma** – *keresőmező;*
   - **Államforma** tábla:
     - **Név** – szöveg, 50 karakter, **kulcs;**
     - **Kormányfő** – *szöveg,* 30 karakter;
   - **Földrész** tábla:
     - **Név** – *szöveg,* 50 karakter, **kulcs.**
4. Kapcsolatok (3.28. ábra):



3.28. ábra. A **Világ országai** adatbázis kapcsolatai

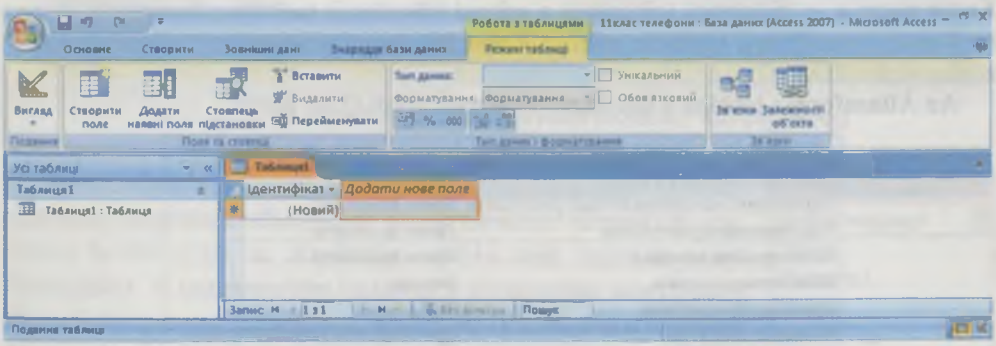
- Az **Államforma** tábla Név mezője és a **Világ országai** tábla **Államforma** mezője között, *egy a többhöz;*
- A **Földrész** tábla **Név** mezője és a **Világ** országai tábla **Földrész** mezője között *egy a többhöz* kapcsolat.

5. Egyéb adatbázis-objektumok:
   - Űrlapok:
     - A **világ országai** űrlap: *Név, Világrész, Terület, Népesség, Létrehozás dátuma, Államforma, Zászló, Gazdasági helyzet, Országhonlap, Diplomáciai kapcsolatok* mezőkkel;
   - Lekérdezések:
     - A **világ országainak** területe lekérdezés: *az Ország neve, Földrész, Terület (földrészenkénti összegzés lehetőségével);*
     - A **világ országainak** államformái lekérdezés: *az Ország neve és Államforma* mezőkkel;
   - Lekérdezések:
     - A **világ országainak** területe és lakossága jelentés: az *Ország neve, Földrész, Terület, Lakosság mezőkkel földrészenként csoportosítva.*

Tábla létrehozása adatlap nézetben

Lássunk hozzá a táblák létrehozásához. E célból hozzunk létre egy üres adatbázist (3.29. ábra), az előző pontban leírtak szerint. Az adattáblák létrehozásának két módja van:

1. Az adatok beírásával **Adattábla nézetben;**
2. A **Tervező** nézet alkalmazásával.



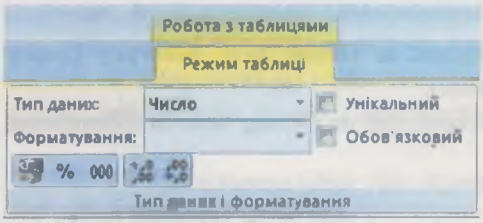
3.29. ábra. Az **Access 2007** ablaka a megnyitott új táblával

Vizsgáljuk meg az első módszert. Ennek különlegessége az, hogy nem szükséges közvetlenül megadni az adattípusokat, mert az **Access 2007** az első bevitt adat alapján automatikusan megpróbálja meghatározni azokat, és alkalmazza az alapértelmezett formátumokat. Csak létre kell hozni a mezőket a megfelelő nevekkel. Az első mező az **Azonosító,** amelyet az adatbázis-kezelő a tábla létrehozása során automatikusan létrehoz. Ez a mező **Számláló** típusú. A mező alapértelmezés szerint kulcs is.

A többi mezőt a következőképpen hozzuk létre:

1. Kattintsunk duplán az **Új mező hozzáadása** címre.
2. Írjuk be a mező nevét vagy szúrjuk be azt a **Vágólapról.**
3. Üssünk le az **Enter** vagy a → gombot.
4. A 2. és 3. pontokat annyiszor ismételjük, ahány új mezőt létre szeretnénk hozni.
5. Fejezzük be a mezők létrehozását a mezőnevek alatti első sorra lépve vagy a ↓ gomb leütésével.

A mezőnevek nem tartalmazhatnak pontokat, felkiáltójeleket, szögletes zárójeleket és rejtett szimbólumokat (például bekezdés-végjeleket).



3.30. ábra. Az **Adattípusok és formátumok** vezérlőelemek

Az adattípus és formátum közvetlen megadására a tábla eszközök ideiglenes **Fül adattípus** és **Formázás** eszköztárát használhatjuk (3.30. ábra). A beállítás módja a következő:

1. Tegyük aktuálissá a mezőt.
2. Válasszuk ki a megfelelő **Adattípust** (3.30. ábra), az **Államforma** tábla **Név** mezője esetében például a *Szöveget.*
3. A **Formátum** mezőben válasszuk ki a megfelelő formátumot (csak a *Szám, Dátum és idő, Pénznem és Igen/nem* adattípusok esetében elérhető).

A **Kötelező** jelölőt bekapcsolva előírhatjuk, hogy az adott mezőt kötelező legyen kitölteni. Az **Egyedi érték** jelölő bekapcsolásával megkövetelhetjük, hogy a mezőbe beírt értékek ne ismétlődhessenek. A csoport többi gombjának használata megegyezik az **Excel 2007**-ben megszokottal.

A következő lépés az ***adatok feltöltése.*** Helyezzük a kurzort a megfelelő mezőbe, majd a billentyűzet vagy a **Vágólap** segítségével vigyük be a megfelelő értéket. Az **Enter** vagy a → billentyű leütésével a rekord következő mezőjére léphetünk. A rekord utolsó mezőjének kitöltése után az **Enter** leütésével a következő rekord első mezőjére léphetünk.

Az **Államforma** táblába a következő értékeket írjuk (3.31. ábra):

Назва	Глава уряду
Президентська республіка	Президент
Парламентсько-президентська республіка	Прем'єр-міністр
Парламентська республіка	Прем'єр-міністр
Конституційна монархія	Прем'єр-міністр
Абсолютна монархія	Монарх
Однопартійна система	Глава партії
Військова диктатура	Диктатор (Військовий керівник)

3.31. ábra. Az **Államforma** tábla

Hasonlóképpen hozzuk létre a **Világrész** táblát a **Név** mezővel, amelynek tartalma *Ázsia, Európa, Amerika, Afrika, Ausztrália és Óceánia, Antarktisz.*

A táblák mentését a következőképpen végezzük:

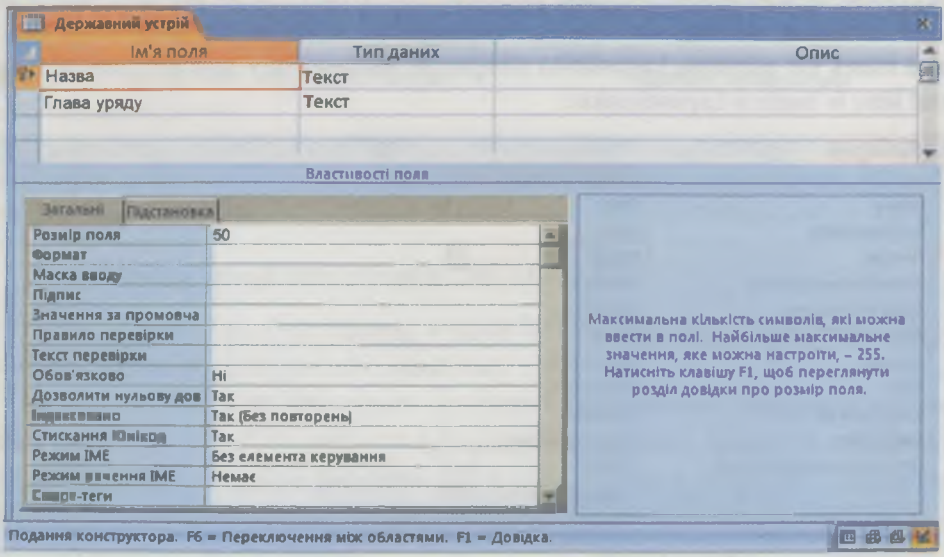
1. Kattintsunk a gyorspanel **Mentés** gombjára.
2. A **Mentés** ablak **Tábla** neve mezőbe írjuk be a tábla nevét (például **Államforma**).
3. Kattintsunk az **Ok** gombra.

Tábla létrehozása tervező nézetben

Összetettebb szerkezetű táblák létrehozásakor, vagy ha valamelyik mező tulajdonságait pontosan be kell állítani, a táblák létrehozásának másik említett módját alkalmazzuk, a **Tervező** nézetet. E célból kattintsunk a ***Létrehozás*** menüpontra, majd pedig válasszuk a ***Tábla Tervező nézetben*** lehetőséget.

Amennyiben a tábla már létezik, a **Nézet** legördülő menüből válasszuk a **Tervező nézet** lehetőséget. Ekkor az **Adattábla** nézetből átváltunk **Tervező** nézetbe.

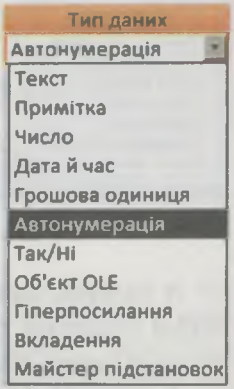
Mindkét esetben megnyílik a tábla **Tervező** nézete (3.32. ábra). Az ablak három oszlopból áll: a **Mező nevéből,** az **Adattípusból** és a **Leírásból.** A bal oldali oszlopba írjuk be a mező nevét. Az adattípus megadása céljából kattintsunk a második oszlopra, majd a legördülő menüből (3.33. ábra) válasszuk ki a megfelelő adattípust.



3.32. ábra Az **Államforma** tábla **Tervező** nézetben **Конструктор**

Az adattípus kiválasztása után a mezőtulajdonságok munkaterületen megváltoztathatjuk az alapértelmezett formátumokat, illetve beállíthatjuk az alapértelmezésben nem használt tulajdonságokat. A munkaterület tartalma a kiválasztott adattípustól függ. Szöveges mező esetében megadhatjuk a **Mezőszélességet,** azaz a mezőbe beírható szimbólumok maximális megengedett számát. **Szám** típusú mező esetében kiválaszthatjuk a fentebb tárgyalt számformátumok egyikét. Ezenfelül beállíthatunk pénznem, fixpontos, százalék, vagy tudományos formátumot és megadhatjuk a tizedesjegyek számát.

A **Dátum és idő,** az **Igen/nem, Pénznem** adattípusok esetében a választható formátumok sok közös jegyet mutatnak az **Excel 2007**-nél megismert formátumokkal. Az **OLE objektumok** estében csak felirat adható meg.



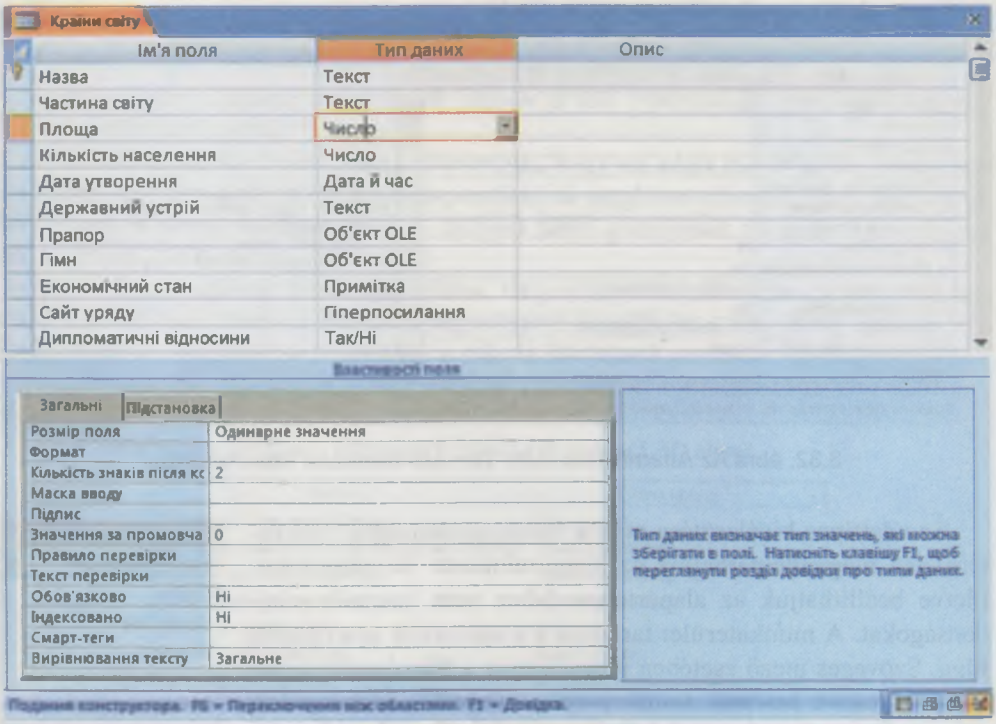
3.33. ábra. Az **Adattípusok** listája

Az egyéb, értékekre vonatkozó korlátozások közül nagyon fontos a kötelező kitöltés. Ha egy mező esetében ezt szeretnénk beállítani, a **Mezőtulajdonságok** területen az **Általános** fülön kapcsoljuk be a **Kötelező** tulajdonságot. Ha ezt megtesszük, a feltöltés során mindaddig nem történik meg az adatok mentése, amíg a mezőt ki nem töltöttük. A kulcs mezők esetében ez a beállítás kötelező.

Ha egy mezőt kulcsnak szeretnénk kijelölni, tegyük azt aktuálissá, majd kattintsunk az eszköztár **Kulcs** eszközére. Ugyanilyen módon lehet a műveletet visszavonni.

Folytassuk a **Világ országai** tábla létrehozását. Hozzuk létre és mentsük a megadott néven a táblát. Hozzuk létre a mezőket, és állítsuk be a korábban meghatározott adattípusokat és formátumokat (3.34. ábra).

Az **Államforma** és **Földrész** mezők esetében a mezők adattípusa megadása a táblák közötti kapcsolatok létrehozását eredményezi. A fenti mezők esetében tehát a **Keresés varázslót** választjuk az adattípusok listájából, amelynek segítségével kiválasztjuk a másik két adattábla kapcsolódó mezőjét. A varázsló (egy speciális program) egyúttal létre is hozza a kapcsolatokat.



3.34. ábra. A **Világ országai**

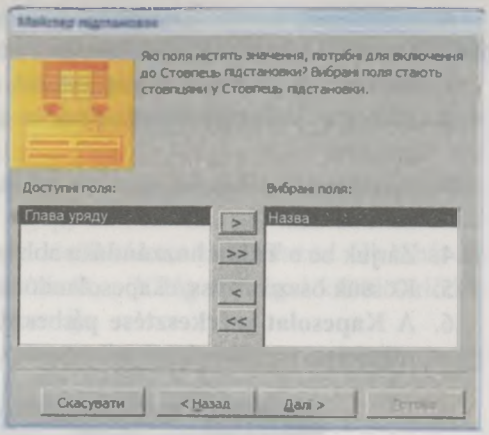
A **varázsló első lépésként** rákérdez, hogy a keresőoszlop elem segítségével a táblákból választunk ki adatokat, vagy egy kézzel szerkesztett listából. Most az első lehetőséggel fogunk élni.

A **második lépés** az adatforrás kiválasztása a már létrehozott két tábla közül. Amennyiben léteznek egyéb objektumok is, azok közül is választhatunk. Az **Állam-**

forma mező estében az **Államforma** táblát, a **Földrész** mező esetében a **Földrész** táblát választjuk.

A **harmadik lépésben** (3.35. ábra) ki kell választani a táblákból azt a mezőt vagy mezőket, amelyek a szükséges adatokat tartalmazzák. Ez most mindkét tábla esetében a név mező lesz. A mező kiválasztása céljából kattintsunk a mező nevére, majd a > gombra.

A további lépésekben a varázslóval beállíthatjuk az adatok rendezését és a mező szélességét. Az utolsó lépésben az értékek többszörös választását engedélyezhetjük a jelölő bekapcsolásával.



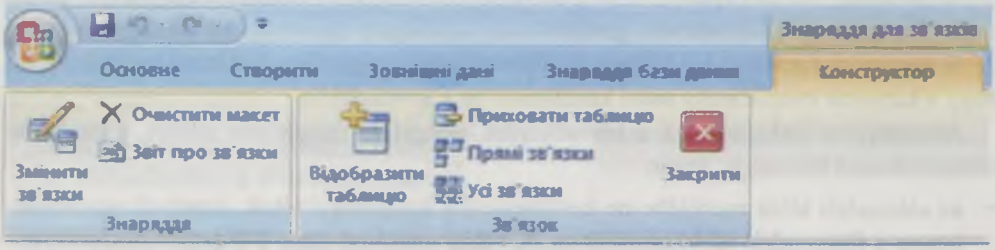
3.35. ábra. A mezők kiválasztása a **Keresés varázslóval**

Ennek utána az **Államforma** és a **Földrész** mezők értékeit a lista elemei közül választhatjuk ki. A lista kibővítése céljából fel kell vennünk az új elemeket az **Államforma** és a **Földrész** táblába.

Kapcsolatok kezelése az Access 2007-ben

Éppen az imént ismerkedtünk meg a táblák közötti kapcsolatok létrehozásának egy módjával, a **Keresés varázsló** alkalmazásával.

Ha meg szeretnénk tekinteni a már létrehozott kapcsolatokat, kattintsunk az **Adatbázis eszközök** menüpontra, majd ott a **Kapcsolatok** eszközre. Megnyílik a **Kapcsolatok** ablak, abban a **Minden kapcsolat** gombra kattintva megnyílik a **Kapcsolatszerkesztő** ideiglenes eszköztár (3.36. ábra) és a táblák közötti kapcsolatokat (3.28. ábra) tartalmazó munkaterület.

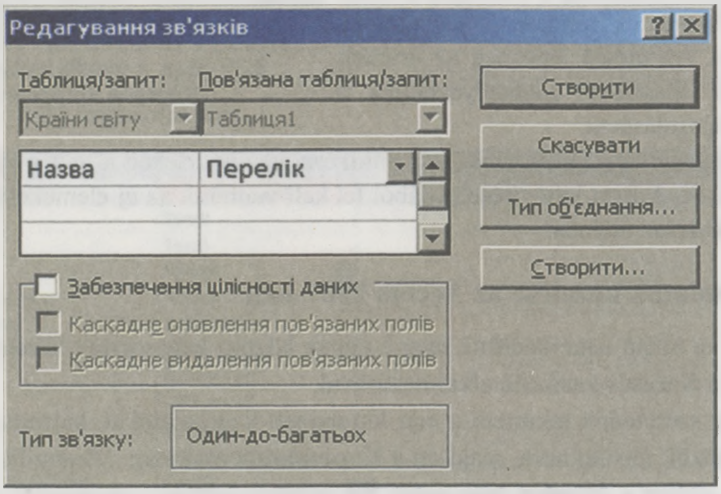


3.36. ábra. A **Kapcsolatszerkesztő** ideiglenes eszköztár

Az **Access 2007**-ben a kapcsolatoknak két típusa létezik az *egy az egyhez* és az *egy a többhöz* kapcsolat. Az *egy az egyhez* kapcsolat olyan mezők között jöhet létre, amelyek mindegyike **Indexelt,** és amelyeknél az ismétlés nem megengedett. Vagyis ha a mezők kulcs tulajdonsággal rendelkeznek. Az *egy a többhöz* kapcsolat olyan mezők között jöhet létre, amelyek egyike **Indexelt** és ismétlés nélküli, a másik viszont megengedi az ismétléseket.

A kapcsolatok létrehozását és szerkesztését a legtöbb esetben a **Kapcsolatok** ablakban végezzük. Új kapcsolat létrehozása a következőképpen történik.

1. Nyissuk meg a ***Kapcsolatok*** ablakot, és kattintsunk a ***Tábla hozzáadása*** gombra.
2. A megnyíló ablakban válasszuk ki a szükséges táblát, és kattintsuk a **Hozzáad** gombra.
3. Ismételjük a tábla hozzáadását mindaddig, amíg minden szükséges táblát hozzá nem adtunk.
4. Zárjuk be a **Tábla hozzáadása** ablakot.
5. Kössük össze az összekapcsolandó mezők neveit.
6. A **Kapcsolat szerkesztése** párbeszédablakban (3.37. ábra) pontosítsuk, hogy a megfelelő mezőket kötöttük-e össze.



3.37. ábra. A **Kapcsolat szerkesztése** párbeszédablak

7. Állítsuk be a kapcsolat típusát.
8. Szükség esetén kapcsoljuk be a **Hivatkozási** integritás megőrzése jelölőt.
9. Kattintsunk a **Létrehozás** gombra.

A kapcsolat szerkesztése közben az **Illesztés típusa** gombra kattintva beállíthatjuk, hogy a kapcsolt mezők mely sorai kerüljenek be a kapcsolatba.

Amennyiben bekapcsoljuk a **Hivatkozási integritás** megőrzése jelölőt, a program automatikusan biztosítja, hogy:

- az alárendelt tábla mezőjébe ne kerülhessenek be olyan értékek, amelyek nincsenek benne a fölérendelt tábla mezőjében. A **Világ országai** tábla **Földrész** mezőjébe tehát nem kerülhet bele olyan érték, amelyik nem szerepel a **Földrész** táblában;
- a fölérendelt táblában nem szerkeszthetők és nem törölhetők olyan adatok, amelyek előfordulnak az alárendelt táblában. A **Földrész** táblából tehát mindaddig nem törölhető **Amerika**, amíg a **Világ országaiban** akárcsak egyetlen, az amerikai kontinensen fekvő ország szerepel;
- a fölérendelt tábla mezője kulcs tulajdonságú legyen;
- a két tábla adattípusa megegyezzen.

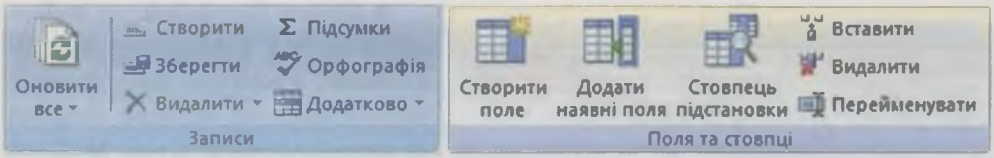
Ha azt szeretnénk, hogy a fölérendelt táblában eszközölt változtatások automatikusan bekerüljenek az alárendelt táblába, kapcsoljuk be a **Kapcsolt mezők kaszkádolt frissítése** jelölőt (3.37. ábra). Ez esetben, ha a fölérendelt táblában a **Katonai diktatúra** bejegyzést **Diktatúrára** változtatjuk, a **Világ országai** táblában a **Katonai diktatúra** összes előfordulása **Diktatúrára** változik majd.

Ha a **Kapcsolt mezők** kaszkádolt törlése jelölőt is bekapcsoljuk, akkor ha a későbbiek során a fölérendelt táblából törlünk egy bejegyzést, az alárendelt táblából automatikusan törlődik minden olyan rekord, amely a törölt elemet tartalmazza. Ha tehát az **Államformák** közül töröljük a **Katonai diktatúrát,** a **Világ országai** közül törlődik minden olyan ország, aminek ez volt az államformája.

Ha a jobb egérgombbal a kapcsolatot jelző vonalra kattintunk, a megnyíló helyi menü megfelelő nevű pontjára kattintva törölhetjük a kapcsolatot.

Táblák szerkesztése és formázása

Az adatbázis használata közben előfordul, hogy a táblákon változtatásokat kell végrehajtani. Az alapvető szerkesztési műveletek közé tartoznak a mező beillesztése, törlése, áthelyezése és átnevezése, rekordok beillesztése és törlése, kulcsmező cseréje. A műveleteket a **Kezdőlap Rekordok** eszköztárával vagy az **Adatlap** nézet **Mezők és Oszlopok** eszköztárával végezzük (3.38. ábra).



3.38. ábra. A **Rekordok** és a **Mezők és oszlopok** eszköztár

Ha be szeretnénk illeszteni egy új oszlopot a táblába, nyissuk meg a táblát adatlap nézetben, tegyük aktuálissá azt az oszlopot, amely elé az újat be szeretnénk szúrni, majd kattintsunk az eszköztár **Beszúrás** gombjára. Az új oszlopnak a már megismert módon nevet kell adnunk. Ha az eszköztár **Új mező** gombját alkalmazzuk, az új mező a mezőlista utolsó elemeként jön létre.

A mező törlése a következő módon történik:

1. Aktualizáljuk a törlendő oszlopot.
2. Kattintsunk az eszköztár **Törlés** gombjára.
3. Hagyjuk jóvá a mező és a benne található adatok törlését.

Ugyanez más módon:

1. Az oszlop nevére kattintva jelöljük ki az oszlopot.
2. Lépjünk a kezdőlapra, majd a **Rekordok** eszköztár **Törlés** eszközével töröljük a mezőt.
3. Hagyjuk jóvá a mező és a benne található adatok törlését.

Törölhetjük a mezőt a jobb egérgombbal aktiválható helyi menü *Törlés* menüpontja segítségével is.

Több mező törlése céljából jelöljük ki ezeket az oszlopokat, majd alkalmazzuk a fentebb leírt módszerek egyikét.

A mezők törlése során nem hagyhatjuk figyelmen kívül, hogy a táblának legalább egy mezőt tartalmaznia kell, továbbá a kulcs mező törlése ilyen módon nem lehetséges. Erre csak **Tervező** nézetben van lehetőség.

Új sor beszúrása céljából töltsük ki a tábla utolsó, üres sorát.

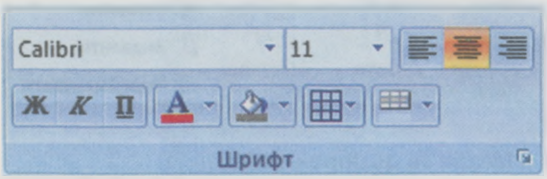
Figyeljük meg, hogy a tábla mindig tartalmaz egy utolsó, üres sort. Ahogy azt elkezdjük kitölteni, azonnal létrejön egy újabb üres sor.

Az új rekord beillesztése céljából használhatjuk a **Kezdőlap** rekordok eszköztárának **Új** gombját. Ezt a módszert leginkább nagy méretű adathalmazok esetében alkalmazzuk.

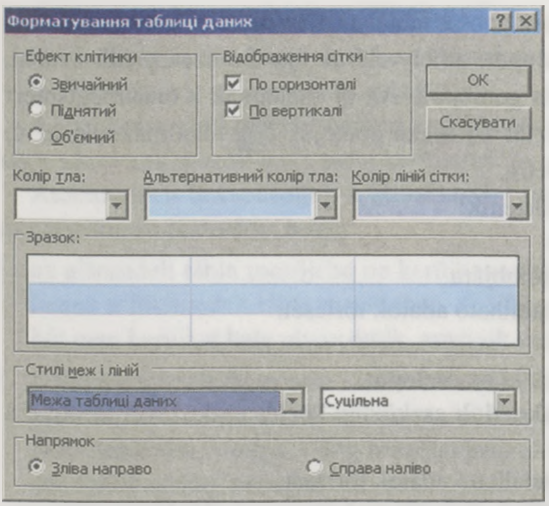
Ha egy rekordot törölni szeretnénk, tegyük azt aktuálissá, majd használjuk a kezdőlap ***Rekord*** eszköztárának ***Törlés*** eszközét.

A mező nevének megváltoztatása céljából kattintsunk duplán a mezőnévre, majd végezzük el a változtatást. A mező tartalmát ugyanúgy szerkeszthetjük, mint a szövegszerkesztőben. Ez alól kivételt képeznek a kapcsolt mezők, amelyek esetében csak a listán szereplő elemek közül választhatunk, valamint a logikai típusú mezők, amelyek esetében a szerkesztés a jelölő ki- és bekapcsolásából áll.

Ha át szeretnénk helyezni egy mezőt, jelöljük ki, majd a mezőnévnél fogva húzzuk azt át a megfelelő helyre.



3.39. ábra. A **Betűtípus** eszköztár



3.40. ábra. **Az adattábla formázása** ablak

A táblázat formázását: az oszlopszélesség és a sormagasság megváltoztatását, a betűtípus megváltoztatását, a kitöltés, az oszlopok elrejtését és megjelenítését az **Excel 2007**-ben megismert módon történik.

A betűtípust a kezdőlap **Betűtípus** eszköztára segítségével változtathatjuk meg (3.39. ábra).

A táblázat kialakításához további eszközöket biztosít az **Adatlap formázása** eszköztár (3.40. ábra). Ezt az ablakot a **Betűtípus** eszköztár jobb alsó sarkában található nyílra kattintva nyithatjuk meg. Itt lehetőség van a cellakiemelés beállítására (*szokásos, kiemelt, süllyesztett*) az írásirány megválasztására (*balról jobbra vagy jobbról balra*), a keretezés, illetve a címsor keretezés beállítására (*folytonos, szaggatott, pontozott, ritkított, pontvonal*).

Figyeljünk fel arra, hogy az egyes cellák külön formázása nem megengedett, a formázási beállítások a tábla egészére vonatkoznak

Ellenőrző kérdések

1•. Mit jelent az egyed-kapcsolat modell leképezése az adatbázisra? Hozzunk fel példákat!
2•. Milyen lépésekből áll az egyed-kapcsolat modell adatbázisra történő leképezése?
3°. Mi az adattáblák létrehozásának két legfontosabb módja?
4°. Melyik mező jön létre automatikusan a tábla létrehozásakor? Milyen ennek a mezőnek az adattípusa?
5•. Ismertessük a tábla létrehozását adatlap nézetben!
6°. Milyen megkötések vonatkoznak a mezőnevekben használható karakterekre?
7°. Hogy mentjük a táblát?
8•. Hogy változtathatjuk meg a tábla nézetét? Hogyan történik a tábla megnyitása **Tervező** nézetben?
9•. Ismertessük a **Tervező** nézet elemeinek rendeltetését!
10°. Hogyan hozunk létre új mezőt **Tervező** nézetben?
11•. Hogyan állítjuk be a mező adattípusát **Tervező** nézetben?
12•. Mit jelent, ha a mező **Kötelező** tulajdonsága **Igen** értékre van beállítva? Mire használjuk ezt a beállítást?
13°. Hogyan állítjuk be a mező kulcs tulajdonságát?
14°. Mit jelent a **Szöveg** típusú mező esetében a 25-ös **Mezőméret**?
15•. Mi a **Keresés varázsló**? Mire használjuk?
16•. Ismertessük a táblák közötti kapcsolatok felépítését a **Keresés varázsló** segítségével!
17*. Ismertessük a táblák közötti kapcsolatok felépítését a **Kapcsolatok** ablak alkalmazásával!
18°. Hogyan történik a táblák szerkesztése?
19°. Hogy történik a cellatartalom formázása a táblában? Ismertessük a legfontosabb szerkesztési műveletek elvégzését!

Végezzük el a feladatokat!

1°. Hozzuk létre **Access 2007**-ben a **Telefonkönyv** adatbázist a következő mezőkkel: családnév, keresztnév, típus (vezetékes, mobil), telefonszám! Töltsük fel az adatbázis osztályunk 5 tagjának adataival. Mentsük az adatbázist **3.5.1.accdb** néven!

2°. Hozzuk létre **Access 2007**-ben a **Világ országai** adatbázist a fentebb leírtak szerint! Töltsük fel az adatbázist 5 ország adataival! Mentsük az adatbázist **3.5.2.accdb** néven!

3•. Hozzuk létre a **Házi audiotéka** adatbázist a következő mezőkkel: lemez neve, előadó, kiadás éve, lejátszás ideje. Mentsük az adatbázist **3.5.3.accdb** néven!

4*. A 3.4. ábrán látható egyed-kapcsolat modell alapján hozzuk létre a **Raktározás** adatbázist. Határozzuk meg a táblák számát, a mezők listáját és adattípusait, az egyes táblák kulcsait és a táblák közötti kapcsolatokat! Induljunk ki abból a feltételezésből, hogy ötféle árucikk van és azokat két raktárban tároljuk. Az **Áru** egyedtípust bővítsük a **Tárolás helye** tulajdonsággal! A táblákat **Tervező** nézetben hozzuk létre. Töltsük fel a táblákat a tanár által biztosított adatokkal! Mentsük az adatbázist **3.5.4.accdb** néven!

12. számú gyakorlati munka Az adott alkalmazási terület egyed-kapcsolat modelljének létrehozása. Adatbázis létrehozása az adatbázis-kezelőben

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat.*

1. A feladat alapján dolgozzuk ki az egyed-kapcsolat modellt és az adatbázist.
   Feladat. Hozzuk létre a **Könyvtár** adatbázist, amely két egyedtípust tartalmaz. A **Tárolási egységet,** amelynek tulajdonságai a ***kód, a név, a szerző, a mennyiség, az ár, a kiadás éve és a címoldal fotója.*** A **Kategóriát,** amelynek egyetlen tulajdonsága a név (ennek lehetséges értékei a *könyv, a hangoskönyv, az újság és a folyóirat*). A két egyedtípus között kapcsolat áll fenn: a kategória **Név** tulajdonsága egy értékéhez a **Tárolási egység** több értéke tartozhat.
2. Hozzuk létre az adatbázist, határozzuk meg a mezők neveit és adattípusait!
3. Hozzuk létre a táblák közötti kapcsolatot a **Keresés varázsló** vagy a **Kapcsolatok** ablak segítségével!
4. Töltsük fel a táblákat a **таблиця 12.docx** fájlból vett adatokkal és a **Тема 3\Практична 12** mappából vett képekkel!
5. A **Tárolási egység** táblában állítsuk be a betűszínt *sötétzöldre,* a háttérszínt *halványsárgára,* az alternatív háttérszínt *világoskékre.*
6. Mentsük az adatbázist **gyakorlati_12.accdb** néven!

3.6. Űrlapok létrehozása, szerkesztése és formázása

1. Nevezzük meg a legfontosabb adatbázis-objektumokat és azok rendeltetését!
2. Ismertessük az adatbázis létrehozásának menetét!
3. Hogyan tölthetjük fel adatokkal a táblákat?
4. Milyen nézetei vannak a táblának? Milyen célt szolgálnak ezek?

Az űrlapok típusai

Az űrlap az adatbázis alapobjektumainak egyike. Hasznos felületet biztosít a felhasználónak az adattáblák feltöltése, vizsgálata és szerkesztése során. Az űrlapot egy vagy több tábla vagy lekérdezés alapján hozhatjuk létre és a felhasználó által szükségesnek ítélt adatokat tartalmazza majd. Az űrlap nemcsak mezőket tartalmazhat, hanem egyéb objektumokat is, például képeket, vezérlőelemeket, diagramokat, feliratokat is. A felhasználó által az űrlapba írt adatok azokban a táblákban tárolódnak majd, amelyek alapján az űrlapot létrehoztuk. Az űrlapon elhelyezhetünk számított mezőket is, amelyek segítségével számításokat végezhetünk a táblák adataival. Az **Access 2007**-ben néhány különböző űrlaptípus áll rendelkezésünkre:

- ***szokásos (oszlopos),*** amelyben a mezők oszlopba (oszlopokba) vannak rendezve. A képernyőn egyidejűleg egy rekord adatai láthatók (3.24. ábra);
- ***táblázatos,*** amely külsőleg nagyon hasonlít a **táblák** adattábla nézetére, de az űrlapnak nem kell tartalmaznia a tábla összes mezőjét és tartalmazhat számított mezőket is. A grafikus mezőket ez az űrlaptípus nem tartalmazza. A képernyőn egyidejűleg több rekord adatai jelennek meg;

- ***többelemes,*** amely annyiban különbözik az előbbi típustól, hogy van címsora, ahol szöveges adatokat, képeket és vezérlőelemeket is elhelyezhetünk. A képernyőn több rekord adatait láthatjuk (3.21. ábra);
- ***kiegyenlített,*** amely lényegében megegyezik a szokásossal azzal a különbséggel, hogy a feliratok a mezők fölött helyezkednek el;
- ***osztott,*** amely a szokásos és a táblázatos alakot egyidejűleg tartalmazza. Szerkesztési műveleteket az űrlapon és a táblázatban egyaránt végezhetünk (3.42. ábra);
- ***táblázatot tartalmazó űrlap,*** amely az előzőhöz hasonlóan egy űrlapot és egy táblázatot tartalmaz, de ezek ebben az esetben nem kapcsolódnak egymáshoz. A táblázat például a dolgozó adatait tartalmazza, az űrlap pedig a dolgozó feladatait;
- ***modális űrlap,*** amely nem tartalmaz adatokat, csak vezérlőelemeket és bizonyos tevékenységeket végezhetünk el a segítségével, például makrókat futtathatunk.

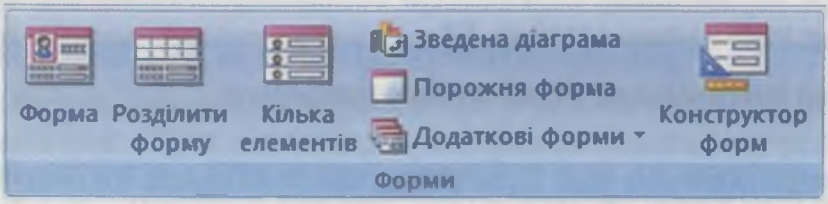
Ahogy a tábláknak, az űrlapoknak is többféle nézete van:

- **Űrlap nézet.**
- **Elrendezés nézet.**
- **Tervező nézet.**
- **Táblázat nézet.**

Az egyes nézetek között a **Nézet** eszköz segítségével válthatunk.

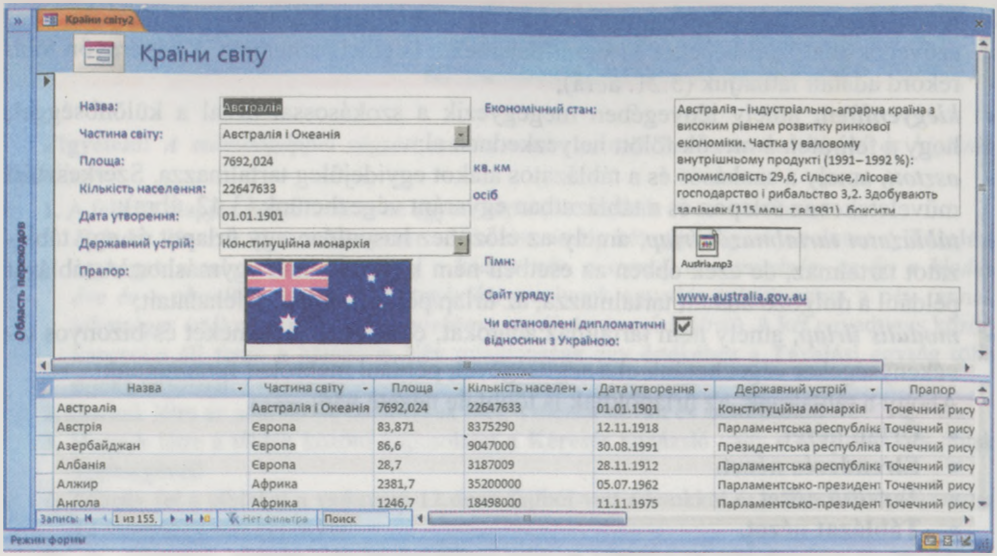
Űrlap létrehozása tábla alapján

Ahogy arról már beszéltünk, űrlapot csak létező tábla vagy lekérdezés alapján hozhatunk létre. Űrlapot a **Létrehozás** menü **Űrlap** eszköztárának (3.41. ábra) segítségével hozhatunk létre:



3.41. ábra. Az **Űrlap** eszköztár

- a gombra kattintva szokásos űrlapot;
- a gombra kattintva osztott űrlapot (3.42. ábra);
- a gombra kattintva többelemes űrlapot;
- a pedig egy üres űrlapot hozhatunk létre, amelyet az elemek áthúzásával szokásos űrlappá alakíthatunk.
- A **Több űrlap** eszköz alkalmazásával:
  - sokféle űrlapot hozhatunk létre az **Űrlap varázslóval;**
  - **táblázatos** űrlapot hozhatunk létre az azonos nevű gombra kattintva;
  - ***modális űrlapot*** hozhatunk létre a **Modális űrlap** gombra kattintva.
- Az **Űrlap tervező** gombra kattintva az **Űrlaptervezés** ideiglenes eszköztár segítségével tetszőleges űrlapot hozhatunk létre.



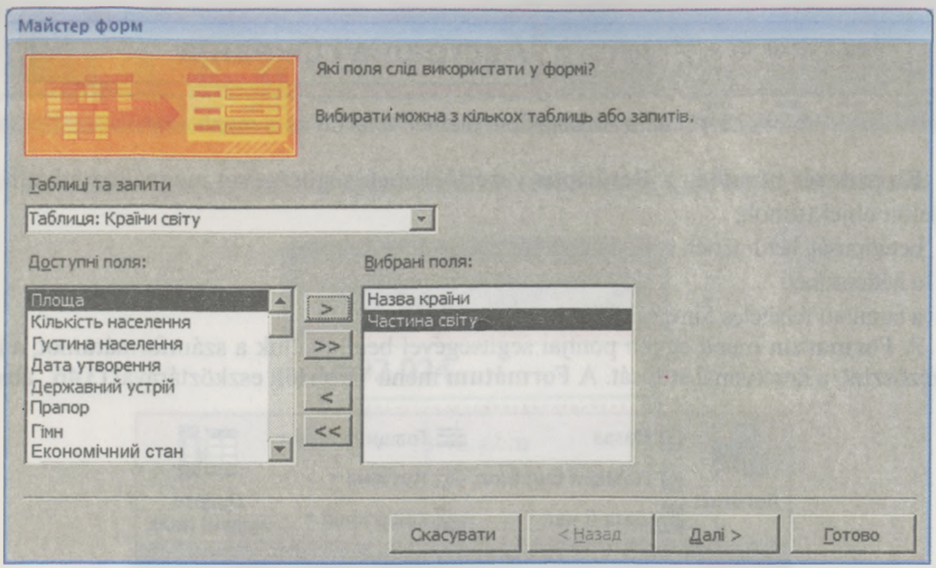
3.42. ábra. A **Világ országai** tábla alapján létrehozott osztott űrlap

Ha az **Űrlap** eszköztár eszközeivel **Szokásos, Osztott, Többelemes** űrlapot, vagy a **Több űrlap** eszközzel **Táblázatos** űrlapot hozunk létre, az vagy az aktuális tábla alapján, vagy a navigációs panelen kijelölt elem alapján fog létrejönni. Az űrlap a létrehozás után leginkább **Elrendezés** nézetben nyílik meg. E nézetben a felhasználó szerkesztheti az űrlapot, megváltoztathatja annak külalakját, a mezők tartalmát azonban nem változtathatja meg. Előbb a gyorspanel **Mentés** gombjára kattintva el kell menteni az űrlapot.

Űrlap létrehozása az Űrlap varázsló segítségével

Az űrlap létrehozásának fent ismertetett módja feltételezi, hogy az űrlap a tábla összes mezőjét tartalmazza majd. A mezők sorrendjét és szélességét is a program határozza meg. Az **Űrlap varázsló** ennél lényegesen több lehetőséget biztosít az űrlap kialakítására. A varázslót a ***Létrehozás*** menü ***Űrlapok*** eszköztárának ***További űrlapok*** eszközére kattintva indíthatjuk el. A további teendők a következők:

1. A **Táblák és lekérdezések** legördülő menüből (3.43. ábra) válasszuk ki a táblát vagy lekérdezést, amely alapján létre szeretnénk hozni az űrlapot.
2. Az **Elérhető** mezők listájából válasszuk ki a szükséges mezőt.
3. A > gombra kattintva helyezzük át a mezőt a **Kiválasztott mezők** listájába.
4. A 2. és 3. pontokat ismételve helyezzük át az összes szükséges mezőt.
5. Amennyiben más tábla vagy lekérdezés adataira is szükség van, válasszuk ki azt is és a már megismert módon helyezzük át a szükséges mezőket.
6. Az előző pontok ismétlésével gyűjtsük ki az összes szükséges mezőt.
7. Kattintsunk a **Tovább** gombra.
8. Válasszuk ki az űrlap típusát (*Szokásos, Táblázatos, Adattábla, Azonos szélességű*).
9. Kattintsunk a **Tovább** gombra.



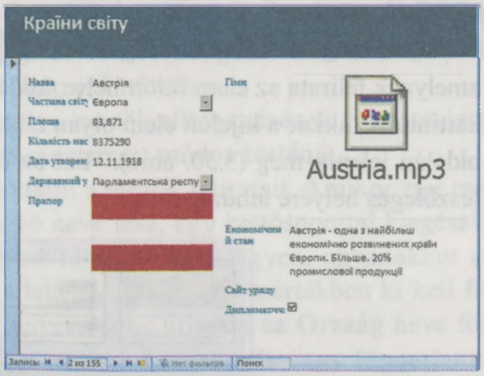
3.43. ábra. Az **Űrlap varázsló** ablaka

10. Válasszuk ki az űrlap stílusát.
11. Kattintsunk a **Tovább** gombra.
12. Adjunk nevet az űrlapnak.
13. Válasszuk ki, mi történjen, miután a varázsló befejezte a működést: *Űrlap megnyitása vagy Szerkesztése.*
14. Kattintsunk a **Kész** gombra.

Űrlap szerkesztése és formázása Elrendezés nézetben

Amikor befejeztük az űrlap létrehozását akár a **Varázsló** segítségével, akár más módon, általában szükség van annak utólagos szerkesztésére. Amikor létrehoztuk a **Varázsló** segítségével a **Világ országai** űrlapot (3.44. ábra), azt tapasztaljuk, hogy a mezők egy része takarva van más mező által, a **Gazdasági helyzet** mező tartalma rosszul olvasható, mivel a mérete túl kicsi, a **Terület** és a **Létrehozás dátuma** mezők mérete pedig túl nagy. Ezért tehát szükség van az űrlap szerkesztésére, amit **Elrendezés** nézetben tehetünk meg.

Elrendezés nézetben minden objektum két összetartozó elem formájában jelenik meg: az egyik a mező neve, a másik maga a mező (3.45. ábra). Mindkét elem a táblázatban helyezkedik el. Az űrlap minden oszlopa egy táblázat. A határvonalak elmozdításával megváltoztathatjuk a táblázat méreteit. Ha egy elemet szeretnénk kiválasztani, jelöljük ki azt, ha az egész táblát, akkor jelöljük ki annak tetszőleges elemét, majd kattintsunk a tábla bal felső sarkában található jelölőt .



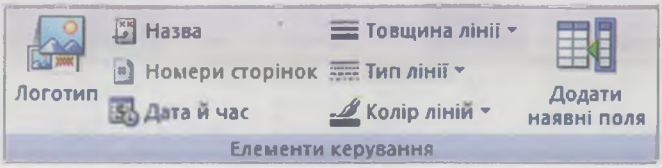
3.44. ábra. A **Világ országai** űrlap

3.45. ábra Az objektum elemei: a felirat és a mező

Elrendezés nézetben a **Betűtípus** vezérlőelemek segítségével megváltoztathatjuk a kijelölt objektumok:

- betűtípusát, betűmértét, betűszínét és igazítását;
- a háttérszínét;
- a tartalom feltételes formázását.

A **Formátum** menü egyéb pontjai segítségével beállíthatjuk a számformátumot, a keretezőszínt, a keretvonal stílusát. A **Formátум** menü **Vezérlők** eszköztárával (3.46. ábra):



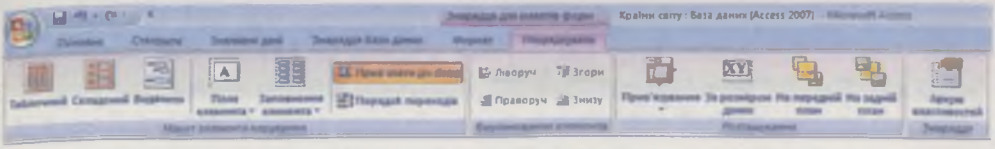
3.46. ábra. A **Vezérlők** eszköztár

- elhelyezhetünk egy **Logót;**
- **Címet** helyezhetünk el az űrlap címsorában;
- elhelyezhetjük az aktuális **Dátumot és időt** a címsorban;
- megváltoztathatjuk az űrlap kijelölt elemének **Keretvonal-stílusát, Vastagságát és Színét.**
- **Új mezőket** adhatunk hozzá az űrlaphoz. A beillesztés a listán található mezők űrlapra való áthúzásával történik.

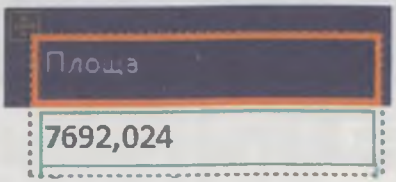
Az **Automatikus formázás** (3.47. ábra) eszköztár segítségével az előre elkészített formátumok egyikét alkalmazhatjuk az űrlapra. Ennek az eszköztárnak a működése megegyezik a **Word 2007**-ben vagy a **PowerPoint 2007**-ben megismerttel.

Az űrlapformázás ideiglenes eszköztár **Igazítás** füléhez tartozó eszközök (3. 48. ábra) segítségével az **Űrlapobjektumok** elhelyezését szerkeszthetjük. Az **Elrendezés** eszköztár segítségével az űrlapobjektumok vagy objektumcsoportok elhelyezését változtathatjuk meg. A **Táblázatos** gombra kattintva a kijelölt elem táblázattá alakul, amelynek felirata az elem fölött helyezkedik el (3.49. ábra). Ha az **Összetett** gombra kattintunk, akkor a kijelölt elem olyan táblázattá alakul át, amelyben a felirat a mező bal oldalán jelenik meg (3.50. ábra). Ezt az objektumot a gombra kattintva az űrlap tesőleges helyére áthúzhatjuk.

3.47. ábra. Az **Automatikus formázás** eszköztár



3.48. ábra. Űrlapobjektumok elhelyezésének szerkesztése **Igazítás** fül segítségével



3.49. ábra

3.50. ábra

A **Törlés** gombra kattintva a kijelölt elemet eltávolíthatjuk az űrlapról. Az objektum törlődik a táblázatból és áthelyeződik az űrlap tetszőleges helyére. Az objektum helyét húzással vagy áthelyezéssel megváltoztathatjuk. A felirat és a mező egymástól függetlenül mozgatható.

A **Margó** eszközre kattintva a lenyíló menüből kiválaszthatjuk, milyen legyen a mező tartalmának behúzása. A lehetséges beállítások: *Nincs, Keskeny, Közepes, Széles.* A mezők közötti távolságot a gombra kattintva állíthatjuk be.

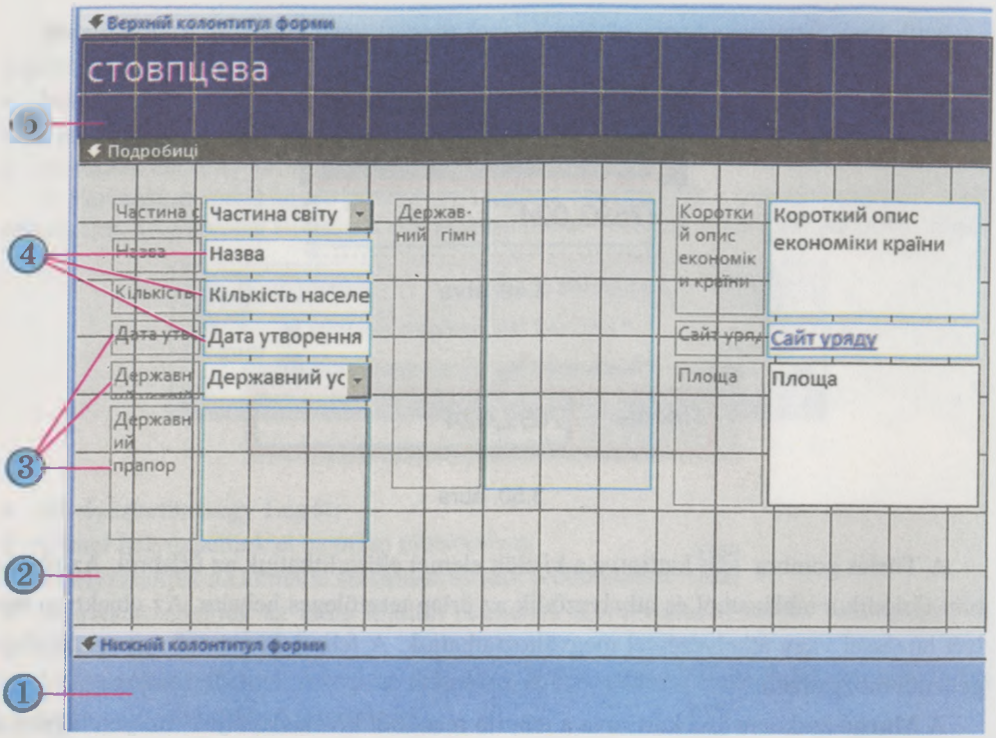
Редагування форми з використанням Конструктора

A **Tervező** nézet az eddig felsoroltaknál szélesebb lehetőségeket nyújt az elemek elhelyezésének kialakításához. A **Tervezető** nézetbe a **Nézet** gombra kattintva léphetünk át. A 3.51. ábrán egy űrlapot láthatunk **Tervező** nézetben.

Az egyik alapművelet **Tervező** nézetben is a mezők elhelyezésének és méretének megváltoztatása. Ez az **Elrendezés** nézetben megismert módon történik.

Gyakori, hogy az űrlapon meg kell változtatni a mezők feliratait. Amikor egy mezőt hozzáadunk az űrlaphoz, a felirata a mező neve lesz, egy kettősponttal kiegészítve. A mezőnév esetében lényeges, hogy az rövid és kifejező legyen, ugyanakkor az űrlapon a mezőfelirat éppen ellenkezőleg, a lehető legteljesebb mértékben ki kell fejezze a mező rendeltetését. A **Név** helyett helyesebb az űrlapon az **Ország** neve feliratot használni, a **Létrehozás** dátuma helyet pedig a **Létrehozás vagy függetlenné válás** dátuma. A felirat tartalmának szerkesztése céljából helyezzük a kurzort a fel-

iratra és a szabványos szerkesztési eszközöket alkalmazva végezzük el a változtatásokat. Vegyük figyelembe, hogy amennyiben a felirat túl hosszú, több sorba tördelhetjük azt. E célból természetesen meg kell változtatni a felirat méreteit is.



1. Alsó hasábcím
2. **Űrlap**
3. A mezők feliratai
4. Mező
5. Felső hasábcím

3.51. ábra. **Űrlap** tervező nézetben

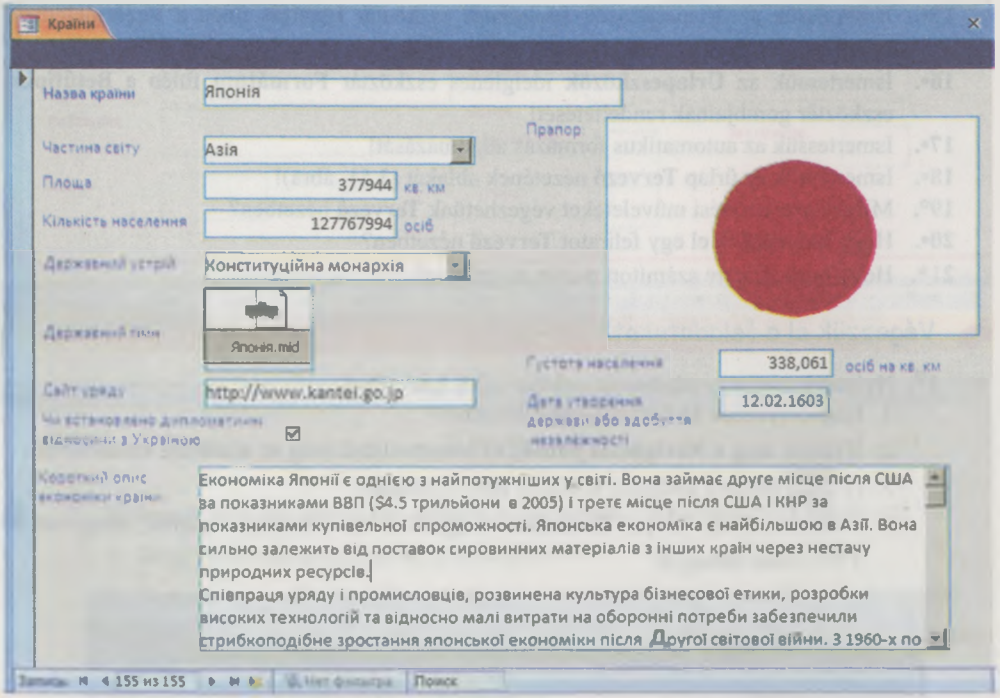
A mezők értelmezését nemcsak a mezők feliratai segíthetik, hanem más, kiegészítő feliratok is. Az országok területe fölött például hasznos lehet elhelyezni egy ilyen feliratot, amely azt mutatná, milyen mértékegységet használunk. Ezt a következőképpen helyezhetjük el:

1. Kattintsunk az űrlaptervező ideiglenes menü **Vezérlőelemek** eszköztárának **Aa Címke** gombjára.
2. Helyezzük el a feliratot az űrlap megfelelő helyére.
3. Írjuk be a *„négyzetkilométer”* szöveget.
4. Az objektum méreteinek megváltoztatásával és áthúzásával pontosítsuk annak elhelyezését az űrlapon.

5. **Űrlap** nézetre váltra győződjünk meg arról, hogy minden az elképzelésünk szerint történt-e.

Ahogy arról már beszéltünk, az űrlapok számított mezőket is tartalmazhatnak. Lássuk, hogy helyezünk el egy, a népsűrűség (a **Lakosság** és a **Terület aránya**) meghatározására szolgáló mezőt. Teendőink a következők lesznek:

1. Kattintsunk az **Űrlaptervező** ideiglenes menü **Vezérlőelemek** eszköztárának szövegmező gombjára.
2. Helyezzük el az elemet az űrlap megfelelő helyére.
3. Az objektum méreteinek megváltoztatásával és áthúzásával pontosítsuk annak elhelyezését az űrlapon.
4. Változtassuk a mező alapértelmezett feliratát **„Népsűrűség"**-re.
5. Írjuk be a mezőbe a népsűrűség kiszámítására szolgáló **=[Lakosság]/[Terület]** képletet.
6. **Űrlap** nézetre váltva győződjünk meg arról, hogy minden az elképzelésünk szerint történt-e.



3.52. ábra. A **Világ országai** űrlap **Űrlap** nézetben

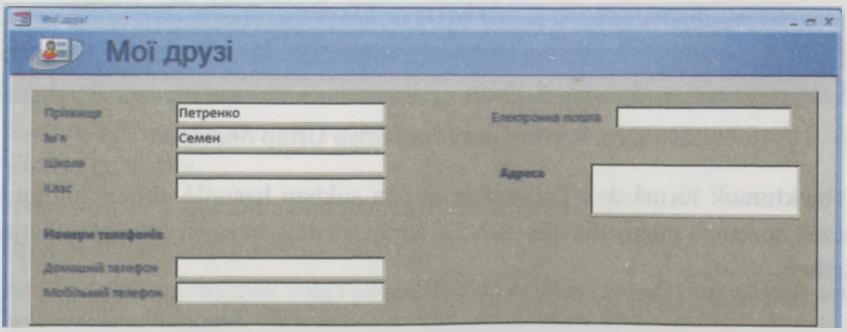
Az objektumok formázása ***Tervező*** nézetben sokban hasonló ahhoz, ahogy azt az **Elrendezés** nézetben megismertük. A 3.52. ábrán a világ országai űrlapot láthatjuk **Űrlap** nézetben.

Ellenőrző kérdések

1°. Mire használjuk az adatbázis-űrlapokat?
2°. Milyen űrlapokat használhatunk **Access 2007**-ben?
3•. Miben különbözik az adattábla az űrlap *adattábla* nézetétől?
4•. Miben különbözik az adattábla űrlaptípus a többelemestől?
5•. Miben különbözik az ***szokásos*** űrlaptípus a ***kiegyenlítettől?***
6•. Mi célt szolgál az ***osztott*** űrlaptípus? Milyen a külalakja?
7°. Milyen nézetei vannak egy űrlapnak **Access 2007**-ben? Hogy váltunk nézetet?
8°. Ismertessük az űrlap létrehozásának folyamatát a **Létrehozás** menü **Űrlap** eszközével!
9°. Hogy történik az űrlap mentése?
10. Hogyan hozunk létre űrlapot egy tábla alapján az **Űrlap varázsló** segítségével?
11°. Milyen űrlaptípusokat hozhatunk létre az **Űrlap** varázslóval?
12°. Milyen nézetekben szerkeszthetjük az űrlapokat?
13•. Hogy változtathatjuk az űrlapelemek elhelyezkedését **Elrendezés nézetben**?
14•. Mire használjuk az űrlapeszközök ideiglenes eszköztár igazítás fülén a **Táblázatos** és **Összetett** gombokat?
15•. Ismertessük az űrlapeszközök **Ideiglenes** eszköztár **Igazítás** fülén a **Vezérlőelemek** eszköztár gombjainak rendeltetését!
16•. Ismertessük az **Űrlapeszközök** ideiglenes eszköztár **Formátum** fülén a **Betűtípus** eszköztár gombjainak rendeltetését!
17•. Ismertessük az automatikus formázás alkalmazását!
18•. Ismertessük az űrlap **Tervező** nézetének ablakát (3.51. ábra)!
19°. Milyen szerkesztési műveleteket végezhetünk **Tervező** nézetben?
20•. Hogy helyezzünk el egy feliratot **Tervező** nézetben?
21*. Hogy hozunk létre számított mezőt az űrlapon?

Végezzük el a feladatokat!

1°. Nyissunk meg egy adatbázist, például a **3.6.1.accdb**-t!
1. Engedélyezzük az összes adatbáziselemet!
2. Nyissuk meg a **Navigációs panelt**, és ismerkedjünk meg az adatbázis tartalmával!
3. A gombra kattintva hozzunk létre új űrlapot!
4. Alakítsuk át az űrlapot **Elrendezés** vagy **Tervező** nézetben olyanná, ahogy azt a 3.53. ábrán láthatjuk!

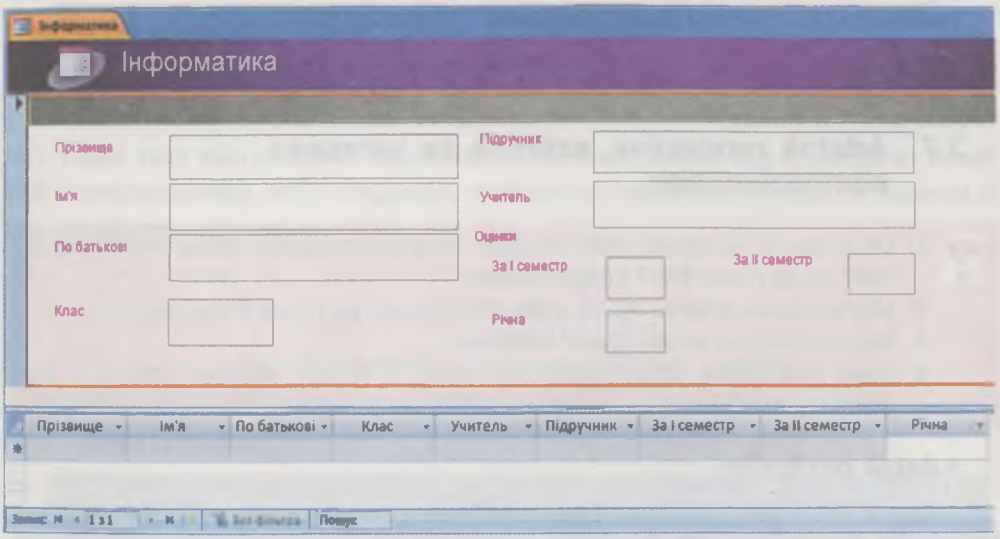


3.53. ábra

5. Töltsük fel a táblát az űrlap alkalmazásával öt rekorddal!
6. Mentsük az adatbázist **3.6.1.accdb** néven!

2*. Nyissunk meg egy adatbázist, például a **3.6.2.accdb**-t:

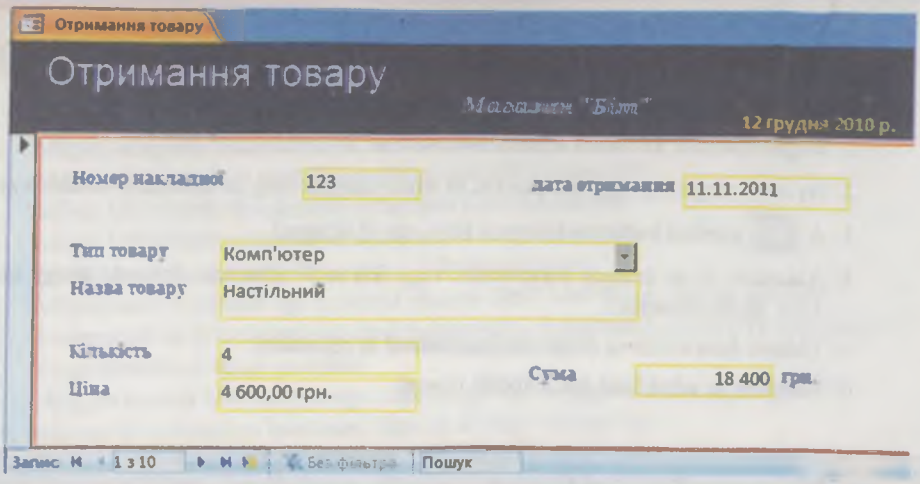
1. Engedélyezzük az összes adatbáziselemet!
2. Nyissuk meg a **Navigációs panelt,** és ismerkedjünk meg az adatbázis tartalmával!
3. A gombra kattintva hozzunk létre egy új űrlapot!
4. Alakítsuk át az űrlapot **Elrendezés** vagy **Tervező** nézetben olyanná, ahogy azt a 3.54. ábrán láthatjuk!
5. Töltsük fel a táblát az űrlap alkalmazásával öt rekorddal!
6. Mentsük az adatbázist **3.6.2.accdb** néven!



3.54. ábra

3*. Nyissunk meg egy adatbázist, például a **Завдання 3.6\зразок 3.6.3.accdb**–ot!

1. Engedélyezzük az összes adatbáziselemet!
2. Nyissuk meg a **Navigációs panelt** és ismerkedjünk meg az adatbázis tartalmával!
3. Vizsgáljuk meg a kapcsolatokat, és állapítsuk meg, milyen mezők vannak összekapcsolva!
4. Az **Űrlap varázsló** segítségével hozzunk létre egy űrlapot!
5. Alakítsuk át az űrlapot **Tervező** nézetben olyanná, ahogy azt a 3.55. ábrán láthatjuk!
6. Helyezzünk el az űrlapon számított mezőket a raktáron levő áru mennyiségének és értékének kiszámítására!
7. Mentsük az adatbázist **3.6.3.accdb** néven!



3.55. ábra

3.7. Adatok rendezése, szűrése és keresése adatbázisokban

1. Mit értünk a rendezés alatt? Hogyan történik az adatok rendezése a **Word 2007** és az **Excel 2007** táblázataiban?
2. Milyen adatokat kereshetünk a **Word 2007**-ben és **Excel 2007**-ben?
3. Hogy működik a szűrés **Excel 2007**-ben?
4. Hogy szűrhetünk **Excel 2007**-ben olyan adatokat, amelyek értéke 0 és 100 közé esik?

Adatok rendezése

Ahogy a **Word 2007**-ben és az **Excel 2007**-ben, az adatbázisokban is lehet adatokat szűrni. Az **Access 2007**-ben az adatok rendezésének sorrendje alapértelmezés szerint növekvő. Ha ezt meg szeretnénk változtatni, akkor:

1. Nyissuk meg a rendezendő adatokat tartalmazó táblát.
2. Helyezzük a kurzort abba a mezőbe, amelyik szerint végre kívánjuk hajtani a rendezést.
3. Kattintsunk a kezdőlapon a **Rendezés** eszköztár **Növekvő** vagy **Csökkenő** gombjára.

Назва	Частина світу	Площа	Кількість населення
Катар	Азія	11	565
Ліван	Азія	10,4	3496
Бахрейн	Азія	0,6	640
Мальдівська Республіка	Азія	0,3	291
Австралія	Австралія і Океанія	7636	19138
Нова Зеландія	Австралія і Океанія	268,7	3778
Соломонові Острови	Австралія і Океанія	29	447
Тонга	Австралія і Океанія	0,7	100

3.56. ábra. A **Világ országai** tábla **Földrész** és **Terület** szerint rendezve

Ha egy tábla rendezve van valamelyik mező szerint, akkor a mező neve mellett egy ↑ vagy ↓ jel látható attól függően, hogy a rendezés növekvő vagy csökkenő. A **Rendezés és Szűrés** eszköztár gombjára kattintva a rendezés megszüntethető. Ha egyidejűleg több mező szerint kívánjuk rendezni a táblát, jelöljük ki ezeket a mezőket (csak szomszédos mezők kijelölése engedélyezett), majd kettintsunk a **Rendezés** eszköztár **Növekvő** vagy **Csökkenő** gombjára. A 3.56. ábrán a **Világ országai** táblát láthatjuk, amely a **Földrész** és a **Terület** szerint van csökkenő sorrendbe rendezve.

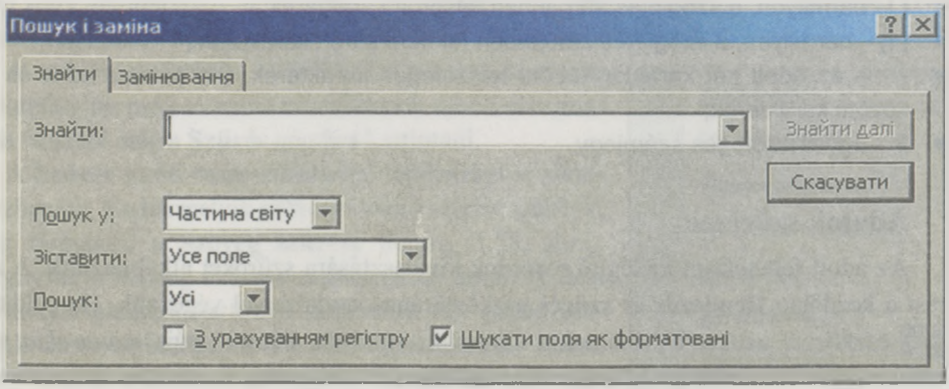
Ha a mezők nincsenek egymás mellett, akkor is elvégezhető a több oszlop szerinti rendezés, ha az oszlopokat egyenként kijelöljük, majd ugyancsak egyenként elvégezzük a rendezést is. Ekkor az egyes mezők rendezése eltérő lehet.

Az űrlapok rendezése ugyanígy történik azzal a különbséggel, hogy az űrlapokat nem lehet több oszlop szerint rendezni egyidejűleg.

Adatok keresése

Ahogy arról már szó volt, az adatbázisok létrehozásának egyik célja az adatok gyors kereshetőségének biztosítása. A felhasználók gyakran végeznek keresést könyvtári, vasúti vagy más internetes adatbázisban. Az **Access 2007**-ben a keresés az **Excel 2007** tanulmányozása során megismert módszerhez hasonlóan történik. A keresést a következőképpen végezzük:

1. Nyissuk meg a fájlt, amelybe keresni szeretnénk.
2. Nyissuk meg a táblát, amelybe keresni szeretnénk.
3. Kattintsunk a ***Kezdőlap Keresés*** eszköztárán a ***Keresés*** gombra.
4. A megnyíló **Keresés** és **Csere** párbeszédablakban (3.57. ábra) adjuk meg a keresendő kifejezést.



3.57. ábra. A **Keresés** és **Csere** párbeszédablak

5. Válasszuk ki a tábla vagy űrlap azon mezőjének nevét, amelyben keresni szeretnénk.
6. Az **Egyezés** menüből válasszuk ki az egyezés módját: *Teljes mező, a Mező tetszőleges része, a Mező eleje.*

7. A **Keresés** menüből válassszuk ki a keresés irányát: *mindenütt, lefelé vagy felfelé.*
8. A megfelelő jelölők beállításával adjuk meg, hogy a kis- és nagybetűk figyelembevételével végezzük-e a keresést vagy alkalmazzunk-e sablonokat.
9. Kattintsunk a **Keresés** gombra.

Ha szeretnénk megtalálni a beállításoknak megfelelő következő értéket is, kattintsunk a **Következő** gombra.

A **Keresés** iránya megmutatja, hogy az aktuális rekordhoz képest milyen lesz a keresés iránya: *Felfelé* – az előző rekord irányába, *Lefelé* – a következő rekord irányába, *Mindenütt* – a teljes táblában.

Adatok cseréje

Az adatok cseréje is a **Keresés és csere** párbeszédablak segítségével történik. A keresésnél megadott paramétereken kívül meg kell még adni, hogy a megtalált adatokat mire kell cserélni. A cserét végrehajthatjuk lépésenként (a **Következő** gombra kattintva) vagy egyszerre (a **Mind** gombra kattintva).

Helyettesítő karakterek használata a keresés során

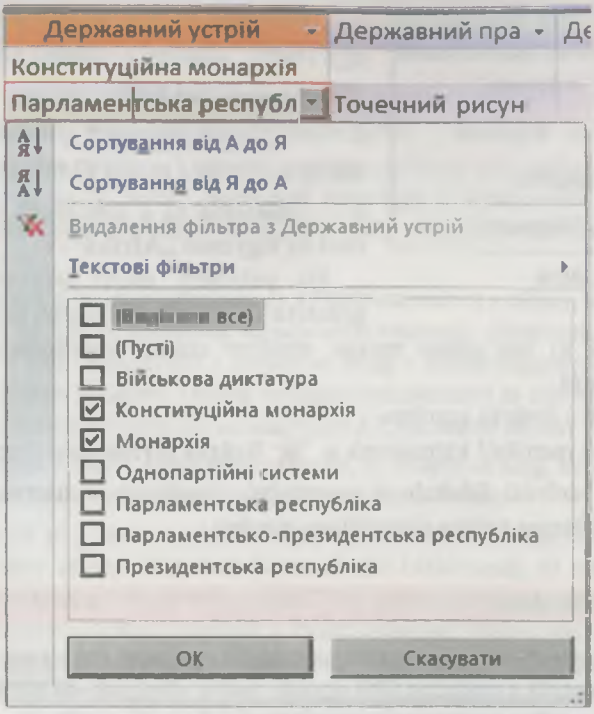
A keresés során az **Access 2007-ben** helyettesítő karaktereket is alkalmazhatunk. Az **Access 2007**-ben használt helyettesítő karakterek megfelelnek az **ANSI-89** szabványnak, ami az **SQL** nyelv egyik szabványa:

- * – tetszőleges mennyiségű bármilyen karakter, beleértve a karakterek hiányát;
- ? – egy tetszőleges karakter;
- [ ]– bármilyen, a szögletes zárójelben felsorolt karakterek;
- [!] – bármilyen, a szögletes zárójelben fel nem sorolt karakterek;
- - – az adott két karakter közötti tetszőleges karakterek, *a-d* például az *a* és a *d* közötti karakterek;
- # – egy tetszőleges számjegy.

Adatok szűrésee

Az adott feltételeket kielégítő rekordok kiválasztására szűrőket alkalmazunk. A szűrést a kezdőlap **Rendezés és szűrés** eszköztárának eszközeivel végezzük. Ha például a eszközzel azokat az államokat keressük, amelyek államformája *monarchia* vagy *alkotmányos monarchia,* a következőket kell tenni:

1. Tegyük aktuálissá azt a mezőt, amelynek értékei szerint a szűrést végezzük.
2. Kattintsunk a **Szűrés** gombra.
3. A listán (3.58. ábra) kapcsoljuk be azokat a jelölőket, amelyek megfelelnek a kitűzött szűrési feladatnak, jelen esetben a *Monarchia és Alkotmányos monarchia* melletieket.
4. Kattintsuk az **OK** gombra.

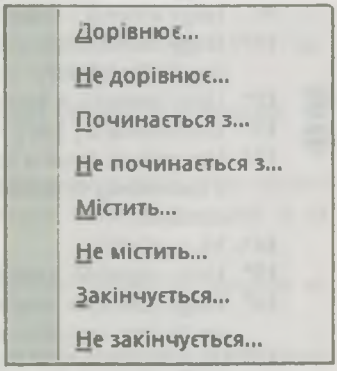


3.58. ábra. Az **Államforma** mező szűrése

A szűrést egyidejűleg több mező szerint is végezhetjük. Ha például azt szeretnénk megállapítani, hogy melyek azok az *Afrikai* országok, amelyek államformája *monarchia vagy alkotmányos monarchia,* az **Államforma** mezőre alkalmazott szűrésen kívül el kell végezni a **Földrész** szerinti szűrést is, *Afrika* paraméterrel.

A szűrést a **Szűrés eltávolítása** gombra kattintva szüntethetjük meg. A szűrési beállítások újbóli alkalmazása céljából elég a **Szűrés** gombra kattintani.

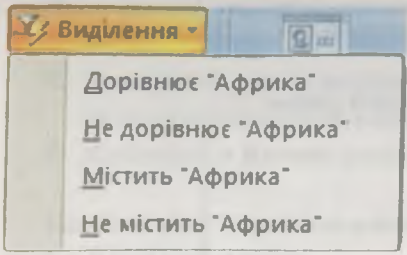
A **Szűrés** során összetett szűrési feltételeket is alkalmazhatunk. Ha bármelyik szöveges mező szerint szűrűnk, elég rámutatni a **Szöveg szűrése** pontra (3.58. ábra), megnyílik a szöveg szűrése terület (3.59. ábra). A lista bármelyik elemére kattintva megnyílik egy párbeszédablak, amelybe beírhatjuk azt a szöveget, amit *keresünk,* vagy amivel a keresett szövegnek *kezdődni*, vagy végződni kell stb.



3.59. ábra. A kiegészítő szűrési feltételek

Ha a mező szám típusú adatokat tartalmaz, akkor a feltételek listáján az *Egyenlő,* a *Nem egyenlő,* a *Több,* a *Kevesebb* és a *Között* szerepel majd, dátum és idő esetében pedig az *Egyenlő,* a *Nem egyenlő,* az *Előtt,* az *Után* és a *Között.*

A kijelöléssel történő szűrést a következőképpen végezzük:



3.60. ábra

1. Jelöljünk ki egy mező tartalmát vagy annak egy részét, amelyre a szűrést végezni szeretnénk, például az *Afrika* szót.

2. A Виділення gombra kattintva gördítsük le a kijelölés menüt (3.60. ábra).

3. Válasszuk ki a szükséges szűrőfeltételt, például az **Egyenlő „Afriká"**- t.

Ha valamely mező szűrését meg kívánjuk szüntetni, a következőképpen kell eljárnunk:

1. Válasszunk ki egy olyan mezőt, amelyre szűrést kapcsoltunk be, például az **Államformát.**
2. Kattintsunk a **Szűrés** gombra.
3. A legördülő menüből kattintsunk a **Szűrés törlése** eszközre.

Ha az összes szűrési feltételt el szeretnénk távolítani, kattintsunk az **Advanced** gombra, majd az Összes szűrés eltávolítása gombra.

Ellenőrző kérdések

1°. Mi a rendezés? Hogyan végezzük a rendezést egy mező értékei szerint?
2°. Hogy végezzük a rendezést több egymás mellett elhelyezkedő mező értékei szerint?
3°. Hogy végezzük a rendezést egymástól távol eső mezők esetében?
4°. Hogy szüntetjük meg a rendezést?
5*. Hasonlítsuk össze a rendezést **Word 2007,** az **Excel 2007** és **Access 2007** esetében! Mi a hasonlóság és a különbség a **Microsoft Office 2007** programokkal végzett rendezések esetében?
6°. Hogy keressük az *informatika* szót, ha az a mező elején található? Ha a mező végén? Ha a mezőn belül bárhol?
7°. Végezhetünk-e keresést egyetlen mezőn vagy egyetlen rekordon belül?
8*. Hogy tekinthetjük meg a keresés során talált összes rekordot?
9*. Hogy történik a teljes mező tartalmának cseréje?
10*. Hogy cserélhetjük a *méter* szót *kilométerre,* ha az a mező tartalmának bármely részében előfordulhat?
11°. Hogy használjuk kereséskor a helyettesítő karaktereket?
12°. Ismertessük a [ ] és [-] helyettesítő karakterek alkalmazását!
13*. Hasonlítsuk össze a keresést **Word 2007**, az **Excel 2007** és **Access 2007** esetében! Mi a hasonlóság és a különbség a **Microsoft Office 2007** programokkal végzett keresések esetében?
14°. Mi a szűrés?
15°. Hogy végezzük a szűrést egy mező értékei alapján?
16°. Hogy jelenítjük meg azokat a rekordokat, amelyeknek van legalább egy olyan mezője, amely nem tartalmaz értéket?
17°. Hogy határozhatjuk meg az adott feltételt kielégítő rekordok számát?
18°. Hogy végezzük a szűrést három mező értékei szerint?
19°. Miben különbözik a **Szöveg,** a **Szám, Dátum** és **Idő** típusú mezők szűrése?
20°. Hogy szüntetjük meg a szűrést? Hogy alkalmazhatjuk újra a szűrőfeltételeket?
21°. Hasonlítsuk össze a keresés és a szűrés eredményeit! Mi a közös és mi az eltérés az ugyanazon paraméterekkel végzett keresés és szűrés eredményeiben?

Végezzük el a feladatokat!

1°. Nyissunk meg egy adatbázist, például a **3.7.1.accdb**-t! Engedélyezzük a teljes adatbázistartalmat! Nyissuk meg a **Világ országai** táblát!

1) Rendezzük a táblát a **Lakosság** szerint *csökkenő sorrendben!* Állapítsuk meg, melyik az az öt ország, amelynek lakossága a legnagyobb! Határozzuk meg, hányadik helyet foglalja el **Ukrajna** a világ országai között a népesség lélekszáma szerint!

2) Rendezzük a táblát a **Lakosság** és a **Földrész** szerint a célból, hogy kideríthessük, melyek az egyes földrészek legnépesebb országai. Állapítsuk meg, milyen sorrendben kell végrehajtani a rendezést, hogy a lehető leggyorsabban választ kapjunk a feltett kérdésre! Melyik mezőnél kell növekvő és melyiknél csökkenő sorrendet beállítani? Mentsük az adatbázist **3.7.1.2.accdb** néven!

3) Rendezzük a táblát **Államforma** szerint, és állapítsuk meg, hány ország államformája a *monarchia* és hányé az *alkotmányos monarchia.*

4) Végezzük el a tábla rendezését **Államforma** és **Földrész** szerint, és állapítsuk meg, hány ország államformája az *elnöki köztársaság* az egyes földrészeken! Állapítsuk meg, milyen sorrendben kell végrehajtani a rendezést, hogy a lehető leggyorsabban választ kapjunk a feltett kérdésre! Melyik mezőnél kell növekvő és melyiknél csökkenő sorrendet beállítani? Mentsük az adatbázist **3.7.1.4.accdb** néven!

2°. Nyissunk meg egy adatbázist, például a **3.7.2.accdb**-t! Engedélyezzük a teljes adatbázistartalmat! Nyissuk meg a **Világ országai** táblát, és hajtsuk végre az adatok keresését és cseréjét!

1) Keressük ki azokat az országokat, amelyek fizetőeszköze a *krusado, a ketsal,* az *arany cordoba* és a *dollár.* Utóbbi esetben a pénzegység neve a mező értékeinek tetszőleges helyén lehet.

2) Cseréljük az *Észak-Amerika* előfordulásait *Amerikára.* Mentsük az adatbázist **3.7.2.2.accdb** néven! Fogalmazzuk meg a csere algoritmusát!

3) Keressük ki azokat az országokat, amelynek területében előfordulnak az 1-es vagy 2-es számjegyek! Keressük ki az 1960-ban létrehozott országokat!

4) Cseréljük az *Közép-* és *Dél-Amerika* előfordulásait *Amerikára!* Mentsük az adatbázist **3.7.2.4.accdb** néven!

3°. Nyissunk meg egy adatbázist, például a **3.7.3.accdb**-t! Engedélyezzük a teljes adatbázistartalmat. Nyissuk meg a **Földrajzi és fizikai térképek** táblát, és végezzük el az adatok szűrését!

1) Válasszuk ki a *karton* vagy *léces* karton alapú *politikai térképeket!* Mentsük az adatbázist **3.7.3.1.accdb** néven!

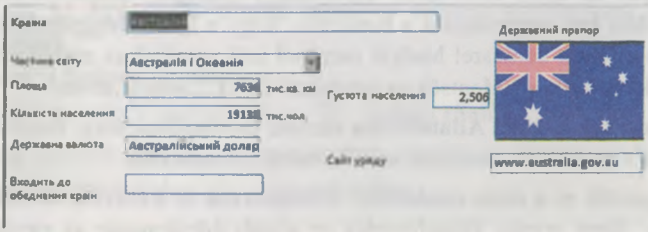
2) Távolítsuk el a szűrőket! Szűrjük ki Ukrajna térképeit, amelyek mértéaránya *M1:2 100 000* és ára meghaladja a *20 hrivnyát!* Hány ilyet találtunk? Mentsük az adatbázist **3.7.3.2.accdb** néven!

3) Távolítsuk el a szűrőket! Keressük ki a *5* és *40* hrivnya közötti árral rendelkező Ukrajna-térképeket! Hány ilyet találtunk? Mentsük az adatbázist **3.7.3.3.accdb** néven!

13. számú gyakorlati munka Űrlapok. Adatok feltöltése. Adatok rendezése, szűrése és keresése

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Nyissunk meg egy adatbázist, például a **Тема 3\Практична 13\Зразок 13-т!**
2. Bővítsük ki az **Országok** táblát az **Egyesülések** mezővel!
3. A tábla alapján hozzuk létre a **Világ országai** űrlapot az ábra szerint! Alkalmazzuk a **Szigorú Automatikus formázást.**
4. A népsűrűség kiszámítása céljából helyezzünk el az űrlapon egy számított mezőt!



5. Az űrlap alkalmazásával keresés segítségével töltsük fel az FÁK-országok és az OPEC (*The Organization of the Petroleum Exporting Countries olajexportáló országok szervezete*) országok esetében az egyesülés mezőket adatokkal! A szükséges adatokat a **Союзи країн.doc** fájlban találjuk meg.
6. Nyissuk meg az **Országok** táblát és rendezzük azt az **Egyesülés** mező szerint! Győződjünk meg arról, hogy az előző pontban helyesen végeztük el az adatok feltöltését!
7. Állítsuk be a következő szűrőfeltételeket: **Terület** – kisebb ezer négyzetkilométernél, **Lakosság** több, mint 1 millió fő.
8. Mentsük az adatbázist **13_gyakorlati.accdb** néven!

3.8. Lekérdezések Access 2007-ben

1. Mi a képlet **Excel 2007**-ben? Mit tartalmazhat a képlet?
2. Mire használjuk a szűrőket **Access 2007**-ben? Hogy alkalmazzuk a szűrőket a tábla adataira?
3. Milyen műveleteket használhatunk az **Excel 2007** képleteiben? Hogy jelöljük ezeket?
4. Ismertessük a többmezős szűrés beállításának menetét!

A lekérdezések fogalma

Az adatbázisok egyik leglényegesebb eszközei a lekérdezések. Segítségükkel nemcsak bizonyos kritériumok szerinti keresést lehet végezni, ahogy azt a szűrés során tettük, de az adatokon végzett műveleteken kívül el is lehet menteni a keresések eredményét.

A felhasználó a lekérdezések alkalmazásával a következő műveleteket végezheti el:

- új táblákat hozhat létre a már létező táblák adatainak elemzése alapján;
- számításokat végezhet el csoportosított adatokon (összegzés, maximum és minimumszámítást stb.);

- új adatok értékeit számíthatja ki a különböző táblákban már meglévő adatok alapján (például kiszámíthatja a népsűrűséget a lakosság számának és az ország területének alapján);
- változtatásokat végezhet a már létező táblák adatain (adatok frissítése, rekordok beszúrása, törlése).

Rendeltetésük alapján a lekérdezések a következők lehetnek:

- **választó lekérdezések,** amelyek a már létező egy vagy több tábla (3.61. ábra) bizonyos feltételeknek megfelelő adatait jeleníti meg (3.62. ábra);
- **kereszttáblás lekérdezések,** amelyek előbb elvégzik az adatok összegzését (összeg, átlag, maximum meghatározása), majd két kritérium alapján csoportosítják azokat, amelyek egyike az oszlopfejléc, a másik pedig a sorfejléc lesz. Legyen egy lekérdezésünk, amely a sportolók három távon elért legjobb eredményeit tartalmazza (3.63. ábra). Ekkor az oszlopfejléc a **Távolság** lesz, a sorfejléc pedig a **Sportoló,** a cellák pedig az elért eredményeket tartalmazzák;
- **frissítő lekérdezések,** amelyek segítségével a felhasználó megváltoztathatja a rekordok tartalmát. Hozzáfűző lekérdezésekkel új rekordokat fűzhetünk a táblához, törlő lekérdezésekkel pedig rekordokat távolíthatunk el azokból.

Результати змагань

Ид	Спортсмен	Дистанція	Результат
2	Голованов Василь	100 м	11,4
3	Петренко Степан	100 м	10,5
4	Федченко Микола	100 м	10,8
5	Хмиз Дмитро	100 м	11,1
6	Суботів Віктор	100 м	10,9
7	Голованов Василь	1500 м	4:01,23
8	Петренко Степан	1500 м	4:02,00
9	Федченко Микола	1500 м	4:03,11
10	Хмиз Дмитро	1500 м	4:02,45
11	Петренко Степан	400 м	54,65
12	Хмиз Дмитро	400 м	51,4
13	Суботів Віктор	1500 м	4:03,22
14	Суботів Віктор	400 м	52,34
15	Голованов Василь	1500 м	4:01,54
16	Голованов Василь	100 м	11,6
17	Федченко Микола	400 м	55,7
18	Федченко Микола	1500 м	4:03,7
19	Голованов Василь	400 м	52,6

3.61. ábra. A sportolók három távon elért legjobb eredményeit tartalmazó táblázat

Результати змагань_перехрестний

Спортсмен	Дистанція	Min-Результат
Голованов Василь	100 м	11,4
Петренко Степан	100 м	10,5
Суботів Віктор	100 м	10,9
Федченко Микола	100 м	10,8
Хмиз Дмитро	100 м	11,1

3.62. ábra. Választó lekérdezéssel kapott táblázat a sportolók 100 méteren elért legjobb eredményeiről

Результати змагань_перехрестний

Спортсмен	100 м	1500 м	400 м
Голованов Василь	11,4	4:01,23	52,6
Петренко Степан	10,5	4:02,00	54,65
Суботів Віктор	10,9	4:03,22	52,34
Федченко Микола	10,8	4:03,11	55,7
Хмиз Дмитро	11,1	4:02,45	51,4

3.63. ábra. Választó lekérdezéssel kapott táblázat a három távon elért legjobb eredményről

- **paraméteres lekérdezések,** amelyek futtatása során a felhasználó paramétereket adhat meg, amelyek értéke szerint az adatfeldolgozás folytatódik majd. Létrehozhatunk például olyan lekérdezést, amely az iskola azon osztályának listáját állítja elő, amelyre a felhasználónak éppen szüksége van. A futtatás során megnyílik egy ablak, amelybe a felhasználó beírja az osztály nevét. Ennek értékétől függően létrejön egy osztály listája.

A lekérdezést elmenthetjük az adatbázisban egy általunk választott név alatt. Amikor a lekérdezést később megnyitjuk, az adatfeldolgozás újból megtörténik és a futtatás eredményét újból láthatjuk a képernyőn.

Kifejezések és műveletek lekérdezésekben

A lekérdezésekben a szűrőfeltételek vagy az adatokkal elvégzendő műveletek beírása során **kifejezéseket** használunk. Az **Access 2007** kifejezései hasonlatosak az **Excel 2007** képleteivel. A kifejezések a következő elemeket tartalmazhatják:

- **azonosítókat** – mezőneveket és az űrlapok vezérlőelemeinek neveit;
- **műveleti jeleket,** amelyek lehetnek *matematikai, relációs, logikai, egyesítő* vagy *speciális* műveleti jelek;
- **függvényeket,** amelyek hasonlóak az **Excel 2007** függvényeihez, bár az **Access 2007** ettől eltérő függvényeket tartalmaz és a függvények kategóriái is eltérőek: *tömb, átalakítás, adatbázis, matematikai, pénzügyi, dátum* stb.
- **állandókat,** amelyek változatlan értéket tartalmazhatnak, például számértékeket, szövegrészleteket;
- **zárójeleket,** a műveletek sorrendjének megváltoztatása céljából.

A kifejezések használata során be kell tartani a következő szabályokat:

- a mezőneveket és egyéb azonosítókat szögletes zárójelbe kell tenni;
- ha egy tábla mezőjére szeretnénk hivatkozni, a tábla neve után felkiáltójelet kell tenni, az **[Áru]![Áru_kód]** hivatkozás például az **Áru** tábla **Áru_kód** mezőjére mutat.
- a szöveges adatokat idézőjelbe kell tenni.

Választó lekérdezések

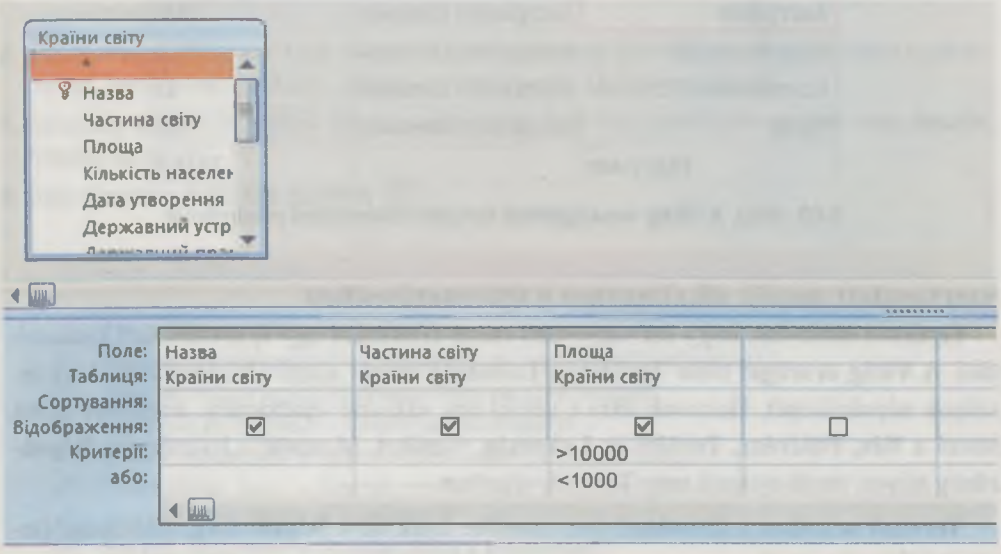
Kövessük végig a választó lekérdezés létrehozását a **Lekérdezés varázsló** segítségével. A **Világ országai** adatbázisból kérdezzük le az országok területeit. Nyissuk meg a **Világ országai** adatbázist, majd a **Létrehozás** menü **Kiegészítő** lapján kattintsunk a **Lekérdezés** varázsló eszközre. Ezután kövessük a varázsló lépéseit:

1. Az **Új lekérdezés** ablakban válasszuk ki az **Egyszerű lekérdezés** varázslót.
2. Az **Egyszerű lekérdezés varázsló** ablakában a **Táblázatok és lekérdezések** listából válasszuk ki azt a táblát, amely alapján létre szeretnénk hozni a lekérdezést, jelen esetben a **Világ országait.**
3. Az **Elérhető mezők** listájából helyezzük át a **Kiválasztott mezők** listájába a szükséges mezőneveket: a **Név,** a **Földrész** és a **Terület** mezőket.
4. Válasszuk ki a megjelenítés típusát: *részletes* (összegzés nélküli) vagy *összegzett (összegek, maximum, minimum, átlag* kiszámítása). Ez, valamint a következő lépés csak számokat tartalmazó mezők esetében értelmezhető.
5. Válasszuk ki az **Összegzés paramétereit** a megfelelő jelölők bekapcsolásával, majd kattintsunk az **Ok** gombra.
6. Adjunk nevet a lekérdezésnek, például a **Világ országainak területe.**

7. Válasszuk ki a folytatás módját: a *lekérdezés megnyitása, vagy a lekérdezés tervének megnyitása.*
8. Kattintsunk a **Kész** gombra.

Ha a *lekérdezés megnyitása lehetőséget* választottuk, a képernyőn megjelenik a kiválasztott mezőket tartalmazó új táblázat.

Ha a második lehetőséget választottuk, akkor a lekérdezés **Tervező** nézetben nyílik meg (3.64. ábra). Ebben a nézetben az ablak felső részében annak a táblának a mezőit láthatjuk, amelynek alapján a lekérdezést létrehoztuk. Ha szükséges, a **Lekérdezés** eszköztár **Tábla** hozzáadása gombjára kattintva bővíthetjük az elérhető táblák és mezők listáját.



3.64. ábra. A **Világ országainak területe Tervező** nézetben

A **Táblák** ablakból válasszuk ki a szükséges táblákat, majd kattintsunk a **Hozzáadás** gombra. Ha egy táblára nincs szükségünk, a jobb gombbal aktiválható helyi menüben kattintsunk a **Tábla** törlése gombra.

A tervező nézet ablakában a táblák alatt a QBE rács látható. A rács a következő sorokat tartalmazza:

- **Mező,** ahová a kiválasztott mezők nevei kerülnek;
- **Tábla,** ahol azoknak a tábláknak a neveit láthatjuk, amelyekhez a kiválasztott mezők tartoznak;
- **Rendezés,** ahol a mezőre alkalmazott rendezési módot válasszuk ki.
- **Megjelenítés,** ahol a jelölő be- illetve kikapcsolásával szabályozzuk, hogy az illető mező kerüljön-e megjelenítésre az eredményhalmazban;
- **Feltétel,** amely az adott mezőre vonatkozó szűrési feltételt tartalmazza;
- **Vagy,** amelyre akkor van szükség, ha összetett szűrési feltételt szeretnénk alkalmazni. A két szűrési feltétel a **VAGY** művelettel lesz összekapcsolva.

Ha egy mezőre vonatkozólag több oszlopban fogalmazunk meg szűrési feltételeket, akkor azok az **ÉS** művelettel lesznek összekapcsolva.

Ha például azt szeretnénk, hogy a lekérdezés csak az **Ausztrália és Óceánia** területén fekvő országokat jelenítse meg, akkor a **Földrész** mező **Feltétel** sorába írjuk be az *="Ausztrália és Óceánia"* feltételt. A beállított szűrési feltétel alkalmazása céljából futtassuk a lekérdezést.

A lekérdezéshez az **Összegzés** gombra kattintva összegző mezőt is hozzáadhatunk. Ha a **Terület** mezőhöz hozzáadunk egy **Összegző** mezőt és ott kiválasztjuk az **Összeg** beállítást, kiszámíthatjuk az országok területeinek összegét (3.65. ábra).

Назва	Частина світу	Площа
Австралія	Австралія і Океанія	7636
Нова Зеландія	Австралія і Океанія	268,7
Соломонові Острови	Австралія і Океанія	29
Тонга	Австралія і Океанія	0,7
Підсумок		7934,4

3.65. ábra. A **Világ országainak** területe lekérdezés eredménye

Szerkesztett mezők alkalmazása a lekérdezésekben

Gyakran előfordul, hogy két vagy több mező értékeiből egy új értéket kell kiszámítani. A **Világ országai** tábla **Terület** és **Lakosság** mezői alapján kiszámíthatjuk az országok népsűrűségét. Hozzunk létre e célból egy választó lekérdezést, amelyhez adjuk hozzá a **Név, Földrész, Terület** és **Lakosság** mezőket. Mentsük a lekérdezést **Népsűrűség** néven, majd nyissuk meg Tervező nézetben.

Tervező nézetben a következő üres mezőbe írjuk be a **Népsűrűség:** [lakosság]\[terület] kifejezést, ahol:

- **Népsűrűség:** – a mező neve;
- A **Lakosság** és a **Terület** azoknak a mezőknek a nevei, amelyeket a kifejezésben használunk;
- \ – az osztás és egészekre kerekítés jele.

A 3.66. ábrán a népsűrűség számított mezőt tartalmazó lekérdezés futtatásának eredményét láthatjuk

Густота населення

Назва	Частина світу	Площа	Кількість населен	Густота населення
Австралія	Австралія і Океанія	7692,024	22510123	2926
Австрія	Європа	83,871	8356707	99484
Азербайджан	Європа	86,6	9047000	103988
Албанія	Європа	28,7	3187009	109896
Алжир	Африка	2381,7	35200000	14777
Ангола	Африка	2381,74	13134	5

3.66. ábra. A **Népsűrűség** lekérdezés futtatásának eredménye

Paraméteres lekérdezések

A felhasználó által a szűrőfelvételbe a futtatás során beírt paramétert a választó, a kereszttáblás és a frissítő lekérdezésekben egyaránt alkalmazhatunk.

Tegyük fel, hogy az országok területét összegző lekérdezésünket úgy szeretnénk átalakítani, hogy ott meg tudjuk változtatni a **Földrészt**, amelyre nézve az összegzést végezzük. Ezt egyelőre csak úgy tudnánk elvégezni, hogy minden futtatás előtt visszatérünk **Tervező** nézetbe és elvégezzük a megfelelő változtatásokat, esetleg valamennyi földrészre létrehozunk egy lekérdezést. Ezt elkerülhetjük paraméteres lekérdezés alkalmazásával. Paraméteres lekérdezést **Tervező** nézetben hozhatunk létre a következőképpen:

1. Tegyük aktuálissá annak a mezőnek a **Feltétel** sorát, amelyre a paramétert alkalmazni szeretnénk.
2. Szögletes zárójelekbe írjuk be azt a szöveget, amit szeretnénk megjeleníteni a paraméter párbeszédablakban, például a következőt: *[A földrész neve].*
3. Szükség esetén a fentiek szerint helyezzük el a megfelelő szövegeket más mezők feltétel soraiban.
4. Kattintsunk a **Futtatás** gombra !.

Ellenőrző kérdések

1°. Mi a lekérdezések rendeltetése?
2°. Melyek a lekérdezésekben alkalmazható leggyakoribb műveletek?
3°. Ismertessük az egyes lekérdezés-típusok rendeltetését!
4°. Hozzunk fel példákat választó lekérdezésekre!
5°. Miben különbözik a választó lekérdezés a kereszttáblástól?
6*. Hasonlítsuk össze a paraméteres lekérdezést az internetes kereséssel! Mi bennük a közös és miben térnek el egymástól?
7°. Mi célt szolgálnak a lekérdezésben a kifejezések?
8°. Mit tartalmazhatnak a kifejezések?
9°. Milyen műveleteket alkalmazhatunk a lekérdezések létrehozása során?
10°. Ismertessük a lekérdezés létrehozásának menetét, ha azt a **Lekérdezés varázslóval** hozzuk létre.
11°. Mi a választó lekérdezés eredménye?
12°. Ismertessük a lekérdezés **Tervezőablakának** szerkezetét! Magyarázzuk el az egyes objektumok rendeltetését!
13° Ismertessük a számított mezőt tartalmazó lekérdezés létrehozásának menetét!
14°. Mi célt szolgálnak a paraméteres lekérdezések?

Végezzük el a feladatokat!

1°. Állítsuk össze a lekérdezések osztályozását!
2°. Nyissunk meg egy adatbázisfájlt, például a **3.8.2.accdb**-t! Az **Egyszerű lekérdezés varázsló** segítségével hozzunk létre lekérdezést a **Földrajzi és történelmi térképek** tábla alapján! A lekérdezéshez adjuk hozzá az **Ország, Régió, Térkép típusa, Méret** és **Ár** mezőket. Váltsunk **Tervező** nézetbe és állítsuk be a következő szűrőfeltételeket:

a **Térkép típusa** – *Politikai*, **Ár** – *kevesebb, mint 45 hrivnya.* Mentsük az adatbázist egy másik mappába **3.8.2.accdb** néven!

3°. Nyissunk meg egy adatbázisfájlt, például a **3.8.3.accdb**-t! A **Egyszerű lekérdezés varázsló** segítségével hozzunk létre lekérdezést a **Földrajzi és történelmi térképek** tábla alapján! A lekérdezéshez adjuk hozzá az **Ország, Régió, Térkép tipusa, Külalak, Ár.** Váltsunk **Tervező** nézetbe és állítsuk be a következő szűrőfeltételeket: a **Térkép típusa**– *illusztrált vagy általános,* **Ár** – *10 és 35 hrivnya között.* Mentsük az adatbázist egy másik mappába **3.8.3.accdb** néven!

4°. Nyissunk meg egy adatbázisfájlt, például a **3.8.4.accdb**-t! Az **Egyszerű lekérdezés varázsló** segítségével hozzunk létre lekérdezést az **Országok** tábla alapján. A lekérdezéshez adjuk hozzá a **Név, Földrész, Létrehozás dátuma, Államforma, Zászló, Himnusz** mezőket. Váltsunk **Tervező** nézetbe és állítsuk be a következő szűrőfeltételeket: **Létrehozás dátuma** – *1990 után,* **Államforma** – *parlamentáris köztársaság.* Mentsük az adatbázist egy másik mappába **3.8.4.accdb** néven!

5°. Nyissunk meg egy adatbázisfájlt, például a **3.8.5.accdb**-t! Az **Egyszerű lekérdezés varázsló** segítségével hozzunk létre lekérdezést az **Áru behozatala** tábla alapján. A lekérdezéshez adjuk hozzá az **Áru típusa, Áru neve, Ár, Mennyiség** mezőket. Bővítsük a lekérdezést **Tervező** nézetben az **Összeg** számított mezővel, amelyben számítsuk ki az **Ár** és a **Mennyiség** mezők szorzatát! Mentsük az adatbázist egy másik mappába **3.8.5.accdb** néven!

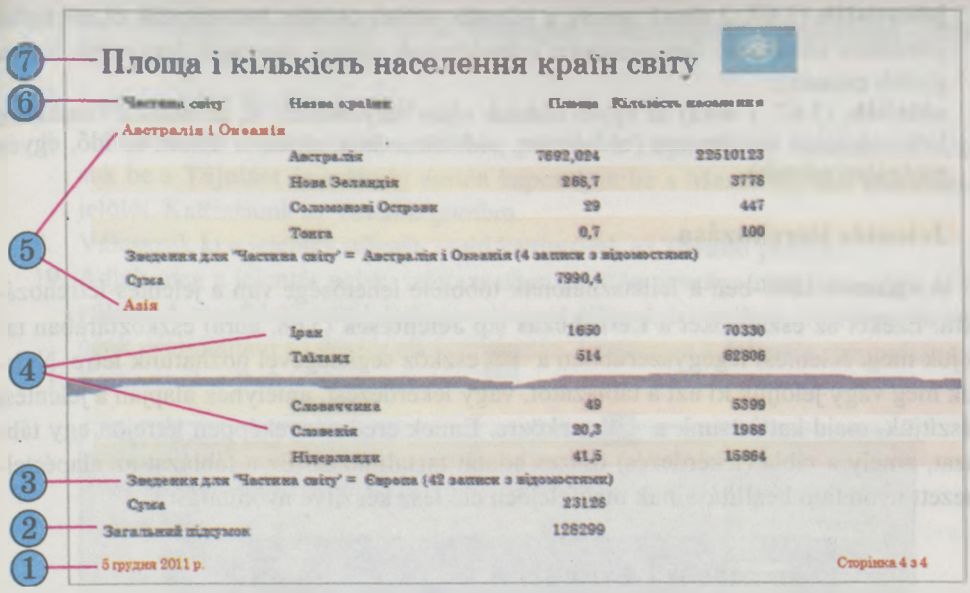
6°. Nyissunk meg egy adatbázisfájlt, például a **3.8.6.accdb**-t! Az **Egyszerű lekérdezés varázsló** segítségével hozzunk létre lekérdezést az **Elektronikus oktatási eszközök** tábla alapján! A lekérdezéshez adjuk hozzá a **Név, Program típusa, Ár, Mennyiség,** az **Oktatási Minisztérium jóváhagyása** mezőket. Bővítsük a lekérdezést **Tervező** nézetben az **Összeg** számított mezővel, amelyben számítsuk ki az **Ár** és a **Mennyiség** mezők szorzatát! A lekérdezés eredményhalmazába csak az oktatási minisztérium által jóváhagyott eszközöket foglaljuk bele. Mentsük az adatbázist egy másik mappába **3.8.6.accdb** néven!

3.9. Jelentések. Jelentések létrehozása Access 2007-ben

1. Hogyan nyomtathatunk dokumentumokat **Word 2007**-ben és **Excel 2007**-ben?
2. Ismertessük a lekérdezés létrehozásának menetét az **Access 2007**-ben!
3. Hogyan hozhatunk létre számított mezőket? Milyen mezők szolgálhatnak a számított mezők alapjául?
4. Ismertessük, milyen szerkesztési műveleteket végezhetünk a lekérdezésekkel az **Access 2007 Tervező** nézetében!

Jelentések szerepe az adatbázisban

Az adatbázisokban a jelentés az adatfeldolgozás eredményének nyomtatására szolgál. A jelentés egy nyomtatásra előkészített elektronikus dokumentum. A jelentés egy kitöltött formanyomtatvány, például az érettségi bizonyítvány pótlapja, a független értékelés eredményeit tartalmazó dokumentum. A jelentés elkészítéséhez felhasználhatjuk a táblák és lekérdezések adatait, azokon csoportosításokat, összegzéseket, rendezéseket végezhetünk, számított mezőket helyezhetünk el. A jelentés az adatokat magyarázó feliratokat, képeket, diagramokat is tartalmazhat.



3.67. ábra. A jelentés legfontosabb objektumai az **Access 2007**-ben

A jelentés elkészítése során ki kell alakítani annak, mint az elektronikus dokumentumnak a megjelenését és meg kell határozni az adatoknak forrását, amelyet be szeretnénk foglalni a jelentésbe.

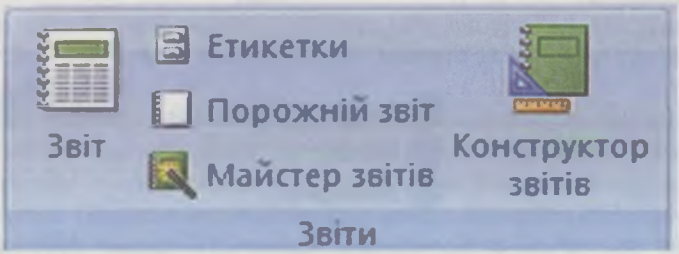
Vizsgáljuk meg a jelentés szerkezetét a **Világ országainak lakossága és területe** jelentés példáján (3.67. ábra), amelyet a **Világ országai** adattábla alapján készítettünk el. E jelentés szakaszai a következők:

- a **jelentésfej** (3.67. 7 ábra), amely a jelentés címét tartalmazza, ezenkívül grafikai objektumokat (logo, embléma, illusztráció), az aktuális időt és dátumot, és vezérlőelemeket helyezhetünk el rajta. A jelentésfej csak a jelentés elején található meg, a jelentésben többször nem ismétlődik;
- az **oldalfej** (3.67. 6 ábra), amely a jelentés oszlopainak neveit tartalmazza. Az oldalfej a **Word 2007** fejléceihez hasonlóan minden oldal tetején megjelenik;
- **csoportfej** (3.67. 5 ábra), amely az objektumcsoportok nevét tartalmazza. A 3.67. ábrán például minden földrész országainak adatai előtt szerepel a földrész neve. A csoportra vonatkozó összesítéseket és vezérlőelemeket tartalmazhatnak;
- **törzs** (3.67. 4 ábra), amely a jelentésben elhelyezett adatokat tartalmazza;
- **csoportláb** (3.67. 3 ábra), amely a csoportra vonatkozó összesítéseket (objektumok száma, egyes mezők összegei stb.) tartalmazza;

- **jelentésláb** (3.67. 2 ábra), amely a jelentés utolsó oldalán helyezkedik el, az egész jelentésre vonatkozó összegzéseket (objektumok száma, összegek, legkisebb, legnagyobb elemek);
- **oldalláb,** (3.67. 1 ábra) az egyes oldalak alján helyezkedik el, az oldalra vonatkozó információkat tartalmazza (oldalszám, oldalak száma, aktuális dátum és idő, egyéb szolgálati adatok).

Jelentés létrehozása

Az **Access 2007**-ben a felhasználónak többféle lehetősége van a jelentés létrehozására. Ezeket az eszközöket a **Létrehozás** lap **Jelentések** (3.68. ábra) eszköztárában találjuk meg. Jelentést legegyszerűbben a eszköz segítségével hozhatunk létre. Nyissuk meg vagy jelöljük ki azt a táblázatot, vagy lekérdezést, amelynek alapján a jelentést készítjük, majd kattintsunk a eszközre. Ennek eredményeképpen létrejön egy táblázat, amely a tábla (lekérdezés) összes adatát tartalmazza. Ez a táblázat az alapértelmezett nyomtató beállításainak megfelelően elő lesz készítve nyomtatásra.



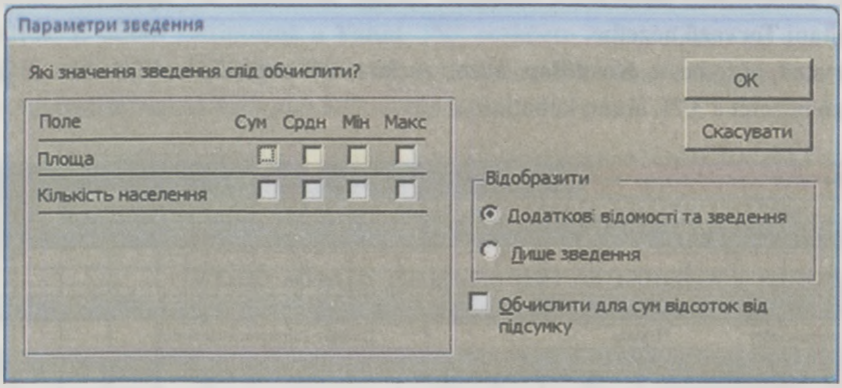
3.68. ábra

A fent leírtaknál lényegesen több lehetőséget biztosít a jelentés kialakításában a **Jelentés varázsló.** Vizsgáljuk meg a jelentés kialakításának mentét a 3.67. ábra alapján. Az eszköztár azonos nevű gombjára kattintva indítsuk el a varázslót. A továbbiakban kövessük a varázsló lépéseit.

1. A **Táblák és lekérdezések** listából válasszuk ki azt az objektumot, amelynek alapján a jelentést készítjük, jelen esetben a **Világ országai** táblát. Az **Elérhető mezők** listájából helyezzük át a kiválasztott mezők listájába a szükséges mezőket: **Név, Földrész, Terület, Lakosság száma.**
2. Szükség esetén válasszunk ki egy másik táblát vagy lekérdezést, és ismételjük meg az előbbi műveletet, majd kattintsunk a **Tovább** gombra.
3. Válasszuk ki a csoportszinteket, jelen esetben a **Földrész** mezőt, majd kattintsunk a **Tovább** gombra.
4. Szükség esetén állítsuk be a rendezést (a csoporton belül).
5. Az **Összesítés** gombra kattintva a megfelelő jelölőket bekapcsolva állítsuk be a szükséges összegzési beállításokat. Állítsunk be összesítést a **Terület** mezőre.

   Figyeljük meg, hogy ha **Kiegészítő összegzési beállítások** területen (3.69. ábra) a **Részletek** és **Összesítés** választógombot kapcsoljuk be, akkor a jelentés a kiválasztott adatokat és az összesített adatokat egyaránt tartalmazza majd, ha pedig a csak **Összegzés** választógombot, akkor csak az összesített adatokat.

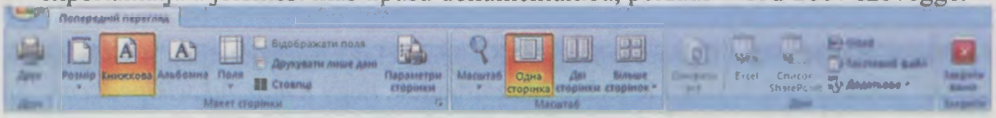
6. Ha szükséges, kapcsoljuk be az **Összegek százalékának kiszámítása** jelölőt. Kattintsunk az **OK** gombra.
7. Kattintsunk a **Tovább** gombra.
8. Válasszuk ki a jelentés elrendezését, például a **Lépcsőzetes** elrendezést. Állítsuk be a **Tájolást** és szükség esetén kapcsoljuk be a **Mezők laphoz méretezése** jelölőt. Kattintsunk az **Tovább** gombra.
9. Válasszuk ki a jelentés stílusát, majd kattintsunk az **Tovább** gombra.
10. Adjuk meg a jelentés nevét, jelen esetben a **Világ országainak lakossága és területe.** A megfelelő jelölő bekapcsolásával állítsuk be, milyen műveletet szeretnénk végrehajtani a jelentés elkészülte után. Legyen ez a **Jelentés megnyitása.**
11. Kattintsunk a **Befejezés** gombra



3.69. ábra. Az **Összegzés beállításai** párbeszédablak

Amikor a jelentés a fenti eljárás szerint elkészült, a képernyőn megjelenik annak nyomtatási képe. **Szalagon** (3.70. ábra) megnyílik a **Nyomtatási kép** ideiglenes eszköztár. Ezeknek az eszközöknek a segítségével a felhasználó:

- megváltoztathatja az oldal méretét, tájolását, a margókat és a hasábok számát;
- megváltoztathatja a méretarányt;
- exportálhatja a jelentést más típusú dokumentumba, például **Word 2007** szöveggé.



3.70. ábra. A **Nyomtatási kép** ideiglenes eszköztár

Amennyiben a jelentésen csak olyan változtatásokat kell elvégezni, amelyeket a **Nyomtatási kép** eszköztár segítségével lehet, a jelentést azonnal ki is nyomtathatjuk. Ezt a **Nyomtatás** gombra kattintva tehetjük meg. Ha a jelentésen egyéb változtatásokat is el kell végezni, akkor a megfelelő gombra kattintva zárjuk be a nyomtatási képet.

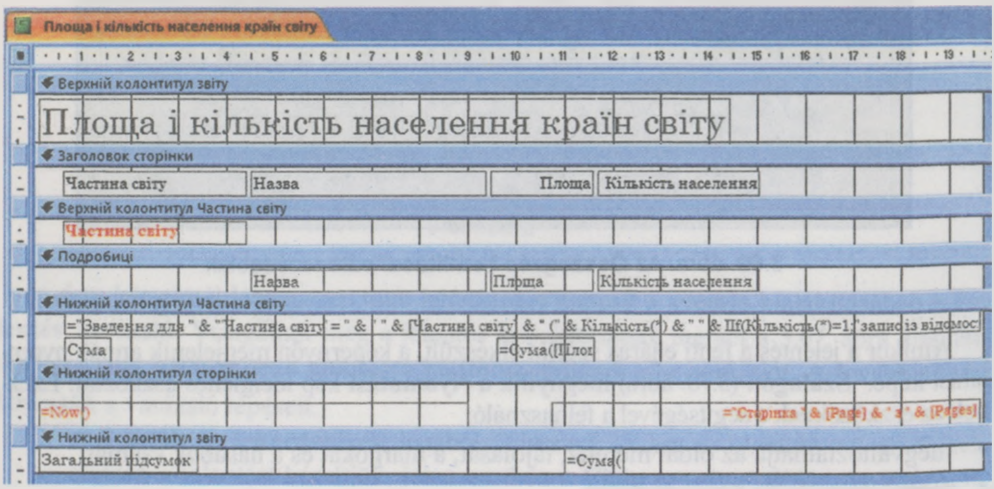
Jelentések szerkesztése és formázása

Az **Access 2007**-ben a jelentések szerkesztését **Elrendezés** nézetben vagy **Tervező** nézetben végezhetjük.

Mindkét nézetbe a **Kezdőlap** nézet eszközének segítségével juthatunk el.

Ha meg szeretnénk változtatni egy oszlop nevét, kattintsunk duplán a megfelelő mezőre és végezzük el a szükséges változtatásokat (a **„Név"** helyett beírjuk az **„Ország neve"** szöveget). Ugyanilyen módon írhatjuk hozzá a **Terület** és a **Lakosság** mezőkhöz a mértékegységeket. A mezőnevek ilyen megváltoztatása több oldalas jelentés esetében nem célszerű. Ez esetben hasznos lenne a mértékegységeket a számértékek mellett elhelyezni. Ezt a változtatást **Elrendezés** nézetben nem végezhetjük el. A csoportláb szerkesztése elrendezés nézetben szintén nehézkes. Az ilyen változatáshoz át kell váltani **Tervező** nézetbe.

Tervező nézetbe a **Kezdőlap** nézet eszközének segítségével jutunk. A jelentés **Tervező** nézetét a 3.71. ábrán láthatjuk.



3.71. ábra. A jelentés **Tervező** nézete

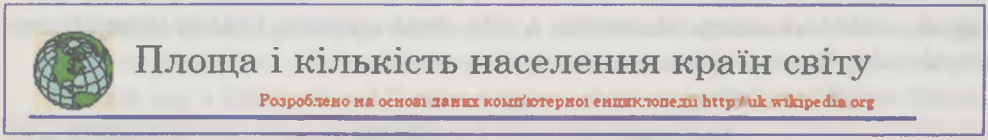
Tervező nézetben a fejlécet szöveggel és képekkel is kiegészíthetjük. *A világ országainak területe és lakossága* fejléchez hozzáírhatjuk, hogy a *http://uk.wikipedia.org internetes enciklopédia adatai alapján.* Ezt a következőképpen érhetjük el:

1. A határvonalakat áthúzva növeljük meg a fejléc méretét.
2. Kattintsunk a **Jelentéstervező** eszköztár **Vezérlőelemek** területén a **Címke** gombra.
3. Helyezzük el a fejléc területén a címkét.
4. Írjuk be a címkébe a szöveget.
5. A **Jelentés** nézetre váltva nézzük meg az eredményt.

Grafikus elemet többféleképpen is hozzáadhatunk a fejléchez. Használhatunk **Logo, Kép, Szabad objektumkeret** és **Kötött objektumkeret** eszközöket. Lássuk, hogyan járunk el **Logo** objektum esetében:

1. Kattintsunk a **Logo** eszközre .
2. A **Kép beillesztése** ablakban válasszuk ki a fejlécbe illesztendő képet, például a **Globe.wmf**-et, majd nyomjunk **OK**-t.
3. A kép és fejléc kereteinek mozgatásával alakítsuk ki a kép elhelyezését a fejlécen.
4. A **Jelentés** nézetre váltva nézzük meg az eredményt.

A többi eszköz használata lényegében nem tér el a fent ismertetettől, de a kép beillesztése esetében meg kell adni a kép helyét és az objektumok beillesztése esetében előbb ki kell választani az objektum típusát. A fejléc részeit vízszintes vonallal is elválaszthatjuk. E célból kattintsunk a **Vonal** eszközre és rajzoljuk meg a vonalat. A megfelelő vezérlőelemekkel beállíthatjuk a vonaltípust, a vonalvastagságot és a vonal színét. Az átszerkesztett fejlécet a 3.72. ábrán láthatjuk.



3.72. ábra. A fejléc szerkesztés után

A láblécbe hasonlóképpen helyezünk el szöveges és grafikus objektumokat. Ha a fejlécbe vagy láblécbe oldalszámot vagy csoportnevet helyezünk el, akkor azok minden oldalon, illetve a csoport elején vagy végén ismétlődni fognak.

A **Jelentéstörzset** (3.67. 4 ábra) **Tervező** nézetben **Részleteknek** nevezik (3.71. ábra). Ha ebben a részben helyezünk el objektumokat, akkor azok a jelentés minden sorában ismétlődni fognak. Ha tehát a **Terület** és a **Lakosság** mezőbe beírjuk a megfelelő mértékegységeket (*a négyzetkilométert* és *főt*) (3.73. *a* ábra), akkor ezek minden sorban ismétlődni fognak (3.73. *b* ábra).

a)

Частина світу	Назва країни	Площа	Кількість населення
Австралія і Океанія			
	Австралія	7692,024 кв. км	22510123 осіб
	Нова Зеландія	268680 кв. км	4377000 осіб
	Соломонові Острови	28896 кв. км	478000 осіб
	Тонга	747 кв. км	120898 осіб

b)

3.73. ábra. Számadatok melletti beírások a jelentésrészletben

Képleteket **számított mezők** segítségével helyeznek el az adatbázis-objektumokban. Helyezzünk el a jelentésben egy ilyen elemet abból a célból, hogy meghatározzuk az egyes csoportokba tartozó elemek számát. Töröljük **Tervező** nézetben a csoportfejből a fentnevezett objektumokat, majd hajtsuk végre a következőket:

1. Kattintsunk a ***Jelentéstervező*** eszköztár ***Vezérlőelemei*** között a ***Számított*** mező elemre.
2. Helyezzük el a láblécben a mezőt.
3. A keret átméretezésével állítsuk be a mező szélességét és magasságát.
4. Írjuk be a mező bal oldali részébe a megfelelő szöveget, például *„Az országok száma"*.
5. Írjuk be a mező jobb oldali részébe az *=count([név])* képletet.
6. Végezzük el a szöveg formázását az alkalmazott stílusnak megfelelően.
7. A **Jelentés** nézetre váltva nézzük meg az eredményt.

A beírt képlet azt jelenti, hogy meg kell számolni a **Név** objektumokat az országok minden csoportjára vonatkozóan. Ugyanebben a láblécben helyezzünk el egy, a területekre vonatkozó *Összegző* (sum) függvényt. A program automatikusan felajánlja *Az országok területének összege* elnevezést. A 3.74. ábrán a jelentés láblécét láthatjuk szerkesztés után.

	Камбоджа	181040 кв. км	14805000 осіб
	Узбекистан	447400 кв. км	28095900 осіб
Кількість країн	35		
Загальна площа країн		27706588	

3.74. ábra. A lábléc szerkesztés után

A jelentés szerkesztése leggyakrabban **Elrendezés** nézetben történik. Ekkor a **Formátum, Rendezés, Oldalbeállítás** és a **Jelentés** eszköztárak állnak rendelkezésünkre. Ezek az eszköztárak a korábban már megismert eszközöket bocsátják a felhasználó rendelkezésére.

A jelentések mentése a létrehozáskor automatikusan megtörténik. Szerkesztés után a jelentést az **Access 2007** szabványos eszközeivel végezhetjük el. A jelentés alapját képező táblák és lekérdezések változásai a jelentés megnyitásakor automatikusan bekerülnek a jelentésbe.

Ellenőrző kérdések

1°. Mi a jelentések szerepe az adatbázisokban?
2°. Hozzunk fel példákat a jelentések alkalmazására!
3°. Ismertessük a jelentés szerkezetét és az egyes elemek rendeltetését!
4°. Hogy hozunk létre jelentést a **Jelentés** eszköz alkalmazásával?
5°. Milyen kiegészítő lehetőségeket biztosít a felhasználó számára a **Jelentés varázsló**?
6°. Milyen lépésekből áll a jelentés létrehozása a **Varázsló** segítségével?

7°. Mire használjuk a jelentés létrehozása során az **Összegzési beállításokat**?

8°. Milyen, a nyomtatás előkészítésére szolgáló eszközök állnak rendelkezésünkre **Nyomtatási kép** nézetben?

9°. Milyen esetekben van szükség a jelentés formázására?

10°. Milyen objektumokat szerkeszthetünk és formázhatunk a fejlécben? Hogyan?

11°. Ismertessük az oldalfej és a csoportfej szerkesztésének sajátosságait!

12°. Ismertessük az oldalláb és a csoportláb szerkesztésének sajátosságait! Általában milyen objektumok helyezkednek el a jelentés ezen részeiben?

13°. Hogy helyezhetünk el grafikus objektumokat a jelentésben? A jelentés melyik részében helyezhetjük el ezeket?

14°. Milyen elemeket tartalmaz a jelentéstörzs? Mivel magyarázhatjuk ezt?

15°. Milyen célból helyeznek el számított mezőket a jelentésekben? A jelentés mely részében helyezzük el ezeket?

Végezzük el a feladatokat!

1°. Nyissunk meg egy adatbázisfájlt, például a **3.9.1.accdb**-t. A **Jelentés** eszköz segítségével hozzunk létre jelentést a **Földrajzi és történelmi térképek** tábla alapján. Formázzuk meg a lekérdezést a 3.75. ábra alapján. A logót az **optdisk.wmf** fájlban találjuk. Mentsük az adatbázist a mappánkba **3.9.1.accdb** néven.

Географічні та історичні карти

Вид карти	Розмір	Масштаб	Вид оформлення	Ціна
Автономна Республіка Крим				
Адміністративна	42 x 30 см	М1:850 000	ламінація	15,00 грн.
Сума				грн.
Європа				
Політична	160 x110 см	М1:3 750 000	ламінація	90,00 грн.
Політична	160 x110 см	М1:3 750 000	папір	24,00 грн.
Політична	110×77 см	М1:5 400 000	ламінація з планками	72,00 грн.
Політична	110×77 см	М1:5 400 000	ламінація	60,00 грн.
Фізична	65 x 45 см	М1:11 000 000	папір	4,50 грн.
Політична	160 x110 см	М1:3 750 000	ламінація з планками	120,00 грн.
Сума				505,50 грн.
Молдова				
Політична	100 x 68 см	М1:35 000	папір	1,00 грн.

3.75. ábra

2°. Nyissunk meg egy adatbázisfájlt, például a **3.9.2.accdb**-t. A **Jelentés** varázsló segítségével hozzuk létre a **Világ országainak lakossága és területe** jelentést a paragrafus

szövege alapján. Végezzük el a jelentés szerkesztését elrendezés és szerkesztő módban. A logót a globe 1. wmf fájlban találjuk. Mentsük az adatbázist a saját mappánkba **3.9.2.accdb** néven.

3*. Nyissunk meg egy adatbázisfájlt, például a **3.9.3.accdb**-t. A jelentés varázsló segítségével hozzunk létre jelentést az áruk bevételezése tábla alapján. Végezzük el a jelentés szerkesztését és formázását a 3.76 tábla alapján. A logót a **wrldmap.wmf** fájlban találjuk. Mentsük az adatbázist **3.9.3.accdb** néven a mappánkba.

Отримання товару

Магазин "Бім"

12 грудня 2010 р.
23:27:28

Номер накладної	Дата отримання	Назва товару	Тип товару	Кількість	Ціна
123	11.11.2011	Комп'ютер	Настільний	4	4 600,00 грн.
123	11.11.2011	Монітор	19"	6	1 567,00 грн.
123	11.11.2011	Сканер	Планшетний	3	437,00 грн.
124	21.11.2011	Комп'ютер	Ноутбук	10	3 656,00 грн.
124	21.11.2011	Фотоапарат	Цифровий	15	1 273,00 грн.
125	23.11.2021	Комп'ютер	Настільний	12	4 398,00 грн.
125	23.11.2011	Принтер	Лазерний	13	954,00 грн.
125	23.11.2011	Принтер	Струменевий	5	673,00 грн.
126	27.11.2011	Комп'ютер	Нетбук	14	3 100,00 грн.
126	27.11.2011	Монітор	22"	7	1 895,00 грн.
123	11.11.2011	Комп'ютер	Ноутбук	5	2 910,00 грн.

Сторінка 1 з 1

3.76. ábra.

14. számú gyakorlati munka Lekérdezések és jelentések létrehozása

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Nyissuk meg a **14.accdb** fájlt.
2. Készítsünk lekérdezést az **Ételek energiatartalma** táblázat alapján. A lekérdezés tartalmazza az **Étel, Kalória, Menü része** mezőket.
3. Tervező nézetben egészítsük ki a **Menü része** mezőt egy szűrőfeltétellel, amely kizárja az eredményhalmazból azokat az ételeket, amelyek nem részei a menünek.
4. A lekérdezés alapján hozzuk létre a 3.77. ábrán látható jelentést.
5. A jelentésfejléc tartalmazza az aktuális dátumot és időt.
6. Végezzünk csoportosítást a **Menü része** mező értékei alapján.
7. Végezzük el csoportonként a **Kalória** mező értékeinek összegzését.
8. Végezzük el a jelentés szerkesztését és formázását.
9. Zárjuk be a lekérdezést és a jelentést.
10. Mentsük az adatbázist **gyakorlati_14** néven a mappánkba.

Меню їдальні на 13 грудня 2011 р.

Страви	Калорійність	Маса
Сніданок		
Кава з молоком	150 ккал	200 г
Млинці із сиром і сметаною	650 ккал	209 г
Вінегрет	60 ккал	100 г
Загальна калорійність страв	860 ккал	
Обід		
Соки ягідні, фруктові, томатні	160 ккал	200 г
Картопля варена	250 ккал	209 г
Риба тушкована з овочами	220 ккал	200 г
Борщ, овочевий суп	46 ккал	100 г
Салат зелений с огірками, помідорами	100 ккал	150 г
Загальна калорійність страв	776 ккал	
Вечеря		
Макарони відварені	330 ккал	209 г
Курка відварена	130 ккал	100 г
Ікра баклажанна	175 ккал	100 г
Чай з цукром	50 ккал	200 г
Загальна калорійність страв	685 ккал	

13 грудня 2011 р. Сторінка 1 з 1

3.77. ábra

4. fejezet

A SZEMÉLYES ÉS CSOPORTOS KOMMUNIKÁCIÓ INFORMÁCIÓS TECHNOLÓGIÁI

Ebben a fejezetben megismerkedünk:

- a multimédiás adatokkal és az azok feldolgozására szolgáló programokkal;
- az audio- és videoklipek létrehozásával;
- a számítógépes kiadványokkal és azok csoportosításával;
- a számítógépes kiadványok objektumainak szerkesztésével;
- a weblapok típusaival, szerkezetével és létrehozásával;
- a weblapok létrehozását segítő szoftverek működésével;
- a webdesign alapjaival;
- a **Web 2.0** technológia szolgáltatásaival és azok válfajaival;
- a blogok létrehozásával, kialakításával, a hozzászólások kezelésével;
- a webes közösségek működésének sajátosságaival;
- az elektronikus dokumentumok integrált kezelésével;
- a dokumentumok megosztott kezelésével.

4.1. A multimédiás adatok fogalma és feldolgozás

1. Nevezzünk meg multimédiás adatok lejátszására szolgáló eszközöket.
2. Mi a fájlformátum? Milyen fájlformátumokat ismerünk?
3. Mi a tömörítés? Milyen fájlformátumok tartalmaznak tömörített adatokat?
4. Az optikai lemezek milyen típusai ismertek? Miben különböznek ezek egymástól?
5. Mi a fájlformátum-konverzió?

A multimédiás adatok fogalma

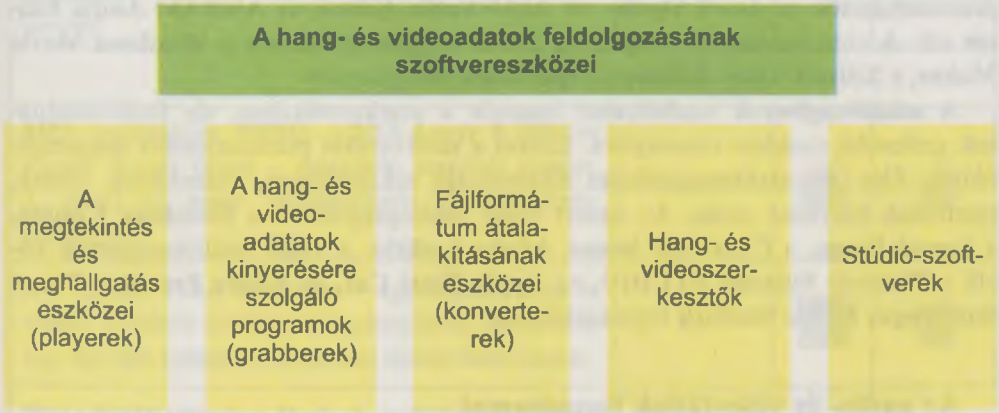
Ahogy azt a 9. osztályban megtanultuk, a multimédiás adatok lejátszásához speciális hardvereszközökre van szükség. Ezek az eszközök a hangszórók, a mikrofonok, a fejhallgatók, projektorok, digitális táblák. Most azonban eljött az ideje annak, hogy pontosítsuk, mit értünk a **multimédiás adatok** alatt.

Multimédiás adatok **alatt olyan adatokat értünk, amelyek érzékelésére több emberi érzékszerv egyidejű használatára van szükség.**

A multimédiás adatok példája a számítógépes prezentáció. A prezentáció szöveget, képeket, audio- és video-adatokat tartalmazhat. Ezért az ilyen prezentációkat gyakran nevezik multimédiásnak. Amikor filmet nézünk, színházi előadáson vagy koncerten vagyunk, multimédiás adatokat érzékelünk.

A multimédiás adatok feldolgozásának szoftvereszközei

A multimédiás adatok feldolgozásának igen sok szoftveres eszköze létezik. Ezek némelyikével már korábban, a 9. és 10. osztályban megismerkedtünk. A **Word 2007** szövegszerkesztőben elkészített szövegek tartalmazhatnak grafikai adatokat, beágyazott audio- és videoadatokat is. Amikor azonban a multimédiás adatok feldolgozásának szoftvereszközeiről beszélünk, leginkább a hang- és videoadatok feldolgozására szolgáló programokra gondolunk. Ezeknek a programoknak csoportosítását tartalmazza a 4.1. ábra.



4.1. ábra. Audio- és videofeldolgozó szoftverek

A megtekintés és meghallgatás szoftvereszközei a multimédiás adatok lejátszásra szolgálnak. Ezeket a programokat **multimédiás lejátszóknak,** rövidebben **médiaplayereknek** nevezzük. Több különböző elterjedt lejátszó-programot ismerünk. Ezek némelyike mind hang- mind pedig videó lejátszására alkalmas, de akadnak specializáltak is (ezeket természetesen **audiolejátszónak** vagy **videolejátszónak** nevezik). A legelterjedtebb multimédialejátszók a **Winamp** (*Windows Advanced Media Player* – **Windows** fejlett médialejátszó), a

Media Player Classic, a QuickTime Player, az Flv Player stb.

Előfordul, hogy a számítógéphez kiegészítő multimédiás eszközöket – mikrofon, web-kamera, TV-tuner, DVD-lejátszó, videomagnó – kapcsolnak. Ezek az eszközök általában az adatbevitelt szolgálják, az adatok fájlokban történő tárolására nem alkalmasak.

A hang- és videofájlok mentését speciális szoftvereszközök, az úgynevezett **grabberek** (*grabber* – az, ami elfog, elfogó) teszik lehetővé. A video lejátszása so-

rán a grabber képes azt átkódolni és fájlba menteni. Az hangok grabbelésére szolgál a **Windows hangrögzítő**, az **Exact Audio Copy**, az **Audio Grabber 3.0**, a **Streamripper**, az **AML Easy Audio Recorder** stb. Az ismertebb videograbberek az **Extra Screen Capture Pro**, a **Wondershare Streaming Video Recorder**, a **Wondershare P2P TV Recorder** stb.

Az audio- és videoformátumok nagy száma miatt a felhasználóknak szüksége lehet olyan szoftverekre, amelyek lehetővé teszik ezek átalakítását. Erre a célra szolgálnak a *konverterek*, például az **Easy Video Converter**, a **MediaCoder**, a **Xilisoft Video Converter**, a **Flash to Video Encoder**, a **Movavi VideoSuite**, a **Direct MIDI to MP3 Converter**, az **AVS Audio Converter.**

Az ***audio-*** és ***videoszerkesztők*** segítségével zenét, hangoskönyvet, hang- és videohatásokat, zenei aláfestést, videofilmeket hozhatunk létre és szerkeszthetünk. Ezekben a programokban lehetőség van az audio- és videorészletek kivágására, másolására, beillesztésére, egyesítésére, azok hosszúságának megváltoztatására. Beilleszthetünk feliratokat, képeket, szinkronizálhatjuk az audio- és videofolyamokat. Ismert zeneszerkesztők az **Anvil Studio**, az **ALO Audio Editor**, az **AKRAM Audio Editor** stb. A videoszerkesztő programok közül a legelterjedtebbek a **Windows Movie Maker**, a **Xilisoft Video Editor**, a **VideoMach** stb.

A ***stúdiószoftverek*** rendeltetése hasonlít a szerkesztőkéhez, de funkcionalitásuk szélesebb azokhoz viszonyítva. Ezeket a szoftvereket professzionális hangstúdiókban, film- és rajzfilmstúdiókban alkalmazzák reklámfilmek, videoklipek, filmek, rajzfilmek készítése során. Az ismert zenei stúdióprogramok a **Steinberg Cubase**, a **Sound Forge**, a **Cakewalk Sonar**, **Adobe Auditio.** A videó stúdióprogramok közül a **Pinnacle Systems STUDIO**, az **Apple Final Cut**, az **Adobe Premiere Pro**, a **SonyVegas Movie Studio** a legismertebbek.

Az audio- és videofájlok formátumai

A 9. osztályban megismerkedtünk a fájlformátum fogalmával, A fájlformátum határozza meg az adatok struktúráját a fájlban. Már megismertünk néhány szöveges és grafikus formátumot. A multimédiás adatoknak igen sokféle formátuma létezik.

A multimédiás fájlformátumok a 4.1. táblázatban vannak összefoglalva.

A 9. osztályban megismerkedtünk az optikai tárolóeszközökkel. Ezeket ma már leginkább adattárolás céljára használjuk, bár a **Philips** és a **Sony** által zeneszámok tárolására voltak kifejlesztve. A tárolás formátuma 1979-ben lett definiálva az úgynevezett red book-ban (*vörös könyvben*). Az ilyen formátumú lemezeket **Audio CD**-nek (audio kompakt lemez) nevezik. A hangadatok úgynevezett sávokba (*track*) kerülnek a lemezen. Egy sáv általában egy zenei részletet (zeneszámot) tartalmaz.

A zenei CD lejátszásához speciális eszközre, CD-lejátszóra van szükség. Később kifejlesztették a videók tárolására alkalmas eszközt és fájlformátumot, a **Video CD**-t (*Video Compact Disk* – video kompakt lemez), majd, a **DVD**-k megjelenésével az audio- és videoadatok tárolására egyaránt alkalmas, ezekhez a lemezekhez alkalmazgató formátumot is.

4.1. táblázat. **Multimédiás fájlok formátumai**

A fájlformátum leírása	*Kiterjesztés*	*Ikon*	
		XP	*Vista*
Tömörítésmentes vagy veszteségmentesen tömörített formátumok			
WAV vagy WAVE (*wavefront audio format* – hullámfront hangformátum) – a **Windows** operációs rendszer alapvető tömörítésmentes formátuma	wav		
MIDI (*Musical Instrument Digital Interface* – zenei eszközök digitális illesztőfelülete) – olyan formátum, amely egy speciális eszköz vagy program – midi **sequencer** – számára tartalmaz utasításokat (*sequentia* – sorozat)	mid, midi		
Veszteségesen tömörített formátumok			
MP3, pontosabban **MPEG -1/2/3,5 Layer 3** (*Motion Picture Expert Group* – mozgókép szakértői csoport, 3. réteg) az egyik legelterjedtebb audioformátum. Fájlcserélő	mp3		
AVI (*Audio Video Interleave* – egybeépített hang és video) –a videók tárolásának legelterjedtebb formátuma. Egy fájl több különböző tömörítési eljárást tartalmazhat	avi		
WMA/WMV (*Windows Media Audio/Video*) a **Windows** operációs rendszer **Microsoft** által kifejlesztett audio/video formátum	wma, wmv		
3GP (*3rd Generation Phone* – harmadik generációs telefon) a harmadik generációs telefonok hálózataiban alkalmazott formátum. A nagy tömörítési arány kis fájlméretet és gyenge hangminőséget jelent.	3gp		
QuickTime (*Quick Time* – gyors idő) – Az **Apple** által kifejlesztett technológián alapuló video és audioformátum	mov		
Flash video – az **Adobe Flash** által kifejlesztett formátum, jellemzője a nagy tömörítési arány.	flv		

Annak ellenére, hogy a felsorolt formátumokat speciális lejátszókra fejlesztették ki, a mai számítógépek nagy többsége alkalmas az audio és videolemezek lejátszására. Az optikai tárolóeszközök formátumait a 4.2. táblázatban foglaltuk össze.

4.2. táblázat. **A multimédiás optikai tárolóeszközök**

A lemezformátum leírása	*Jel*
Audio CD – a zenei CD-k formátuma. Egy sáv minimális hossza 2 s, a sávok maximális száma 99. A lemez maximális lejátszási ideje 74 perc.	
Video CD – a CD-n tárolt videók formátuma. Egy lemez maximálisan 74 perc gyenge minőségű videót tartalmazhat.	
DVD Audio – DVD-n tárolt hangformátum. Az ilyen formátumú hang minősége a több alkalmazott csatorna miatt nagyon jó.	
DVD-Video – a DVD-n tárolt videók formátuma. Egy lemez maximálisan 120 perc jó minőségű videót tartalmazhat.	

Mivel az audio- és videofájlok tömörítése különböző algoritmusok szerint történhet, azok lejátszásához speciális szoftverek, kódoló/dekódoló programok, röviden kodekek szükségesek. A kodekek leginkább a lejátszók részei, de különálló szoftvercsomagként is forgalomba kerülnek, mint például a **K-Lite Codec.**

A Windows Media Player

A **Windows Media Player** a **Windows** operációs rendszer beépített lejátszója. A 2.4. ábrán a program ablakát láthatjuk.

A **Windows Media Player** segítségével a következő műveleteket hajthatjuk végre a médiafájlokon:

- kereshetünk a médiafájlok könyvtárában és fájlokat adhatunk ahhoz;
- különböző formátumú médiafájlokat játszhatunk le és hallgathatunk meg;
- fájlformátum-konverziót hajthatunk végre a program által támogatott formátumok között;
- audio- és videofájlokat menthetünk a 2.4. táblázatban ismertetett optikai tárolóeszközökre;
- internetes tárolókon (***on-line tárolókon***) elhelyezett médiafájlokat tölthetünk le és játszhatunk le, amennyiben megvannak a megfelelő jogosultságaik;
- médiafájlokat szinkronizálhatunk a számítógépünk és a kiválasztott hordozható eszköz között.

1. **Előre/hátra** gomb
2. Menüsor
3. On-line források megnyitása gomb
4. **Lista ablaktábla megjelenítése**
5. Információs munkaterület
6. Teljes képernyős megjelenítés
7. Kompakt mód
8. Hangerőszabályzó
9. Lejátszás-szabályzók
10. Előrehaladás-jelző
11. Lejátszott fájl
12. Navigációs panel

2.4. ábra. A **Windows Media Player** ablaka

A program automatikusan megkeresi a multimédiás tartalmakat a megadott mappákban és felveszi azokat a médiatárba. A médiatárban nyilván van tartva az egyes audiofájlok előadója, az album neve, a zeneszám címe, a stílus és a felvétel éve, a videofájlok stílusa és a szereplő színészek neve.

Ha meg szeretnénk változtatni a keresés beállításait, a következőképpen járunk el:

1. Kattintsunk a **Műsortár** gomb alatti nyílra.
2. Kattintsunk a **Hozzáadás a műsortárhoz** parancsra.
3. Válasszunk a **Saját személyes mappák és a mások általunk hozzáférhető mappái** közül.
4. Kapcsoljuk be a megfelelő **Speciális beállításokat.**
5. Kattintsunk a **Hozzáadás** gombra.
6. Válasszuk ki azokat a mappákat, amelyeket hozzá szeretnénk adni a figyelési listához.

A program ezután figyelni fogja a beállított mappák médiafájljait és hozzáadja az ott megjelenő új fájlokat a műsortárhoz.

A felhasználó saját lejátszási listákat hozhat létre a kedvenc zenéiből, amelyeket aztán a szükséges időpontban – például amikor bemutatja az előadót, vagy áttekintést ad annak bizonyos albumairól – lejátszhat. A listát a következőképpen hozzuk létre:

1. Kattintsunk a **Navigációs** munkaterület fölött a **Kategória kiválasztása** gombra.
2. A legördülő listából válasszuk ki, hogy *Zenéket, Képeket, Videókat, Felvett TV-műsorokat* vagy egyéb médiafájlokat szeretnénk-e megnyitni a munkaterületen.
3. Kattintsunk a **Lista ablaktábla megjelenítése** gombra.
4. Húzzuk át a listába megfelelő objektumokat.
5. Ismételjük a 4. pontot annyiszor, ahányszor szükséges.
6. Kattintsunk a Зберегти список відтворення gombra.
7. Adjunk nevet a listának.
8. Kattintsunk a **Lista mentése** gombra.

A mentett lista neve megjelenik a **Navigációs területen**. A felhasználó több lejátszási listát is létrehozhat.

Az audio- és videofájlokat a következőképpen játszhatjuk le:

1. A **Navigációs területen** válasszuk ki a következő objektumok egyikét:
   - A **Lejátszási listák** egyikét;
   - A műsortár valamelyik csoportját.
2. A **Munkaterületen** kattintsunk egy album vagy egy fájl nevére.
3. Kattintsunk a **Lejátszás** gombra.

Amennyiben egy videofájlt választottunk ki, a program automatikusan átvált a videó lejátszására. A lejátszást a **Lejátszás-vezérlő** gombokkal irányíthatjuk (4.2. 9 ábra). A gombok rendeltetését a 4.3. táblázatban találjuk.

Szükség esetén a lejátszási ablakot kompakt méretűre változtathatjuk. Ehhez elegendő a **Kompakt méret** (4.2. 7) gombra kattintani. A kompakt méretű ablakot a 4.3. ábrán láthatjuk.

4.3. ábra. A kompakt lejátszási ablak

4.3. táblázat. **A Windows Media Player vezérlőelemei**

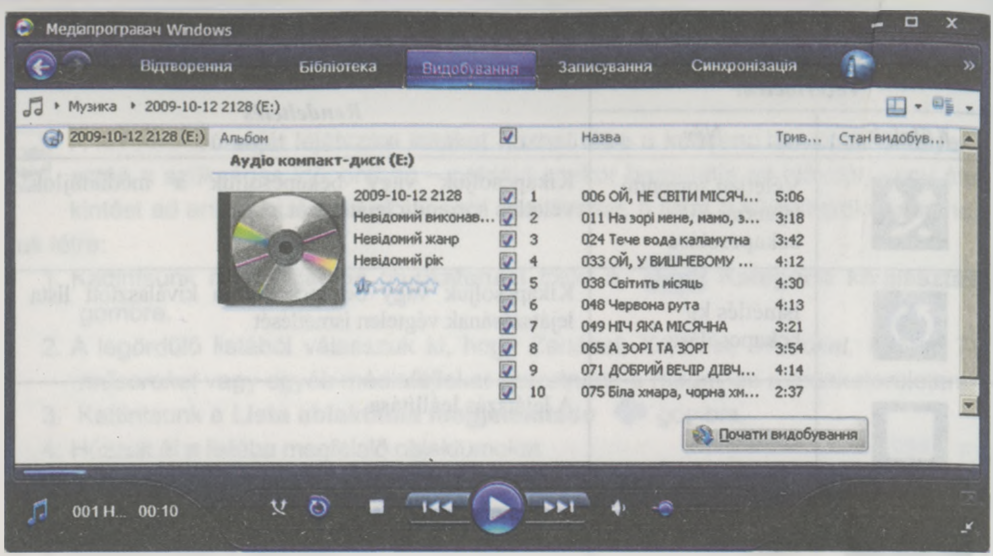
Vezérlőelem		*Rendeltetés*
Külalak	*Név*	
	Véletlen sorrendű lejátszás ki/ bekapcsolása	Kikapcsoljuk vagy bekapcsoljuk a médiafájlok véletlen sorrendű lejátszását
	Ismétlés ki/ bekapcsolása	Kikapcsoljuk vagy bekapcsoljuk a kiválasztott lista lejátszásának végtelen ismétlését
	Leállítás	A lejátszás leállítása
	Előző	Az előző fájl lejátszása
	Következő	A következő fájl lejátszása
	Lejátszás	Lejátszás indítása
	Szünet	A lejátszás szüneteltetése. A szüneteltetés idejére a gomb lejátszás gombra változik

Audio- és videofájlok formátumának átalakítása

Mivel az audio- és videofájlok meglehetősen sok formátumban léteznek, gyakran van szükség a fájlformátumok átalakítására. Ahogy azt a 9. osztályban megtanultuk, ezt a folyamatot **fájlformátum-konverziónak** nevezzük.

A konverzió megtörténhet a fájl megnyitása során vagy ha egy audio-CD tartalmát lemezre mentjük. A **Windows Media Player** egyik konverziós szolgáltatása az **Audio CD, Video CD, DVD Audio** és **Dvd Video** formátumú lemezek és lemezképek bemásolása és **WMA, WMV** vagy más formátumú fájlokká konvertálása. A műveletet a következőképpen hajthatjuk végre:

1. Helyezzünk egy audio-vagy video-CD-t a meghajtóba.
2. Kattintsunk a **Bemásolás** gombra. Megnyílik a 4.4. ábrán látható ablak.
3. Nyissuk meg a **Bemásolás** listát és a kattintsunk az **Egyéb beállítások** parancsra. A **Zene beállítása** fülön állítsuk be:

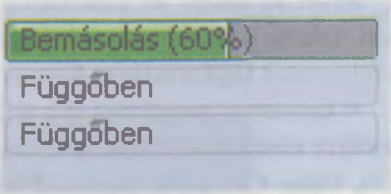


4.4. ábra. A **Windows Media Player Bemásolás** lapja

- a célfájl formátumát;
- a mappát, ahová a konvertált fájlokat meneti szeretnénk. Alapértelmezés szerint a **C:\Documents and Settings\ felhasználó\Dokumentumok\Zene.**
- a zeneszámok mely tulajdonságait szeretnénk megjeleníteni az **Információs területen,** például a zeneszám címét és előadóját.

4. Kapcsoljuk be a jelölőket azoknak a zeneszámoknak a neve mellett, amelyeket szeretnénk bemásolni.
5. Kattintsunk a Почати видобування gombra.
6. Szükség esetén kapcsoljuk be a bemásolt fájlok másolásvédelmét vagy kapcsoljuk ki azt.
7. Kapcsoljuk be azt a jelölőt, amely szerint tisztában vagyunk a szerzői jogok lehetséges megsértésével.
8. Várjuk meg a konverzió befejezését.

A konverzió folyamatáról a **Bemásolás** állapot oszlop (4.5. ábra) tájékoztat. A Зупинити видобування gombra kattintva a felhasználó bármikor leállíthatja a konverziót.

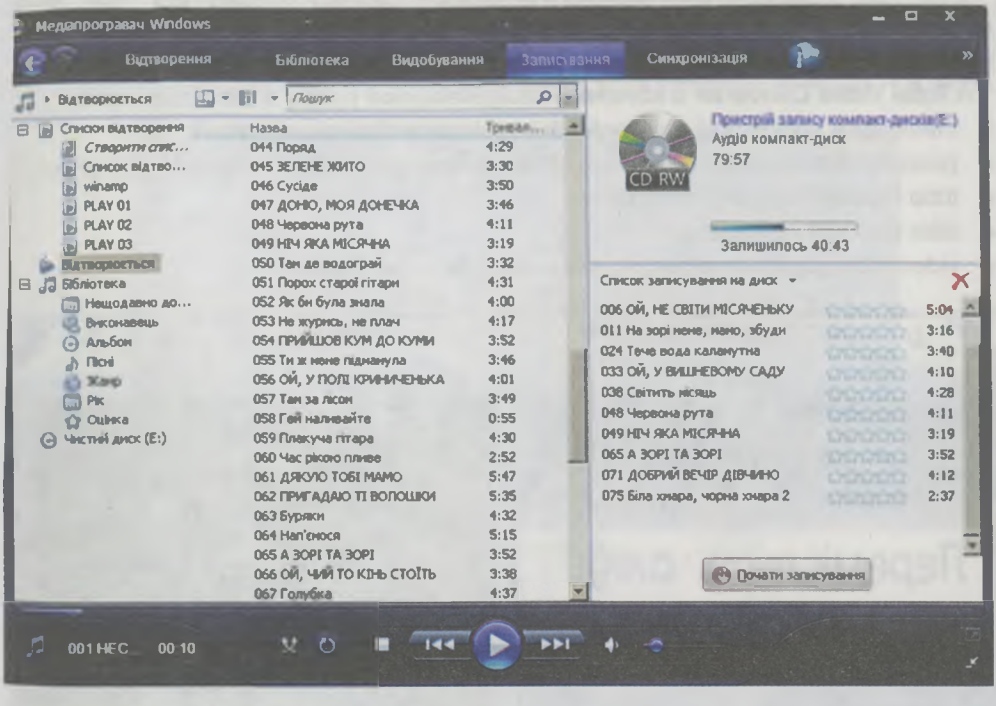


4.5. ábra. A bemásolás állapota

A bemásolással ellentétes folyamat a médiafájlok audio- vagy video-CD-re történő felírása. Ezt a **Windows Media Player**-ben **Írás**-nak nevezik. Az ilyen lemezekre akkor lehet szükség, ha a multimédiás tartalmakat olyan eszközökön szeretnénk lejátszani, amelyek a számítógépes formátumokat nem támogatják, például régi típusú DVD-lejátszókon.

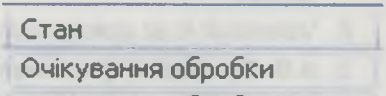
Az Audio-CD létrehozása a következőképpen történik:

1. Helyezzünk egy üres írható CD-t a meghajtóba.
2. Kattintsunk az **Írás** fülre (4.6. ábra).
3. Húzzuk át a felírandó fájlokat az **Írás munkaterületre** (az ablak jobb oldala).
4. Az **Írás munkaterület** indikátora alapján határozzuk meg a hanganyag hosszát.



4.6. ábra. A **Windows media** player írás ablaka

5. Kattintsunk az **Írási lista** gombra, majd az **Írási lista átnevezése** parancsra.
6. Írjuk be a mezőbe a lemez nevét (amennyiben előzőleg az írási listába áthúztuk a lemez nevét is, úgy ezt a lépést kihagyjuk).
7. Szükség esetén kattintsunk az **Írás** gomb legördülő menüjében az **Egyéb beállítások** parancsra, ahol az **Írás** lapon:
   - beállíthatjuk az írás sebességét;
   - növelhetjük a lemezre írt hanganyag hosszát a minőség rovására;
   - beállíthatjuk a lemez kiadását a felírást követően.
8. Kattintsunk az gombra.
9. Várjuk meg a konverzió és a lemezre írás folyamatának befejezését (4.7. ábra).
10. Amennyiben nem kívánunk több példányt felírni, a gombra kattintva törölhetjük az írási listát.



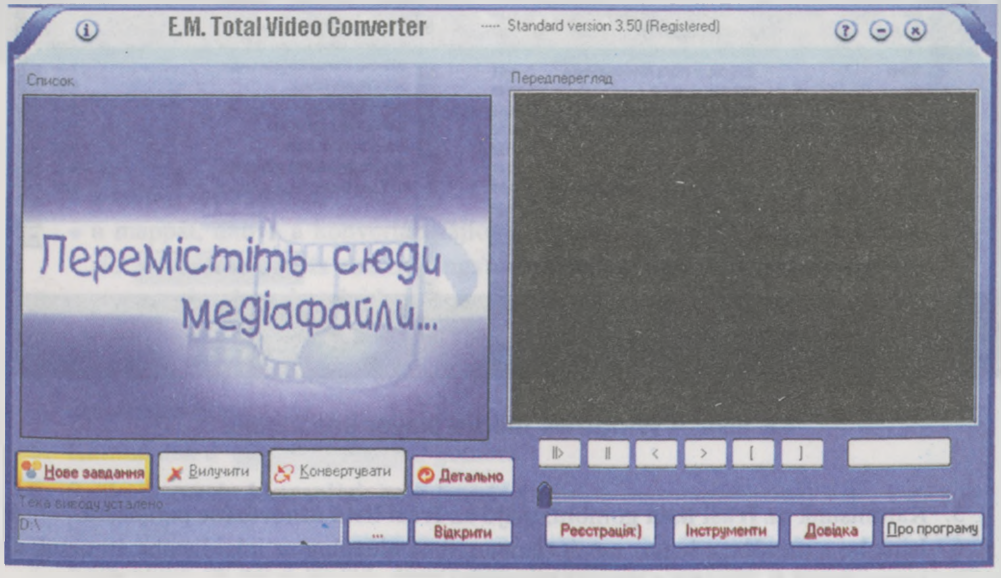
4.7. ábra. Az írás folyamata

A lejátszó programokon kívül nagyon sok speciális szoftver létezik, amelyek segítségével elvégezhetjük a fájlformátumok konverzióját. Az egyik ilyen a **Total Video Converter** (általános videokonverter), amelyet a http://www.effectmatrix.com webhelyről tölthetünk le. A magyar nyelvű változat a http://download.chip.eu/hu webhelyen érhető el.

A program a nevének nem teljesen megfelelően nemcsak az **MP4, AVI, 3GP, MPG, MPEG, WMV, FLV, DIVX videóformátumokat,** hanem a **CDA, MP3, MP2, WAV, WMA** hangformátumokat is kezeli.

A **Total Video Converter** a következő szolgáltatásokat nyújtja a felhasználóknak:

- médiafájlok konverziója egy sor különböző kodek alkalmazásával;
- prezentációs folyamatok kezelése sablonok vagy videoklipek váltása segítségével;
- több médiafájl egyidejű átalakítása;
- több videofájl egyesítése;
- videofájlok vágása és a részek elmentése különböző fájlokba.



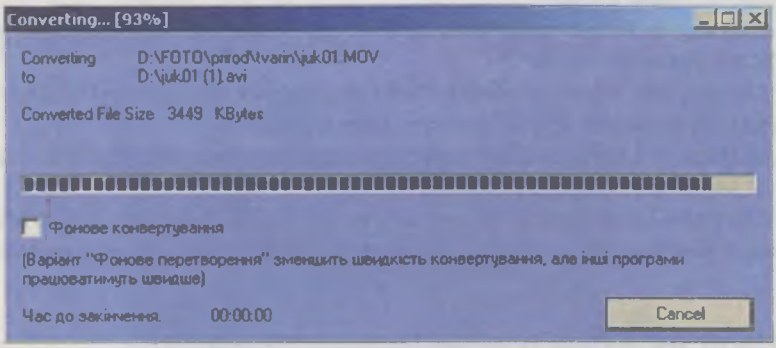
4.8. ábra. A **Total Video Converter** ablaka

A **Total video Converter** ablakát a 4.8. ábrán láthatjuk. Vizsgáljuk meg, hogy történik a MOV formátumú fájl átalakítása **AVI** formátumúvá a program segítségével.

1. Kattintsunk az **Új feladat** gombra, majd a **Імпорт медіафайлів** parancsra.
2. Válasszuk ki az átalakítandó fájlt.
3. A **Вибір формату конвертування** ablakban válasszuk ki a médiafájlok típusát, jelen esetben a videofájlokat.
4. Válasszuk ki a célfájl formátumát, jelen esetben az AVI-t.
5. A listából válasszuk ki az alkalmazandó kodeket, most a WMV AVI-t.
6. Szükség esetén a gombra kattintva változtassuk meg a célmappát.

7. Amennyiben szükséges, a **Властивості** linkre kattintva állítsuk be a célfájl tulajdonságait.
8. Kattintsunk a Конвертувати gombra.
9. Várjuk meg az átalakítási folyamat befejezését.

A fájlok átalakítása a fájl mérete, az alkalmazott tömörítési eljárások és a számítógép teljesítménye függvényében elég sokáig tarthat. A **Converting** ablakban a megfelelő jelölő bekapcsolásával elérhetjük, hogy a folyamat a háttérben történjen (4.9. ábra).



4.9. ábra. Az átalakítás folyamatának haladása

Az audiofájlok átalakítása hasonlóképpen történik.

Ellenőrző kérdések

1°. Milyen adatokat nevezünk multimédiásnak? Hozzunk fel példákat!
2°. Milyen forrásai lehetnek a multimédiás adatoknak!
3°. Milyen programok tartoznak a multimédiás adatokat feldolgozó szoftverekhez?
4°. Ismertessük az audio- és videotartalmakat feldolgozó szoftverek osztályozását!
5°. Mik a médialejátszók?
6°. Ismertessük a **Windows Media Player** szolgáltatásait!
7°. Milyen üzemmódjai vannak a **Windows Media Player**-nek?
8°. Hogy készítünk **Audio-CD**-t **Windows Media Player** segítségével?
9°. Hogy alakíthatunk át egy **Audio-CD**-t zenei fájlokká a **Windows Media Player** segítségével?
10°. Hogy hozunk létre lejátszási listát **Windows Media Player** segítségével?
11°. Mire használjuk a műsortárat?
12°. Mik a grabberek?
13°. Mik a konverterek?
14°. Miben különböznek a zenei- és videoszerkesztők a stúdiószoftverektől?
15°. Nevezzünk meg fájl- és lemezformátumokat!
16°. Miben különbözik a WAV fájl az MP3-tól?
17°. Mik a kodekek? Mi a rendeltetésük?
18°. Ismertessük a fájltípus-konverzió folyamatát **Total Video Converter**-ben?

Végezzük el a feladatokat!

1°. Vizsgáljuk meg a számítógépünkre telepített szoftvereket! Állapítsuk meg, hogy ezek közül melyik alkalmas multimédiás adatok feldolgozására! A programok súgójának segítségével derítsük ki ezek rendeltetését.

2°. Az operációs rendszer eszközeinek segítségével keressünk médiafájlokat a számítógépünkön! Állapítsuk meg a talált fájlok formátumát!

3°. Írjuk fel egy adott mappában található audiofájlokat a **Windows Media Player** segítségével **Audio CD**-re!

4°. Alakítsuk át **Windows Media Player**-el egy **Audio CD** két tetszőleges zeneszámát WMA formátumú fájllá!

5*. Hozzunk létre **Windows Media Player** segítségével egy lejátszási listát! A listába vegyük fel a tanárunk által megnevezett mappák fájljait!

6*. Alakítsuk át a tanárunk által megnevezett mp3 formátumú fájlokat WMA formátumúvá! Mentsük a fájlokat a megadott mappába!

7*. Végezzük el a tanárunk által megnevezett két MOV formátumú fájlt WMV formátumúvá konverter alkalmazásával. Mentsük az átalakított fájlt a saját mappánkba!

4.2. Hang és videó hozzáadása az oldalalapú prezentációkhoz

1. Mi az oldalalapú prezentáció? Milyen objektumokat tartalmaz?
2. Milyen audio- és video fájlformátumokat ismerünk?
3. Milyen programokkal dolgozhatók fel a hang- és videofájlok?
4. Mik a grabberek?
5. Mi az animáció? Hogy adhatunk animációs hatást a diához?

Ahogy azt már megtanultuk, az oldalalapú prezentációk a szöveges és képi anyagokon kívül hang- és videoadatokat is tartalmazhatnak. Mielőtt ezeket az objektumokat elhelyeznénk a diákon, általában szerkesztésre szorulnak. Ez azért van így, mert a **PowerPoint 2007** csak a WMA, WAV, MP3, MIDI, AU és AIFF hangformátumokat és a WMV, AVI, MPEG és ASF videoformátumokat kezeli. Az ettől eltérő formátumú fájlokat beillesztés előtt konvertálni kell.

A másik hang- és videoanyagok előkészítése során szükség lehet még azok rögzítése vagy grabbelése.

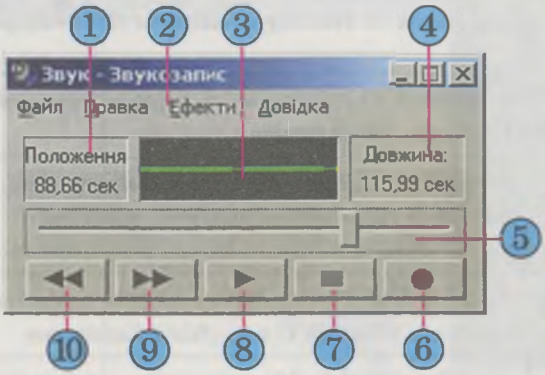
Hang rögzítése

A hang rögzítésének egyik legegyszerűbb módja a **Hangrögzítő** program alkalmazása. Ez a program a **Windows** operációs rendszer része. A programot a ***Start*** ⇒ ***Programok*** ⇒ ***Kellékek*** ⇒ ***Szórakozás*** ⇒ ***Hangrögzítő*** paranccsal indíthatjuk.

A **Hangrögzítő** hangfelvételek rögzítésére és szerkesztésére szolgál. A rögzített hangot WAV formátumú tömörítésmentes fájlba menthetjük. Egy felvétel időtartama maximum 60 másodperc lehet.

A **hang felvételéhez** szükséges egy, a hangkártya megfelelő bemenetéhez csatlakoztatott mikrofon. A felvétel menete következő:

1. Hajtsuk végre a ***Fájl-Új*** parancsot (4.10. ábra).
2. Kattintsunk a **Felvétel** gombra.
3. Mondjuk a mikrofonba a szöveget vagy játsszuk le valamely megfelelő eszközön a zenét.
4. Állítsuk le a felvételt a **Leállítás** gombra kattintva.
5. A felvételt a **Lejátszás** gombra kattintva hallgathatjuk meg.
6. A felvételt a ***Fájl –Mentés*** paranccsal fájlba menthetjük.



1. A felvétel aktuális pozíciója
2. Menüsor
3. A hang erőssége
4. A felvétel hossza
5. Pozíciójelző csúszka
6. **Rögzítés** gomb
7. **Leállítás** gomb
8. **Lejátszás** gomb
9. **Ugrás a felvétel végére**
10. **Ugrás a felvétel elejére**

4.10. ábra. A **Hangrögzítő** ablaka

Ha egy régebbi felvételt szeretnénk **meghallgatni,** a következőképpen járunk el:

1. Hajtsuk végre a ***Fájl*** ⇒ ***Megnyitás*** parancsot.
2. Válasszuk ki a lejátszandó fájlt.
3. Kattintsunk a **Lejátszás** gombra.

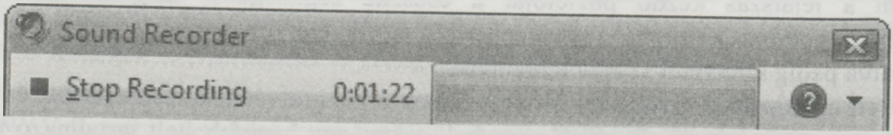
A lejátszást a **Leállítás** gombra kattintva szüneteltethetjük, majd a lejátszást a **Lejátszás** gombra kattintva ugyanebből a pozícióból folytathatjuk. Szükség esetén a lejátszás kezdő pozícióját a csúszka segítségével megváltoztathatjuk. A gombra kattintva visszaléphetünk a felvétel elejére, a gombra kattintva pedig a felvétel végére léphetünk.

A **Hangrögzítő** segítségével a felhasználók a hangfelvételek szerkesztését és formázását is elvégezhetik. E célból meg kell nyitni egy hangfelvételt tartalmazó fájlt, majd hajtsuk végre a 4.4. táblázatban összefoglalt tevékenységek közül a szükségeset.

4.4. táblázat. **Hangfelvételek szerkesztése és formázása Hangrögzítővel**

A művelet neve	*A végrehajtás módja*
Létező felvétel folytatása vagy cseréje	Húzzuk a csúszkát a már létező felvétel végére (folytatás) vagy abba a pozícióba, ahonnan a hozzáírást kezdeni szeretnénk (csere) és kattintsunk a Felvétel ● gombra. A felvételt a **Leállítás** ■ gombra kattintva állíthatjuk le
Hangerő megváltoztatása	Kattintsunk az ***Effektusok*** ⇒ ***Hangerő növelése (100%-kal)*** vagy ***Effektusok*** ⇒ ***Hangerő csökkentése (25%-kal)*** parancsra
A lejátszás sebességének megváltoztatása	Kattintsunk az ***Effektusok*** ⇒ ***Sebesség növelése*** (25%-kal vagy ***Effektusok*** ⇒ ***Sebesség csökkentése (25%-kal)*** parancsra
Felvétel lejátszása visszafelé	Kattintsunk az ***Effektusok*** ⇒ ***Lejátszás visszafelé*** parancsra, majd a **Lejátszás** ► gombra
A felvétel minőségének megváltoztatása	Kattintsunk az ***Fájl – Tulajdonságok*** parancsra, majd a **Konverzió** gombra és válasszuk ki a megfelelő minőséget
A fájl formátumának megváltoztatása	Kattintsunk az ***Fájl*** ⇒ ***Mentés*** másként parancsra, majd a **Módosítás** gombra és válasszuk ki a megfelelő formátumot
A felvétel egy részének törlése	Húzzuk a csúszkát abba a pozícióba, amelyig vagy amelytől kezdve törölni szeretnénk, majd hajtsuk végre a ***Szerkesztés*** ⇒ ***Törlés az aktuális pozícióig (aktuális pozíciótól)*** parancsot
Felvétel kiegészítése egy másik fájlból	Húzzuk a csúszkát abba a pozícióba, amelytől kezdve be szeretnénk szúrni a másik fájl tartalmát, majd hajtsuk végre a ***Szerkesztés*** ⇒ ***Fájl beszúrása*** parancsot
Másik fájl tartalmának rákeverése (mixelése) a felvételre	Húzzuk a csúszkát abba a pozícióba, amelytől kezdve rá szeretnénk keverni a másik fájl tartalmát, majd hajtsuk végre a ***Szerkesztés*** ⇒ ***Fájl rákeverése*** parancsot. A rákeverés során a két hangfelvétel szuperponálódik

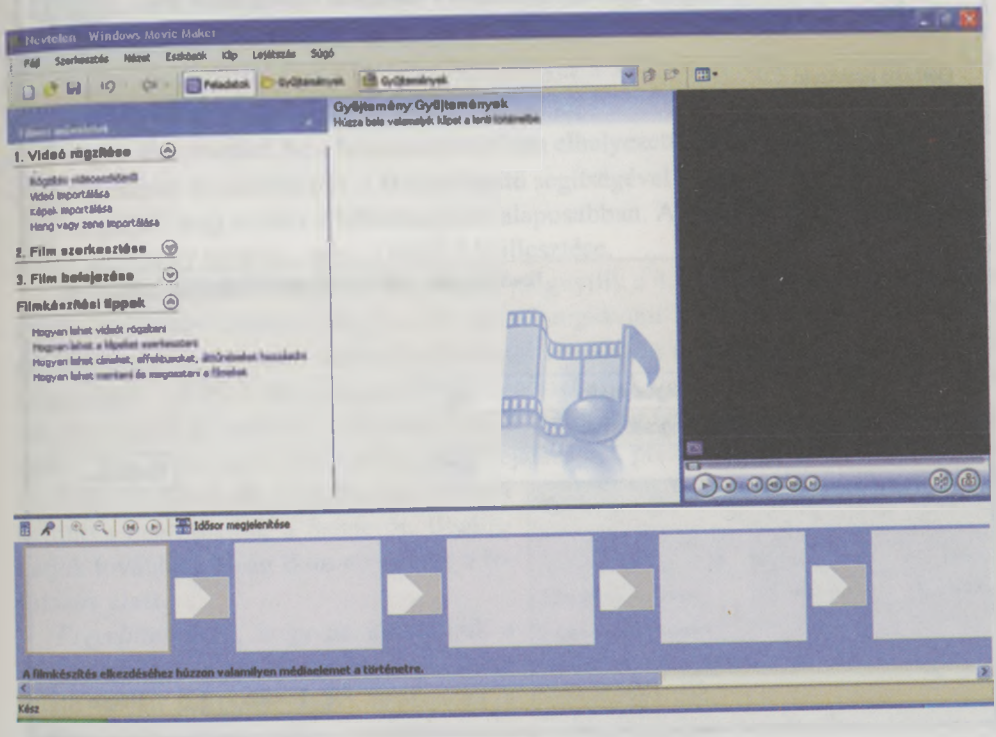
A **Windows Vista** operációs rendszerben az XP-hez képest egy megváltozott **Hangrögzítő** program tartozik (4.11. ábra), amelyből hiányoznak a szerkesztési lehetőségek.



4.11. ábra. A **Windows Vista** hangrögzítője

Videó rögzítése

A videókat egy sor különböző programmal rögzíthetjük. A Windows operációs rendszer rendelkezik beépített videórögzítő programmal, ez a **Windows Movie Maker** (*movie maker* – filmrögzítő). A program segítségével szerkeszthetjük is a videóinkat. A programot a ***Start*** ⇒ ***Minden program*** ⇒ ***Windows Movie Maker*** paranccsal indíthatjuk. A program ablakát a 4.12. ábrán láthatjuk.



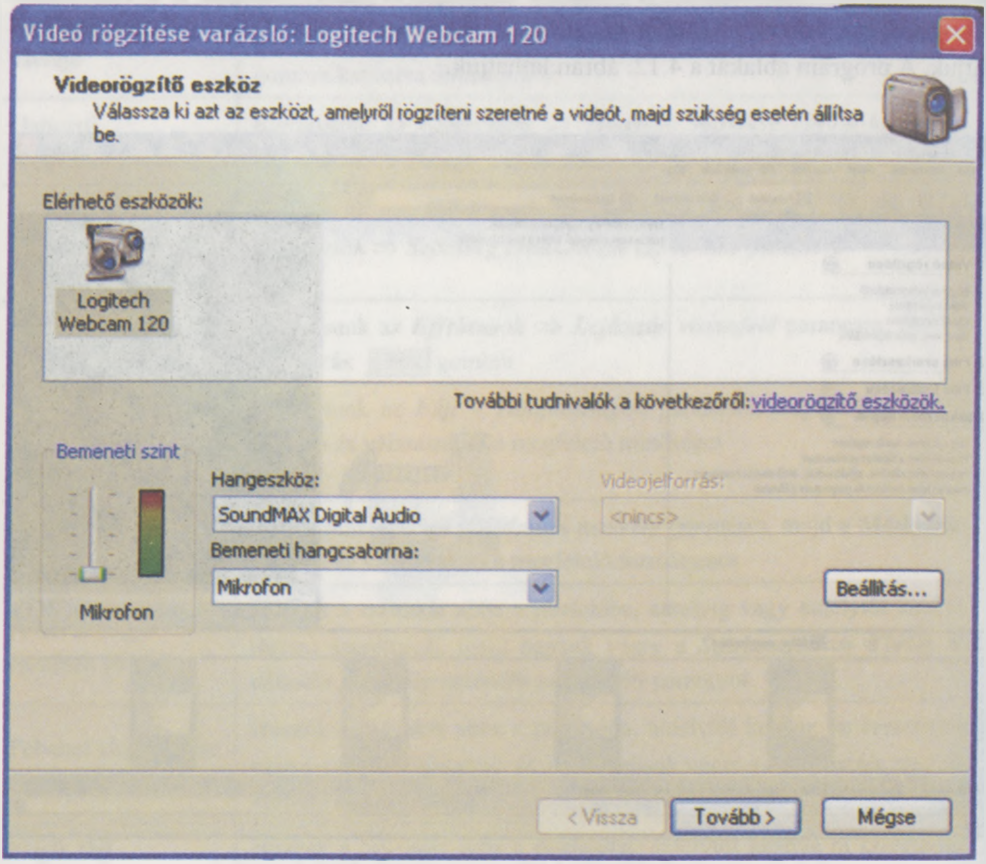
4.12. ábra. A **Windows Movie Maker** ablaka

A program segítségével hangot és videót egyaránt felvehetünk, a kettőt külön-külön is. A számítógéphez a felvétel előtt természetesen csatlakoztatni kell a megfelelő beviteli eszközt. A videó rögzítése a következőképpen történik:

1. Hajtsuk végre a ***Fájl*** ⇒ ***Videó rögzítése*** parancsot.
2. Az elérhető videórögzítők közül válasszuk ki azt az eszközt, amelyről fel szeretnénk venni a videót (4.13. ábra).
3. Állítsuk be a hangrögzítő eszközt és a bemeneti hangcsatorna jelszintjét.
4. A **Tovább** gombra kattintva lépjünk a **Rögzítő varázsló** következő ablakába.
5. Adjuk meg a létrehozandó fájl nevét és elérési útját.
6. A **Tovább** gombra kattintva lépjünk a **Rögzítő varázsló** következő ablakába.
7. Állítsuk be a videó minőségét.
8. Az **Egyéb beállítások** választógombra kattintva a listából válasszuk ki a megfelelő beállítást (a beállítás részleteiről az ablak bal alsó sarkában a beállítás részle-

te területről tájékozódhatunk, az ablak jobb alsó sarkában pedig a rendelkezésre álló és a szükséges lemezterületet láthatjuk).

9. A **Tovább** gombra kattintva lépjünk a **Rögzítő varázsló** következő ablakába.

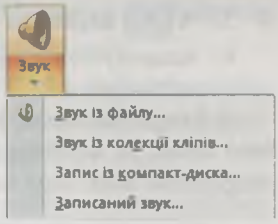


4.13. ábra. A videórögzítő varázsló ablaka

10. Kapcsoljuk be az alábbi jelölők közül a szükségeseket:
    - hozzunk-e léte a klipeket a varázsló után;
    - legyenek-e elnémítva a hangszórók a rögzítés során;
    - kapcsoljunk-e be időkorlátot a rögzítésre.
11. Kattintsunk a **Rögzítés indítása** gombra.
12. A felvétel befejeztével kattintsunk a **Rögzítés leállítása** gombra.
13. A 10–11. lépéseket ismételve rögzítsük a jeleneteket.
14. Kattintsunk a **Befejezés** gombra.

Létrejött a megadott nevű fájl a felhasználó által megadott mappában. A továbbiakban ezt a fájlt szerkeszthetjük, konvertálhatjuk, más fájlba illeszthetjük.

Hangobjektumok hozzáadása az oldalalapú prezentációkhoz



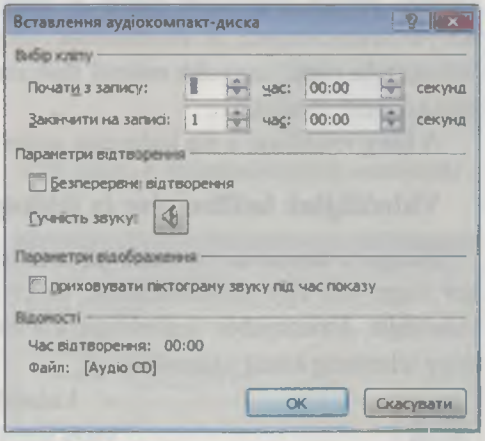
4.14. ábra. A **Hang** eszköz menüje

Hangot a ***Beszúrás*** ⇒ ***Hang*** paranccsal helyezhetünk el a prezentáció egy diáján. Ekkor megnyílik egy párbeszédablak, amelynek segítségével kiválaszthatjuk a szükséges hangfájlt. Amennyiben legördítjük a **Hang** eszköz menüjét (4.14. ábra), a következő négy lehetőség közül választhatunk:

- **Hangfájlt** illeszthetünk be a helyi számítógépen vagy helyi hálózatban tárolt fájlok közül;
- Hangot illeszthetünk be **ClipArt**-ból, amelyet a **Microsoft Office** gyűjteményből vagy a **Webes gyűjteményből** választhatunk ki;
- Hangot illeszthetünk be a lemezmeghajtóban elhelyezett **Audio CD**-ről;
- Közvetlenül beilleszthetjük a **Hangrögzítő** segítségével készített felvételt.

Vizsgáljuk meg ezeket a lehetőségeket alaposabban. A hangfájl beillesztése lényegében ugyanúgy történik, mint a képfájl beillesztése.

Ha **audio CD-ről illesztünk be** hangot, megnyílik a 4.15. ábrán látható párbeszédablak. Itt kiválaszthatjuk, hogy a CD mely hangsávjait kívánjuk beilleszteni, egész pontosan melyik sáv hányadik másodpercétől kezdődjön a beillesztés és pontosan hol érjen véget. Az ablak bal alsó sarkában ellenőrizhetjük, hogy a beállításaink pontosan milyen hosszú **Játékidőt** jelentenek majd. A **leállításig ismétli** jelölőnégyzetet bekapcsolva kezdeményezhetjük, hogy a hang lejátszása a prezentáció bemutatása alatt folyamatos legyen. A **Hangszóró** gombra kattintva beállíthatjuk a hangerőt. Beállíthatjuk továbbá a **hang ikon elrejtését a lejátszás alatt.**

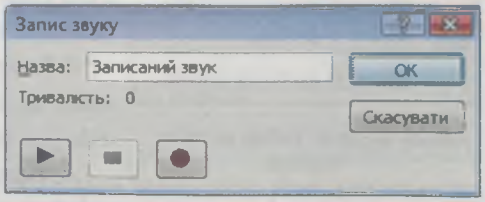


4.15. ábra. A zenei **CD behelyezése** párbeszédablak

Figyeljünk arra, hogy ha kicseréljük a meghajtóban a CD-t, akkor az új CD megfelelő sávjai kerülnek majd lejátszásra a vetítés során. Amennyiben a vetítés során nincs CD a meghajtóban, úgy természetesen a hang lejátszása elmarad.

Ha a diához a felhasználó kommentárjait szeretnénk beilleszteni, a **Hangrögzítőben** (4.16. ábra) adjuk meg a hangobjektum nevét, majd végezzük el a hang rögzítését a következőképpen:

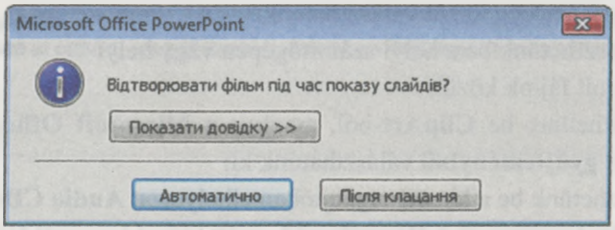
1. Kapcsoljunk mikrofont a számítógéphez. Amennyiben a számítógép beépített hangszóróval rendelkezik, azt is használhatjuk.
2. Kattintsunk a ● **Rögzítés** gombra.
3. Mondjuk a mikrofonba a szöveget.
4. Kattintsunk a ■ **Leállítás** gombra.



4.16. ábra. **Hangrögzítés** ablak

5. A ▶ **Lejátszás** gombra kattintva meghallgathatjuk a felvételt.
6. Amennyiben a szövegünk jól lett rögzítve, kattintsunk az **OK** gombra. Ha nem, ismételjük a 2–5. lépéseket.

A hangobjektumok beillesztése során utolsó lépésként megnyílik a 4.17. ábrán látható ablak. Itt állíthatjuk be, hogy a zene lejátszása **Automatikusan** történjen-e a dia megnyitása után, vagy az ikonra kattintással kívánjuk-e elindítani.



4.17. ábra. A hang lejátszásának beállítása ablak

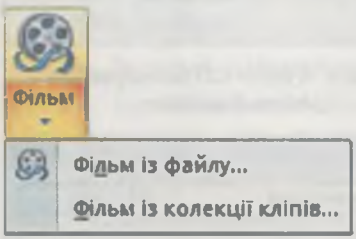
A hang beillesztése után a dián a megjelenik a hang ikon, amennyiben annak elrejtését nem kapcsoltuk be. Ha az ikonra kattintunk, a **Szalagon** megnyílik a **Hang** fül és a **Paraméterek** ideiglenes eszköztár, amelynek segítségével elvégezhetjük a hang szerkesztését és formázását.

Alapértelmezés szerint a 100 Kbyte-ot meghaladó hangfájl nem ágyazódik a prezentációba, hanem csatolódik ahhoz. A beágyazható wav fájl maximális mérettét 50 000 KByte-ra emelhetjük, de ez jelentősen megnöveli a prezentáció méretét. Amennyiben a felhasználó ennél nagyobb méretű fájlt szeretne beágyazni, a prezentáció-készítő program a kérést elutasítja.

A hang lejátszása a dia lejátszása során történik majd meg.

Videófájlok beillesztése és feldolgozása

Hangot a ***Beszúrás*** ⇒ ***Videó*** paranccsal helyezhetünk el prezentáció egy diáján. Ekkor megnyílik egy párbeszédablak, amelynek segítségével kiválaszthatjuk a szükséges videófájlt. Amennyiben legördítjük a **Videó** eszköz menüjét (4.18. ábra), a következő négy lehetőség közül választhatunk:



4.18. ábra. A **Videó** eszköz menüje

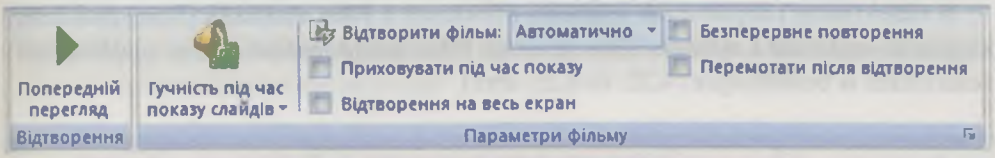
- **Videófájlt** illeszthetünk be a helyi számítógépen vagy helyi hálózatban tárolt fájlok közül;
- **Videót illeszthetünk be ClipArt-ból,** amelyet a **Microsoft Office** gyűjteményből vagy a **webes gyűjteményből** választhatunk ki.

A hang beszúrásához hasonlóan a videó beillesztésekor be kell állítani, hogy a lejátszás automatikusan induljon vagy kattintásra.

A videó objektum elhelyezése után a dián megjelenik egy téglalap alapú terület, amelyben a videó lejátszásra kerül. A **Méretezőgombok**

vagy a **Formátum** ideiglenes eszköztár **Méret** csoportjának **Magasság** és **Szélesség** eszközének segítségével megváltoztathatjuk ennek méreteit és elhelyezését.

A videó objektum kijelölése után a **Szalagon** megjelenik a **Film** kiegészítő eszköztár, amelyen megtaláljuk a **Beállítások** ideiglenes fület. Itt találhatjuk azokat az eszközöket, amelyek segítségével beállíthatjuk az objektum paramétereit (4.19. ábra).

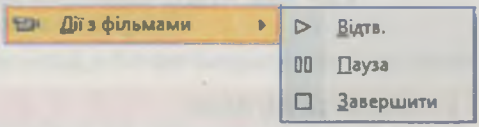


4.19. ábra. A **Film** eszköztár

Ugyanezeket a beállításokat elérhetjük, ha a **Film beállításai** csoport **Megnyitás** gombjára kattintunk, majd megnyitjuk a megfelelő ablakot.

Animációs hatások hozzárendelése a videó- és hangobjektumokhoz

A hang- és videó-objektumokhoz ugyanúgy adunk hozzá animációs hatásokat, mint a grafikai objektumokhoz, csak ez esetben a hang objektumot, vagy a videó lejátszására szolgáló téglalap alapú területet kell kijelölni előtte. Az **Effektus hozzáadása** lista ezen objektumok esetében tartalmaz egy **Hangműveletek** és egy **Videóműveletek** parancsot is (4.20. ábra). Mindkét esetben megnyílik egy **Lejátszás, Szünet** és **Leállítás** gombot tartalmazó eszköztár.

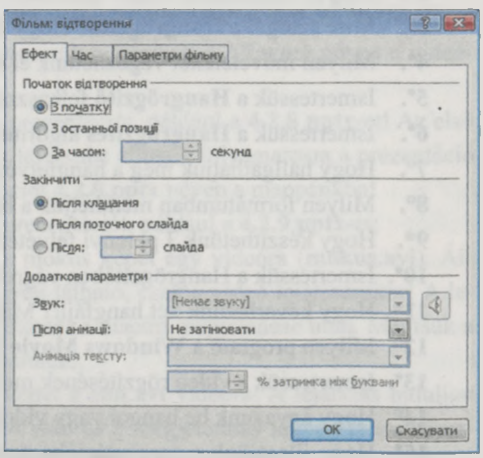


4.20. ábra. A Videóműveletek eszköztár

A kiválasztott animációs hatások beállítását a helyi menü segítségével végezzük. Kiválasztjuk a hatástól függő listát, amelynek beállításait el kívánjuk végezni. Videó-objektumok esetében ez a **Lejátszás**, a **Szünet** és a **Leállítás** lehet.

A **Film lejátszása** hatás esetében a **Hatás** fülön a felhasználó a következő beállításokat végezheti el (4.21. ábra):

- beállíthatja, hogy a lejátszás *az elejétől, az utolsó pozíciótól vagy egy megadott időponttól* kezdődjön-e;
- beállíthatja, hogy a leállítás *egérkattintásra, az aktuális dia után, vagy a megadott számú dia után* történjen-e;
- a szöveges objektumokhoz és képekhez hasonlóan beállíthatja, hogy mi történjen az animáció után.



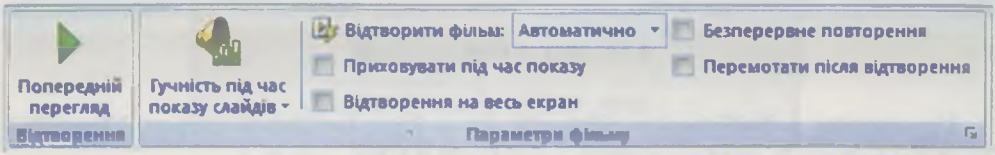
4.21. ábra. **Hatás** ablak **Film lejátszása** fül

A **Film lejátszása hatás** esetében a **Film beállításai** fülön felhasználó a következő beállitásokat végezheti:

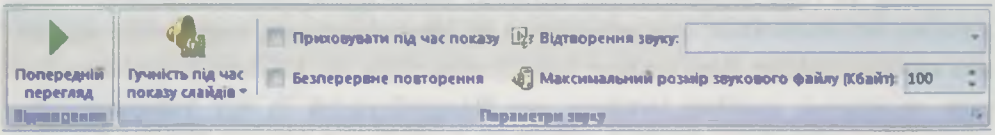
- beállíthatja a hangerőt;
- bekapcsolhatja az elrejtést, ha nem történik lejátszás;
- bekapcsolhatja a teljes képernyős megjelenítést.

A hang animációs effektusok beállítása gyakorlatilag ugyanúgy történik, mint a videó objektumok esetében, de itt nem értelmezhető a teljes képernyős megjelenítés.

A videó (hang) objektumok lejátszási effektusait a **Film eszközök** (Hang eszközök) kiegészítő eszköztár **Lejátszás** fülén található **Film tulajdonságai** (Hang tulajdonságai) vezérlőkkel is beállíthatjuk (4.22. és 4.23. ábra).



4.22. ábra. A videó objektumok lejátszási effektusai a **Film eszközök** kiegészítő eszköztár segítségével



4.23. ábra. A hang objektumok lejátszási effektusai a **Hang eszközök** kiegészítő eszköztár segítségével

Ellenőrző kérdések

1°. Milyen formátumú hangot illeszthetünk be a **PowerPoint 2007** prezentációba?
2°. Milyen formátumú videót illeszthetünk be a **PowerPoint 2007** prezentációba?
3°. Mire szolgál a **Hangrögzítő**? Hogy indítjuk?
4°. Milyen műveleteket végezhetünk el a **Hangrögzítővel**?
5°. Ismertessük a **Hangrögzítő** felhasználói felületét!
6°. Ismertessük a **Hangrögzítés** algoritmusát!
7°. Hogy hallgathatjuk meg a hangfelvételt a **Hangrögzítővel**?
8°. Milyen formátumban menthetjük a hangfelvételt a **Hangrögzítőben**?
9*. Hogy készíthetünk 3 percnyi felvételt a **Hangrögzítőben**?
10°. Ismertessük a Hangrögzítőben végrehajtható szerkesztési műveleteket!
11*. Hogy keverhetünk két hangfájlt? Mikor használjuk ezt a műveletet?
12°. Milyen program a **Windows Movie Maker**? Hogy indítjuk el?
13°. Ismertessük s videó rögzítésének menetét **Windows Movie Maker**-ben.
14°. Hogy ágyazunk be hangot vagy videót a prezentáció diájába?
15°. Hogy illesztünk a prezentációba zenét CD-ről? Milyen CD-t használhatunk ebben az esetben?

16*. Hasonlítsuk össze a következő két lehetőséget: kommentárt illesztünk a prezentáció egy diájába a **Hangrögzítővel**, vagy egy már rögzített hangot illesztünk be. Mikor célszerű az egyik vagy másik módszert alkalmazni?

17*. A videó milyen tulajdonságait változtathatjuk meg a prezentáció lejátszása során és hogy végezzük ezt?

18*. Milyen különbség van a prezentáció egy diájába illesztett hangfájlok és videófájlok tulajdonságai között?

19*. Hogy érhetjük el, hogy az audio CD egy sávján rögzített zenét hallhassunk a prezentáció lejátszásának teljes ideje alatt?

Végezzük el a feladatokat!

1°. Vegyük fel a kedvenc versünket a hangrögzítő segítségével. Mentsük el a felvételt **4.2.1.wav** néven a mappánkba.

2°. Nyissuk meg a **Hangrögzítőben** a tanárunk által megadott fájlt, például a **4.2.2.wav**-ot! Bővítsük a fájlt egy verssel a saját előadásunkban! Mentsük a fájlt **4.2.2.wav** néven a mappánkba!

3°. Vegyük fel a kedvenc versünket a hangrögzítő segítségével! Folytassuk a felvételt egy másik vers első versszakával! Mentsük a felvételt **4.2.3.wav** néven a **Zenéink** mappába!

4*. Nyissuk meg a **Hangrögzitőben** a tanárunk által megadott fájlt, például a **4.2.4wav**-ot! Állítsuk a csúszkát az első versszak végére. Töröljük a fájlnak a megmaradt részét. Egészítsük ki a felvételt egy másik vers első versszakával és mentsük a munkánkat **4.2.4.wav** néven a mappánkba!

5*. A **Hangrögzítő** súgójából derítsük ki, mit értünk a felvétel hangerején, és hogy változtathatjuk azt meg!

6°. Kapcsoljunk a számítógéphez egy webkamerát és egy mikrofont. Vegyünk fel ezek segítségével egy videót, amelyen a kedvenc szépirodalmi művünkből mondunk egy részletet! Mentsük a felvételt **4.2.6.wmv** néven a mappánkba!

7. Egy osztálytársunk segítségével készítsük el egy világirodalmi szöveg felvételét a **Windows Movie Maker** segítségével! Mentsük a felvételt **4.2.7.wmv** néven a videóink mappájába!

8°. Nyissunk meg a tanárunk által megadott prezentációt, például a **4.2.8 pptx**-et! Az első diába szúrjuk be a **4.2.8.mid** hangfájlt! A lejátszást állítsuk folyamatosra a prezentáció lejátszásának végéig! Mentsük a prezentációt **4.2.8.pptx** néven a mappánkba!

9°. Nyissunk meg a tanárunk által megadott prezentációt, például a **4.2.9 pptx**-et:

a) A mókusról szóló dián cseréljük le a mókus képét egy videóra (**mókus.avi**). Állítsuk be, hogy a videó ablaka ne legyen látható, csak a videó lejátszásakor! A lejátszás induljon automatikusan, közvetlenül a diacím megjelenése után. Mentsük a prezentációt **4.2.9.pptx** néven a mappánkba!;

b) A sünről szóló dián cseréljük a sün képét a **sun.avi** videóra! A lejátszás induljon 1 másodperccel a szöveg megjelenése után és a megjelenítés legyen teljes képernyős! Mentsük a prezentációt **2.4.9_1.pptx** néven a mappánkba!

10°. Nyissunk meg a tanárunk által megadott prezentációt, például a **4.2.10 pptx**-et! Ismerkedjünk meg a prezentáció tartalmával! Készítsünk rövid ismertetőt a modern számítógép jellemzőiről és rögzítsük azt a hangrögzítővel! A felvételt ágyazzuk be a prezentáció diáiba:

a) ismertessük a *monitorokat és a szkennereket.* A hangfelvétel induljon a megfelelő dia megjelenésével egyidejűleg. A diákhoz ne állítsunk be lejátszási időt. Mentsük a prezentációt **4.2.10.pptx** néven a mappánkba;

b) ismertessük a *processzorokat.* A lejátszás induljon egérkattintásra (bal gomb). A diák lejátszási idejét ne változtassuk meg. Mentsük a prezentációt **4.2.10_1.pptx** néven a mappánkba.

4.3. Videofilmek (időalapú prezentációk) készítése és szerkesztése

1. Véleményetek szerint miben különbözik a mozifilm a videofilmtől? Milyen eszközök szükségesek egy videó megtekintéséhez?
2. Mi a forgatókönyv? Hol alkalmazzák?
3. Mire használhatjuk a Windows Movie Makert? Hogy indíthatjuk el?
4. Hogy végezzük a videó és a hang rögzítését? Milyen programok szükségesek ehhez? Hogy nevezzük a programok ezen típusát?

A videó létrehozásának általános követelményei

A **videofilm** kifejezés a **mozifilm** (az angol *cine*–film) kifejezésből származik, amely alatt általában a filmművészeti alkotást értjük. A videó a rögzítés módjában különbözik a mozifilmtől. A mozifilm fényképészeti filmre, a videó kazettára vagy CD-re van rögzítve. A rövid videókat vagy mozifilmeket klipeknek nevezik (*clip* – vágás). Az utóbbi időkben a klipek igen elterjedtek a zenei iparban, ahol zenés **videoklipeknek** nevezik ezeket.

Szerkezetüket és elkészítésük technológiáját tekintve az **időalapú prezentációkat** olyan videoklipeknek tekinthetjük, amelyek egy új ötlet, projekt, gyártmány vagy áru bemutatására szolgálnak. Elkészítésükhöz videoszerkesztő programra van szükség.

Az ilyen prezentáció elökészítésének ***első lépése*** a forgatókönyv megírása. A forgatókönyv valamely rendezvény lebonyolításának részletes terve. A **forgatókönyv** még egy igen rövid klip esetében is meghatározott szerkezettel (kompozícióval) bír. A forgatókönyv részletes kell legyen, amely a tevékenységek sorrendjét másodperces pontossággal írja le.

A ***második lépés*** a jelenetek rögzítése a forgatókönyv előírásai szerint. Elkészítik a szükséges fényképeket, rajzokat, kiválasztják a zenei aláfestést, felveszik a kísérőzenét. Előfordulhat, hogy a videofilm valamely előzetes videoanyag alapján készül. Egy történelmi film, egy jelentős személyiség életét bemutató film, de akár az osztályunk életét bemutató film is nagy részben dokumentumfilmek és fotók alapján készül.

A ***harmadik lépés*** **a montírozás**, amely során a jeleneteket egységes egésszé – videofilmmé – alakítják át. A montírozás során a felvett jeleneteket szerkesztik, pontosítják a

forgatókönyvet, hozzászerkesztik a hangot és a feliratokat, szöveges kommentárokkal látnak el egyes képkockákat stb.

A forgatókönyv elkészítése

Vizsgáljuk meg egy, a folyók mentén élő madarakról szóló film forgatókönyvének elkészítését. A film az általános iskolák harmadik osztályosainak készül, az **Állatok** témakör tanulmányozásának segítése céljából. Ismerkedjünk meg a rendelkezésre álló videó- és fotóanyagokkal (4.5. táblázat).

4.5. táblázat. **A videofilmben szereplő fájlok listája**

Fájl neve	*Mappa*	*Tartalom*	*Hosszúság, s*
лисуха.avi	**Тема 4\Відео**	Videó	42
лелека.avi	**Тема 4\Відео**	Videó	31
лисуха01.jpg	**Тема 4\Фото**	Fénykép	–
лисуха02.jpg	**Тема 4\Фото**	Fénykép	–
лелека01.jpg	**Тема 4\Фото**	Fénykép	–
лелека02.jpg	**Тема 4\Фото**	Fénykép	–
лелека.wav	**Тема 4\Звук**	Hang	4
лисуха.wav	**Тема 4\Звук**	Hang	6,75

Állítsuk össze a videó forgatókönyvét oly módon, hogy a lejátszás megfeleljen az általános iskola harmadik osztályosai életkori sajátosságainak. A forgatókönyv **kulcskockák** – a videó tartalmi elemeinek leírásit tartalmazó kockák – sorozatából áll.

1. jelenet. **Címlap::**
   - Cím – *Ukrajna madarai,* alcím – *3. osztály számára;*
   - háttérszín – *zöld,* betűszín – *fehér;*
   - hosszúság – *4 s*;
   - hang – *nincs;*
   - videóhatás – *szemcsézettség;*
   - átmenet a következő kockára – *szem hatás.*

2. jelenet. **A gólya fényképe:**
   - fájl – *gólya02.jpg;*
   - felirat – *Gólya,* alcím – *(gilice, czakó, eszterág);*
   - hosszúság – *4 s;*
   - hang – **gólya.wav,** hangzás hossza – *4 s;*
   - átmenet a következő kockára – *forgószél hatás.*

3. jelenet. **A gólya fényképe:**

- fájl – *gólya01.jpg;*
- hosszúság – *4 s;*
- hang – *nincs*
- átmenet a következő kockára – *függőleges pillangó hatás.*

4. jelenet. **Videó a gólyáról:**

- fájl – *gólya01.avi;*
- hosszúság – *26,14 s* (teljes hossz – *31 s*, az elejéről 4,86 s; törlendő, az eredeti hang törlendő;
- hang – *kommentár,* hossza 22,14 s;
- átmenet a következő kockára – *belépés balról, alulról hatás.*
- a kommentár szövege: *„Láttatok egy madarat és meghallgattátok annak hangját, pontosabban csőrének kopogását, a kelepelést. Ez a madár a gólya, népi nevén gilice, czakó vagy eszterág. A gólya tolla fehér, a szárnyán fekete. Lába hosszú, csőre hosszú és piros vagy sárgáspiros. Ukrajnában a gólya a tavasz hírnöke."*

5. jelenet. **A kacsa fényképe:**

- fájl – *kacsa02.jpg;*
- felirat – *Eurázsiai vagy közönséges szárcsa;*
- hosszúság – *4 s;*
- hang – *kacsa.wav* fájl, 2 s-mal a kép megjelenése után kezdve, hosszúsága 6,75 s, folytatódik a következő kockán;
- átmenet a következő kockára – *belépés balról, alulról hatás.*

6. jelenet. **A kacsa fényképe:**

- Fájl – *kacsa01.jpg;*
- Hosszúság – *4 s;*
- Hang – a *kacsa.wav* fájl, az előző kockáról folytatva, a fénykép eltűnésével egyidejűleg fejeződik be;
- Átmenet a következő kockára – *diafragma hatás.*

7. jelenet. **Videó a szárcsáról:**

- fájl – *szárcsa.avi*
- hosszúság – *30 s* (teljes hossz 42 s, a végéről 12 s-ot le kell vágni, az eredeti hang megmarad a kommentár hátttereként);
- átmenet a következő kockára – *átlós áttűnés hatás;*
- a kommentár szövege: *„Az eurázsiai vagy közönséges szárcsa egy kistestű vízimadár, amely Ukrajna egész területén elterjedt. Könnyen megismerjük fehér csőréről és a homlokán látható fehér foltról. Életének nagy részét vízen tölti, amiért gyakran mocsári vagy vízityúknak is nevezik".*

8. jelenet. **Zárójelenet:**

- felirat – *1. sz. Zsitomiri Líceum, Jövendő stúdió;*
- kamera – *Mihajlenko Sztepan;*
- szöveg – *Kovalenko Okszana;*
- hang – *Ivanenko Mikola;*
- *a 10. A osztály tanulói a 3. oszt számára;*
- háttér – *sárgászöld, betűszín –fehér;*
- hosszúság – *10 s.*

A videó létrehozása

Hozzuk létre a videót **Windows Movie Maker**-ben. E célból ismerkedjünk meg alaposabban a program felhasználói felületével (4.24. ábra). Ha a **Gyűjtemények** Сборники gombra kattintunk, akkor az ablak bal oldalában megnyílik az azonos nevű munkaterület (4.24. 3 ábra), amely három szabványos mappát tartalmaz: **Videoeffektusok, Videoáttűnések, Gyűjtemények.** Ha megnyitjuk a **Videoeffektusok** mappát, a **Tartalom** táblán (4.24. 4 ábra) megnyílik a videoeffektusok listája (*elforgatás, elmosás, fényerő csökkentése, növelése* stb.), amelyeket az egyes kockákra alkalmazhatunk. A **Videoáttűnések** mappa az egyes kockák közötti átmenetekre alkalmazható hatásokat tartalmazza. A **Gyűjtemények** mappa grafikai, hang- és videofájlokat tartalmaz, amelyeket beilleszthetünk a saját videónkba.

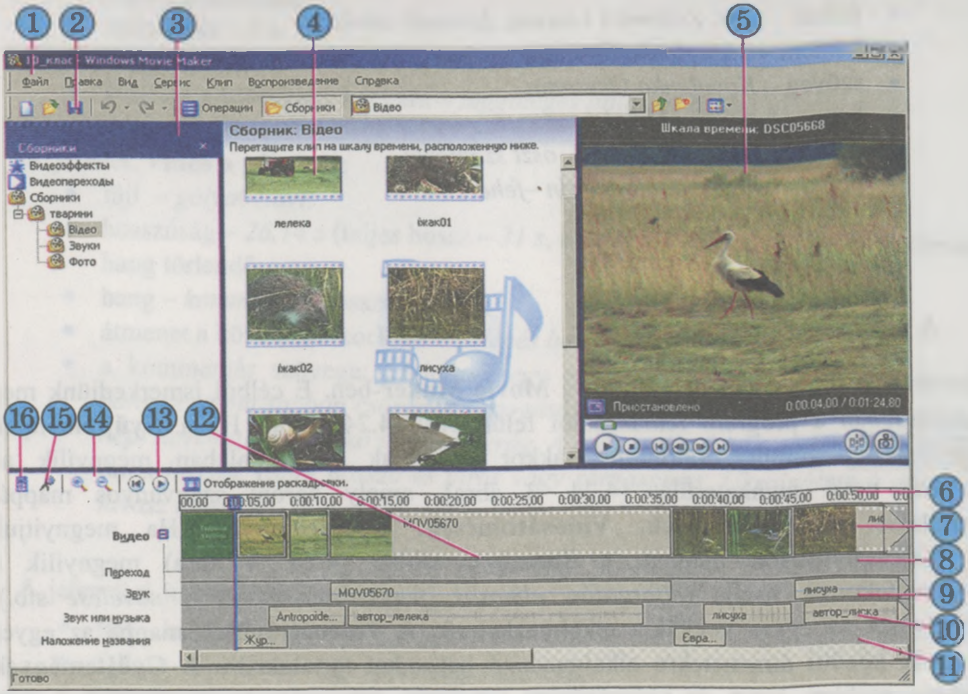
Az ablak jobb oldali részében találjuk a lejátszás ablakot (4.24. 5 ábra), ahol lehetőségünk van a létrehozott film megtekintésére. Az ablak alsó részét a **Történet** panel foglalja el (4.24. 12 ábra). A panelt kétféle módban használhatjuk: **történet** és **idősor** módban. Az üzemmódokat az **Idősor megjelenítése** Отображение шкалы времени, illetve **Történet megjelenítése** Отображение раскадровки gombokkal válthatjuk.

A **Történet panel** idősor módban hangsávot (4.24. 9 ábra), videosávot (4.24. 7 ábra), zenei sávot (4.24. 10 ábra) feliratsávot (4.24. 11 ábra) tartalmaz. A munkaterület felső része tartalmazza az időskálát (4.24. 6 ábra).

Történet üzemmódban a **Montírozás sáv** kockákat tartalmaz: azokat a videó és grafikai elemeket, amelyekből a videó áll.

Az **Időskála** módot leginkább a multimédiás objektumok lejátszásának beállítására használjuk, megadhatjuk a lejátszási időket, átmeneteket rendelhetünk az egyes jelenetekhez és a jelentek közötti átmenetekhez.

A ***videovágás első lépéseként a Fájl – Új*** projekt paranccsal vagy az **Új projekt** gombra kattintva hozzunk létre egy új projektet. A projekt hivatkozásokat tartalmaz a multimédiás objektumokra, tartalmazza azok tulajdonságait, valamint egyéb, a film felépítésére vonatkozó más információkat. A **Windows Movie Maker** a projektet **MWMM** (*Microsoft Windows Movie Maker*) formátumban tárolja a projektet. A felhasználónak időnként mentenie kell a projektet, hogy műszaki vagy más problémák előfordulása esetében ne veszítsük el az addig elvégzett munka eredményét.



1. Menüsor
2. Eszköztár
3. **Gyűjtemények** panel
4. Taralom tábla
5. Lejátszás ablak
6. Idősor
7. Videósáv
8. Átmenetek
9. Hangsáv
10. Zenei sáv
11. Feliratok
12. Montírozás sáv
13. Lejátszásvezérlők
14. Időskála méretarány
15. Kommentár rögzítésének bekapcsolása
16. Hangerőszabályzó bekapcsolása

4.24. ábra. A **Windows Movie Maker** ablaka

A projekthez tartozó fájlokat célszerű egy gyűjteményben elhelyezni. A fájlok csoportosítása céljából a gyűjteményben mappákat hozhatunk létre. A 4.24. ábrán az állatok mappa tartalmazza a **Видео, Звуки** és **Фото** mappákat. Mappákat az **Új gyűjteménymappa** gombra kattintva vagy a **Gyűjtemény** helyi menü megfelelő pontjára kattintva hozhatunk létre.

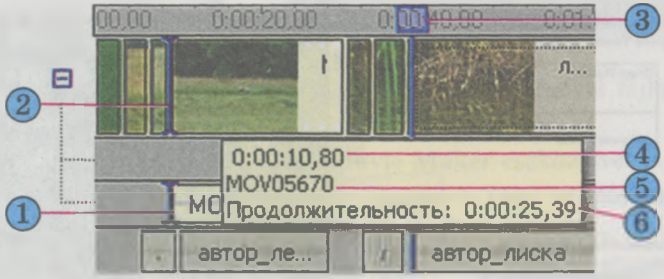
Második lépésként át kell húzni a gyűjtemény objektumait a szerkesztőterületre. Ezt a következőképpen végezhetjük el:

1. Váltsunk **Történet megjelenítése** módba.
2. A megírt forgatókönyvnek megfelelően húzzuk át a **Відео** és **Фото** mappa objektumait a **Szerkesztőterület** fekete téglalapjaira (4.25. ábra).
3. Váltsunk át idősor nézetre.
4. A forgatókönyvnek megfelelően helyezzük el a hangobjektumokat a hangsávon. Ügyeljünk a helyes kezdési időpontok betartására.
5. Állítsuk be az objektumok lejátszási idejét.

4.25. ábra. A **Szerkesztőterület** történet megjelenítés módban

Ha a videóobjektumot idősor módban húzzuk a szerkesztőterületre, akkor áthúzáskor az egérmutató mellett megjelenik a ⊞jel, amely az objektum elhelyezését mutatja, a hangsávon pedig az objektum helyét mutató Ijel (4,26. 1, 2 ábra). A mutató melletti téglalapból leolvasható a lejátszás ideje (például *0 h 0 m 10,8 s*), valamint annak az objektumnak a neve, amely után az objektumot elhelyezzük (például *MOV05670*) és annak lejátszási ideje (*0 h 0 m 25,39 s*).

Grafikai objektumok esetében az alapértelmezett lejátszási idő 4 s. Ha ezt meg szeretnénk változtatni, az objektum jobb oldali határvonalának áthúzásával tehetjük meg (4.27. ábra). A lejátszás hosszát növelhetjük és csökkenthetjük is.



1. Az objektum beillesztésének helye a hangsávon
2. Az objektum beillesztésének helye a videósávon
3. Az aktuális objektum kezdete
4. A film teljes hossza és a beillesztett objektum kezdő időpontja
5. A szomszédos objektum neve
6. A beillesztett objektum lejátszásának ideje

4.26. ábra. Az objektum beillesztésének helye és jelölése a **Szerkesztőterületen**

4.27. ábra

Videóobjektumok esetében a lejátszás ideje megegyezik annak hosszával. A lejátszás idejét a grafikai objektumokhoz hasonlóan itt is megváltoztathatjuk, de itt az objektum mindkét szélét mozgathatjuk. Ha a bal oldalt mozgatjuk, akkor a részlet nem az elejétől kerül lejátszásra, ellenkező esetben pedig nem kerül lejátszásra a végéig. Ha a beállításokat pontosabban szeretnénk megadni, használjuk a **Nagyítás/kicsinyítés** eszközöket az **időskálán.**

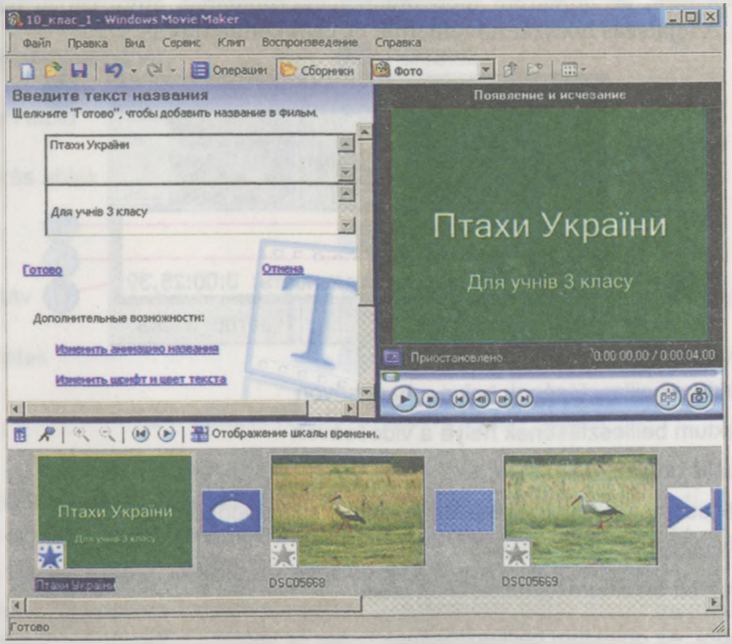
Amikor a videoobjektum határait mozgatjuk, a lejátszási ablakban láthatjuk is az aktuális részletet. Ilyen módon láthatjuk is a szerkesztési művelet eredményét. A folyamat megfordítható, vagyis a határvonalakat az ellenkező irányban is mozgathatjuk. Ugyanakkor a lejátszás ideje nem haladhatja meg a fájl lejátszási idejét.

A hangobjektumok szerkesztése a videókhoz hasonló módon történik, csak azokat a hang- vagy zenei sávon kell elhelyezni.

A videó- és hangobjektumok szerkesztése (kezdési időpontjának és hosszának megváltoztatása) során megtörténik a videosáv és a hangsáv ***szinkronizálása*** is.

A szerkesztés eredményét időről időre le kell ellenőrizni a lejátszás ablakban.

A ***szerkesztés következő lépése*** a feliratok és kommentárok elhelyezése. A film bevezető részének feliratait láthatjuk a 4.28. ábrán.



4.28. ábra. Feliratozás **Windows Movie Maker**-ben

A filmen az alább ismertetett módon helyezhetünk el feliratokat:

1. Hajtsuk végre az ***Eszközök – Cím és stáblista*** parancsot.
2. A megnyíló panelen válasszuk ki, **Hol szeretnénk elhelyezni a feliratot.** A feliratot a film elejére, végére, valamint a kijelölt kockán vagy az előtt helyezhetjük el.
3. A megnyíló beviteli mezőbe írjuk be a megfelelő szöveget.
4. Amennyiben szükséges, a nem grafikus objektumokon elhelyezett felirat esetében a **Betűtípus és szín** hivatkozásra kattintva adjuk meg a *háttérszínt, betűszínt, betűtípust,* a **Cím animációjának módosítása** linkre kattintva pedig rendeljünk animációs hatást a felirathoz.
5. A **Lejátszás** gombra kattintva győződjünk meg arról, hogy minden beállítást jól végeztünk el.
6. Kattintsunk a **Kész, cím hozzáadása** a filmhez hivatkozásra.

A létrehozott feliratok vagy a videósávra vagy pedig a feliratsávra kerülnek. A feliratsávra azok a feliratok kerülnek, amelyeket egy grafikai elemen helyeztünk el. Más esetben a feliratok a videósávon helyezkednek el.

Amennyiben az szükséges, a fenti eljárást megismételve újabb feliratokat helyezhetünk el a filmen belül vagy annak végén.

A felirat lejátszásának ideje alapértelmezés szerint 3,5 s, amit a grafikus objektumoknál megismert módon megváltoztathatunk. A filmen belül elhelyezett feliratok esetében azok jobb és bal oldala egyaránt áthúzható.

Az objektumok sorrendjét a kockák áthúzásával tetszőlegesen megváltoztathatjuk.

Az objektumok áthelyezése után mindig ajánlott megnézi annak eredményét a lejátszás ablakban.

Ha minden objektumot elhelyeztünk és a lejátszási idejüket beállítottuk, ***negyedik lépésként*** a filmet kommentárokkal és zenei kísérlettel látjuk el.

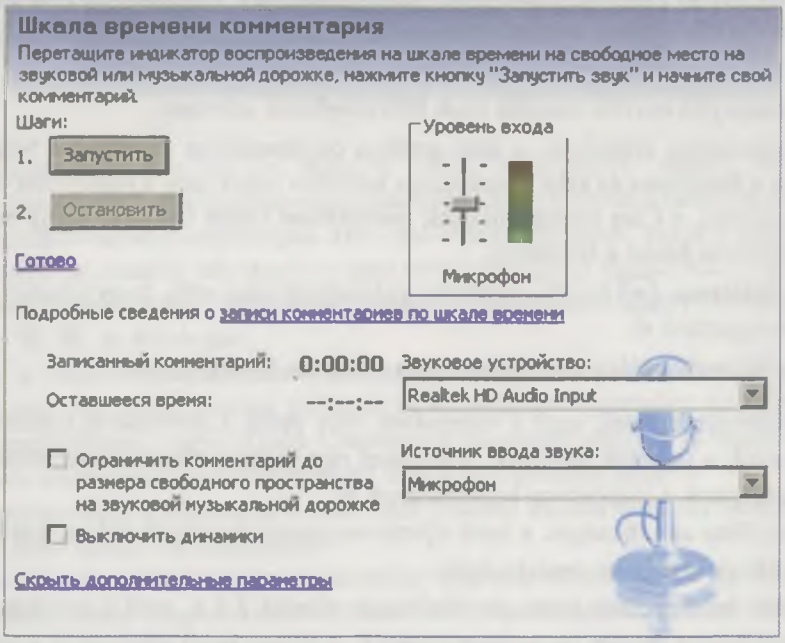
A kísérőzenét és a kommentárokat a **Hangrögzítő** segítségével fájlba menthetjük, majd a már megismert módon elhelyezhetjük a megfelelő sávon.

A kommentárt rögzíthetjük a **Windows Movie Maker** eszközeivel is, ha végrehajtjuk az ***Eszközök – Kísérőszöveg az idősorhoz*** parancsot, vagy az idősor bal oldalán található ugyanilyen nevű gombra kattintunk.

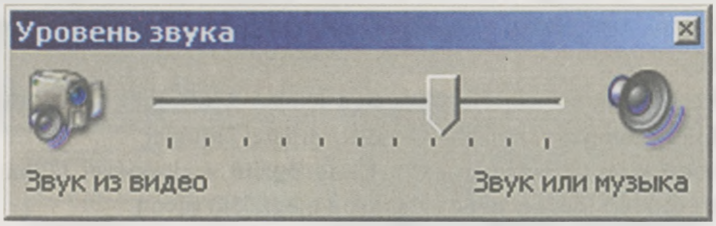
Ha a kísérőszöveg felvétele munkaterületen A **több beállítás megjelenítése** hivatkozásra kattintunk, lehetőségünk lesz a kísérőszöveg paramétereinek pontosabb beállítására. A megnyíló területen (4.29. ábra) lehetőségünk van például a **Kísérőszöveg korlátozására oly módon, hogy az elférjen a hang/zene sávon rendelkezésre álló helyen,** ha az azonos nevű jelölőt bekapcsoljuk. Ennek célja az, hogy a kísérőszöveg ne keveredjen a videó saját hangjával. Ha a jelölőt bekapcsoljuk, a program automatikusan leállítja a kísérőszöveg felvételét a megfelelő pillanatban.

Ha bekapcsoljuk a **Hangszóró elnémítás**a jelölőt, elkerülhetjük, hogy a felvételt torzítsa a hangsáv esetleges lejátszása. Hasznos lehet a **Bemeneti szint** beállítása a csúszka segítségével.

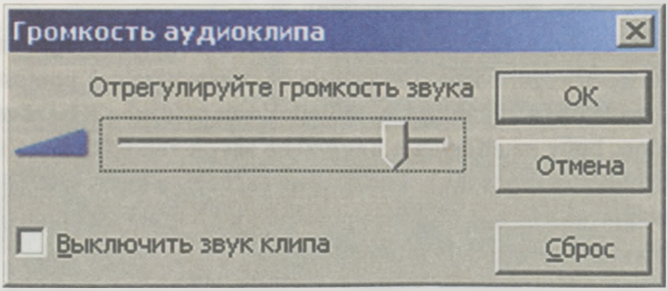
A felvételt a **Kísérőszöveg lejátszása** gombbal indítjuk és a **Kísérőszöveg leállítása** gombbal állítjuk le. A felvételt a **Kész** hivatkozásra kattintva helyezzük el az idősávon.



4.29. ábra. A **Kísérőszöveg felvétele** munkaterület



4.30. ábra. A **Hangerő szabályozása** ablak



4.31. ábra. Az **Audioklip hangereje** ablak

A létrehozott kísérőszöveg felkerül a hangsávra.

Ezután össze kell hangolni a különböző részletek hangerejét. A program erre több eszközt is biztosít. Az **Eszközök** menü **Hangszintek** pontjára kattintva megnyílik a 4.30. ábrán látható párbeszédablak, amelynek segítségével a teljes videó hangerejét szabályozhatjuk.

Ha egy konkrét részlet hangerejét szeretnénk szabályozni, használjuk a ***Klip – Hang – Hangerő*** parancsot. Az **Audioklip hangereje** párbeszédablakban (4.31. ábra) elvégezhetjük a beállítást.

Ha egy klip hangját ki szeretnénk kapcsolni, jelöljük ki azt a **Szerkesztőterületen,** majd hajtsuk végre a ***Klip – Hang – Elnémítás*** parancsot, vagy az objektum helyi menüjét.

A hangerő beállítását elvégeztük, célszerű megvizsgálni a szerkesztés eredményét a lejátszás ablakban.

Ötödik lépésként a videóeffektusok és a videóáttűnések beállítása következik. Ezt **Történet megjelenítése** módban végezhetjük el.

Videóeffektusok hozzáadása céljából nyissuk meg a **Gyűjtemények** panelen a **Videóeffektusok** mappát, majd húzzuk át a kiválasztott hatást a megfelelő ★ kockára. Egy kockához több effektust is rendelhetünk. A lejátszás ablakban megtekinthetjük a szerkesztés eredményét. Ha az effektust el szeretnénk távolítani, kattintsunk a megfelelő effektusra a ★ kockán,majd üssük le a **Delete** gombot.

Ha áttűnést szeretnénk elhelyezni két részlet között, a **Gyűjtemények** panel **Videóáttűnések** mappájából húzzuk át a kiválasztott hatást a két részlet közötti téglalap alakú területre. A szerkesztés eredményét most is a lejátszás ablakban ellenőrizhetjük. A hatás eltávolítása céljából jelöljük ki azt, majd üssük le a **Delete** gombot.

Ne felejtsük el a szerkesztés eredményét ellenőrizni a lejátszás ablakban.

Utolsó lépésként **fájlba kell menteni a filmet.** E célból:

1. Hajtsuk végre ***a Fájl – Filmfájl mentése*** parancsot.
2. A **Mentés varázsló** első ablakában válasszuk ki, hogy a filmet:
   - a felhasználó számítógépére szeretnénk menteni;
   - CD-re szeretnénk felírni;
   - el szeretnénk küldeni e-mail-ben;
   - weblapon szeretnénk elhelyezni;
   - DV-kamerára szeretnénk felírni.
3. Kattintsunk a **Tovább** gombra.
4. Amennyiben helyi gépre mentünk, a következő ablakban adjuk meg a fájl nevét és a mentés mappáját, majd kattintsunk a **Tovább** gombra.
5. Adjuk meg film minőségét meghatározó paraméterek értékét, majd kattintsunk a **Tovább** gombra.
6. Várjuk meg, amíg a mentés megtörténik.

A film WMV formátumban lesz elmentve. A fájlt ezentúl ugyanúgy használhatjuk, mint bármelyik más fájlt, tehát lejátszhatjuk, másolhatjuk, konvertálhatjuk, prezentációban helyezhetjük el stb.

Ellenőrző kérdések

1°. Mi a videofilm? Miben különbözik a mozifilmtől?
2°. Hogy történik a forgatókönyv kidolgozása?
3°. Ismertessük a videó létrehozásának folyamatát?
4°. Hogyan hoznánk létre egy családi videó forgatókönyvét?
5°. Mi a kulcskocka? Milyen adatokat tartalmaz?
6°. Mit értünk a montírozás alatt?
7°. Milyen tevékenységek előzik meg a montírozást **Windows Movie Maker**-ben?
8°. Mire szolgálnak **Windows Movie Maker**-ben a **Videóátmenetek, Videóhatások** és **Gyűjtemények** mappák?
9°. Milyen üzemmódjai vannak **Windows Movie Maker**-ben a **Szerkesztőterület**-nek?
10°. Milyen rendeltetése van az egyes sávoknak **Idősor módban**?
11°. Mit tartalmaz a projektfájl **Windows Movie Maker**-ben?
12°. Hogy helyezünk el grafikai objektumokat a videofilmen?
13°. Hogy helyezünk el hangobjektumokat a videón?
14°. Hogy tilthatjuk le a videó hangsávjának lejátszását?
15°. Milyen sajátosságai vannak a grafikai, hang- és videóobjektumok lejátszásának?
16°. Hogyan történik a feliratok elhelyezése a videón? Milyen lehetőségeink vannak a feliratok elhelyezése során?
17°. Hogy formázzuk a szöveges objektumokat **Windows Movie Maker**-ben?
18°. Hogy helyezünk el kommentárokat és zenei kíséretet a videón?
19°. Ismertessük a videofájl mentésének folyamatát **Windows Movie Maker**-ben.
20*. Hasonlítsuk össze a **PowerPoint 2007**-ben és a **Windows Movie Maker**-ben létrehozott prezentációk lehetőségeit!
21*. Létrehozhatunk-e **PowerPoint2007**-ben videóhoz hasonló prezentációt **PowerPoint2007**-ben? A választ indokoljuk meg!

Végezzük el a feladatokat!

1°. Hozzuk létre az „Ukrajna madarai" filmet a fenti forgatókönyv szerint. A szükséges fájlok a **Тема 4\Завдання 4.3.** mappa **Відео, Звук** és **Фото** mappákban találhatók. Mentsük a videót **4.3.1.wmv** néven.

2°. Hozzunk létre **Windows Movie Maker**-ben egy időalapú prezentációt Leszja Ukrainkáról. Használjuk fel a **Тема 4\Завдання 4.3\Українка** videófilmet. A prezentáció tartalmazzon bevezető képkockákat és kísérőszöveget. Keressünk kísérőzenét, amely a prezentáció teljes lejátszási ideje alatt hallható legyen a háttérben.

3°. Hozzunk létre **Windows Movie Maker**-ben egy időalapú prezentációt a XX, század elejének ukrán állami vezetőiről. Használjuk fel a 10. osztályos történelemkönyv anyagait, valamint a **Тема 4\Завдання 4.3\Скоропадський** és **Тема 4\Завдання 4.3\Петлюра Mappák**-ban található fájlokat. A kísérőszöveget és a zenei aláfestést hozzuk létre önállóan. Mentsük a fájlt **4.3.3.wmv** néven.

15. számú gyakorlati munka **Videófilm (időalapú prezentáció) készítése**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Hozzunk létre **Windows Movie Maker**-ben egy környezetvédelmi témájú videoklipet!
2. Hozzunk létre forgatókönyvet, figyelembe véve az alábbiakat!
   - 1. jelenet. **Cím**
     - cím – *Óvjuk hazánk természetét*, alcím – *„Oktatófilm;*
     - háttér – *sárga*, betűszín – *sötétzöld;*
     - animáció – *videó hatás, a szövegre;*
     - lejátszás ideje – *4,40 s;*
     - hang – *nincs;*
     - átmenet – *körök.*
   - 2. jelenet. **Fotó a szemétről.**
     - fájl – *смитмя 01.jpg;*
     - felirat – *„Mit tegyünk, hogy ilyen ne legyen?;*
     - felirat kezdete – *a felirat a kép megjelenése után 2 s-mal jelenik meg;*
     - lejátszás ideje – *3 s;*
     - animáció – *szivárványos;*
     - hang – *FINCL_02.MID; lejátszás ideje 5 s;*
     - átmenet – *forgószél.*
   - A következő jelenetekben mutassunk be egy környezetvédelmi problémát és annak lehetséges megoldásait. A videoklip hossza legyen 3–5 perc.
3. A videó készítése során használhatjuk a **Тема 4\Завдання 4.3\Екологія** mappa fájljait. Felhasználhatunk egyéb forrásokat is.
4. Szükség esetén végezzük el a videó- és audióadatok szerkesztését és szinkronizálását!
5. Mentsük a videót **gyakorlati15.wmv** néven!

4.4. A számítógépes kiadványszerkesztés alapjai. A Microsoft Office Publisher 2007

1. Milyen ismereteink vannak a nyomdatechnikáról? Milyen nyomtatott termékeket állítanak elő a nyomdaipari vállalatok?
2. Mikor találkoztunk először nyomtatott kiadványokkal?
3. Hogyan osztályozzuk a szövegszerkesztőket? Mik azok a kiadványkezelő rendszerek?
4. Mi a sablon? Hogyan használjuk a sablonokat **Word 2007**-ben és **Power Point 2007**-ben?

A számítógépes kiadványok típusai és azok rendeltetése

A publikáció (*publicatio* – közzététel) alatt valamely kiadvány megjelentetését értjük. A kiadványt magát is publikációnak nevezzük.

A megjelentetés módjától függően megkülönböztetünk ***nyomtatott*** és ***elektronikus*** kiadványokat. A nyomtatott kiadványok közé soroljuk a könyveket, brosúrákat, folyóiratokat, újságokat, bulletineket, füzeteket, nyomtatott kártyákat. stb. Az elektronikus kiadványok alatt az interneten elhelyezett, rádióban elhangzott vagy televízióban bemutatott dokumentumokat értjük.

A nyomtatott kiadványokat hosszú ideig csak nyomdákban készítették. A poligráfia (a görög πολὺ – sok, sokat, γραφειν – írni szavakból) volt az az iparág, amelynek termékei voltak a nyomtatványok. A kiadványokat a kiadókban készítették elő nyomtatásra. A nagy példányszámú nyomdai sokszorosítás ma is hasonló módon történik. A kiadók elkészítik a könyvek, újságok, folyóiratok makettjét, a nyomdaipari vállalatok pedig nyomtatják, szükség esetén bekötik azokat.

A személyi számítógépek és a nagy teljesítményű lézer- és tintasugaras nyomtatók elterjedése lehetővé tette az úgynevezett ***asztali kiadványszerkesztő*** rendszerek elterjedését. Az ilyen rendszerek hardvereszközei a személyi számítógép, a nyomtató (printer, fénymásoló), az adatbeviteli eszközök (szkennerek), szoftvereszközei pedig a kiadvány makettjének elkészítésére szolgáló programok. Ezt mind elhelyezhetjük a felhasználó asztalán, innen az elnevezése: **desktop publishing** (asztali kiadványszerkesztő) rendszerek.

Ahogy azzal a 9. osztályban már megismerkedtünk, a kiadványok makettjének elkészítésére szolgáló szoftvereket ***kiadványkezelőknek*** nevezzük. Ilyen kiadványszerkesztő a **QarkXPress,** a **Corel Ventura,** az **Adobe PageMaker,** az **Adobe InDesign.**

Ezek professzionális programok, amelyeket leginkább a nagyméretű (több száz oldalas, több ezres példányszámú) kiadványok előkészítésére használnak. Kisvállalatok, oktatási intézmények számára kiadványok előkészítésére megfelelhet a szövegszerkesztő, vagy egy egyszerűbb felületű kiadványszerkesztő, mint a **Microsoft Publisher,** a **Page Plus,** az **Avery DesignPro,** a **Paraben's Label Builder** vagy a **Publish It!.**

A szövegszerkesztőtől eltérően a kiadványszerkesztők rendeltetése nem a szöveges objektumok bevitele és szerkesztése, hanem a tördelés.

A *tördelés* a szöveges és grafikus objektumok oldalakon (hasábokon) történő elhelyezésének folyamata, amelynek meg kell felelnie a kiadvány tervezési és műszaki követelményeinek.

A tördelés rendeltetése szerint lehet ***könyv- és folyóirat-tördelés, újságtördelés és eseti tördelés.*** Az eseti tördelést kis példányszámú és oldalszámú kiadványok: hirdetések, űrlapok, prospektusok, névjegykártyák tördelésénél alkalmazzák.

A számítógépes technológiák segítségével elkészített kiadványokat nevezzük ***számítógépes kiadványoknak.***

Az asztali kiadványszerkesztők legfontosabb szolgáltatásai

Az asztali kiadványszerkesztők biztosítják:

- a szöveges és grafikus objektumok elhelyezésének automatikus elhelyezését a kiadványon belül (***makettezés***);

- más alkalmazói szoftver segítségével készített szöveg és grafika importálását;
- a szöveg feldolgozását stílusok, fejlett karaktertávolság és sorközkezelés alkalmazását;
- listák és hivatkozások kezelését;
- betűtípusok nyomdai szabványoknak megfelelő kezelését;
- a kiadvány reprodukálást a képernyőn annak az eszköznek megfelelően, amelyen a kiadvány nyomtatásra kerül;
- a kiadvány nyomdai alkalmazásra alkalmas fájlba történő mentését;
- a kiadvány nyomtatását különböző alapszínekben (színelválasztás);
- a kiadvány nyomtatását különböző nyomtatókon.

A fenti lista a nyomdai lehetőségek, a szoftver rendeltetése függvényében változhat.

A Microsoft Office Publisher 2007 kiadványszerkesztő általános jellemzése

A **Microsoft Office Publisher 2007** kiadványszerkesztő rendeltetése, ahogy arról már szóltunk, kis példányszámú számítógépes kiadványok elkészítése kisvállalati és személyes használatra.

A **Publisher 2007** a professzionális kiadványszerkesztőkhöz képest egyszerűsített funkcionalitással bír. Eseti tördelést biztosít többféle sablon felhasználásával. Kétféle üzemmódja van: a ***nyomtatott*** és az ***elektronikus*** kiadványkészítés.

A **Publisher 2007** alapértelmezett formátuma a **PUB.** Ha a fájlt valamely régebbi verzió (1998 és 2000) formátumában mentjük, előfordulhat, hogy a **2007**-es verzióval készített kiadványok nem minden beállítása marad meg.

A **Publisher 2007** alapobjektuma a **kiadvány**, ami oldalak gyűjteménye. Az oldalakon szöveges és grafikus objektumokat helyezhetünk el, ami elektronikus kiadványok esetében videóobjektumokkal és hiperhivatkozásokkal egészül ki. A kiadvány objektumainak tulajdonságai gyakorlatilag nem különböznek a **Word 2007** és a **PowerPoint 2007** objektumainak tulajdonságaitól. A kiadványszerkesztő minden objektumon meghatározott szerkesztési és formázási műveletet képes végezni.

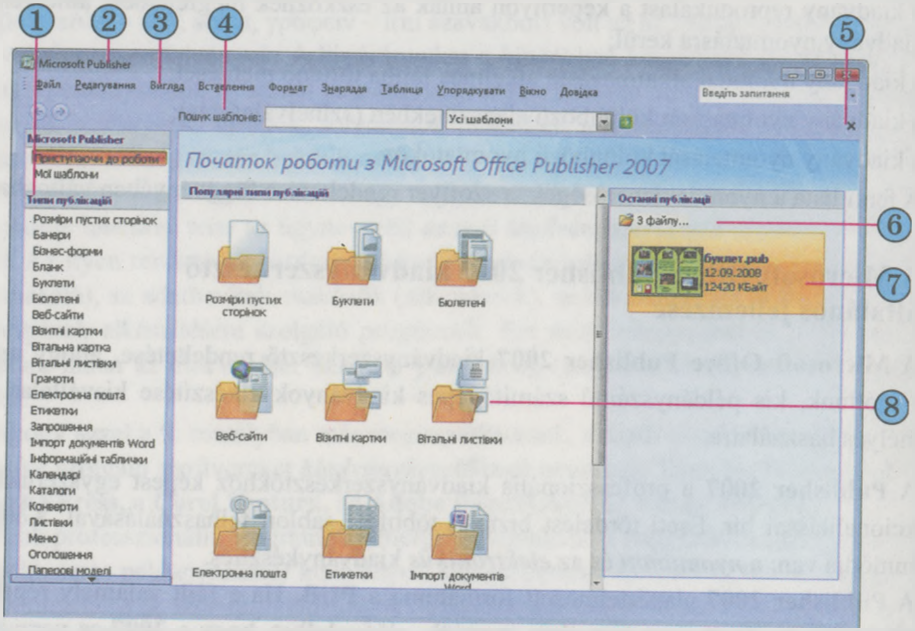
A Publisher 2007 indítása és felhasználói felülete

A **Publisher 2007**-et többféleképpen indíthatjuk. Ezek közül a legelterjedtebbek az alábbiak:

- a ***Start – Minden program – Microsoft Office – Publisher 2007*** paranccsal;
- az **Asztal** parancsikonjára vagy a **Gyorsindítás panel** gombjára kattintva;
- duplán kattintva egy **Publisher** kiadvány ikonjára .

Ha az első két módszert választjuk, megnyílik a program 4.32. ábrán látható induló ablaka, amely három részre van felosztva.

Az ablak bal oldalán (4.32., ábra) a **Publisher**-ben létrehozható kiadványok listáját találjuk. Ha ezek egyikére, például a **Programfüzetek**-re kattintunk, a középső munkaterületen megnyílnak az adott kiadványtípushoz tartozó sablonok. Ekkor a jobb oldali részben a **Legutóbb használt** kiadvány helyett megnyílik a választott kiadványtípus testreszabására szolgáló ablak. Itt pontosíthatjuk a kiadvány beállításait.



1. **Kiadványtípusok** listája
2. **Címsor**
3. **Menüsor**
4. Eszköztár
5. A katalógus bezárása gomb
6. **Fájlból** gomb
7. A legutóbbi kiadvány
8. Kiadványtípusok terület

4.32. ábra. A **Publisher 2007** nyitóablaka

A szükséges sablonok kiválasztását meggyorsíthatjuk a **Sablonok keresése** mező alkalmazásával. A keresést végezhetjük a *helyi számítógépen vagy a Microsoft Office Online webhelyen.*

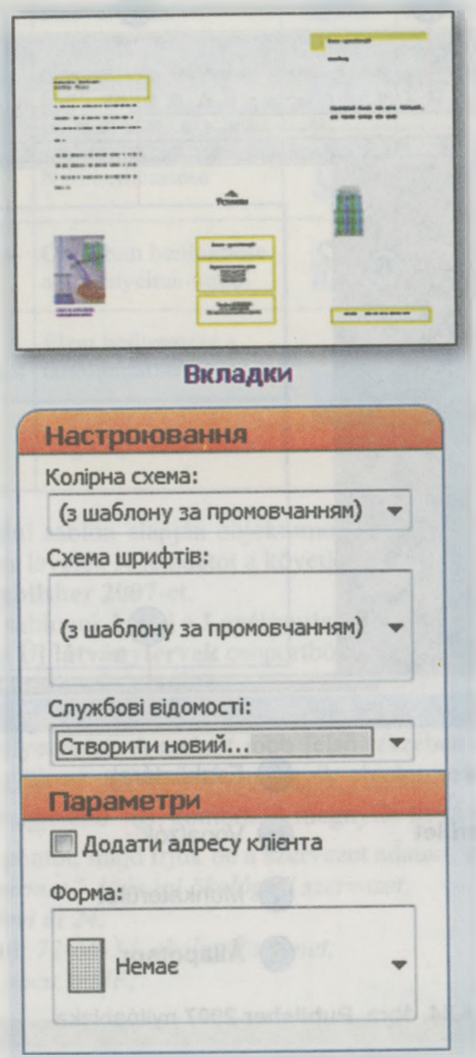
Új kiadvány létrehozása

Az új kiadvány létrehozását elkezdhetjük a program nyitóablakában, amellyel fentebb már megismerkedtünk.

A kiadványtípus és a sablon kiválasztása után el kell végezni a tulajdonságok beállítását (4.33. ábra), ahol kiválaszthatjuk a színsémát, a betűtípussémát (vagy megtarthatjuk az alapértelmezetteket), megadhatunk üzleti adatokat (egyéni és vállalati adatokat), amelyek automatikusan bekerülnek a sablonba, majd a kiadványtípustól függően egyéb

paramétereket adhatunk meg. A **Létrehozás** gombra kattintva átléphetünk a kiadvány szerkesztésére.

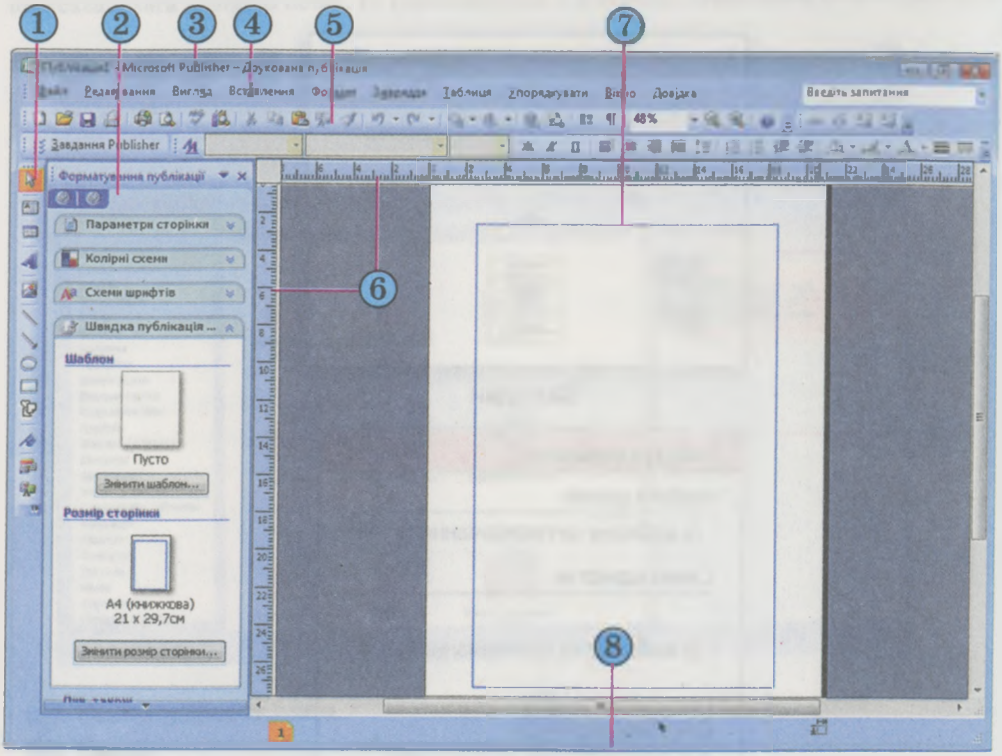
A felsorolt tevékenységek elhagyhatók, ha a katalógus bezárása gombra kattintunk (4.32., 5 ábra). A nyitóablak ekkor bezáródik, majd megnyílik egy valamelyest megváltozott nyitóablak (4.34. ábra), amelynek felülete eltér a **Microsoft Office 2007** csomag már megszokott felületétől.



4.33. ábra. A kiadvány testreszabása terület

A program nyitóablakában az **Objektum eszköztár,** a feladat **Munkaterület kiadvány formázása** eszköztára segítségével a felhasználó önállóan hozza létre a kiadvány makettjét. A ***Fájl – Létrehozás*** parancsot végrehajtva vagy a **Kiadvány formázása** eszköztár **Sablon módosítás** gombjára kattintva visszatérhetünk a sablon alapján történő kiadványtervezéshez.

Ahogy a **PowerPoint 2007**-ben, a **Publisher 2007**-ben is létrehozhatunk kiadványokat már létező kiadványok alapján. Ehhez a nyitóablakban kattintsunk a **Fájlból** hivatkozásra, majd válasszuk ki a fájlt, amely alapján az új kiadványt létre szeretnénk hozni. Ugyanezt elérhetjük a **Fájl – Megnyitás**, vagy a ***Fájl – Új dokumentum – Fájlból*** paranccsal is.



1. **Objektum** eszköztár
2. **Feladatok munkaterület**
3. **Címsor**
4. **Menüsor**
5. Eszköztárak
6. Vonalzók
7. Munkaterület
8. **Állapotsor**

4.34. ábra. **Publisher 2007** nyitóablaka

Objektumok beillesztése és szerkesztése

Az objektumok beillesztése leginkább az **Objektumok** eszköztár (4.6. táblázat) eszközeinek segítségével történik. A művelek többsége hasonló a **PowerPoint 2007**-ben megismertekre. Ahogy a **PowerPoint 2007**-ben, a szöveges objektumok szövegdobozokban kerülnek elhelyezésre.

4.6. táblázat. **Az Objektumok eszköztár gombjainak rendeltetése**

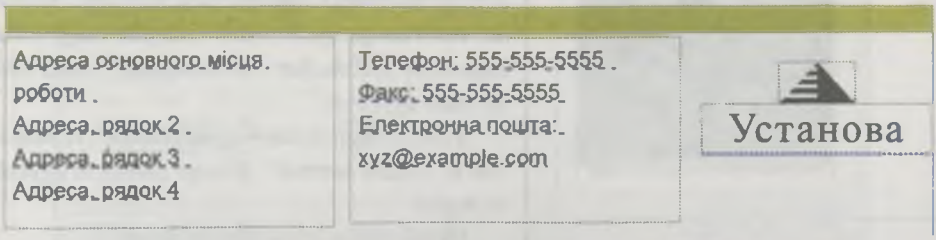
Gomb	*Név*	*Rendeltetés*	*Gomb*	*Név*	*Rendeltetés*
	Objektum kijelölése	A kiadvány objektumainak kijelölése		Szövegdoboz	Szövegdoboz beillesztése a kiadványba
	Táblázat beszúrása	Táblázat elhelyezése a kiadványban		**WordArt** beszúrása	**WordArt** objektum beillesztése
	Képdoboz	Grafikai objektum beillesztése		Vonal	Egyenes vonal beillesztése
	Nyíl	Nyíl beillesztése		Ellipszis	Ellipszis beillesztése
	Látványelem-tár	Objektum beillesztése a látványelem-tárból		Alakzatok	Alakzat beillesztése
	Elem a tartalomtárból	Elem beillesztése a tartalomtárból		Könyv-jelző	Könyvjelző beszúrása
	Téglalap	Téglalap beillesztése			

Hozzuk létre például sablon alapján objektumok elhelyezésével egy ifjúsági ökológiai szervezet hivatalos levelét. A feladatot a következőképpen kezdjük:

1. Indítsuk el a **Publisher 2007**-et.
2. Válasszuk ki a sablonok közül a **Levélpapírfejléc** kategóriát.
3. Válasszuk ki az **Új látványtervek** csoportból a **Színes szalag** sablont.
4. Kattintsunk a **Létrehozás** gombra.

Létrehoztuk a kiadvány makettjét, amely helyőrzőket tartalmaz a szöveges és grafikai információk elhelyezésére. A makett jobb felső részében néhány objektumot találunk (4.35. ábra). Vigyük az egérmutatót az egyik objektumra és kattintsunk az ekkor az objektum mellett megjelenő gombra. A megnyíló listában válasszuk ki az **Üzleti adatok szerkesztése** pontot, majd írjuk be a szervezet adatait, például:

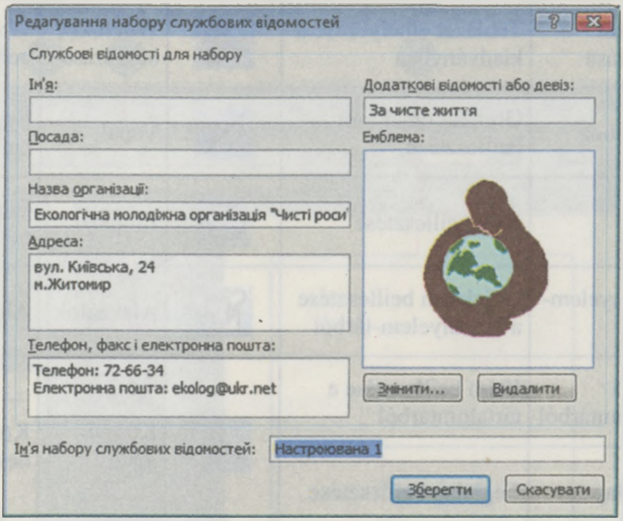
- Cégnév: *„Tiszta harmat" ifjúsági ökológiai szervezet;*
- Cím: *Zsitomir, Kijevi út 24.*
- Telefon, fax, e-mail: *72-66-34, ekolog@ukr.net;*
- Embléma: *Чистий ранок.WMF;*



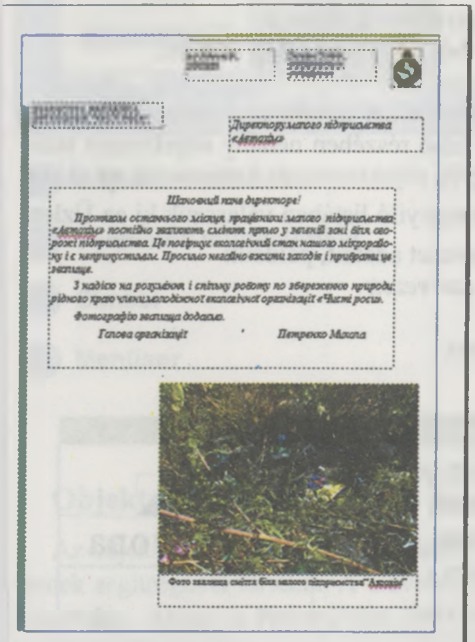
4.35. ábra. Objektumok a szervezet levelének makettjén

A cég emblémáját a **Módosítás** gombra kattintva változtathatjuk meg, ami után válasszuk ki a szükséges fájlt, jelen esetben a **Тема 4\Завдання 4.4\Чисті роси.WMF.**

A 4.36. ábrán ezt az ablakot láthatjuk a szerkesztés elvégzése után. Kattintsunk a **Mentés,** majd a **Kiadvány frissítése** gombra.



4.36. ábra. Az **Üzleti adatok szerkesztése** ablak



4.37. ábra. A levél makettje

Az üzleti adatok el lesznek mentve, a későbbiek során automatikusan ezek kerülnek be a sablonokba. A levél címzettjének beírása céljából helyezzünk el egy szövegdobozt az embléma alá, majd írjuk bele a szöveget, például: *„Az Autochem kisvállalat igazgatójának”*. A levél szövegét szintén szövegdobozban helyezzük el:

„Tisztelt igazgató úr!

Az utóbbi hónapok során az Önök vállalatának dolgozói sorozatosan a vállalat kerítése melletti zöld övezetbe helyezik el a szemetet. Ez lényegesen rontja városrészünk ökológiai helyzetét, ezért megengedhetetlen. Kérjük, hogy minél hamarabb intézkedjen a felhalmozott szemét eltakarítása ügyében.

A lehető legteljesebb együttműködés reményében a „Tiszta harmat” ifjúsági ökológia egyesület tagjai.

Mellékeltük a szemétlerakó képét.

A szervezet vezetője, Mikola Petrenko.”

A szöveg alatt helyezzük el a **Тема 4\Завдання 4.4\Сміття.jpg.** fájlt. E célból az **Objektumok** eszköztáron kattintsunk a **Képdoboz** eszközre, a megnyíló ablakban kattintsunk a **Kép fájlból** pontra, majd rajzoljunk körül egy téglalap alakú területet, ahol a képet el szeretnénk helyezni. Az ezután megnyíló párbeszédablakban válasszuk ki a szükséges fájlt.

A kép alatt helyezzünk el egy képfeliratot, például: *Az Autochem vállalat szemétlerakója,* majd a **Csoportba foglalás** gombra kattintva foglaljuk a feliratot és a képet egy csoportba. A makett így kialakított képét láthatjuk a 4.37. ábrán.

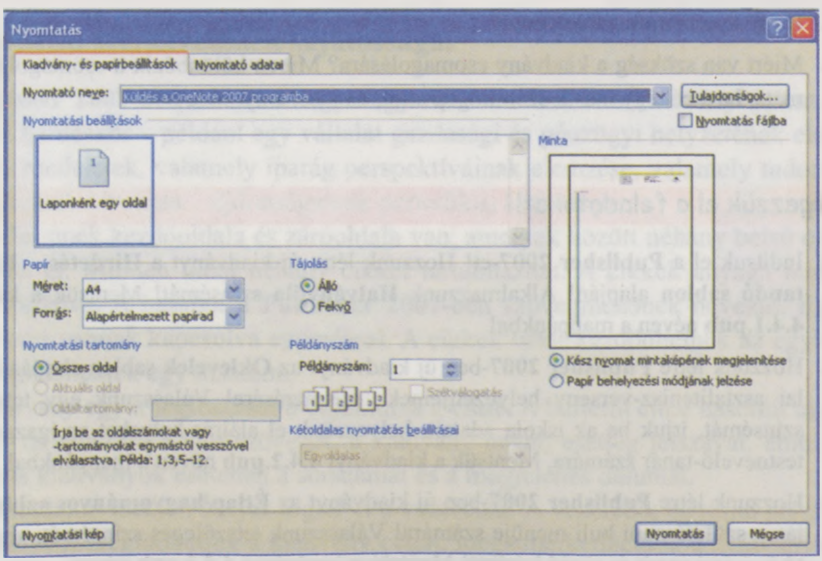
Kiadvány mentése és nyomtatása

A kiadvány mentése céljából hajtsuk végre a ***Fájl*** ⇒ ***Mentés*** parancsot, válasszuk ki a mentés helyét és adjuk meg a fájl nevét, majd kattintsunk a **Mentés** gombra.

A nyomtatás folyamata nem különbözik a **Word 2007** tanulmányozása során megismerttől. A nyomtatás paramétereit a ***Fájl*** ⇒ ***Nyomtatóbeállítás*** paranccsal adhatjuk meg. Ugyanezt megtehetjük a ***Fájl*** ⇒ ***Nyomtatás*** paranccsal is (4.38. ábra).

A **Word 2007** és a **Publisher 2007 Nyomtatás** ablakai abban különböznek, hogy utóbbinál létezik egy nyomtatási kép eszköz, továbbá némileg különböznek az egy lapra történő több oldal nyomtatásának beállítására szolgáló eszközök.

A **Publisher 2007**-ben a **PowerPoint 2007**-hez hasonlóan lehetőség van a fájlok csatolására. Ha a kiadványt átvisszük egy másik számítógépre, a csatolt objektumok nem kerülnek megjelenítésre. A probléma megoldása az objektumok csomagolása, amit a ***Fájl*** ⇒***Úticsomag*** ⇒ ***Átvitel más számítógépre /átvitel nyomdába*** paranccsal hajthatunk végre. Mindkét esetben a **Csomagolás varázsló** nyílik majd meg, ahol lépésről lépésre beállíthatjuk a csomagolás paramétereit (megmutatjuk a csatolt fájlok helyét és nevét), amelynek végén egy **ZIP** formátumú tömörített állományt kapunk. Ha másik számítógépre visszük a kiadványt, akkor nincs szükség például a betűtípusfájlok csatolására a tömörített állományhoz.



4.38. ábra. A **Nyomtatás** ablak

Ellenőrző kérdések

1°. Mi a kiadvány?

2°. Mi a nyomtatott kiadvány? Hozzunk fel példákat!

3°. Mi az elektronikus kiadvány. Hozzunk fel példákat!

4*. Milyen szerepet játszanak a kiadvány elkészítésében a nyomdaipari vállalatok, kiadók?

5°. Mi az asztali kiadványszerkesztő rendszer?

6*. Milyen elemei vannak az asztali kiadványszerkesztő rendszereknek? Mire szolgálnak?

7°. Mi a tördelés?

8*. Milyen típusai vannak a tördelésnek? Hol alkalmazzák ezeket?

9°. Mi a számítógépes kiadvány? Hozzunk fel példákat!

10*. Milyenek lehetnek a kiadványok szerkezetük szerint? Hol alkalmazzák ezeket?

11*. Ismertessük az asztali kiadványszerkesztők szolgáltatásait!

12*. Mi a **Publisher 2007** rendeltetése?

13*. Milyen formátumban menthetők a publikációk? Miben különböznek ezek?

14°. Ismertessük a **Publisher 2007** felületét!

15*. Ismertessük a **PowerPoint 2007** nyitóablakát!

16*. Hogy hozunk létre új kiadványt **Publisher 2007**-ben?

17*. Hasonlítsuk össze a szöveges és grafikai objektumok kezelését **PowerPoint 2007**-ben és **Publisher 2007**-ben! Mi közöttük a hasonlóság és mi a különbség?

18°. Mire szolgál a kiadványok objektumai melletti ⓘ gomb?

19°. Ismertessük a mentés folyamatát **Publisher 2007**-ben!

20°. Hogy nyomtatjuk a kiadványt?

21*. Miért van szükség a kiadvány csomagolására? Miben különbözik a csomagolás folyamata, ha a cél egy másik számítógép attól, ha a cél a nyomda?

Végezzük el a feladatokat!

1°. Indítsuk el a **Publisher 2007**-et! Hozzunk létre új kiadványt a **Hirdetés – Felmutatandó sablon** alapján! Alkalmazzunk **Halványlila** színsémát! Mentsük a kiadványt **4.4.1.pub** néven a mappánkba!

2*. Hozzunk létre **Publisher 2007**-ben új kiadványt az **Oklevelek** sablon alapján az iskolai asztalitenisz-verseny helyezettjeinek elismerésére! Válasszunk egy tetszőleges színsémát, írjuk be az iskola adatait, helyezzünk el aláírás-helyeket az igazgató és a testnevelő-tanár számára. Mentsük a kiadványt **4.4.2.pub** néven a mappánkba!

3*. Hozzunk létre **Publisher 2007**-ben új kiadványt az **Étlap/hagyományos sablon** alapján a születésnapi buli menüje számára! Válasszunk tetszőleges színsémát, vegyük fel rá 5–6 étel nevét és rövid leírását! Mentsük a kiadványt **4.4.3.pub** néven!

4°. Hozzunk létre **Publisher 2007**-ben a saját névjegykártyánkat! Helyezzük el rajta a fényképünket és az iskolánk adatait! Nyomtassuk ki a névjegykártyát, 10-et egy oldalra. Mentsük a kiadványt **4.4.4.pub** néven a mappánkba!

5*. Hozzuk létre a **Levélpapírfejléc/Elemes** sablon alapján iskolánk fejléces levélpapírját! A levélpapír tartalmazza az iskola nevét, címét, telefonszámát és emblémáját! A színsémát harmonizáljuk városunk/régiónk zászlajának színeivel! Mentsük a kiadványt **4.4.5.pub** néven a mappánkba!

4.5. Szöveges és grafikai objektumok kezelése Publisher 2007-ben. Információs bulletin és füzet készítése

1. Mi a dizájn alapja?
2. Milyen követelményeket támasztanak egy kiadvány színtervével szemben?
3. Mik a stílusok?
4. Hogy hozunk létre új kiadványt **Publisher 2007**-ben?

Információs bulletin készítése

Ahogy azt már tudjuk, a kiadványok kétféle objektumot tartalmazhatnak: szövegdobozokat és képeket. Vizsgáljuk most meg ezek kezelését és hozzunk létre egy információs bulletint **Publisher 2007**-ben.

A *bulletin* (a francia *bulletin* – közlemény, jelentés szóból) rövid hivatalos közleményt, hivatalos kiadványok gyűjteményét vagy bizonyos kérdéskör újdonságainak periodikus kiadványát jelenti.

A bulletin szerkezetének sajátosságai

Publisher 2007-ben a bulletin egy vagy több oldalból álló kiadványt jelent, amely bizonyos kérdéskör – például egy vállalat gazdasági és pénzügyi helyzetének elemzése, hivatalos rendeletek, valamely iparág perspektíváinak elemzése, valamely tudományterület cikkeinek kivonata – újdonságainak periodikus kiadványa.

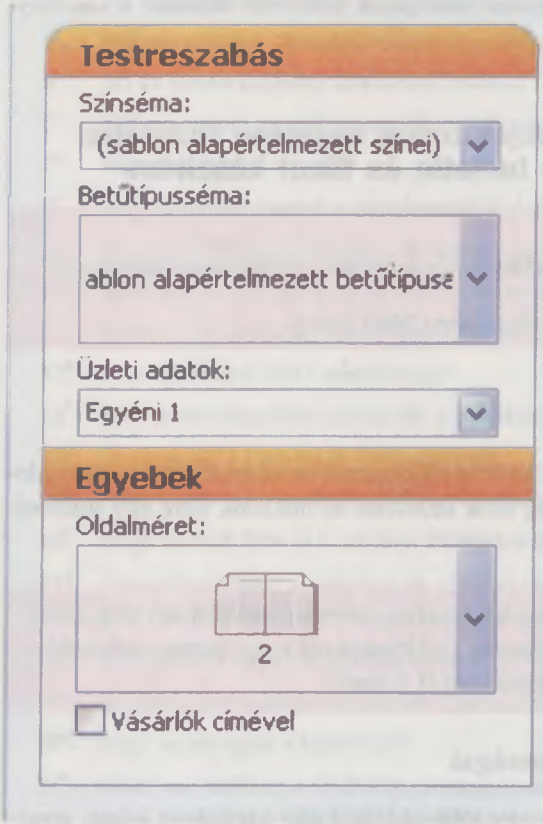
A bulletinnek kezdőoldala és záróoldala van, amelyek között néhány belső oldal helyezkedhet el. Minden oldal néhány cikket tartalmazhat. A cikkek anyaga hasábokba vannak tördelve. A hasábot a **Publisher 2007**-ben szövegmezőnek nevezik. Egy cikk mezői össze vannak kapcsolva egymással. A cikkek tehát kezdődhetnek az egyik oldalon és végződhetnek egy másikon.

A kiadvány egyik legfontosabb objektuma a **címe.** A bulletin címe hasonlít egy újság címéhez, tartalmazza a cím szövegét, a kiadvány nevét, esetleg jelszavát, emblémáját, periodikus kiadványok esetében a sorszámát és a megjelenés dátumát.

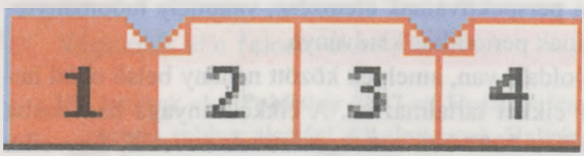
A címoldalon kiemelt helyet foglal el a vezércikk. A vezércikk egyfajta programadó cikk, amelyből megtudhatjuk a kiadvány célját, megismerkedhetünk a kiadványban tárgyalt problémákkal, amelyek a továbbiakban részletezve és pontosítva lesznek.

A záróoldalon foglal helyet a kolofon, ami a szerkesztők és tervezők nevét tartalmazza, valamint egyéb szolgálati adatot, mint a példányszám, a szerkesztőség és a nyomda címe.

Bulletin létrehozása sablon alapján



4.39. ábra. A kiadvány beállításai panel



4.40. ábra. A kiadvány oldalainak sorszáma

A bulletin létrehozásának folyamata a prezentációhoz hasonlóan a kiadvány szerkezetének és külalakjának megtervezésével kezdődik. Először is meg kell ismerkedni az adott kiadványtípus sablonjaival, meg kell határozni a kiadványban tárgyalandó kérdések körét, majd össze kell gyűjteni a kiadvány anyagát: nevet kell a kiadványnak adni, meg kell írni a cikkek szövegeit, ki kell keresni a szükséges illusztrációkat, grafikai elemeket stb. Ennek utána hozzáláthatunk a bulletin létrehozásához.

Publisher 2007-ben először is ki kell választanunk a megfelelő sablonkategóriát, például a **Hírleveleket**. A megfelelő sablon kiválasztása után el kell végezni a kiadvány tulajdonságainak pontos beállítását (4.39. ábra): ki kell választani a színsémát, a betűtípussémát, meg kell adni az üzleti adatokat. Ugyanezen a panelen be kell állítani az **Oldalak számát.** Vegyük figyelembe, hogy a nyomtatás alapértelmezetten kétoldalas, tehát ha egy oldalt választunk, akkor a bulletinnek két oldala lesz, ha pedig kettőt, akkor négy.

Ha elvégeztük a beállításokat, kattintsunk a **Létrehozás** gombra. Ezzel át is léptünk a bulletin makettjének szerkesztéséhez és formázásához.

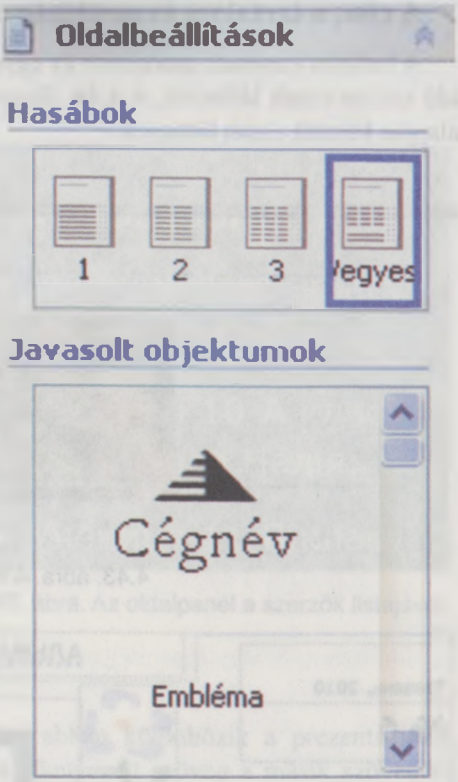
A program ablakának **Státuszsorában** láthatjuk a kiadvány oldalainak sorszámait (4.40. ábra). Hogy megtekintsünk egy adott oldalt, kattintsunk annak sorszámára. A belső oldalak (tehát most a 2. és 3.) együtt jelennek meg. Az aktuális oldal számát sötétebb szín jelöli.

Oldalbeállítások

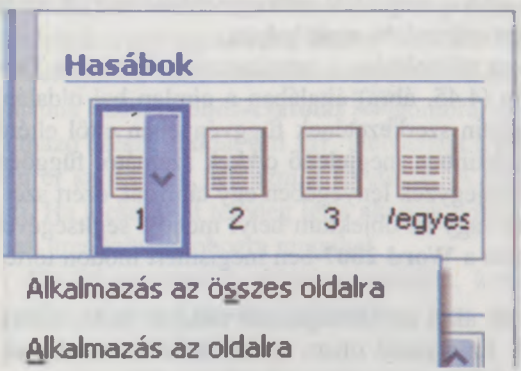
Kattintsunk az **Oldalbeállítások** melletti nyilakra a **Kiadvány** formázása munkaterületen. Ekkor megnyílik az azonos nevű ablak (4.41. ábra). A **Hasábok** mezőben beállíthatjuk, hány hasábos tördelést szeretnénk alkalmazni az aktuális oldalon vagy a kiadványban. E célból gördítsük le a megfelelő hasábszám melletti listát és válasszuk ki, hogy a beállítást az aktuális oldalra vagy az egész kiadványra szeretnénk érvényesíteni (4.42. ábra). Ha a **Vegyes** gombra kattintunk, az egyes cikkek különböző számú hasábra lesznek tördelve.

A **Színséma** munkaterületen megváltoztathatjuk a **Kiadvány** létrehozása során beállított színsémát, a **Betűtípusséma** munkaterületen pedig a kiválasztott betűtípussémát.

A **Hírlevél** munkaterületen megváltoztathatjuk a kiadvány sablonját és beállíthatjuk az oldalméretet. E célból kattintsunk az **Oldalméret módosítása** gombra. A megnyíló **Oldalbeállítás** ablakban választhatunk a szabványos oldalméretek közül vagy megadhatjuk egyéni beállításainkat. Ugyanitt állítjuk be a margóméreteket is.



4.41. ábra. Az **Oldalbeállítások** panel

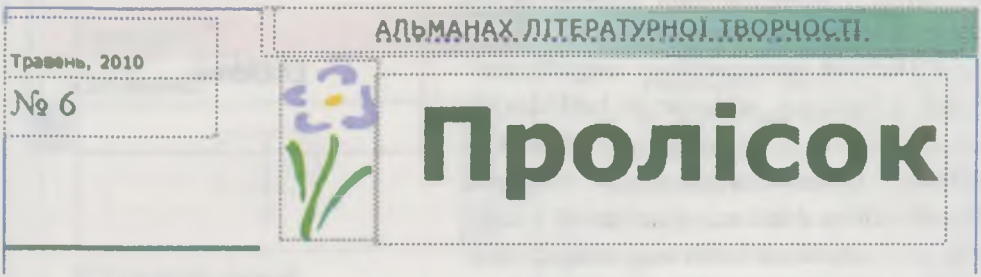


4.42. ábra. A hasábok számának beállítása

A cím, a tartalom és az oldalpanel szerkesztései

A bulletin címének szerkezete az egyes sablonokban hasonló. A 4.43. ábrán a **Nyáridő** sablon címét láthatjuk. A 4.44. ábrán a *«Пролісок» irodalmi almanach* az e sablon alapján készült címét láthatjuk.

4.43. ábra. A **Nyáridő** sablon címe

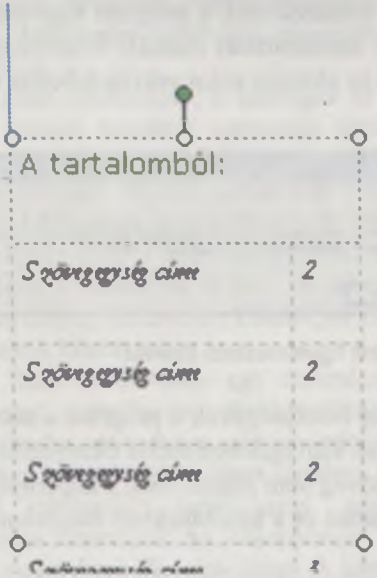


4.44. ábra. A **«Пролісок»** irodalmi almanach címe

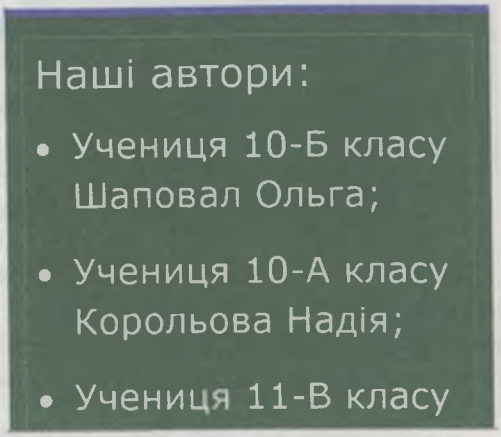
A címet úgy változtatjuk meg, hogy a sablon megfelelő szövegmezőibe beírjuk az új tartalmakat, töröljük a képeket és új képeket szúrunk be azok helyett.

A bulletin címoldala – különösen, ha az többoldalas – tartalmazza a kiadvány **Tartalomjegyzékét.** A tartalomjegyzék-sablon (4.45. ábra) általában a címlap bal oldalán, az első cikk mellett kap helyet, bár a bulletin szerkezetének függvényében ettől eltérő elhelyezés sem kizárt. A felhasználó a bulletinben megjelenő cikkek számától függően megváltoztathatja a sorok számát. A tartalomjegyzék lényegében egy táblázat, ezért szerkesztése és formázása is a **Táblázat** menü vagy az objektum helyi menüje segítségével történik. A szöveg szerkesztése és formázása a **Word 2007**-ben megismert módon történik.

A *Nyáridő* sablonban a tartalomjegyzék alatt az **Oldalpanelt** találjuk (4.46. ábra), amely a *Fontos közlemények* címet viseli. Ez a panel olyan rövid frázisokat tartalmaz, amelyek kiemelik a bulletinben megjelenő információk jellegzetességeit, felhívva ezekre olvasó figyelmét. Irodalmi újság esetében például információkat tartalmazhat azokról a szerzőkről, amelyek most publikálnak először. A tartalom szerkesztését a már megismert módon végezzük: beírjuk a megfelelő szöveget a szövegmezőkbe.



4.45. ábra. A tartalomjegyzék-sablon



4.46. ábra. Az oldalpanel a szerzők listájával

A bulletin cikkeinek szerkesztése

Szövegelhelyezés szempontjából a kiadvány abban különbözik a prezentációtól, hogy a kiadványban az egyik szövegdobozban elhelyezett szöveg a másik szövegdobozban folytatódhat. Ezt nevezzük a szöveg átfolyásának. A szöveg átfolyhat másik hasábba, de akár másik oldalba is. A program az **Átugrás a következő szövegdobozra** jel segítségével tudatja a felhasználóval, hogy az adott szövegdoboz tartalma egy másik szövegdobozban folytatódik. Ha erre kattintunk, akkor átlépünk abba a szövegdobozba, ahol a folytatást találjuk.

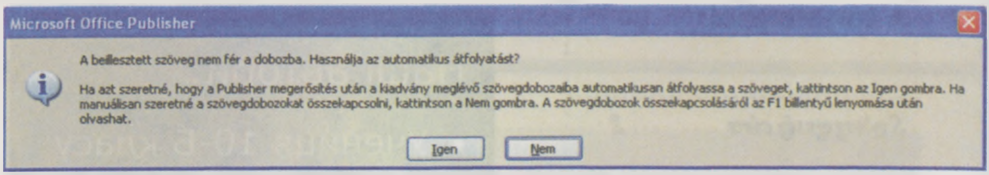
A sablonokban már be vannak állítva a szövegdobozok közötti átfolyások. Ha a felhasználó önállóan hozza létre a kiadvány makettjét, akkor kénytelen beállítani a szöveg átfolyását a szövegdobozok között. Ez a művelet a ***szövegdobozok csatolása.***

Jelöljük ki azt a szövegdobozt, amely a folytatandó szöveget tartalmazza, majd kattintsunk **Szövegdoboz-csatolás** gombra. Az egérmutató egy lefelé mutató nyilat tartalmazó csésze alakját ölti. Helyezzük a kurzort a fölé a szövegmező fölé, amelyikben el szeretnénk helyezni a folytatást. A kurzor egy megdöntött csésze alakját ölti. Ha a csatolást tényleg létre akarjuk hozni, kattintsunk a szövegdobozra. A csatolás csak üres szövegdobozra mutathat.

Ha fel szeretnénk oldani a csatolást, kattintsunk az **Előrecsatolás feloldása** gombra. A gomb nevével ellentétben nem az előző, hanem a következő szövegdoboz csatolása oldódik fel.

A cikk szövegét általában valamely szövegszerkesztőben, például **Word 2007**-ben készítjük elő és a **Vágólap** segítségével helyezzük el a szövegdobozokban. Ha a sablon szövegdobozára kattintunk, a több hasábban elhelyezett szöveg egésze kijelölésre kerül. A **Beillesztés** végrehajtása után a szövegdoboz teljes tartama le lesz cserélve. Ha a szö-

veg nem fér el a szövegdobozban (vagy csatolt dobozokban), a program figyelmeztető üzenetet jelenít meg (4.47. ábra) és felajánlja az **automatikus** csatolás lehetőségét. Ha ezt elfogadjuk, a program automatikusan elvégzi az aktuális utáni szövegdobozok csatolását mindaddig, amíg a szöveg el nem fér.



4.47. ábra. A szöveg túlcsordulására figyelmeztető üzenet

Ha nem kívánunk élni az automatikus csatolás lehetőségével, a program a szöveget a csatolt szövegdobozokban helyezi el.és az utolsó szövegdoboz mellé elhelyezi a **Túlcsordulás jelét** [A ...]. Az el nem helyezhető szöveg nem jelenik meg a képernyőn, de természetesen megmarad a számítógép memóriájában és a későbbiekben megjelenítésre kerülhet.

A felhasználó a következőképpen érheti el, hogy a teljes szöveg megjelenítésre kerüljön:

- törölheti a szöveg lényegtelennek ítélt részét;
- formázhatja a szöveget, megváltoztatva a betűméretet vagy a bekezdés-tulajdonságokat;
- megváltoztathatja a szövegdoboz méretét;
- egy újabb szövegdobozt csatol a cikkhez.

A kiadványban a fényképek és rajzok mellett képaláírást szokás elhelyezni. Az **Objektumok** eszköztár **Szövegdoboz** eszközére kattintva helyezzünk el egy szövegdobozt a kép mellett, majd írjuk bele a megfelelő szöveget. Ezután jelöljük ki a képet és a szövegdobozt, majd kattintsunk a **Csoportosítás** gombra. A program elvégzi az objektumok csoportosítását, ami után azok együtt mozognak a kiadványban, a képnek tehát felirata lesz.

A grafikus objektumok szerkesztését **Publisher 2007**-ben ugyanúgy végezzük, mint **Word 2007**-ben vagy **PowerPoint 2007**-ben. Az egyetlen lényeges különbség, hogy a beillesztett kép körbefuttatásának stílusa a **Word 2007**-ben a **Szöveggel egy, PowerPoint 2007**-ben a **Szöveg elé, Publisherben** pedig **Négyzetes.**

Publikációk létrehozásakor be kell tartanunk a szerzői jogokat. Valamennyi cikk és grafikus objektum esetén fel kell tüntetni azok szerzőjének nevét. Ha más kiadványból szeretnénk átvenni dokumentumokat, akkor ki kell kérni a szerzők engedélyét.

Füzet létrehozása

A **Füzet** olyan kiadvány, amely egy oldal hajtogatásával készül, amelyet úgy olvashatunk, hogy azt ne kelljen teljes egészében szétnyitni. Ilyen formátumban leginkább ismertetők, prospektusok, koncertprogramok, kiállítások, reklámok készülnek. Jellemzően A4-es méretben, fekvő tájolásban, háromhasábos tördeléssel készülnek.

A hasábok elválasztó-vonalai mentén hajtogatjuk. Jellemzően kétoldalas nyomtatásban készülnek.

A füzet elkészítése, a szöveges és grafikus elemek kezelése ugyanúgy történik, mint a bulletin esetében. Indítsuk el a **Publisher 2007**-et, az indulóablakban (vagy a ***Fájl – Létrehozás*** paranccsal) hozzunk létre új kiadványt, a sablonkategóriák közül válasszuk a **Brosúrát.** A füzet és a bulletin közötti lényegi különbség külalakjuk és rendeltetésük között van.

A füzet rendeltetése egy tanintézmény, szervezet, személy, áru, akció reklámozása. Figyelembe véve, hogy minden információ egyetlen lapra kerül, a füzet tartalma olyan rövid szövegeket tartalmazhat, amelyek pontosan ismertetik az objektumot, ezenfelül pedig képi illusztrációkat. A képek jó minőségűek kell legyenek és kevés, jól elkülöníthető objektumot kell tartalmazniuk.

4.48. ábra. Füzetcímlapok

A füzetnek általában van címlapja és hátlapja. A címlap tartalmazza a füzet címét, témáját, valamint a témára utaló képet vagy rajzot. A füzetet megjelentető személy vagy szervezet adatait is tartalmazhatja. A 4.48. ábrán néhány címlapot láthatunk.

A hátlapon a kiadványt megjelentető szerkesztőség vagy magánszemély elérhetőségeit találjuk.

Az interneten elég sok olyan füzetet megtalálhatunk, amelyeket érdemes megvizsgálni. Ehhez elég a keresőmezőbe beírni a reklámfüzet vagy brosúra szót, majd pedig a találati listát megvizsgálni.

Ellenőrző kérdések

1°. Mi a bulletin?

2°. Hogy hozunk létre bulletint **Publisher 2007**-ben?

3°. Hogy állítjuk be a kiadvány paramétereit?

4°. Hogy nézhetjük meg a bulletin egy oldalának tartalmát?

5°. Hogyan állíthatunk be különböző tördelést a bulletin egyes oldalaira?

6°. Milyen adatokat szokott tartalmazni a bulletin címlapja?

7°. Hogy készítjük a bulletin **Tartalomjegyzékét**?

8°. Mit tartalmaz a tartalomjegyzék alatt elhelyezkedő **Oldalsó panel**?

9°. Hogy változtatjuk meg a hasáb méretét, illetve hogy mozdíthatjuk el a hasábot?

10°. Hogy adhatunk új szövegdobozt a kiadványhoz?

11°. Hogy tudhatjuk meg, hogy a szövegdobozban elhelyezett szövegnek van-e folytatása más szövegdobozban?

12°. Mi a szövegátfolyás? Hogy állíthatjuk be, illetve hogy törölhetjük?

13°. Mit jelent a szövegdoboz mellett a A ··· jel? Hogy távolíthatjuk el?

14°. Hogy végezzük a kép és a szövegdoboz csoportosítását?

15°. Mi a füzet? Mi a rendeltetése?

16*. Mik a bulletin és a füzet közötti alapvető különbség? Hol alkalmazzák ezeket a kiadványokat?

Végezzük el a feladatokat!

1°. Indítsuk el a **Publisher 2007**-et! Hozzunk létre egy új hírlevelet a *Dagerrotípia* sablon alapján! Alkalmazzuk a *Tengerkék* színsémát és a *Levéltári* betűtípus-sémát! A kiadvány legyen *egyoldalas*. Mentsük a kiadványt **4.5.1.pub** néven!

2°. Hozzunk létre **Publisher 2007**-ben egy brosúrát az elemes sablon alapján! A színséma legyen *Elemes*, a betűtípusséma *Alapítvány*. Bővítsük a sablont egy űrlappal a *Látványelemtárból. Töltsük fel az űrlapot saját elérhetőségeinkkel* és távolítsuk el belőle a fölösleges mezőket! Mozdítsuk el az űrlapot, hogy az oldal alsó felét foglalja el! Mentsük a kiadványt **4.5.2.pub** néven!

3°. Hozzunk létre **Publisher 2007**-ben egy brosúrát az osztályunkról! A színsémát és betűtípus-sémát válasszuk ki önállóan! A címlap tartalmazza iskolánk fényképét, a település nevét és a kiadás évét. Mentsük a kiadványt **4.5.3.pub** néven a mappánkba!

4°. Hozzunk létre **Publisher 2007**-ben egy bulletint egy olyan íróról, akinek munkásságát jól ismerjük. A bulletin legyen *egyoldalas*. A címlap tartalmazza az író fényképét és rövid életrajzát, a cikkek pedig tartalmazzanak részleteket az író műveiből. A hátsó borítóra vigyük fel a saját adatainkat, mint a bulletin kiadóját. Mentsük a kiadványt **4.5.4.pub** néven a mappánkba!

5°. Nyissuk meg a tanárunktól kapott fájlt, például a **Тема 4\Завдання 4.5\зразок 4.5.5.pub.-ot!** A **Тема 4\Завдання 4.5\зразок 4.5.5.doc** fájl információit és a **Тема4\ Завдання 4.5** fotóit felhasználva fejezzük be a kiadvány kialakítását!

6*. Hozzunk létre **Publisher 2007**-ben egy bulletint, amely az iskolánk profilját írja le! Válasszunk hozzá színsémát és betűtípussémát. A cikkekben indokoljuk meg, miért célszerű az adott profilt választani. Mentsük a kiadványt **4.5.6.pub** néven a mappánkba!

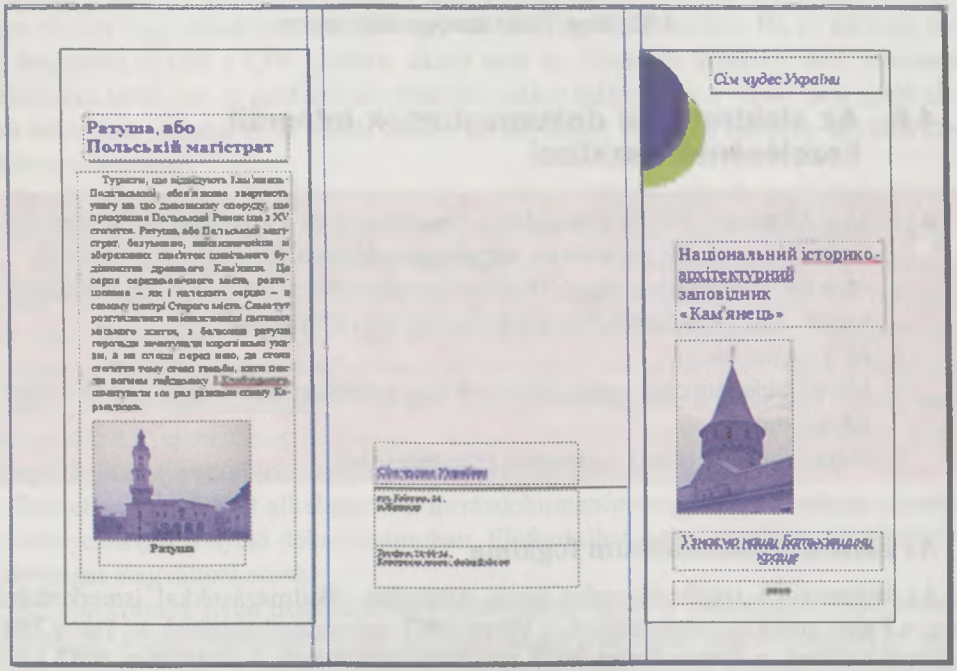
16. számú gyakorlati munka **Információs bulletin és füzet létrehozása**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Hozzunk létre **Publisher 2007**-ben bulletint „Ukrajna 7 csodája” címmel! Használjuk a *Szüretidő* sablont vagy készítsünk saját makettet! A bulletin első két lapjának mintáját (4.49. ábra) a **Тема 4\Практична 16** mappában találjuk **Практична 16_1.jpg** és **Практична 16_2.jpg** néven.
2. A bulletin létrehozása során használjuk fel a **Тема 4\Практична 16** mappa szöveges és grafikus fájljait!
3. Mentsük a bulletint **16.1_gyakorlati.pub** néven!
4. Hozzunk létre füzetet Ukrajna azon természeti vagy kulturális emlékhelyeiről, amelyek az előbbi bulletinben szerepeltek! Használjuk a *Halvány* sablont vagy készítsünk saját makettet! A füzet első két lapjának mintáját (4.50. ábra) a **Тема 4\Практична 16** mappában találjuk **Практична 16_3.jpg** és **Практична 16_4.jpg** néven.
5. A bulletin létrehozása során használjuk fel a **Тема 4\Практична 16** mappa szöveges és grafikus fájljait!
6. Mentsük a bulletint **16.2_gyakorlati.pub** néven!

a) b)

4.49. ábra. Bulletin szerkesztési sablon



a)

4.50. ábra. Füzet szerkesztési sablon

b)

4.50. ábra. Füzet szerkesztési sablon

4.6. Az elektronikus dokumentumok integrált kezelésének eszközei

1. Mi a **Vágólap**? Milyen műveleteket végezhetünk el a segítségével? Miben különbözik az operációs rendszer **Vágólapja** a **Microsoft Office** vágólapjától?
2. Mi a fájlformátum-konverzió? Milyen formátumban menthetünk egy dokumentumot? Egy táblázatot? Egy prezentációt? Egy adatbázist?
3. Mi a parancsikon?
4. Milyen objektumokat helyezhetünk el egy szöveges állományba? Hogy tehetjük ezt meg?
5. Milyen formátumban menthetünk egy web-lapot?

Az összetett dokumentum fogalma

Az informatika tanulmányozása során különféle alkalmazásokkal ismerkedtünk meg: a **Paint** grafikus szerkesztővel, a **Word 2007** szövegszerkesztővel, az **Excel 2007** táblázatkezelővel, a **Power Point 2007** prezentációkészítővel, a **Publisher 2007** kiadványszerkesztővel, az **Access 2007** adatbázis-kezelővel. Másként szólva különféle elektronikus dokumentumok – szövegek, elektronikus táblázatok, számítógépes prezentációk, adatbázisok stb. – kezelésével ismerkedtünk meg.

Az elektronikus dokumentumok kezelése során gyakran előfordul, hogy ezekben egyéb szoftverekkel készített részleteket kell elhelyezni.

Előfordul például, hogy egy **Word 2007**-ben készített szöveges dokumentumba el kell helyezni egy, a **Paint** segítségével készített raszteres képet, egy **Excel 2007**-ben készített elektronikus táblázatot, vagy az **Access 2007**-ben készített adatbázis egy tábláját. Más esetben az **Access 2007**-ben meg kell nyitni egy, az **Excel 2007**-ben már elkészített táblázatot, esetleg egy prezentációban előre elkészített hanganyagot és videoklipet kell elhelyezni. Ezekben az esetekben *összetett* vagy *integrált* dokumentumot hozunk létre.

Az *összetett (integrált)* dokumentum alatt olyan dokumentumot értünk, amely más szoftverek segítségével készített részeket tartalmaz.

Azt a programot (dokumentum, fájl), amelyből a beillesztett részlet származik, ***forrásnak***, amelybe ezeket beillesztettük, ***nyelőnek*** nevezzük.

Különböző forrásból származó adatok beillesztése

A **Windows** operációs rendszer különböző lehetőségeket biztosít a felhasználóknak összetett dokumentumok készítésére. Ezek némelyikével már megismerkedtünk. Ilyen a drag-and-drop (húzd és vidd) technológia és a **Vágólap** alkalmazása.

Több, **Windows** operációs rendszer alatt futó program esetében lehetséges a szükséges részlet vagy objektum áthúzása egy másik program ablakába. Ha az áthúzás során lenyomva tartjuk a **Ctrl** gombot, akkor nem az objektum áthelyezésére, hanem a másolására kerül sor. A módszer alkalmazása akkor ajánlott, ha a forrás és a nyelő ablaka közel helyezkedik el. E célból ezeket az ablakokat érdemes előzőleg egymás közelében elhelyezni.

Az objektumokat a **Windows** operációs rendszer **Vágólapjának** segítségével is áthelyezhetjük. E célból az objektumot a forrásból rá kell másolni a **Vágólapra** (Másolás), majd be kell illeszteni a nyelőbe (**Beillesztés**). A **Microsoft Office 2007** használatakor a **Microsoft Office** vágólapját használjuk, amelyben, mint tudjuk, 24 objektumot lehet elhelyezni. Ez akkor igazán hasznos, ha több objektum áthelyezésre van szükség, mert ekkor a forrásból a másolandó objektumokat áthelyezhetjük a vágólapra, majd onnan beilleszthetjük a nyelőbe. A művelet elvégzése során mind a forrásban, mind pedig a nyelőben meg kell legyen nyitva az **Office Vágólap**. A vágólapot a **Kezdőlap Vágólap Gyorselérési** eszköztár gombra kattintva nyithatjuk meg.

Ezeket az eszközöket alkalmazva a forrásdokumentum egy objektumának másolatát elhelyezhetjük a nyelő dokumentumban. Előfordulhat, hogy az objektum egyes tulajdonságai megváltozhatnak.

Az objektumok ilyen másolása vagy áthelyezése során a dokumentumokat kezelő programok, a dokumentumok vagy az objektumok példányai között semmilyen kapcsolat nem jön létre. A továbbiakban az objektum kezelését az integrált dokumentumot kezelő alkalmazás végzi. Ha például egy szöveges dokumentumban egy, a **Paint**-ben készített képet illesztünk, akkor a későbbiekben nem szerkeszthetjük tovább ezt a

képet a **Paint Ecset, Radír, Festékszóró** vagy egyéb eszközeivel. A beillesztés után csak a szövegszerkesztő képszerkesztési eszközeit használhatjuk.

A forrás és a nyelő közötti kapcsolat hiánya azt eredményezi, hogy az eredeti objektum változásai nem látszanak a nyelőben elhelyezett objektumon. Ha egy szöveges dokumentumban elhelyezünk egy **Excel** táblázatkezelőben elkészített diagramot és az eredeti táblázatban megváltoztatjuk az adatokat, amely alapján a diagram készült, ez a szövegben elhelyezett diagramon semmilyen változást nem okoz.

Az OLE technológia

A fent ismertetett technológiákon kívül, amelyek alkalmazása nem mindig célszerű, az objektumokat az **OLE** technológia segítségével is elhelyezhetjük az összetett dokumentumokban.

Az **OLE** (*Object Linking and Embedding* – objektum csatolása és beágyazása) olyan módon helyezi el az objektumokat a céldokumentumban, hogy eközben biztosítja annak kapcsolatát a forrásdokumentummal és az azt kezelő programmal.

Az OLE technológia különböző típusú objektumok más típusú dokumentumokban történő elhelyezését biztosító technológia.

Az **OLE** technológia az objektumok kétféle beillesztési módját támogatja:

- ***objektumok beágyazása*** – amikor is a céldokumentumban elhelyezett objektum-másolat és az azt kezelő program között kapcsolat jön létre, ugyanakkor az eredeti objektum és a másolat között nem. Ez esetben az eredeti objektum változásait nem láthatjuk a másolaton, ugyanakkor a másolatot továbbra is az eredeti objektumot kezelő szoftver segítségével dolgozhatjuk fel.
- ***objektumok csatolása*** – ekkor a céldokumentumba nem egy objektum kerül be, csak egy hivatkozás az eredeti dokumentumra. Ekkor az eredeti objektum változásai automatikusan belekerülnek a céldokumentumba. Az objektum kezelését értelemszerűen a forrásobjektumot kezelő szoftver segítségével végezzük.

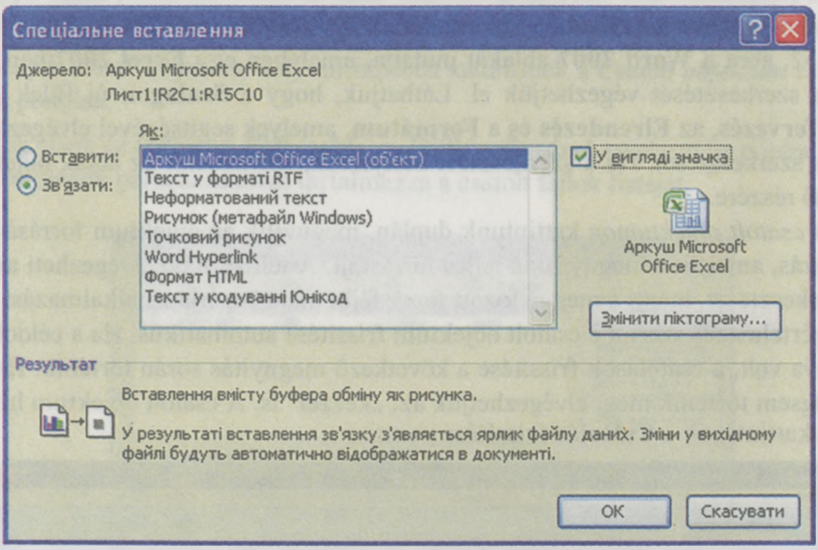
A dokumentum olvasása vagy nyomtatása során gyakorlatilag nem tudjuk megkülönböztetni, milyen módon illesztettük be az objektumot a dokumentumba (kivéve természetesen, ha az objektum ikonként látszik). A beillesztés módja csak a beillesztett objektum szerkesztésének módjára van hatással, valamint az összetett dokumentum méretére. A beágyazott objektum az összetett dokumentumban helyezkedik el, ezért növeli annak méretét. A csatolt objektum nem tartozik az összetett dokumentumhoz, az csak egy hivatkozást tartalmaz az objektumra. Ez esetben sokkal kisebb lesz az összetett dokumentum mérete.

Az **OLE** technológiát nem minden alkalmazás támogatja. Vannak alkalmazások, amelyek csak forrásként, mások csak nyelőként működnek. A **Microsoft Office 2007** alkalmazásai teljes egészében támogatják az **OLE** technológiát, a **Paint** csak forrásként szerepelhet.

Objektumok csatolása és beágyazása Microsoft Office 2007-ben

A **Microsoft Office 2007**-ben az objektumok csatolása és beágyazása a **Vágólap** segítségével az ***Irányított beillesztés*** paranccsal történik az alábbi algoritmus szerint:

1. Jelöljük ki a forrásdokumentumban az objektumot (dokumentumrészletet), majd hajtsuk végre a ***Kezdőlap*** ⇒ ***Vágólap*** ⇒ ***Másolás*** parancsot.
2. Helyezzük a kurzort a céldokumentumban a beillesztés helyére, majd hajtsuk végre a ***Kezdőlap*** ⇒ ***Vágólap*** ⇒ ***Beillesztés*** ⇒ ***Irányított beillesztés*** parancsot.
3. A megnyíló irányított beillesztés párbeszédablakban kapcsoljuk be a megfelelő rádiógombot:
   - beágyazás esetében a **Beillesztés**-t;
   - csatolás esetében a **Csatolás**-t (4.51. ábra).
4. A **Miként** területen állítsuk be a beillesztett objektum formátumát.



4.51. ábra. Az **Irányított beillesztés** párbeszédablak

A kiválasztott típus függvényében a program különböző eszközöket biztosít a felhasználónak az objektum-másolat kezelésére. A lehetőségek az eredeti objektumtól függően különbözőek lehetnek.

Ha azt szeretnénk, hogy a másolat tulajdonságai megegyezzenek az eredeti objektum tulajdonságaival, akkor válasszuk az *Objektumként* lehetőséget.

5. Az **Ikonként látszik** jelölő beállításával válasszuk ki az objektum megjelenítési módját a céldokumentumban:
   - ha a jelölő nincs bekapcsolva, az objektum ugyanúgy jelenik meg, mint az eredeti dokumentumban;
   - ha a jelölő be van kapcsolva, akkor az objektum helyett egy ikon jelenik meg (az ikon külalakja megváltoztatható). Az ikonra duplán kattintva megnyílik az eredeti objektum.
6. Kattintsunk az **OK** gombra.

Az objektumok használata során tartsuk szem előtt az alábbiakat:

- az eredeti objektum kezelésére alkalmas szoftvernek telepítve kell lenni azon a számítógépen, ahol az összetett dokumentumot meg szeretnénk nyitni;
- a forrásfájlnak és a célfájlnak mentve kell lenni;
- az összetett dokumentumok vagy a forrásdokumentumok áthelyezése a csatolások ***megszakadásához*** vezet. Az objektumok ez esetben nem jelennek meg az összetett dokumentumban.

A beágyazott és csatolt objektumok kezelése Microsoft Office 2007-ben

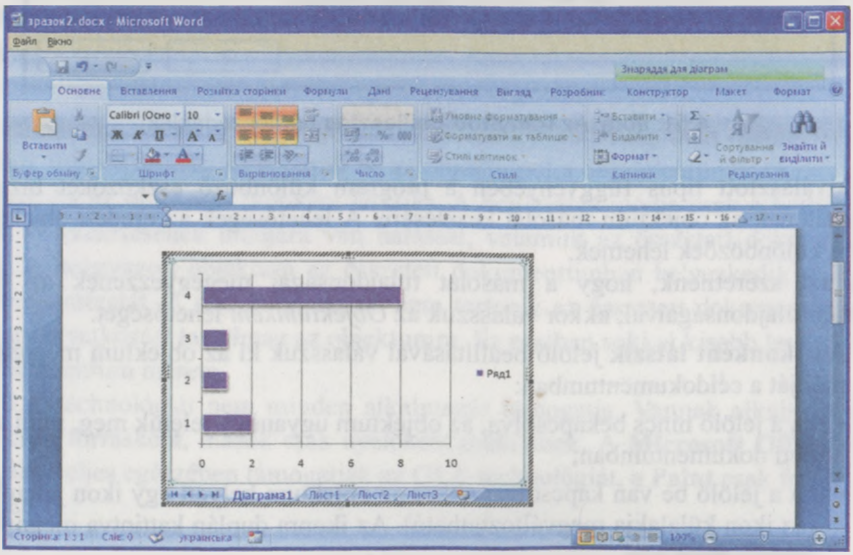
A beágyazott és csatolt objektumok kezelésének megvannak a maga sajátosságai.

Ha duplán kattintunk a *beágyazott objektumra*, megnyílik egy ablak, amely a beágyazott objektumot és a forrásprogram eszközeit tartalmazza, amelyekkel elvégezhetjük az objektum szerkesztését.

A 4.52. ábra a **Word 2007** ablakát mutatja, amelyben egy **Excel 2007**-ben készült diagram szerkesztését végezhetjük el. Láthatjuk, hogy a **Szalagon** új fülek jelentek meg, a **Tervezés**, az **Elrendezés** és a **Formátum**, amelyek segítségével elvégezhetjük a diagram szerkesztését. Ha a szerkesztést befejeztük, kattintsunk az ablak objektumon kívül eső részére.

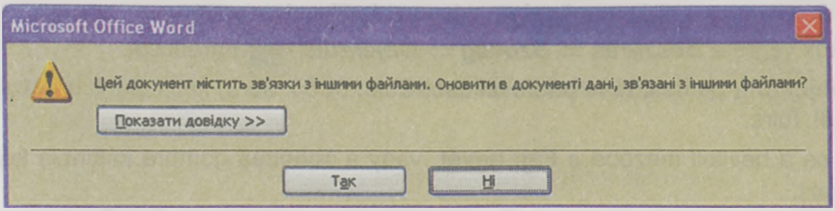
Ha a *csatolt objektumon* kattintunk duplán, megnyílik az objektum forrását kezelő alkalmazás, amelyben megnyílik a teljes forrásfájl. A felhasználó elvégezheti az objektum szerkesztését, menti a megváltozott forrásfájlt, majd bezárja az alkalmazást.

Alapértelmezés szerint a csatolt objektum frissítése automatikus. Ha a céldokumentum zárva volt, a csatolások frissítése a következő megnyitás során történik. Ha a frissítés mégsem történik meg, elvégezhetjük azt „kézzel" is. A csatolt objektum helyi menüjében kattintsunk a **Csatolás frissítése** pontra.



4.52. ábra. A forrás-szoftver kiegészítő ablaka a beágyazott objektum szerkesztésekor

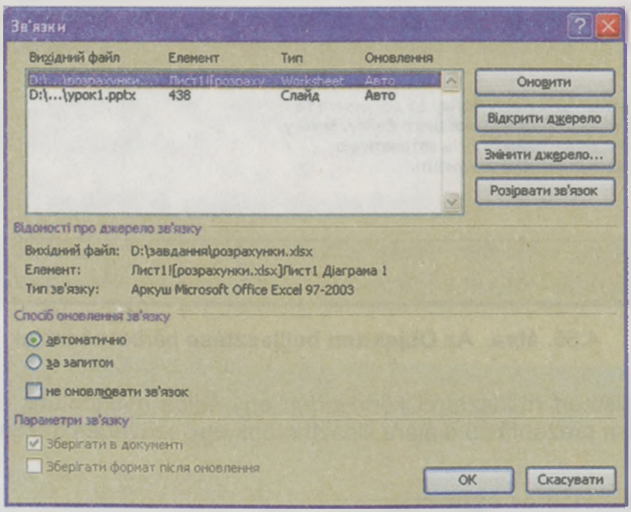
A szoftverkörnyezetet beállíthatjuk úgy is, hogy a csatolt objektumok frissítése ne legyen automatikus, csak a dokumentum megnyitása során történhessen meg a felhasználó kezdeményezésére (4.53. ábra). A felhasználónak ellenőrzése alatt kell tartani a dokumentumok csatolását azok megszakadása esetében.



4.53. ábra. A csatolások frissítésére vonatkozó kérés

A dokumentum kapcsolatainak figyelését, frissítését, beállítását és törlését a **Csatolások** ablakban (4.54. ábra) végezhetjük el. Az ablakot kétféleképpen nyithatjuk meg:

- bármelyik csatolt objektum helyi menüjében kattintsunk a ***Csatolt objektum*** ⇒ ***Csatolások*** pontjára;
- hajtsuk végre az ***Office*** ⇒ ***Előkészítés*** ⇒ ***Fájlhivatkozások szerkesztése*** pontra.
  A **Csatolások** párbeszédablak tartalmazza a csatolt fájlok listáját.



4.54. ábra. A **Csatolások** párbeszédablak

Ha kiválasztunk egy tetszőleges fájlt, megtudhatjuk:

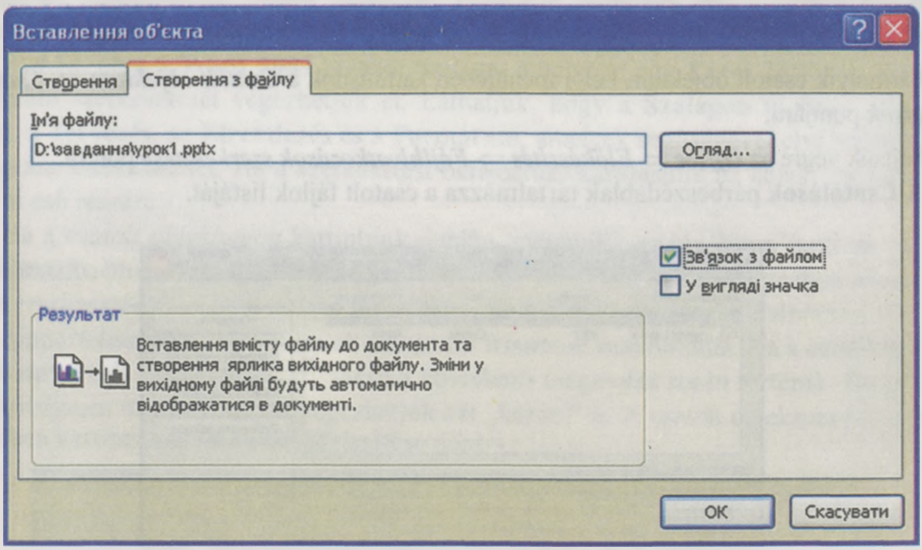
- a forrásfájl nevét;
- az objektum típusát a forrásfájlon belül;
- a csatolás típusát;
- a frissítés módját (*automatikus* vagy *kézi*).

A megfelelő jelölők beállításával megváltoztathatjuk a csatolás tulajdonságait.

Az **OLE** technológia segítségével egy dokumentumban elhelyezhetünk egy másik fájl tartalmát, vagy létrehozhatunk egy új objektumot egy másik szoftver eszközeivel. Az ilyen beillesztést beágyazásként és csatolásként is létrehozhatjuk.

Egy másik fájl tartalmát a következőképpen helyezzük el a dokumentumban:

1. Vigyük a kurzort a dokumentumban a leendő beillesztés helyére.
2. Hajtsuk végre a ***Beszúrás*** ⇒ ***Szöveg*** ⇒ ***Objektum*** parancsot.
3. Az **Objektum beillesztése** párbeszédablakban (4.55. ábra) kattintsunk a **Létrehozás fájlból** fülre.
4. Írjuk be a beviteli mezőbe a **Fájl nevét**, vagy a **Tallózás** gombra kattintva keressük meg azt.
5. Kapcsoljuk be a **Csatolás** és az **Ikonként látszik** jelölőket szükség szerint.
6. Kattintsunk az **OK** gombra.



4.55. ábra. Az **Objektum beillesztése** párbeszédablak

A most megismert módszerrel könnyedén egy fájlba gyűjthetünk néhány szöveges állományt, vagy a prezentáció diájára illeszthetünk egy szöveges dokumentumot.

Fájlok importja és exportja

Az alkalmazások közötti adatátvitelt gyakran célszerű az *import*, illetve *export* műveletek segítségével megoldani, amennyiben az alkalmazások támogatják ezt a megoldást.

Az **import** művelet azt jelenti, hogy az alkalmazás olyan formátumú fájlt nyit meg, amely formátum nem az alkalmazás saját formátuma. Az importálás után az alkalmazás ugyanúgy kezeli a fájlt, mintha a saját szabványos formátumában lenne. Az **export** művelet azt jelenti, hogy a fájlt olyan formátumban mentjük, ami nem a programunk saját formátuma. Lényegében mind az import, mind az export során *fájlformátum-konverzió* történik, ami részleges adatvesztéssel is járhat.

Import (_importo_ – behozatal) olyan fájl megnyitása, amelynek formátuma eltér a szoftver alapértelmezett formátumától.

Export (_exporto_ – kivitel) a fájl mentése olyan formátumban, amely eltér a szoftver alapértelmezett fájlformátumától.

Vizsgáljuk most meg az export és import műveletek néhány sajátosságát a **Microsoft Office 2007** rendszerben.

1. példa. Szöveges fájlok importálása **Microsoft Office 2007**-ben.

Excel 2007-ben egy szöveges fájl importálása a fájl megnyitása művelet segítségével történik, amelyhez a **Fájltípus** listából a *Szövegfájlok* lehetőséget kell választani. Importálni csak a TXT formátumú szöveges állományokat lehet.

A fájltípus kiválasztása után elindul az **Importálás varázsló**, amelyben be kell állítani, tartalmaz-e tagolásjeleket az importálandó fájl és ha igen, milyeneket, meg kell adni a kódolást, valamint az oszlopok adattípusát és formátumát. A varázsló futásának eredményeként a szöveges állomány **Excel 2007**-táblázatba importálódik.

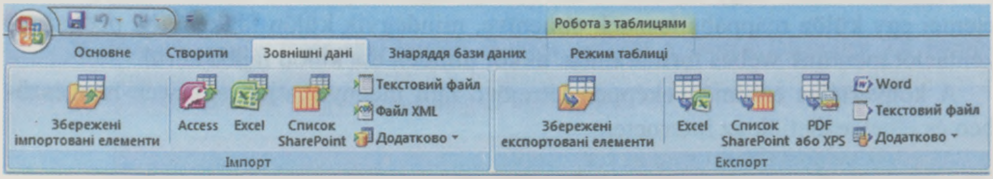
Hasonlóképpen működik az **Access 2007 Importálás varázslója**, de abban meg kell adni az importálandó szöveg nyelvét és kódolását is.

A **Power Point 2007**-ben több szöveges formátum importálása lehetséges (TXT, RTF, DOCX). Az importálás eredményeképpen olyan prezentáció jön létre, amelynek diáin címkeként az importált fájl sorait tartalmazza.

Publisher 2007-ben TXT, RTF és DOCX típusú fájlokat nyithatunk meg és **Word** dokumentumokat importálhatunk.

2. példa. Fájlok importja és exportja **Access 2007**-ben.

Az **Access 2007** adatbázis-kezelő által támogatott adatcsere különleges a **Microsoft Office 2007**-en belül. Az adatcsere eszközeit a **Külső adatok** fülön találjuk (4.56. ábra).



4.56. ábra. A Külső adatok eszköztár

Az **Import** vezérlőelemek segítségével más **Access 2007** (gomb) adatbázisból, **Excel 2007** táblázatból (gomb), **RTF** szövegfájlból (gomb) importálhatunk adatokat, az **Export** vezérlőkkel pedig elektronikus táblázatba (gomb), **Word 2007** szöveges dokumentumba (gomb) stb. exportálhatjuk az aktuális adatbázis adatait.

Ha **Excel** munkafüzetet szeretnénk importálni az **Access 2007** adatbázisba, használjuk az **Külső** adatok eszköztár **Import** csoportjának **Excel** gombját. Az **Importálás**

varázsló segítségével egy munkalap vagy egy névvel ellátott cellacsoport tartalmát importálhatjuk.

Ha az **Access 2007**-ből szeretnénk exportálni **Excel 2007**-be, válasszunk ki egy táblát, űrlapot vagy jelentést, majd hajtsuk végre a ***Külső adatok*** ⇒ ***Export*** ⇒ ***Excel*** parancsot. A program **Excel** munkafüzet formátumban menti az általunk kiválasztott objektumot a megadott mappába. A fájl neve alapértelmezés szerint megegyezik a kiválasztott objektum nevével.

Hasonlóképpen történik az exportálás szöveges dokumentumba is.

3. példa. Adatexport **Power Point 2007**-ben.

A prezentációkat kétféle módon exportálhatjuk szöveges állományba (***Office*** ⇒ ***Közzététel*** ⇒ ***Emlékeztetők létrehozása a Microsoft Office Word programban***):

- jegyzetek formájában, ahol a prezentáció diái grafikus vázlat formájában kerülnek a szövegbe, amelyek mellett jegyzetsorok helyezkednek el. Az egy oldalra elhelyezendő diák száma és a jegyzetek elhelyezése beállítandó;
- vázlat formájában, amikor a diák szöveges objektumai egymás után kerülnek be a szövegbe.

Hasznos lehet még a prezentáció képként történő mentése, amikor is az egyes diák külön képként kerülnek mentésre (a fájlformátum választható), amelyek egy, a prezentációval azonos nevű mappába kerülnek. E célból hajtsuk végre az ***Office*** ⇒ ***Mentés másként*** parancsot, majd válasszunk egy képformátumot.

Dokumentumok közzététele a weben

Ahhoz, hogy a **Microsoft Office** programban készült fájlokat közzétehessük a világhálón, nem feltétlenül kell ismerni a HTML nyelvet, elegendő a fájlt weblap formátumban menteni. Ezt a **Mentés másként** – *Weblap* paranccsal tehetjük meg.

Előfordulhat, hogy amikor weblap formátumba mentünk, egyes formázási elemek elvesznek. Ezért mentés előtt célszerű annak webes elrendezését megtekinteni: e célból kattintsunk a státuszsor **Webes elrendezés** gombjára.

Jegyezzük meg, hogy amikor webes formátumba mentünk, a dokumentum grafikus elemei egy külön mappába lesznek elmentve, mindegyik külön fájlként, ha pedig prezentációt mentünk webes formátumba, akkor minden dia külön fájlba kerül.

A konvertálás eredményeképpen létrejövő fájlt megnyithatjuk egy web-böngészőben és elhelyezhetjük az interneten.

Ellenőrző kérdések

1°. Milyen dokumentumot nevezünk összetettnek?

2°. Hozzunk fel példákat olyan dokumentumokra, amelyeket több szoftver segítségével hozunk létre!

3°. Milyen objektumokat illeszthetünk be egy szöveges állományba? Egy prezentációba? Egy elektronikus táblázatba?

4°. Milyen módon illeszthetünk be objektumokat egy elektronikus dokumentumba?

5°. Melyik dokumentumot tekintjük az objektumok forrásának és melyiket cél (nyelő) dokumentumnak?

6°. Hogy végezzük el az objektumok beillesztését a **Microsoft Office 2007 Vágólap** segítségével?

7°. Mi az **OLE** technológia? Mi a rendeltetése?

8°. Miben különbözik az objektumok beágyazása és csatolása?

9°. Miben különbözik a **Vágólap** segítségével végrehajtott beillesztés az **OLE** technológiától?

10°. Mit értünk a csatolás megszakadása alatt? Mikor történhet ez meg? Hogy hozhatjuk helyre?

11*. Vállalati hálózatokban gyakran hoznak létre olyan szabványos objektumokat (nyomtatványok űrlapjait, logók, sablonok, fájlsablonok stb.), amelyeket a vállalat minden alkalmazottja csatolt objektumként használ. Mi az **OLE** mechanizmus ilyen alkalmazásának előnye? Mi a hátránya?

12*. Egy diagramot publikációba illesztve abban csak egy ikon jelent meg. Mi történhetett, és hogy lehet orvosolni a problémát?

13*. Egy dokumentum három részlete három fájlba lett elmentve. Hogy lehet ezeket gyorsan egyesíteni?

14°. Miben különbözik a beágyazott és csatolt objektumok szerkesztése?

15°. Mi az exportálás és az importálás műveletek lényege?

16°. Mit értünk a dokumentum webes publikációja alatt?

Végezzük el a feladatokat!

1. Hasonlítsuk össze az objektumok elektronikus dokumentumokba illesztésének módjait, és töltsük ki az alábbi táblázatot!

Jellemző	*Beágyazott objektum*	*Csatolt objektum*
Az összetett dokumentum fájlmérete		
Csatolás megszakadása		
Szerkesztési eszközök		

2°. Nyissuk meg a tanár által kijelölt fájlokat, például a **Тема4\Завдання 4.6 mappa Зразок 4.6.2-1.docx, Зразок 4.6.2-2.docx** fájljait! Hasonlítsuk össze a tartalmukat és a fájlok méretét! Magyarázzuk meg az eltéréseket!

Fájl neve	*Fájl mérete*	*Beillesztés módja*
Зразок 4.6.2-1.docx		
Зразок 4.6.2-2.docx		

3°. Nevezzük meg a beágyazás és csatolás előnyeit és hátrányait!

Jellemző	*Beágyazott objektum*	*Csatolt objektum*
Előnyei		
Hátrányai		

4°. Nyissuk meg a **Microsoft Office** csomag fájljait, és határozzuk meg, milyen formátumba importálhatunk, illetve exportálhatunk fájlokat az egyes alkalmazásokból!

Program	*Export formátumok*	*Import formátumok*
Word 2007		
Excel 2007		
Power Point 2007		
Access 2007		
Publisher 2007		

5°. Nyissuk meg a tanárunk által megnevezett szöveges dokumentumot és elektronikus táblázatot, például a **Тема4\Завдання 4.6 mappa текст.docx** és **таблиця.xlsx** fájljait! Helyezzünk a szöveges állományba egy diagramot a munkafüzetből a vágólap segítségével! Mentsük a szöveges állományt **вправа 4.6.5.docx** néven a mappánkba! Változtassuk meg a diagramot a munkafüzetben, majd vizsgáljuk meg, milyen hatással van ez a változás a szöveges dokumentumra! Magyarázzuk meg a tapasztalatainkat!

6°. Nyissuk meg a tanárunk által megnevezett fájlokat – egy prezentációt és egy képet – (például a **Тема4\Завдання 4.6 mappa малюнок.bmp** és **презентація.pptx** fájljait)! A prezentáció első diájára illesszük be a képet, a másodikba ágyazzuk be, a harmadikhoz pedig csatoljuk. Mentsük a prezentációt **вправа4.6.6.pptx** néven! Változtassuk meg a képet egy képszerkesztővel! Nyissuk meg a prezentációt, és figyeljük meg annak változásait!

7°. Nyissuk meg a tanárunk által megnevezett szöveges állományt (például a **Тема4\Завдання 4.6\текст 4.6.7.docx** fájlt)! Próbáljuk meg megváltoztatni a beillesztett objektumokat! Határozzuk meg, milyen módon kerültek ezek beillesztésre! Döntéseinket indokoljuk meg!

8°. Nyissuk meg a tanárunk által megnevezett prezentációt, például a **Тема4\Завдання 4.6\зразок 4.6.8.pptx**-et! Mentsük a prezentációt HTML, JPG, RTF formátumban! Tegyük közzé a prezentációt jegyzetek formájában! Vizsgáljuk meg a létrehozott állományokat!

9°. Nyissuk meg a tanárunk által megnevezett szöveges dokumentumot, például a **Тема4\Завдання 4.6\текст 4.6.9.docx**-et! Tekintsük meg a dokumentum csatolásait! Töröljük a csatolásokat! Mentsük a fájlt HTML, RTF, TXT formátumban a mappánkba! Nyissuk meg a létrehozott fájlokat!

10°. Hozzuk létre **Завдання 4.6.10.** adatbázist, majd importáljunk abba egy elektronikus táblázatot (például a **Тема4\Завдання 4.6\таблиця 4.6.10.xlsx**-et)! Mentsük a fájlt a saját mappánkba **вправа 4.6.10.accdb** néven!

11°. Nyissuk meg a tanárunk által megnevezett adatbázist (például a **Тема4\Завдання 4.6\база 4.6.11.accdb**-t)! Exportáljuk a táblát elektronikus táblázatba, a jelentést pedig szöveges állományba! Mentsük a fájlokat **вправа 4.6.11.xlsx** és **вправа 4.6.11.docx** néven!

12*. Importáljuk a megadott szöveges dokumentumot (**Тема4\Завдання 4.6\текст 4.6.12.rtf**) adatbázisba és elektronikus táblázatba! Mentsük ezeket a fájlokat **вправа 4.6.12.xlsx** és **вправа 4.6.12.accdb** néven!

17. számú gyakorlati munka **Adatok kezelése több szoftvereszközzel**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiénia szabályokat!*

1. Nyissuk meg **Paint**-ben a **Тема4\Завдання 4.6\Практична 17\Квітка.jpg.** képet!
2. Nyissuk meg **Excel 2007**-ben a **Тема4\Завдання 4.6\Практична 17\Кількість.xlsx.** diagramot!
3. Nyissuk meg **Access 2007**-ben a **Тема4\Завдання 4.6\Практична 17\Олімпійці.accdb.** adatbázist!
4. Nyissuk meg a **Power Point 2007**-et, és hozzunk létre egy diát, amelybe illesszük be a megnyitott dokumentumok objektumait: a képet beágyazott objektumként, a diagramot csatolt objektumként, a tanulók névsorát a **Vágólap** segítségével beillesztett objektumként!
5. Mentsük a prezentációt PPTX, RTF, JPG és HTML formátumban a mappánkba!
6. Exportáljuk az **Олімпійці.accdb** adatbázist elektronikus táblázatba és szöveges állományba!
7. Mentsük a létrehozott fájlokat **Олімпійці.xlsx** és **Олімпійці.txt** néven a mappánkba!
8. Zárjuk be az összes alkalmazást!

4.7. Weblaptípusok. A weblapok osztályozása

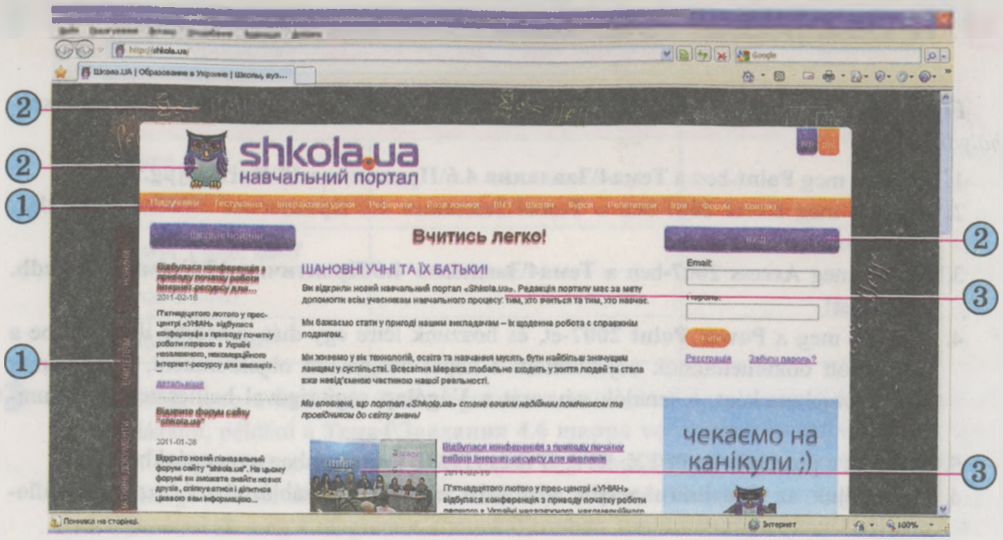
1. Milyen dokumentumokat nevezünk összetettnek?
2. Mi a weboldal, a weblap, a webportál? Mi a különbség a felsorolt fogalmak között?
3. Milyen nyelvet használnak a weblapok létrehozásakor?
4. Milyen típusú információt helyezhetünk el egy weboldalon?
5. Milyen részekből áll egy domain-név? Egy URL-cím?

Weboldaltípusok

A weboldal egy tipikus összetett dokumentum, mivel képeket, animációkat, videókat, zenei részleteket tartalmazhat. A weboldal, ami a **WWW** szolgáltatás információs erőforrása. A 9. osztályban megismerkedtünk a weboldalak kezelésével, megtanultuk, hogyan kell keresni az interneten. Most a weboldalak létrehozásával fogunk megismerkedni. A weboldalak létrehozása során figyelembe kell venni azok típusát és szerkezetét, valamint annak a webhelynek felépítését, amelynek része lesz a weboldal.

Tekintsük meg a 4.57. ábrán látható weboldalt. A következő szerkezeti elemeket különböztethetjük meg:

- **tartalom** – a weboldal tartalmi része, erre koncentrálódik a felhasználó figyelme; szöveget, grafikát, videót, hangfájlokat tartalmazhat;
- **navigációs elemek** – lehetővé teszik, hogy átléphessünk a webhely más lapjaira vagy más weblapokra;
- **dizájn-elemek** – a weboldal kialakítását, látványát, tartalmának strukturáltságát biztosítják.



4.57. ábra. A weboldal elemei

Rendeletetésük szerint a weblapok a következő csoportokba sorolhatók:

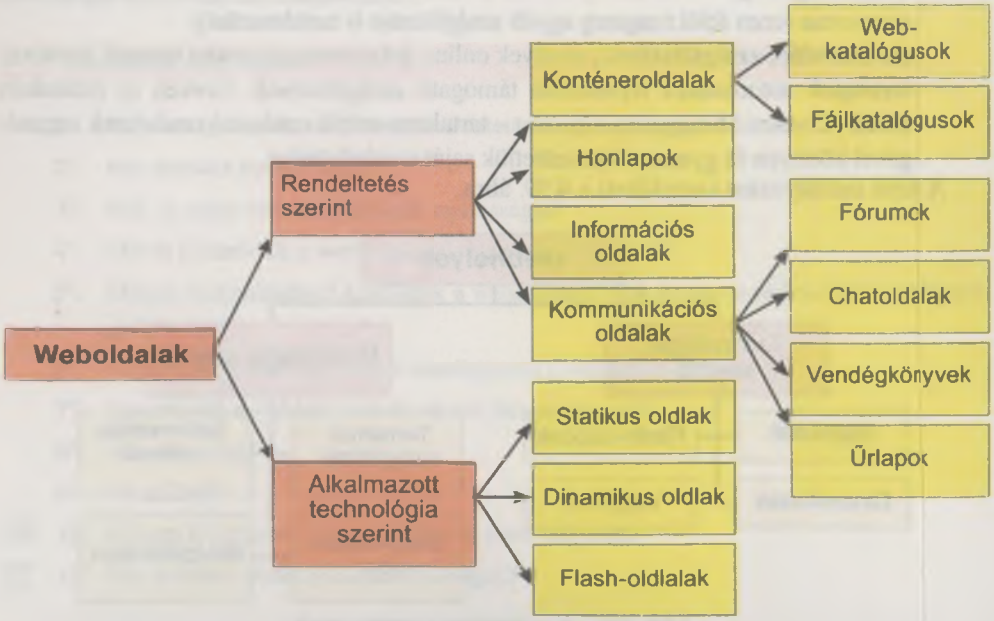
- **honlapok** – azok a weboldalak, amelyekre az adott webhely meglátogatása során először eljutunk. Ide mutat a webhely URL-címe. Legtöbbször a webhely tematikáját, rendeltetését mutatja be, néha a weboldal fejlesztőit is bemutatja, valamint megismertet a webhely egyéb oldalain található tartalmakkal;
- **információs weblapok** – egy témát bemutató weboldal, szöveget, képeket, egyéb információkat tartalmaz. A honlap egy speciális információs oldal;
- **konténer-oldalak** – hivatkozásokat tartalmaz más webes erőforrásokra:
  - **web-katalógusok** – az adott webhely weblapjaira vagy más webhelyekre mutató hivatkozásokat tartalmaz;
  - **fájlkatalógus** – letölthető fájlokra mutató hivatkozásokat tartalmaz;
- **kommunikációs (interaktív) lapok,** amelyek lehetővé teszik a felhasználók kommunikációját egymással és a webhely fejlesztőivel:
  - **fórumoldalak** – amelyeken a felhasználók megvitathatnak egy témát;
  - **chatoldalak** – csevegési szolgáltatást nyújtanak a felhasználóknak;
  - **vendégkönyvek** – segítségükkel az oldal látogatói megjegyzéseket írhatnak, leginkább a webhely fejlesztői számára.
  - **űrlapoldalak,** amelyek segítségével közvélemény-kutatást, áru-megrendelést végezhetünk

stb.

A weblapok a **létrehozás során alkalmazott technológia** szerint a következő csoportokba sorolhatók:

- **statikus oldalak,** amelyek HTML alapon készülnek. Az oldalak tartalma ugyanaz az összes látogató számára. Az ilyen weboldalakon nem tervezik a tartalom vagy a külalak megváltoztatását. Az oldal nem igényli semmilyen speciális szoftvertermék telepítését sem a szolgáltató sem pedig a felhasználó számítógépén. Jellemzően oktatási anyagokat, történelmi ismereteket, múzeumi kiállítási tárgyak ismertetését tartalmazzák.
- **dinamikus oldalak** speciális programozási nyelvek, **PHP** (*Personal Home Page Tools* – személyes honlapszerkesztő eszköz), **ASP** (*Active Server Pages* – aktív szerveroldalak), **PERL** (*Practical Extraction and Report Language* – gyakorlati lekérdező és jelentéskészítő nyelv) segítségével készülnek. Gyakran megújuló információkat tartalmazó oldalak készülnek ezzel a módszerrel, például híroldalak, webáruházak stb. Az ilyen oldalakat meglátogató felhasználók eltérő tartalmakat találhatnak;
- **flash-oldalak** (lobbanás), amelynek alapját az **Adobe Flash** animációs technológia képezi. Az ilyen oldalak feltűnőek, színesek, rengeteg animációt, hanghatást tartalmaznak. A weboldal külalakja az animáció lejátszása során megváltozik, de hatással lehet rá az egérmutató elhelyezkedése is. Az egyedi dizájn valódi látogató-csalogató. Az ilyen oldalak gyakran alkalmazzák tervezők, művészek, fényképészek, hogy munkáikat hatásosan mutathassák be. A flash-oldalakat ugyanakkor elég nehéz megváltoztatni, ezért gyakran megújuló tartalmak bemutatására nem alkalmas. A flash-oldalak fájlmérete meglehetősen nagy, ezért letöltésük nehézkes lehet. Megtekintésükhöz beépülő-modulokat kell telepíteni a böngészőnkhöz.

A weboldalak osztályozását a 4.58. ábra foglalja össze



4.58. ábra. A weboldalak osztályozása

A webhelyek osztályozása

A különböző webhelyek **szerkezete** eltérő lehet: eltérhetnek az oldalak számában, tematikus szervezésükben, belső kapcsolataikban. A webhelyeknek mindig van egy „főoldala", amely az összes többihez kapcsolódik. Az egyes fejezeteknek is lehet kezdőlapja, amely „felfelé" kapcsolódik a kezdőoldalhoz. Az oldalak számát és rendeltetését a tartalom mennyisége és jellege is befolyásolja.

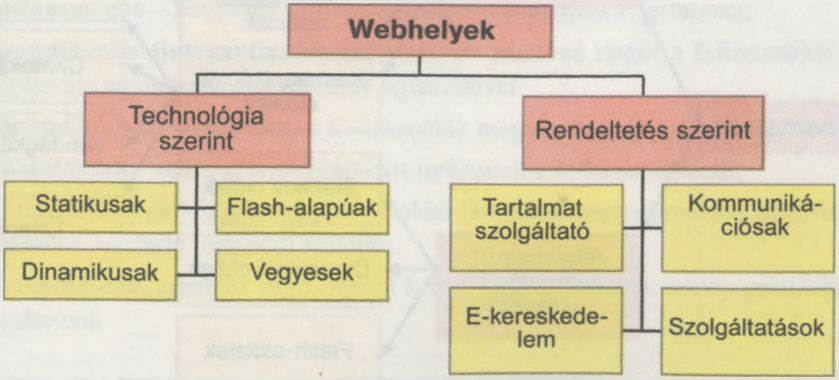
A webhelyeket különböző kritériumok szerint osztályozhatjuk. Lássunk néhány lehetséges osztályozást.

Az **alkalmazott technológia szerint** a webhelyek lehetnek **statikusak** (minden oldal statikus), **dinamikusak** (minden oldal dinamikus), **flash-alapúak** (minden oldal flash-alapú) és **vegyesek.**

Rendeltetésük szerint a webhelyek lehetnek:

- **információsak:** tematikus oldalak, híroldalak, elektronikus könyvtárak, enciklopédiák, szótárak, katalógusok, fájl-tárolók, médiatékák stb.;
- **kommunikációsak:** fórumok, chatoldalak, közösségi oldalak, on-line játékok oldalai, ismerkedési oldalak stb.;
- **elektronikus kereskedelmet biztosító webhelyek:** internet-áruházak, elektronikus fizetési lehetőséget biztosító rendszerek, bankok webhelyei, különböző fizetett szolgáltatásokat (idegennyelv-tanulás, pszichológiai szolgáltatások stb.) biztosító oldalak;
- **on-line szolgáltatásokat biztosító webhelyek:** internetes levelezőszolgáltatók, keresőszerverek, fordítógépek. Az on-line szolgáltatások közé tartozik még:
  - a **hosztingszolgáltatás,** amely lehetővé teszi weblapjaink elhelyezését a szolgáltató szerverein (ezen felül rengeteg egyéb szolgáltatást is tartalmazhat);
  - **on-line office szolgáltatások,** amelyek online dokumentumkezelést tesznek lehetővé;
  - weblapok automatikus fejlesztését támogató szolgáltatások. Ezeken az oldalakon **CMS** (*Content Management System* – tartalomvezérlő rendszer) rendszerek segítségével könnyen és gyorsan fejleszthetjük saját webhelyünket.

A fenti osztályozást szemlélteti a 4.59. ábra.



4.59. ábra. A webhelyek osztályozása

Webhelyek értékelése

Az interneten szörfölve minden bizonnyal felfigyeltünk arra, mennyivel különböznek az egyes weboldalak megjelenésükben, tartalmukban, célirányosságukban. Egyes oldalakon hosszasan elidőzünk, másokat szinte azonnal elhagyunk. Az, hogy mennyi időt töltünk el egy webhelyen, erősen függ a hely vonzóerejétől, dizájnjától, kezelhetőségétől. A webhely értékelése során a látogató a következő szempontokat veszi figyelembe:

- a hely információtartalmát, vagyis azt, hogy az ott található információt mennyire tekinti hasznosnak, mennyire ismétli a webhely más források tartalmát;
- a frissítés gyakoriságát, vagyis azt, mennyire felügyeli az oldal tulajdonosa annak tartalmát.
- az interaktivitást, vagyis azt, mennyire van lehetősége a felhasználónak arra, hogy a webhely tartalmára vonatkozó megjegyzéseket, kéréseket, kérdéseket fogalmazzon meg a fejlesztő vagy tulajdonos felé;
- milyen nyelveken elérhető a webhely tartalma;
- a webhely külalakját;
- a **használhatóságot,** vagyis azt, mennyire kényelmes és egyszerű a webhely használata, érthető annak szerkezete, minden, ami komfortossá teszi a böngészést ezen a webhelyen.

stb.

Az, hogy a látogató mennyi időt tölt el a webhelyen és milyen gyakran tér oda vissza, meghatározza a webhely népszerűségét és azt, milyen helyet foglal el az adott webhely a keresőszerverek népszerűségi listáján.

Ellenőrző kérdések

1°. Milyen szerkezeti elemeket különböztetünk meg egy weboldalon?

2°. Mit értünk a tartalom alatt?

3°. Mik az egyes weboldal-típusok sajátosságai?

4°. Miben különbözik a webhely honlapja a többi oldaltól?

5°. Milyen technológiával készülnek a weboldalak? Mik az egyes technológiák sajátosságai?

6°. Milyen ismertetőjelek szerint osztályozzuk a webhelyeket?

7°. Ismertessük a webhelyek 4.59. ábrán látható típusait!

8°. Mi a hoszting?

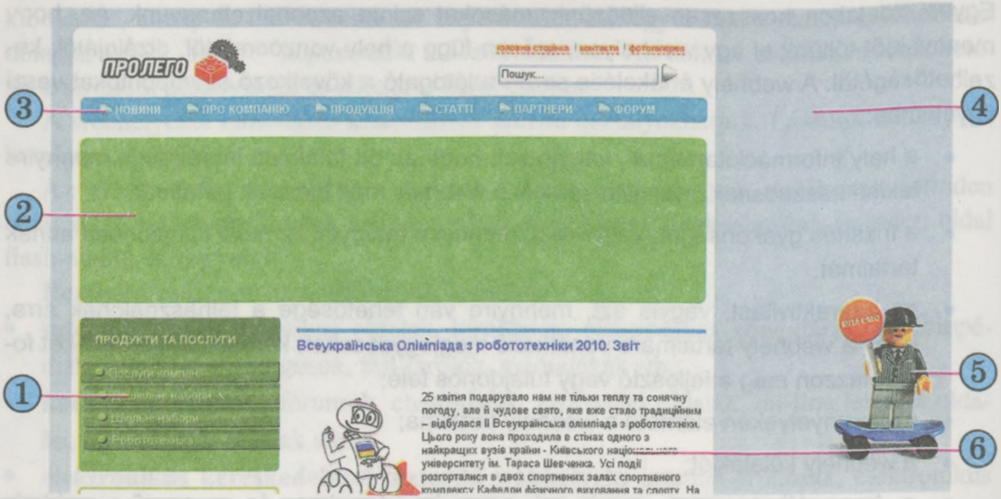
9°. Mi a CMS?

10°. Milyen kritériumok szerint értékelik a webhelyeket?

11°. Mit értünk a webhely használhatóságán?

Végezzük el a feladatokat!

1°. Állapítsuk meg, milyen tartalmi elemeket jelöltünk meg a weboldalon (4.60. ábra)!



4.60. ábra

Elem	*Tartalom*	*Navigáció*	*Dizájn*
Az elemek sorszáma			

2•. Nyissuk meg a következő weblapokat és határozzuk meg azok típusát!

Cím	*Típus*	
	Rendeltetés szerint	*Technológia szerint*
http://ukrainiancomputing.info/ada_u.html		
http://www.museum-ukraine.org.ua/index.php		
http://www.litforum.net.ua/showthread.php?t=144&page=70		
http://mncat.mikanewton.info		
http://children.kmu.gov.ua/history/persons/1358.html		

3•. Látogassuk meg az alábbi weblapokat! Határozzuk meg, melyik lap milyen típushoz tarozik!

	Cím
1	http://www.profeo.ua
2	http://www.freemail.ukr.net/ua
3	http://www.kancelarka.com.ua
4	http://formula.co.ua

	Típus
A	Információs oldal
B	E-kereskedelmi oldal
C	On-line szerviz
D	Kommunikációs oldal

4*. Látogassuk meg az alábbi weboldalakat! Határozzuk meg, milyen technológiával készültek!

	Cím
1	http://veselka-mebli.com
2	http://www.icfcst.kiev.ua/museum/museum-map_u.html
3	http://top.bigmir.net/show
4	http://www.brevis.kiev.ua

	Típus
A	Vegyes
B	Dinamikus
C	Flash
D	Statikus

5*. Határozzuk meg, milyen típusú webhelyre lehet jellemző:

a) a webhely 1–5 oldalt tartalmaz és tartalmazza a fejlesztő elérhetőségeit;

b) a webhely interneten megrendelhető áruk katalógusát tartalmazza;

c) a webhely nagy mennyiségű információt tartalmaz, továbbá szolgáltatásokat és kommunikációs lehetőségeket biztosít;

d) a webhely a szerző saját ötleteit tartalmazza, a felhasználóknak lehetősége van kommentárokat írni a tulajdonos számára.

6*. Látogassuk meg a webhelyet és határozzuk meg annak típusát!

URL-cím	*Típus*	
	Technológia szerint	*Rendeltetés szerint*
http://www.chl.kiev.ua		
http://www.onlandia.org.ua/ukr		
http://ticapac.pp.ua		
http://www.man.gov.ua		

7*. Értékeljük az alábbi webhelyeket!

Kritérium	*URL-cím*			
	http://www.nenc.gov.ua	*http://www.4uth.gov.ua/index.htm*	*http://www.osvita.com*	*http://mamajeva-sloboda.ua*
Mennyire informatív?				
Mikor volt utoljára frissítve?				
Tartalmaz-e interaktív eszközöket?				
Milyen nyelveken elérhető a tartalom?				
Vizuálisan vonzó				
Mennyire kényelmes a használata?				

4.8. A weblapok létrehozásának lépései. A webdizájn alapjai

1. Mik a számítógépes feladatmegoldás lépései?
2. Mi a dizájn? Mik a prezentáció dizájnjának elvei?
3. Mi a stílus?
4. Mik a kompozíció legfontosabb törvényei? A színválasztásé? Az ergonómiáé?
5. Mi a tördelés? Milyen jelentőséggel bír a kiadványszerkesztés során?

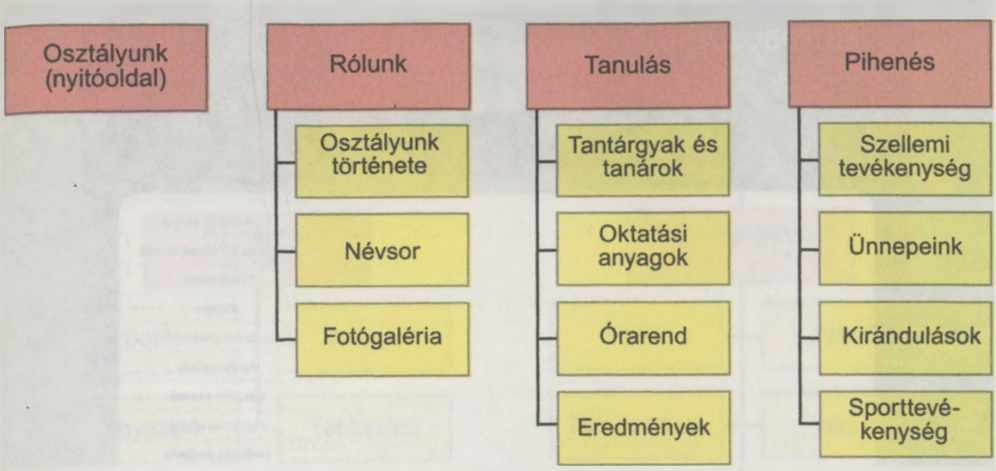
A weblapok létrehozásának lépései

Manapság gyakorlatilag minden vállalat, szervezet, oktatási intézmény rendelkezik saját weblappal. Egy weblapnak hasznát lehet venni az oktatás, a kommunikáció, az üzleti tevékenység szervezése stb. során. A weblap megtervezését megrendelhetjük egy erre szakosodott stúdiótól vagy elvégezhetjük önállóan is. A tervezés több lépésből áll. Ezek a lépések sokban hasonlítanak a számítógépes feladatmegoldás lépéseihez.

1. **A feladat megfogalmazása.** Ennek során meghatározzuk, mi célból van szükség a weblapra, milyen témákat szeretnénk ott bemutatni, milyen lesz a weblap típusa, elemezzük a hasonló tematikájú weblapokat. A fejlesztőnek ismernie kell:
   - a weblap létrehozásának célját;
   - a weblap tematikáját;
   - a weblap típusát: honlap, fórum, internet-áruház, portál stb.;
   - miben különbözik a létrehozandó weblap a hasonlóktól;
   - a potenciális látogatók körét: korát, nemét, érdeklődési körét stb.;
   - milyen szolgáltatásokat biztosít majd a weblap: fórum, chat, keresőszolgáltatás, e-mail stb.;
   - a weblap fejlődési perspektíváit.
2. **A weblap és az egyes oldalak szerkezetének meghatározása.** Ennek során meg kell határozni, milyen fejezetekből áll majd a weblap, milyen navigációs elemek alkalmazását tervezzük, milyen oldalakból áll majd és azok hogyan kapcsolódnak egymáshoz. Végeredményben meg kell kapjuk a **weblap térképét**: egy diagramot, amely a weboldalak hierarchiáját ábrázolja az azok között fennálló kapcsolatrendszerrel együtt, vagyis a weblap **belső szerkezetét.**

Lássuk például osztályunk weblapjának szerkezetét (4.61. ábra), amelynek látogatói osztályunk tagjai és azok szülei lesznek.

Ebben a lépésben meg kell még tervezni a weblap **külső szerkezetét** is. A weblapon belül egységes stílus alkalmazása ajánlott, ezért meg kell tervezni, milyen elemekből épülnek fel az oldalak, hol lesznek ezek az elemek elhelyezve, lesznek-e reklámok és ha igen, hol lesznek ezek elhelyezve, hol lesznek a navigációs elemek, milyen lesz a menürendszer, lesz-e látogatás-számláló és ha igen, hol.



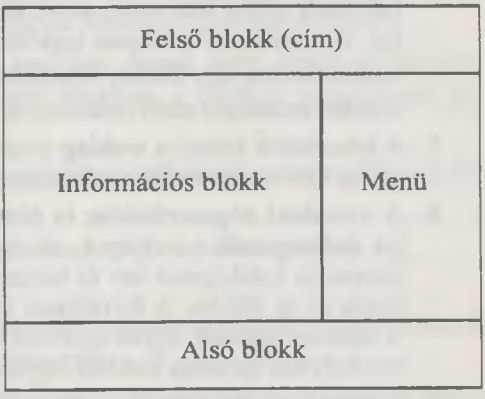
4.61. ábra. Az osztályweblap szerkezete

A weblapok általában a következő elemeket tartalmazzák:

- felső blokk (**cím**) – a címet, logót tartalmazza;
- **menü** – a weblap bejárását biztosítja;
- **információs blokk** – központi helyet foglal el az oldalon, a lényeges információkat tartalmazza;
- **alsó blokk** – szerzői jogokat, elérhetőségeket tartalmazza.

A külső szerkezetet **modulháló** formájában ábrázoljuk (4.62. ábra). Ha az oldal frissítése automatikus lesz, a külső szerkezetet oldalsablon formájában is elkészíthetjük.

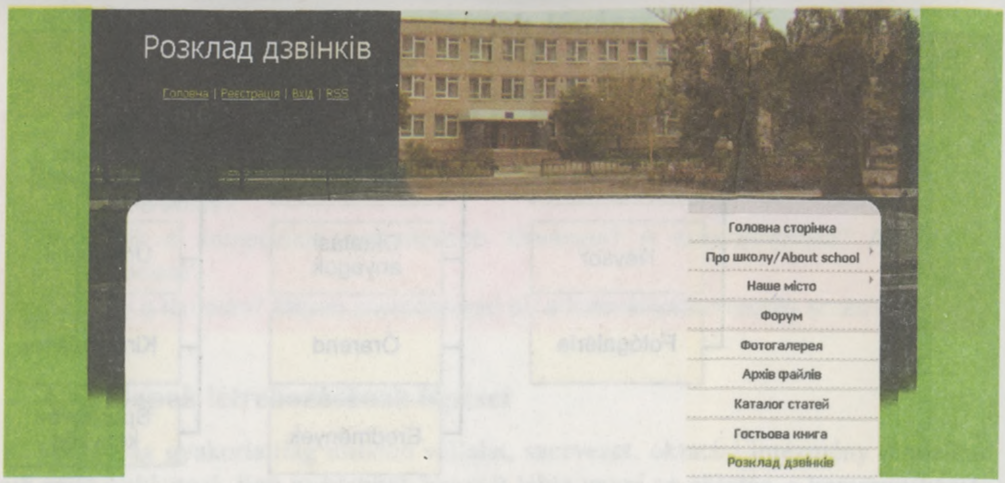
3. **Az oldalak makettjének megtervezése.** A makett alapját az előző lépésben megtervezett oldalszerkezet alkotja. A makett tartalmazza az oldal szöveges és grafikus elemeinek méreteit, az alkalmazott színeket és betűtípusokat, amelyek meghatározzák az **oldal stílusát** (4.63. ábra). Fontos, hogy a stílus összhangban legyen a weblap rendeltetésével, a potenciális látogatóinak ízlésével, támogassa a tartalom kellemes érzékelését.



4.62. ábra. Modulháló

4. **A weblap létrehozása és tördelése.** A weblapokat általában **HTML** nyelven írják. A létrehozás során el kell végezni az oldal tördelését is. Ahogy azt már megtanultuk, a **tördelés** során helyezzük el az oldalakon a szöveges és a grafikus elemeket, mindezt olyan módon, hogy összhangban legyen a makettel.

A jól tördelt weboldal ugyanúgy néz ki minden böngészőben, gyorsan betöltődik, könnyen szerkeszthető, frissíthető. Ebben a lépésben töltjük fel az oldalakat információkkal.



4.63. ábra. Példa az oldal makettjének megtervezésére

5. Ha a terv interaktív elemek elhelyezését is tartalmaza (keresőmotor, szavazógép, fórum stb.), akkor a következő lépés az **oldal programozása** lesz.
6. A **weblap publikálása (elhelyezése az interneten).** A fejlesztés során a weblap a fejlesztő helyi számítógépén található. Amikor elkészültünk, a weblapot el kell helyezni egy hoszting-**szolgáltatást** biztosító szerveren. A publikálás során a weblap domain-nevet kap. Ezután az oldal minden érdeklődő számára elérhetővé válik, amennyiben a hozzáférés nem korlátozott.

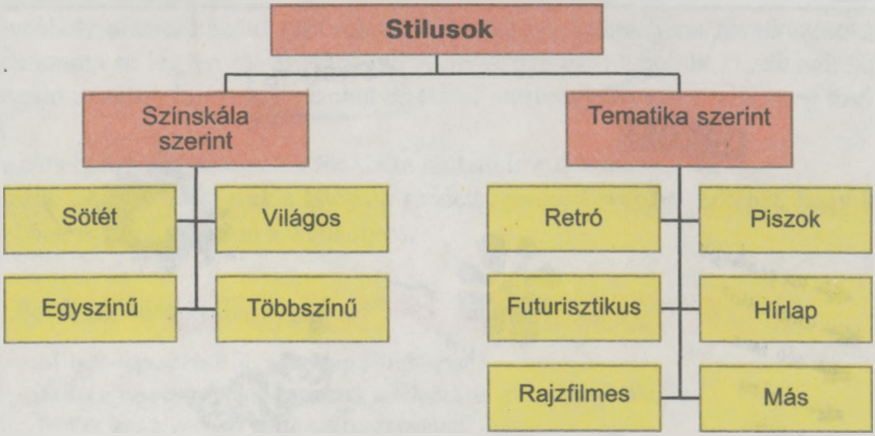
   Léteznek olyan szerverek, amelyeken ingyenesen helyezhetjük el a weblapjainkat. Ez esetben a weblapon legtöbbször reklámcsíkot helyeznek el. Weblapunkat természetesen egy fizetős szolgáltatást nyújtó szerveren is elhelyezhetjük, ahol minden szükséges szoftvereszköz is rendelkezésünkre áll.
7. A közzétételt követi a **weblap tesztelése,** amelynek során kiderülhetnek a tervezés és kivitelezés során esetlegesen elkövetett hibák.
8. **A weboldal népszerűsítése és támogatása.** Ahhoz, hogy a látogatók megtalálják és látogassák a weblapot, célszerű azt regisztrálni a keresőszolgáltatóknál, az internetes katalógusokban és hasznos, ha más oldalakon hivatkozásokat helyezzenek el az oldalra. A folyamatot **népszerűsítésnek, optimalizálásnak** nevezik. A népszerűsítésnek egyéb eszközei is vannak, de a legfontosabb mégis az, hogy a webhelyünk tartalma érdekes legyen a látogatók számára.
9. Ahhoz, hogy biztosítsuk webhelyünk töretlen népszerűségét, állandóan új, érdekes anyagokkal kell azt bővíteni, frissíteni. Előfordulhat, hogy az oldalt **újra kell tervezni.** Ezek a munkák alkotják az oldal **karbantartását.**

A webdizájn alapjai

A weboldalak létrehozása során fontos szerepe van a webdizájnnak. Ez a tervezésnek az a része, amely a weblapok megjelenésével foglalkozik. Mint minden tervezési tevékenység során, a webes tervezés során is be kell tartani bizonyos kompozíciós elveket, amelyek

lényegében megegyeznek a 10. osztályban a prezentáció tervezése során megismert kompozíciós, színelméleti és ergonómiai elvekkel.

A webdizájn fontos eleme a stílus. A dizájnterv valamely stílusra épül (4.64. ábra).



4.64. ábra. Dizájn-stílusok

Amikor **színstílust** (színsémát) választunk, fontos figyelembe venni, milyen a weblapok különböző korú látogatóinak színérzékelése:

- a **sötét** stílus többnyire sötét árnyalatok alkalmazását jelenti. Az ilyen stílust jól érzékelik a fiatalok, de kellemetlenül hathat az idősebb látogatókra;
- a **világos** stílus leginkább világos árnyalatokat alkalmaz, minden korosztály jól érzékeli, de az ilyen stílus bizonyos felhasználóknak túlontúl egyszerűnek tűnhet;
- az **egyszínű** stílus esetében valamely szín árnyalati vannak jelen, amely természetesen nem fehér vagy fekete. A választott szín általában a webhely tulajdonosát jellemzi;
- a **sokszínű** stílus több színt alkalmaz. Leginkább gyerekeknek szánt webhelyek kialakításánál alkalmazzák.

Nagyon sokféle weblap-stílust különböztethetünk meg **tematika szerint.** Lássunk néhány példát:

- **Retró stílus** – a weboldal kialakításában a múlt századra jellemző tárgyak, bútorok képeit, az akkori idők jellemző színeit alkalmazzák;
- **Piszok** (grunge) **stílus** – az elemek elhelyezése aszimmetrikus, ami kaotikus hatást kelt, mintha a weblapot figyelmetlenül szerkesztették volna (4.65. ábra).
- **Futurisztikus stílus** – a weblap kialakításban jövőbeli tárgyak, robotok, mechanizmusok kapnak kiemelt szerepet;
- **Hírlap stílus** – a weblap egy folyóirat stílusában van kialakítva, jól kiemelt címekkel, nagy felbontású képekkel;
- **Rajzfilmes stílus** – rajzfilmekre jellemző elemeket tartalmaz, alapvetően vidám hatást kelt.

4.65. ábra. Példa a **Piszok** stílusra

A fenti felsorolás csak töredékes, mivel bármely témához meg lehet találni az arra jellemző sajátos arculatot.

Az arculat tervezése során az esztétikai szempontokon kívül mindenképpen szerepet kap a weblap rendeltetése. A látogatónak hamar meg kell értenie, mi célból hozták létre a weblapot, könnyen fel kell ismernie annak tartalmát, intuitívan rá kell jönnie a navigálás módjára. A legnépszerűbb weblapok elemzése azt mutatja, hogy minél egyszerűbb a dizájn, annál könnyebben tájékozódik a felhasználó a weblapon. Ha a weboldal grafikával túlzsúfolt, a látogató könnyen figyelmen kívül hagyhatja azokat a lényeges tartalmi elemeket, amelyek miatt az oldalt meglátogatta.

Mindent egybevetve a weboldalak kialakítása során célszerű a következő szabályok betartása:

- A weblap színstílusában juttassuk kifejezésre a saját ízlésünket, a weblap témáját és a potenciális látogatók preferenciáit. Ne használjunk túl sok színt.
- Ügyeljünk arra, hogy a háttér és a szöveg színe legyen kontrasztos, ez megkönnyíti az olvasást.
- A szöveget tagoljuk bekezdésekre, a bekezdések között hagyjunk térközöket.
- Válasszunk olyan betűméretet, ami kényelmes olvasást tesz lehetővé, ne legyen se túl kicsi se túl nagy. A címek legyenek betűmérettel kiemelve. A betűstílus minden oldalon legyen ugyanaz.
- A címeket igazítsuk középre, a szöveg legyen kiegyenlített.

- A strukturált szöveget helyezzük el táblázatokban. A szöveg és a grafika táblázatba foglalása érdekes kompozíciós lehetőségeket rejt.
- Ne használjunk túl sok animációt, zenét és grafikát. Ezek túlzott használata figyelemelterelő hatással bír.
- A webhely oldalai közötti kapcsolatok biztosítására használjunk hivatkozásokat, de ezek száma se legyen túlságosan sok. A hivatkozásokat emeljük ki színnel, hogy a látogató azonnal észrevegye azokat és lássa, melyeket látogatott már meg ezek közül.
- A webhely logóját alakítsuk a főoldalra mutató hivatkozássá.
- Hozzuk létre és bocsássuk a látogató rendelkezésére a weblap térképét, hogy könnyebben tájékozódhasson a webhelyen.

Ellenőrző kérdések

1°. Milyen lépésekből áll a weblap létrehozása?
2°. Milyen tevékenységek tartoznak a feladat megfogalmazásához?
3°. Mit értünk a weblap belső szerkezete alatt?
4°. Milyen összetevői vannak a külső szerkezetnek?
5°. Milyen objektumok tulajdonságait tartalmazza a makett?
6°. Mi a tördelés? Mi a tördelés eredménye?
7°. Mi a hoszting? Kik a hoszting-szolgáltatók?
8°. Mit takar a webhely népszerűsítése és karbantartása?
9°. Magyarázzuk meg a stílusok csoportosítását!
10*. Miért kell a weblap fejlesztése során figyelembe venni a potenciális látogatók preferenciáit?
11°. Ismertessük a webfejlesztés szabályait!

Végezzük el a feladatokat!

1°. Látogassuk meg a http://pets.uz.ua,http://palace.dp.ua, http://www.odnoklasnyk.org.ua, http://smakuje.livejournal.com webhelyeket, és válaszoljunk az alábbi kérdésekre.

a) Milyen céllal hozták létre a webhelyet?
b) Mi a weblap tematikája?
c) Milyen típusú a weblap?
d) Milyen eredeti elemeket tartalmaz a weblap?
e) Kik a webhely potenciális látogatói?
f) Milyen szolgáltatásokat biztosít a webhely?
g) Milyen, az adott webhelyen megismert ötletet alkalmaznánk saját weblapunk fejlesztése során?

2°. Nézzük meg a http://www.kazka.in.ua, http://www.npblog.com.ua, http://maque.org.ua, http://neyrat.net webhelyeket, és hasonlítsuk össze azok arculatát! Elérhetők-e a webhelyek térképei? Mennyire teszi ez kényelmessé a webhely használatát?

3°. Nézzük meg a http://platfor.ma, http://starodub.org.ua, http://molode.com.ua, http://www.4uth.gov.ua, http://osobystist.com webhelyeket! Hasonlítsuk össze ezek külső szerkezetét! Határozzuk meg, tartalmaznak-e ezek a helyek cím-blokkot, információs blokkot, menüt!

4*. Válasszunk egyet a következő témák közül: *Veszélyes állatok, Kozmikus kutatások, Rekordok, Tudományos felfedezések, Művészeti szalon.*

a) Válasszunk egy weboldaltípust a témához.

b) Hozzuk létre szövegszerkesztőben a weblap belső szerkezetét és mentsük el **4.8.4.docx** néven.

c) Hozzuk létre grafikus szerkesztőben a weblap külső szerkezetét és mentsük azt **4.8.4.bmp** néven.

5•. Nyissuk meg az **ArtPoint** (http://artpoint.com.ua). weblapot! Elemezzük a korporatív stílusban készült munkákat! Figyeljük meg, milyen módon kapcsolódik a téma az alkalmazott színsémához! Mi módon lehetne jobban kihangsúlyozni webhelyünk stilisztikai egységét?

6°. Nyissuk meg a **Файнохост** (http://www.faynohost.com.ua/design-templates.html webhelyet, és vizsgáljuk meg a sablonokat! Figyeljük meg a sablonok külső szerkezetét! Mely sablonokat alkalmaznánk saját webhelyünk fejlesztése során?

7•. Nyissuk meg az alábbi webhelyeket, és határozzuk meg, milyen dizájnstílust alkalmaztak a fejlesztők!

URL-cím	*Stílus*	
	Színséma	*Tematikus stílus*
http://www.nbg.kiev.ua		
http://1000plastinok.net		
http://www.vovazilvova.net		
http://fcdynamo.kiev.ua		
http://www.janda.ru		

8*. Keressünk az interneten 3–4 olyan webhelyet, amelynek témája megegyezik a 4. feladatban választott témával! Értékeljük, mennyire tartották be a fejlesztők a webes fejlesztés szabályait! Milyen, az adott lapon alkalmazott ötletet használnánk a saját webhelyünk fejlesztése során!

4.9. A webfejlesztés automatizálásának eszközei

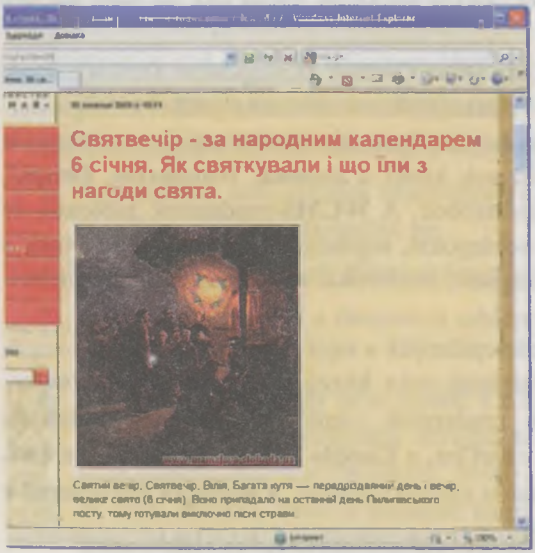
1. Mi a hoszting? Kik a hoszting-szolgáltatók?
2. Milyen kritériumok alapján osztályozzuk a weblapokat? Milyen típusokat ismerünk?
3. Milyen blokkokból áll egy weboldal?
4. Milyen grafikus- audio- és videó-fájlformátumokat ismerünk? Ezek közül melyik tömörített?
5. Milyen módon változtathatjuk meg a fájlformátumokat?

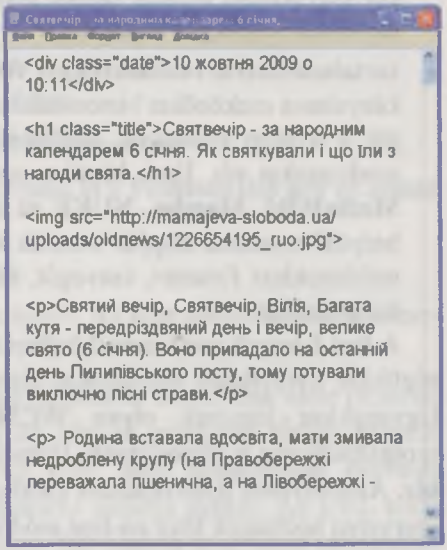
A webfejlesztés eszközei

9. osztályban már megtanultuk, hogy a weboldal egy szöveges állomány, amelynek kiterjesztése az oldalak közötti mozgásra szolgáló hiperhivatkozásra utal. A weboldalak **HTML** nyelven készítik. Az ezen a nyelven megírt fájlok kiterjesztése **htm** vagy **html**. Az ilyen fájlok **HTML-kódot** tartalmaznak, amelyek kétféle adatból állnak:

- az oldalon megjelenítendő szövegből;
- parancsokból, amelyek a megjelenítendő szövegrész **formátumát** határozzák meg. Ezeket a parancsokat **tag**-eknek (címkéknek) is nevezik.

A böngészőben megnyitott weblap **HTML**-kódját megtekinthetjük, ha az **Internet Explorer**-ben végrehajtjuk a ***Nézet*** ⇒ ***Forrás*** parancsot. **Mozilla Firefox** böngésző esetében ez a parancs a ***Nézet*** ⇒ ***Oldal forrása, Operában Menü*** ⇒ ***Oldal*** ⇒ ***Fejlesztőeszközök, Google Chrome***-ban ***Eszközök*** ⇒***Forrás.***





4.66. ábra. Egy weboldal részlete és **HTML-kódja**

A 4.66. ábrán egy weboldal részletét láthatjuk és annak HTML kódját.

A weboldalt különböző szoftvereszközök segítségével készíthetjük:

- **szövegszerkesztőkkel,** amelyben a felhasználó beírhatja a szöveget és a **tag**-eket is. Erre a **Jegyzettömb,** az **Edit Plus,** a **Homsite,** a **HTML Pad** stb. szövegszerkesztőket használják. Az elkészült oldalt **htm** vagy **html** kiterjesztéssel kell menteni. A felhasználónak természetesen jól kell ismerni a **HTML**-kódokat. A webfejlesztésnek ez a módja nagyon lassú, azonban a kis fájlméretnek köszönhetően letöltésük igen gyors;
- **általános rendeltetésű szoftvereszközökkel,** például a **Microsoft Office** programjaival, amelyekben lehetséges a dokumentum mentés **HTML** formátumban. Ilyenek

a **Word 2007**, a **Power Point 2007**, a **Publisher 2007**, amelyekben a dokumentumot weblapként (**.htm, *.html*) menthetjük. Ez esetben a weboldal létrehozásához nem szükséges a HTML nyelv ismerete. Az így készült fájlok mérete meghaladja a szövegszerkesztőben írt fájlok méretét. Ennek oka a tag-ek többszöri alkalmazása a dokumentum azonos formátumú elemeinél;

- **speciális webszerkesztőkkel**, amelyek rendeltetése a weboldalak létrehozása. Ilyenek az **Adobe Dreamweaver**, a **Microsoft FrontPage**, a **SharePoint Designer**, a **WYSIWYG Web Builder**, a **KompoZer** stb. Ezek a szoftverek speciális eszköztárakkal rendelkeznek statikus és dinamikus oldalak fejlesztéséhez. Használatuk nem igényli a **HTML** nyelv ismeretét. Ezeket a programokat szokás **WYSIWYG-szerkesztőknek** (*What You See Is What You Get* – amit látsz, azt kapsz) is nevezni. A segítségükkel létrehozott weboldalakat olyannak látjuk a szerkesztőben, amilyennek a böngészőben látni fogjuk. Az előállított **HTML**-kód sokkal korrektebb, mintha **Office** programokkal állítottuk volna elő, de még mindig tartalmaz felesleges elemeket;
- **tartalomvezérlő rendszerekkel WCMS** (*Web Content Management System*). Ezek kényelmes eszközöket biztosítanak a felhasználók számára, amelyekkel kidolgozhatják a weblap külalakját, feltölthetik azt tartalommal, kialakíthatják a navigációs rendszereket stb. Ezek közül a rendszerek közül a **Joomla, Wordpress, Drupal, MediaWiki, Mambo, NUKE** az ismertebbek. A **WCMS**-rendszerek sablonok és beépülő-modulok alapján készítik a weblapokat, segítségükkel könnyen adhatunk a weblapokhoz fórumot, csevegőt, szalagokat, statisztikai eszközöket, egyéb dinamikai elemeket.

A fent felsorolt szoftvereszközöket feltelepíthetjük a saját számítógépünkre. A segítségükkel kivitelezett weblapokat létrehozásuk után közzé kell tenni az interneten. Ugyanakkor léteznek olyan **WCMS** rendszerek, amelyek ingyenes hosztingszolgáltatásokat is biztosítanak. Ilyenek az **uCoz, a Google Sites, a Prom.ua, az Ua7.biz.** Amennyiben ilyen rendszer alkalmazása mellett döntünk, a weblapot közvetlenül a szerveren hozhatjuk létre on-line módban.

Automatikus webfejlesztés

Az on-line webfejlesztés és tartalomvezérlés a következő lépések megtételét feltételezi:

1. Regisztrálnunk kell a szerveren.
2. Ki kell választani a weblap nevét és az alkalmazandó sablont.
3. Létre kell hozni a weboldalakat és a navigációs elemeket.
4. Fel kell tölteni az oldalakat tartalommal.

Vizsgáljuk meg ezeket a lépéseket részletesebben a **Google** ingyenes szolgáltatás példáján. A szolgáltatás neve **Google Sites** (Google webhelyek).

Mielőtt hozzáfognánk a weblap létrehozásához, regisztrálnunk kell a **Google** világában. E célból:

1. Nyissuk meg a www.google.com.ua lapot.
2. Kattintsunk a ***Bejelentkezés,*** majd a ***Hozzon létre most fiókot*** linkre.
3. Töltsük ki a **Fiók létrehozása oldal** mezőit, majd kattintsunk az **Elfogadom. Készüljön el a fiókom** gombra.
4. Aktiváljuk a bejelentkezésünket. Kattintsunk az oldalon arra a linkre, amelynek segítségével bejelentkezhetünk a frissen regisztrált postafiókunkba.
5. Kattintsunk a **Belépés** linkre.
6. Töltsük ki a **Felhasználói** név és **Jelszó** mezőket.
7. Kattintsunk a **Bejelentkezés** gombra.
8. Kattintsunk a **Webhelyek** hivatkozásra.

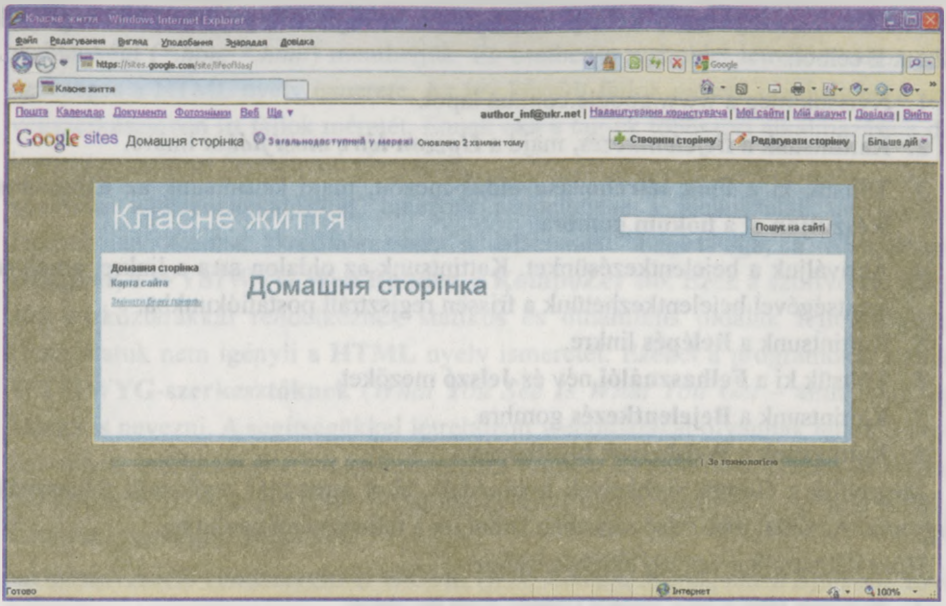
Megnyílik a **Google** webhelyek nyitóoldala, ahol most már regisztrált felhasználók vagyunk. Az oldal jobb felső sarkában láthatjuk a felhasználói nevünket.

Hozzáláthatunk a weblap létrehozásához:

1. Nyissuk meg a http://www.google.com.ua oldalt.
2. Kattintsunk a ***Továbbiak,*** majd a ***Webhelyek*** hivatkozásra.
3. Amennyiben nem voltunk bejelentkezve, töltsük ki a **Felhasználói név** és **Jelszó** mezőket, majd jelentkezzünk be.
4. Kattintsunk a **Webhely létrehozása** gombra.
5. Válasszunk ki egyet a felajánlott sablonok közül. Az üres sablon az alapértelmezett.
6. Töltsük ki a **Webhely** neve mezőt. Ez lehet például az osztályunk élete (*Класне життя*). Ezzel egyidejűleg létrejön az URL cím is, amelynek alakja: **https://sites.google.com/site/<webhely_neve>.** Az URL nem tartalmazhat ékezetes és cirill betűket, ezért a rendszer angol betűkké alakítja ezeket. Az URL tehát **https://sites.google.com/site/klasnezitta** lesz. Ezt később meg lehet változtatni. Az URL-nek egyedinek kell lennie.
7. Válasszuk ki a **Témát.** Alapértelmezés szerint a sablonbeli *Alapértelmezett* téma van kiválasztva. Válasszuk ki például a *Sportosat.*
8. Írjuk be a beviteli mezőbe **a rajzon látható kódot.**
9. Kattintsunk a **Webhely létrehozása** gombra.

A böngésző ablakában megjelenik a létrehozott honlap. A honlap jobb felső sarkában láthatjuk a kiválasztott nevet. Az ablak bal oldalán találjuk a navigációs panelt, középső része tartalmazza az információs blokkot, azaz a **Főoldalt** (4.67. ábra).

A navigációs panel jelenleg két hivatkozást tartalmaz: a **Főoldalt** és a **Webhelytérképet.** A webhelytérkép jelenleg egyetlen hivatkozást tartalmaz, ami a főoldalra mutat.



4.67. ábra. A frissen létrehozott honlap

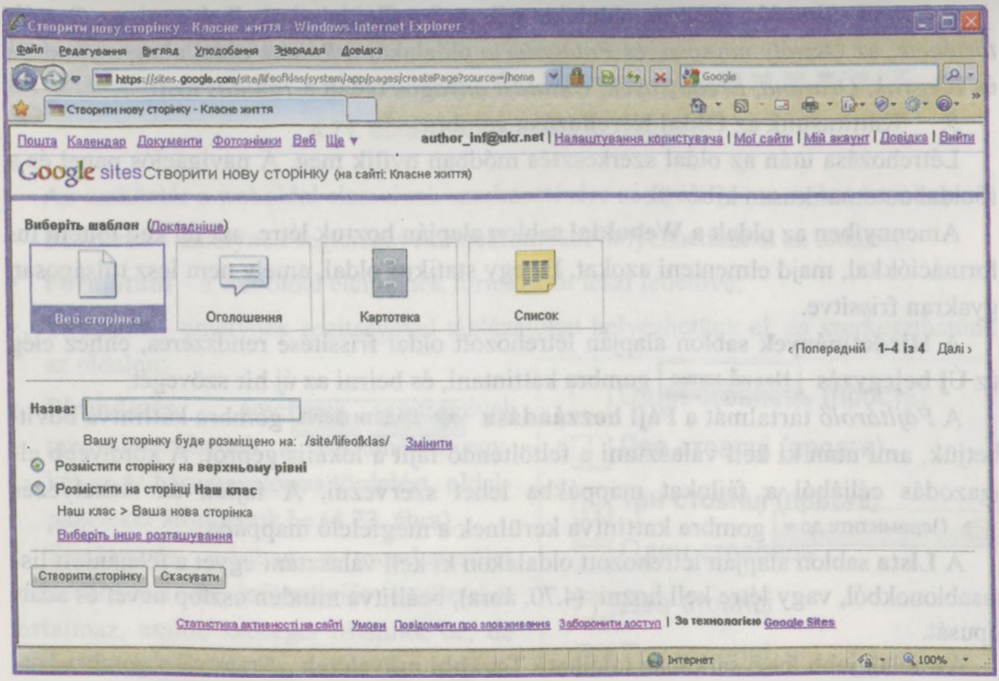
Weboldalak létrehozása és beállítása

Mivel a frissen létrehozott webhely csak egyetlen oldalt tartalmaz, szükség lehet új oldalak létrehozására.

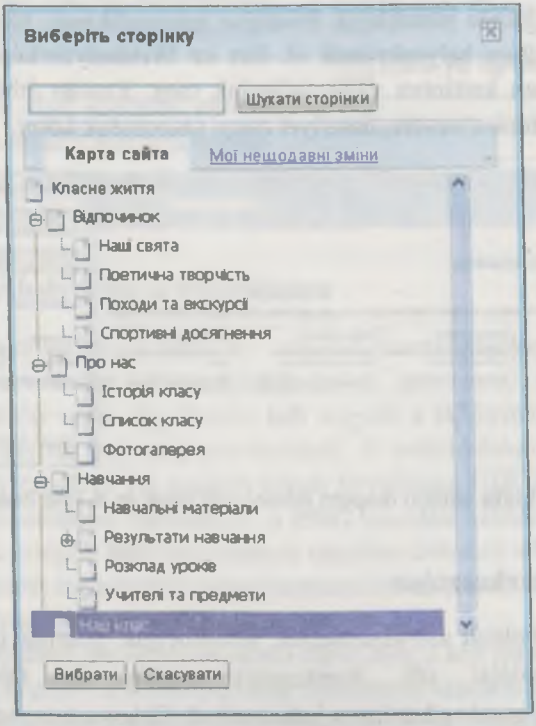
1. Kattintsunk az ablak jobb felső sarkában található **Oldal hozzáadása** Створити сторінку gombra.
2. Válasszuk ki a létrehozandó oldal sablonját (4.68. ábra):
   - **Weboldal** – szöveget, képet tartalmazó oldal. Tematikus információs oldal létrehozására ajánlható.
   - **Hirdetmények** – kronologikus sorrendű szöveges információk (hírek) elhelyezésére alkalmas, amelyek közül a legfrissebb kerül a lap tetejére. Lehetőség van az egyes hírek külön oldalként történő megnyitására.
   - **Fájltároló** – linkeket tartalmaz a megnyitható fájlokra. A fájlok mappákba vannak csoportosítva. Tipikus konténeroldal, amely fájlkatalógust tartalmaz.
   - **Lista** – többmezős, strukturált adatok megjelenítésére alkalmas lap. A listát az egyes mezők szerint rendezhetjük.

Az *Osztály történetének* elhelyezésére a Weboldal a megfelelő választás (4.68. ábra), a *Tanárok és tantárgyak* elhelyezésére a **Lista** a legalkalmasabb, a *Verseket* **Hirdetményeken,** az Oktatási anyagokat pedig **Fájltárolóban** célszerű elhelyezni.

3. Töltsük ki a **Név** mezőt.
4. Határozzuk meg az oldal helyét a webhely szerkezetében. Az **Oldalt elhelyezhetjük a legfelső szinten,** vagy egy **Meghatározott oldal alá.** Ha ezek közül egyik sem megfelelő, kattintsunk a **Válasszon másik helyet** linkre. Az első esetben a főoldalra kerül az adott oldalra mutató link, a második esetben pedig a kiválasztott oldalra. Ha a harmadik lehetőség mellett döntöttünk, akkor ki kell választani a megfelelő oldalt (4.69. ábra), ahová az új oldalra mutató link kerül.



4.68. ábra. **Új oldal létrehozása** lap



4.69. ábra. **Oldal választása** ablak

A Rólunk, Tanulás, Pihenés oldalakat célszerű a főoldal alatt elhelyezni, az *Osztály története,* az *Osztály névsora,* és *Fotógaléria* oldalakat a *Rólunk* oldal alatt, a *Tanárok* és *Tárgyak, Órarend, Eredmények, Oktatási anyagok* oldalt a *Tanulás* alatt.

5. Kattintsunk az **Oldal létrehozása** gombra.

Létrehozása után az oldal szerkesztés módban nyílik meg. A navigációs panel és a főoldal automatikusan kibővül.

Amennyiben az oldalt a **Weboldal** sablon alapján hoztuk létre, azt fel kell tölteni információkkal, majd elmenteni azokat. Ez egy statikus oldal, amely nem lesz túlságosan gyakran frissítve.

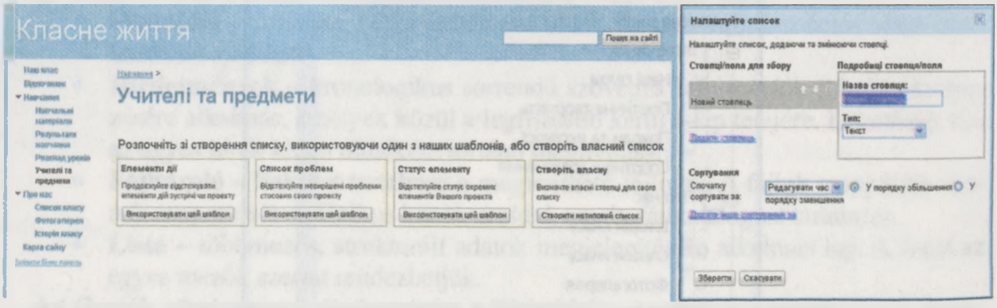
A **Hirdetmények** sablon alapján létrehozott oldal frissítése rendszeres, ehhez elég az **Új bejegyzés** Новий запис gombra kattintani, és beírni az új hír szövegét.

A *Fájltároló* tartalmát a **Fájl hozzáadása** Додати файл gombra kattintva bővíthetjük, ami után ki kell választani a feltöltendő fájlt a lokális gépről. A könnyebb eligazodás céljából a fájlokat mappákba lehet szervezni. A fájlok az **Áthelyezés** Перемістити до gombra kattintva kerülnek a megfelelő mappába.

A **Lista** sablon alapján létrehozott oldalakon ki kell választani egyet a felajánlott listasablonokból, vagy létre kell hozni (4.70. ábra), beállítva minden oszlop nevét és adattípusát.

Az ablak jobb felső sarkában található **További műveletek** Більше дій gombra kattintva megváltoztathatjuk az oldal sablonját, törölhetjük az oldalt, szabályozhatjuk a hozzáférési beállításokat stb.

A webhely navigációs paneljának frissítése automatikusan történik. Az oldalak a panelen abc-sorrendben helyezkednek el. Ezt az **Webhelytérkép,** majd az **Oldalak kezelése** hivatkozásra kattintva változtathatjuk meg. Ezután áthúzhatjuk az oldalak nevét annak az oldalnak a nevére, amellyel össze szeretnénk kötni.



4.70. ábra. A **Lista** sablon alapján létrehozott oldal és a lista beállítása ablak

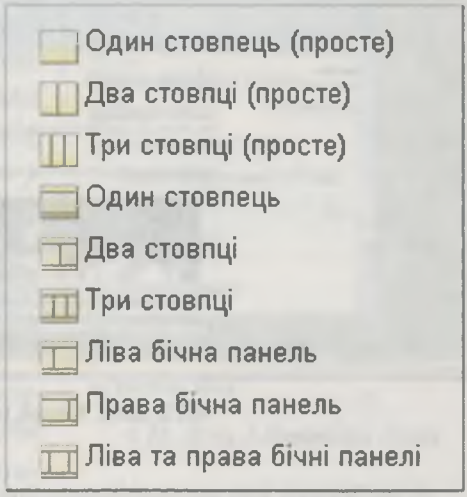
Weboldalak szerkesztése

A létrehozott weboldalt szerkeszthetjük, feltölthetjük információval, megváltoztathatjuk a modulhálóját stb. Szerkesztés módba az **Oldal szerkesztése** Редагувати сторінку gombra kattintva juthatunk el. Ekkor a böngésző ablakában megnyílik a szerkesztés eszköztár (4.71. ábra).

4.71. ábra. A szerkesztés eszköztár

Az eszköztár a weboldal elemeinek szerkesztésére szolgáló parancsokat tartalmaz:

- **Beszúrás** – amelynek segítségével objektumokat helyezhetünk el az oldalon;
- **Formátum** – a weboldal elemeinek formázását teszi lehetővé;
- **Táblázat** – amelynek segítségével táblázatokat helyezhetünk el, és szerkeszthetünk az oldalon;
- **Elrendezés** – amelynek segítségével megváltoztathatjuk a modulhálót: egy, két vagy háromoszlopos tördelést, oldalpaneleket állíthatunk be (4.72. ábra).



4.72. ábra. Weboldalak elrendezése

Szerkesztés módban a munkaterület címterületeket és információs területeket tartalmaz, amibe szöveget írhatunk be, de használhatjuk a **Vágólapot** is. Ha megváltoztatjuk az oldal címét, a navigációs panel automatikusan frissülni fog. Az információs területre beírt szöveget az eszköztár segítségével formázhatjuk.

A formázás ideje alatt időnként elmentésre kerül az oldal aktuális állapota, a piszkozat. Ha befejeztük a szerkesztést, kattintsunk a **Mentés** Зберегти gombra a végleges változat mentése céljából.

Objektumok elhelyezése a weboldalon

A weboldalon különféle objektumokat – képeket, hivatkozásokat, vízszintes vonalakat, **Google**-dokumentumokat – helyezhetünk el.

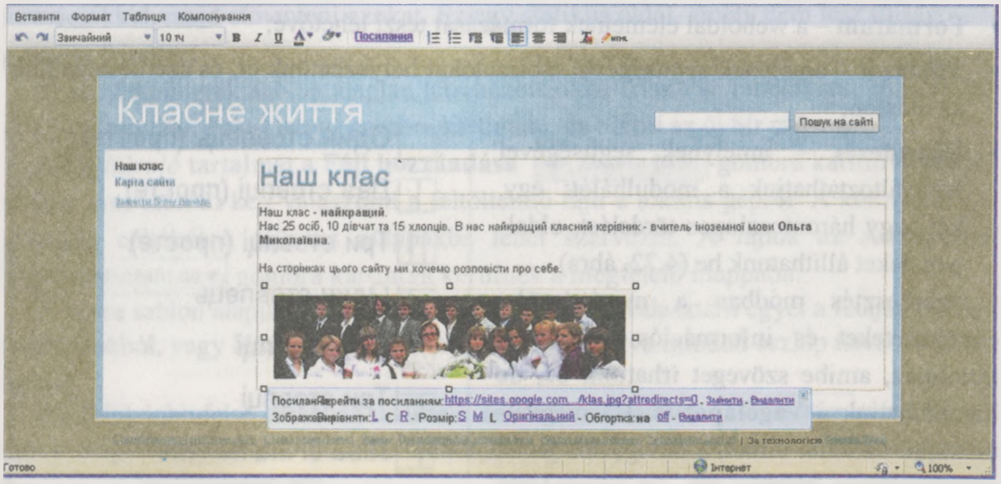
A képek beillesztése során figyelembe kell vegyük a fájlformátumokat. Ezek sajátosságaival a 9. osztályban már megismerkedtünk. A weboldalakon általában **JPG** formátumú képeket helyeznek el. Az animált képek formátuma **GIF**. Ezt a lehetőséget jellemzően az oldal színesítésére használják. A **PNG** speciális hálózati formátum, raszteres képek tárolására szolgál. Más formátumok esetében célszerű a kép elhelyezése előtt egy grafikus szerkesztő segítségével megváltoztatni annak formátumát:

1. Hajtsuk végre a ***Beszúrás – Kép*** parancsot.
2. A **Kép hozzáadása** párbeszédablakban adjuk meg a kép forrását:
   - **feltöltött képek** – a webhelyen már elhelyezett képek közül választunk, vagy a **Tallózás gombra kattintva** kiválasztunk egyet a helyi számítógépen tárolt képek közül.

- **Internetcím (URL)** – az adott URL-címen elhelyezkedő fájl.

3. Válasszuk ki a megfelelő fájlt vagy adjuk meg az URL-jét.
4. Kattintsunk az **OK** gombra.

A kép beillesztése után megnyílik a szerkesztés panel, amelynek segítségével megadhatjuk a kép elhelyezkedését és a méretét (4.73. ábra). A panel hivatkozásinak segítségével a képet **L** – bal oldalon, **C** – középen, **R** – jobb oldalon helyezhetjük el. Beállíthatjuk az **S** – kicsi, **M** – közepes, **L** – nagy (Eredeti) méretet is.



4.73. ábra. Képek beillesztése a weboldalra

A beillesztett kép automatikusan linkelődik ahhoz a fájlhoz, amely a képet tartalmazza. Ha erre a hivatkozásra kattintunk, a kép eredeti méretben nyílik meg a böngésző ablakában. Ha az **Ugrás** a linkre sorban a **Módosítás** linkre kattintunk, lecserélhetjük a képet.

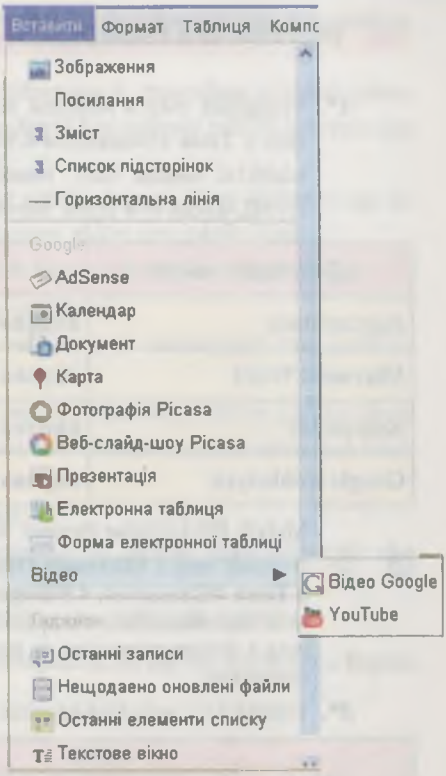
A **Beszúrás** menü (4.74. ábra) segítségével a **Google** szolgáltatások által létrehozott objektumokat (térképeket, naptárakat, táblázatokat, űrlapokat), **Picasa** fotókat vagy vetítéseket helyezhetünk el. Ezek az objektumok **gadgetek** (barkácsolás, kütyü, bigyó) – speciális programok, amelyek rendeltetése bizonyos objektumok megjelenítése a weboldalon – segítségével kerülnek az oldalakra.

Videót a **Google videó** vagy a **YouTube** forrásokból csatolhatunk az oldalhoz. E célból hajtsuk végre a ***Beszúrás – Videó*** parancsot, válasszuk ki a videó forrását, majd adjuk meg az URL címet.

A hiperhivatkozás segítségével léphetünk át lapunkról egy másik webes erőforrásra. A beszúrás- link paranccsal illeszthetjük be. A megnyíló párbeszédablakban válasszuk ki az oldalt, vagy adjuk meg annak az oldalnak az URL-jét, amelyre a hivatkozás mutat. Az oldalon a link annak az oldalnak a nevén lesz elhelyezve, ahová a hivatkozás mutat. A hivatkozást összeköthetjük egy szövegrésszel vagy egy képpel. Jelöljük ki a szövegrészt vagy képet, majd kattintsunk a **Link**-re Посилання az eszköz-

táron. Ezután meg kell adni annak az objektumnak a nevét vagy URL-jét, ahová a hivatkozás mutat.

Az olyan fájlokat, amelyek elhelyezése az oldalon nem lehetséges (flash, hang stb.), szintén elhelyezhetjük a webhelyünkön. A mentett fájlok a szerveren tárolódnak, a weboldalon pedig a **Beillesztett fájlok** részben olyan hivatkozások kerülnek be, amelyekre kattintva a fájlokat le lehet tölteni a helyi gépre, le lehet törölni, vagy meg lehet nyitni, amennyiben annak formátuma megegyezik a **Google** valamelyik formátumával. A fájlok feltöltése céljából kattintsunk az oldal alján a **Beillesztett fájlok** linkre, majd pedig **Fájl megnyitása** ablakban ki kell választani a helyi gépen a megfelelő fájlt. Ezután a fájl felmásolódik a szerverre. Ezután a **Törlés, Megnyitás** és **Letöltés** hivatkozásra kattintva elvégezhetjük a szükséges műveletet. A honlapra fájlokat nem helyezhetünk el.



4.74. ábra. A **Beszúrás** menü

A feltöltendő fájlok esetében célszerű megvizsgálni annak formátumát, és szükség esetén elvégezni annak átalakítását (különösen audio- vagy videofájlok esetében) valamilyen tömörített formátumra, például **MP3**-ra, **AVI**-ra vagy **MP4**-re.

Ellenőrző kérdések

1°. Milyen eszközök segítségével hozhatunk létre weboldalakat? Melyiknek milyen előnye és hátránya van?
2°. Mi a HTML-kód? Milyen adatokat tartalmaz? Hogy tekinthetjük meg?
3°. Milyen webszerkesztőket ismerünk? Mik az alkalmazásuk sajátosságai?
4°. Mik a CMS rendszerek? Nevezzünk meg néhány ismertebb CMS-t!
5°. Mely CMS-eket használhatjuk on-line módban? Mik az alkalmazásuk sajátosságai?
6°. Milyen lépésekből áll a weblap automatizált létrehozása?
7°. Milyen sablonokat használhatunk a weblap **Google webhelyek** eszközzel történő végrehajtása során?
8°. Hogy helyezünk el képet egy weblapon? Milyen forrásokat használhatunk ehhez?
9°. Hogy kötünk össze egy hiperhivatkozást egy szövegrésszel?
10°. Hogy helyezzünk el egy videoobjektumot a weboldalon? Milyen forrásokat használhatunk?
11°. Mi a gadget?
12°. Milyen audio- és videoformátumok ajánlatosak webes elhelyezés céljából? Mivel magyarázhatjuk ezt?
13*. Miben különbözik a kép beillesztésének művelete annak feltöltésétől?

Végezzük el a feladatokat!

1°. Vizsgáljuk meg a tanárunk által megnevezett HTML-oldalakat, például azokat, amelyek a **Тема 4\Завдання 4.9\Вправа 1** mappában találhatók! Ezeket különböző eszközökkel hozták létre. Hasonlítsuk össze a weblapok tartalmát a böngészőben, a HTML-kódot és a fájlok méretét! Töltsük ki a táblázatot!

Létrehozás módja	*Fájl neve*	*Fájlméret*
Jegyzettömb	**вправа 4.4.1.1.html**	
Microsoft Word	**вправа 4.4.1.2.html**	
KompoZer	**вправа 4.4.1.3.html**	
Google webhelyek	**вправа 4.4.1.4.html**	

Melyik fájl letöltése történik meg a leggyorsabban?.

2*. Nyissuk meg a **Microsoft Office** különböző alkalmazásaival készült dokumentumokat a **Тема 4\Завдання 4.9\Вправа 2 mappából!** Mentsük ezeket ugyanolyan néven, de html formátumban a mappánkba! Figyeljük meg a fájl ikonjának változását! Nyissuk meg a dokumentumokat, és hasonlítsuk össze az eredeti fájlokkal! Vessük össze a fájlméreteket!.

3°. Töltsük ki a weboldal-készítés eszközeit összehasonlító táblázatot!

Eszköz	*HTML ismereteket igényel*	*Oldalfeltöltési és szerkesztési eszközöket tartalmaz*	*Használata kényelmes*	*Hosztingszervert igényel*
Szövegszerkesztő				
Microsoft Office				
Webszerkesztő				
CMS				

4°. Regisztráljunk a **Google** szerveren a fentebb ismertetett eljárás szerint!

5°. Hozzunk létre **Google** webhelyet a 4.61. ábrán látható vázlat szerint! Válasszuk egy tetszőleges sablont. Az *Osztály története oldalt weboldal,* a *Tantárgyakat és tanárokat lista,* a *Verseket Hirdetmény,* a *Tananyagokat fájltároló* sablon alapján hozzuk létre. A többi oldal sablonját válasszuk ki önállóan. Tudassuk tanárunkkal az oldal URL-jét.

6°. Hozzunk létre a **Google** szerveren egy *Művészeti szalon* nevű webhelyet!

1. Az oldal URL-je legyen a nevével megegyező.
2. Alkalmazzunk egy tetszőleges sablont.
3. Az oldal belső szerkezete legyen a **Тема 4\Завдання 4.9\Вправа 6\схема.docx** fájlban leírt.
4. Töltsük fel a főoldalt adatokkal, amely ismerteti az oldal rendeltetését, az oldalak tartalmát és a fejlesztő adatait!

5. Hozzunk létre egy oldalt a *Weboldal* sablon alapján, és helyezzük azt el a webhely belső szerkezetének megfelelően!
6. Töltsük fel az oldalt a **Тема 4\Завдання 4.9\Вправа 6,** mappában található szöveges és grafikus adatokkal! A szöveg és a megfelelő kép egymás mellett helyezkedjenek el! Alkalmazzunk *Kétoszlopos* tördelést!
7. Helyezzünk el hivatkozásokat az oldalon oly módon, hogy a képre kattintva egy információs oldal nyíljon meg, ha pedig a szövegre, akkor egy másik oldal!
8. Töltsük fel az oldalra a **Тема 4\Завдання 4.9\Вправа 6\Файли** mappa fájljait!
9. Küldjük el a tanárunknak az oldal URL-jét e-mail-ben.

7*. Nyissuk meg az előző feladatban létrehozott webhelyet, és változtassuk meg a következőképpen:

1. Adjunk hozzá egy *Események* nevű oldalt, amelet *Hirdetmény* sablon alapján készítünk! Helyezzünk el egy hirdetést arról a kiállításról, amely a szalonban lesz lebonyolítva!
2. Töltsük fel a főoldalra a szalon nyitva tartását!
3. Az információs oldalakat tördeljük *Két oszlopba.* Az oldalak alján helyezzük el a fejlesztő elérhetőségeit!

8*. Hozzunk létre a **Google** webhelyeken egy weblapot a leendő szakmánkról!

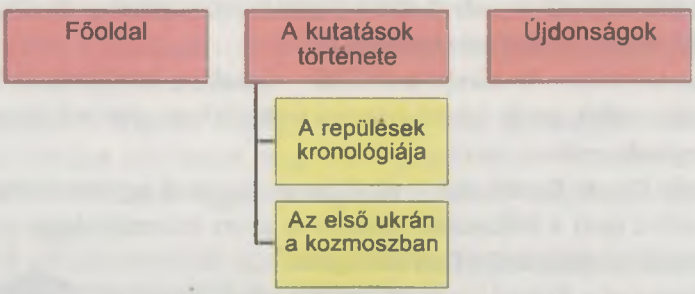
9*. Egy webszerkesztőben hozzuk létre a saját weblapunkat, és tegyük közzé azt a **TopUa** (http://www.topua.net) hoszting-szerveren!

18. számú gyakorlati munka **Weblap automatizált létrehozása**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

Hozzunk létre a **Google** webhelyeken egy weblapot a *Kozmosz ukrán kutatóiról*!

1. Az oldal URL-je egyezzen meg annak címével!
2. Válasszunk ki egy, a témának megfelelő sablont!
3. A *Főoldalra* helyezzünk el információkat a webhely többi oldaláról, és az oldal fejlesztőjéről!
4. A webhely szerkezete legyen a következő:



A kutatások történetét, Az első ukrán a kozmoszban – weboldal, az *Újdonságokat hirdetmény,* a *Kronológiát Lista* sablon alapján hozzuk létre.

5. A *kutatások története* oldalt töltsük fel adatokkal a **Тема 4\Практична 18\космічні дослідження.docx** fájlból!
6. A *kutatások története* oldalon helyezzük el hivatkozást az első mondatra, amely az *Újdonságok* oldalra mutat!
7. *Az első ukrán a kozmoszban* oldalon helyezzük el a **Тема 4\Практична 18\Попович,** mappában található információkat oly módon, hogy a kép az oldal közepén helyezkedjen el, a szöveg pedig a kép két oldalán. Alkalmazzunk háromhasábos (egyszerű) tördelést!
8. Helyezzünk el *Az első ukrán a kozmoszban* oldalon egy hivatkozást a képre, amely a http://h.ua/story/285307 URL-re mutat!
9. Szerkesszük meg *A repülések kronológiáját* úgy, hogy abban legyen helye a repülés dátumának, az űrhajós nevének, az űrhajó típusának és egy rövid leírásnak. Töltsük fel az oldalt 3–4 repülés adataival, amelyeket a **Тема 4\Практична 18\польоти.docx** fájlban találunk!
10. Helyezzünk el az újdonságok oldalon egy hírt a legutóbbi repülésről a http://space.vn.ua oldalról!
11. Töltsük fel az *Újdonságok* oldalra a **Тема 4\Практична 18\Файли тарра** fájljait!
12. Közöljük a tanárunkkal az oldal URL-jét!

4.10. A Web 2.0 technológia. Webes közösségek. Blogok létrehozása

1. Mik a gadgetek?
2. Hogy hozunk létre és hogy szerkesztjük a **Wikipédia** szócikkeket?
3. Hogy keressük meg a szükséges szócikket a **Wikipédián**?
4. Mi a fórum és a chat? Milyen a fórum szerkezete?
5. Hogy írunk hozzászólást egy fórumra? Hogy írunk e-mailt?

A Web2.0 technológia áttekintése

Az utóbbi időkben egyre inkább teret hódítanak az interneten a **Web 2.0** technológia néven ismertté lett szolgáltatások.

A **Web2.0** kifejezés néhány olyan technológiát takar, amelyek a modern webes erőforrások alapját képezi. Ezek közé tartoznak az:

- **AJAX** (*Asynchronous JavaScript and XML* – aszinkron JavaScript és XML, webes programozási nyelv), amely lehetővé teszi a weboldal egy részének dinamikus frissítését a böngészés során;
- **RRS** (*Really Simple Syndication* – ténylegesen egyszerű együttműködés), amely lehetővé teszi a felhasználó könnyű és gyors informálódását az őt érdeklő oldalak frissítéseiről – **futó szalagcím**;

- **FOAF** (*Friend of a Friend* – a barát barátja), amely lehetővé teszi, hogy a felhasználó új információkhoz jusson más felhasználóktól, akiket a *barátai listájára* bejegyzett;

- **trackback** (visszakövetés) – lehetővé teszi, hogy amennyiben egy weblapon hivatkozást teszünk egy webes erőforrásra, azon automatikusan létrejöjjön az ellenkező irányú hivatkozás.

A klasszikus webes erőforrásokra (statikus weboldalak, fórumok, chatek), amelyeket ma **Web 1.0** néven említenek, az a jellemző, hogy egy fejlesztő által vannak létrehozva, a tulajdonos által vannak feltöltve, a többi felhasználó (látogató) pedig csak olvashatja ezeket. Az információtartalom sokszor szerzői jog által védett. Az információhoz jutás egyetlen módja ez esetben, ha megnyitjuk a megfelelő oldalt. A webes erőforrás elsősorban információforrásként jut szerephez.

A **Web 1.0**-val szemben a **Web 2.0**-ra jellemző, hogy:

- a **web elsősorban platformként** szerepel, amelyen különböző dokumentumokat (prezentációkat, elektronikus táblázatokat, animációkat) hozhatunk létre akár on-line, akár a helyi gépen. Sok **Web 2.0** erőforrás programként működik egy weboldalon: ezek a **web-kiegészítők**. Ezek alkalmazása felmenti a felhasználókat attól, hogy szoftvereket kelljen telepíteni a helyi számítógépen. Jó példa erre az on-line szótár és fordító, amelyekhez nemcsak számítógépekről, de mobiltelefonokról is hozzáférhetünk;
- **az internetes erőforrások keveredése** (mash-up), amikor az erőforrások funkcionalitása egy másik erőforrás révén bővül, ugyanakkor a régebbi fejlesztésekbe beépülnek az újak. Egyre több webhelyen találhatunk videólejátszó gadgetet, a videókat és a lejátszást vezérlő programokat pedig valahol egy másik webelyen találjuk;
- **a kollektív agy technológia**, amikor egy tartalom fejlesztésében és értékelésében több felhasználó is részt vesz, mintegy közös projektet valósítva meg. A felhasználók már nemcsak fogyasztói a más által közétett tartalmaknak, hanem alkotói. Ennek példája a **Wikipédia**, amelyen minden érdekelt felhasználó részt vehet a tartalom javításában és megújításában;
- **a folkszonómia** (***folk*** –nép, ***taxonomy*** – rendezési elv), ami egy közösségi értékelési rendszer, amelyben az erőforrásokhoz tag-eket (címkéket) rendelhetünk. A címkéket a felhasználók választják a tartalom megjelölésére. A keresést a címkékből kialakuló címkefelhő könnyíti meg.
- **web-hírügynökség.** A **Web 2.0** erőforrások dinamikusak, a tartalmak különböző URL-ekről származnak, ezekhez kommentárokat lehet fűzni. Az **RSS** és a **FOAF** technológiáknak köszönhetően azokat a webhelyeket, amelyekre előjegyeztük magunkat, továbbá a barátaink webhelyeit nem kell meglátogatnunk, mivel ezek változásairól a hírszalagról értesülünk;
- **a szocializáció.** A **Web 2.0** elősegíti az olyan webes közösségek kialakulását, amelyek közös témákat vitathatnak meg, közös projektekben vehetnek részt, eközben a résztvevők létrehozhatják egyedi profiljaikat is, amely megkülönbözteti őket a közösség más tagjaitól.

A **Web 2.0** szolgáltatásainak egyik rendeltetése, hogy segítse a felhasználók közös tevékenységét és kommunikációját az interneten.

A webes szolgáltatásokat a felhasználók tevékenysége szerint csoportosíthatjuk. Tekintsük át ebből a szempontból a **Web 2.0** szolgáltatásokat (4.7. táblázat).

4.7. táblázat. **A web 2.0 szolgáltatások csoportosítása a felhasználók tevékenysége szerint**

Tevékenység	*Web 2.0-ás szolgáltatás*	*Rövid ismertető*
Információkeresés	**Szociális keresőrendszerek: Flexum** (http://flexum.ru), **Open Directory Project** (http://www.dmoz.org), **Quintura** (http://quintura.ru) Szociális keresőrendszerek	A felhasználók feltöltik az adatbázist azoknak az internetes erőforrásoknak a címeivel, amelyeket érdekesnek találtak, ezeket címkékkel látják el, ily módon létrehoznak egy listát, amelyen belül keresni lehet. A létrehozott listákat keresőoldalakon helyezhetik el, amelyek segítségével a felhasználók a listára felkerült erőforrások között kereshetnek
Kollektív hivatkozáslista létrehozása	**Közösségi hivatkozáslisták:** (http://bobrdobr.ru) (http://del.icio.us), (http://memori.ru)	A felhasználók hivatkozásokat hoznak létre az internetes erőforrásokra és címkékkel látják el azokat. Ha ezek között szeretnénk keresni, elég rákattintani a megfelelő címkére és megnyílik az arra mutató hivatkozások listája.
Közösségi tartalomfejlesztés	**Wiki oktatás:** (http://www.eduwiki.uran.net.ua), **Kijev webenciklopédiája** (http://wek.kiev.ua), **Wikiszótár** (http://uk.wiktionary.org)	Az oldal felhasználói a regisztrációt követően új oldalakat hozhatnak létre, szöveges és multimédiás tartalmakat tölthetnek fel, szerkeszthetik a saját és más felhasználók cikkeit, olvashatják a szerkesztés történetét, más (belső és külső) oldalakra mutató hivatkozásokat helyezhetnek el
	Hosztingszolgáltatás beépített CMS rendszerrel: (http://www.ucoz.ua (https://sites.google.com), (http://blog.i.ua)	A felhasználók közvetlenül a szolgáltató szerverén hoznak létre oldalakat és töltik fel azokat adatokkal. Az oldalak kész sablonok alapján készülnek. Az oldal látogatóinak általában lehetősége van arra, hogy kifejtse a véleményét az oldal tartalmával kapcsolatban
	On-line office: https://docs.google.com (http://toffix.ru), (http://www.fengoffice.com	A felhasználók az internetre mentik elektronikus dokumentumaikat. A dokumentumokat több felhasználó is olvashatja és szerkesztheti egyidejűleg. A dokumentum tulajdonosa szabályozza a többi felhasználó jogait

A 4.7. táblázat folytatása

Tevékenység	*Web 2.0-ás szolgáltatás*	*Rövid ismertető*
Kognitív térképek létrehozása	**Gliffy:** (http://www.gliffy.com), **Bubbl.us** (http://bubbl.us), **Mapul** (http://www.mapmyself.com)	A felhasználók on-line módban **kognitív térképeket** – bizonyos ismeretterület objektumait és azok kapcsolatait szemléltető animált képeket – hoznak létre, amelyek szemléletesen ábrázolják az adott terület felépítését
Közösen használt dokumentumok tárolása	**Közösségi médiatárak:** on-line fotótárak (http://flamber.ru), Közösségi videótárak (http://www.rutube.com), közösségi dokumentumtárak (http://www.scribd.com), közösségi hangtárak **Bee.fm** (http://www.bee.fm)	A felhasználók a szerveren tarthatják a saját dokumentumaikat, címkékkel láthatják el azokat, on-line olvashatják más felhasználók dokumentumait, nézhetik a képeit, videóit, kommentálhatják azokat. A fájlok kényelmesen kereshetők, közösen használhatók, más oldalakra integrálhatók
	Fájlcserélők: (file-hoszting) (http://www.shareua.com), (http://upload.com.ua), (http://fileshare.in.ua)	A felhasználók a szerverre menthetik fájljaikat. A fájlok különböző típusúak lehetnek és egyedi URL-címmel rendelkeznek. Az URL ismeretében a fájlokat más felhasználók is letölthetik. Így valósul meg a fájlcsere
Egyéni és közösségi tevékenységek szervezése	**On-line naptárak:** (http://www.google.com/calendar), (http://calendar.yandex.ru)	A felhasználók a naptárba felvehetnek eseményeket egyéni és közösségi használat céljából, figyelmeztetőket kaphatnak az események közeledtéről, kommentálhatják az eseményeket
	Szervezők: (http://organizeit.ru), (http://www.businessitonline.com)	Az on-line szervezők naptárból, címtárból, jegyzetfüzetből, tervezőből, emlékeztetőrendszerből, üzenet- és SMS-küldő rendszerből állnak. A felhasználók közösen használhatják a felsorolt eszközöket
	On-line projektmanagerek: (http://projects.zoho.com), (http://www.easyprojects.net)	Segítségükkel a felhasználók közös projekteket bonyolíthatnak le, tervezhetik azokat és felügyelhetik a végrehajtásukat, kicserélhetik véleményüket a folyamatban lévő ügyekkel kapcsolatban, jelentéseket készíthetnek

A 4.7. táblázat folytatása

Tevékenység	*Web 2.0-ás szolgáltatás*	*Rövid ismertető*
Helymeghatározás és útvonaltervezés	(http://maps.google.com), (http://www.panoramio.com), (http://earth.google.com)	A felhasználók objektumokat kereshetnek a föld felszínén, meghatározhatják azok földrajzi helyzetét, fényképeket csatolhatnak, kommenteket írhatnak az objektumokhoz, a megadott pontokon áthaladó útvonalat tervezhetnek, megtekinthetik a földfelszín műholdképeit
Kommunikáció	**Blogok (*web log* – webnapló):** (http://blogosphere.com.ua), (http://blogoreader.org.ua), (http://blogs.pravda.com.ua)	A **blogok** olyan weblapok, amelyeken személyes véleményünket, gondolatainkat fejthetjük ki valamely eseményre, személyre, alkotásra vonatkozóan. A bejegyzéseknek saját URL-je van, ezért a keresőszerverek segítségével kereshetőek. A blogon elvben bárki készíthet bejegyzést, amennyiben a blog szerzője – a blogger – ezt engedélyezi
	Közösségi hálók: (http://connect.ua), (http://www.profeo.ua), (http://vkontakte.ru), (http://my.mail.ru)	Ezeken az oldalakon a felhasználók közös érdeklődésű csoportjai (jellemzően közös érdeklődési kör, munkahely, iskola, hobbi köti össze őket), kommunikálhatnak egymással. Az oldalak a regisztráció és a kommunikáció során töltődnek fel tartalommal. A felhasználók és a bejegyzéseik között keresést végezhetünk

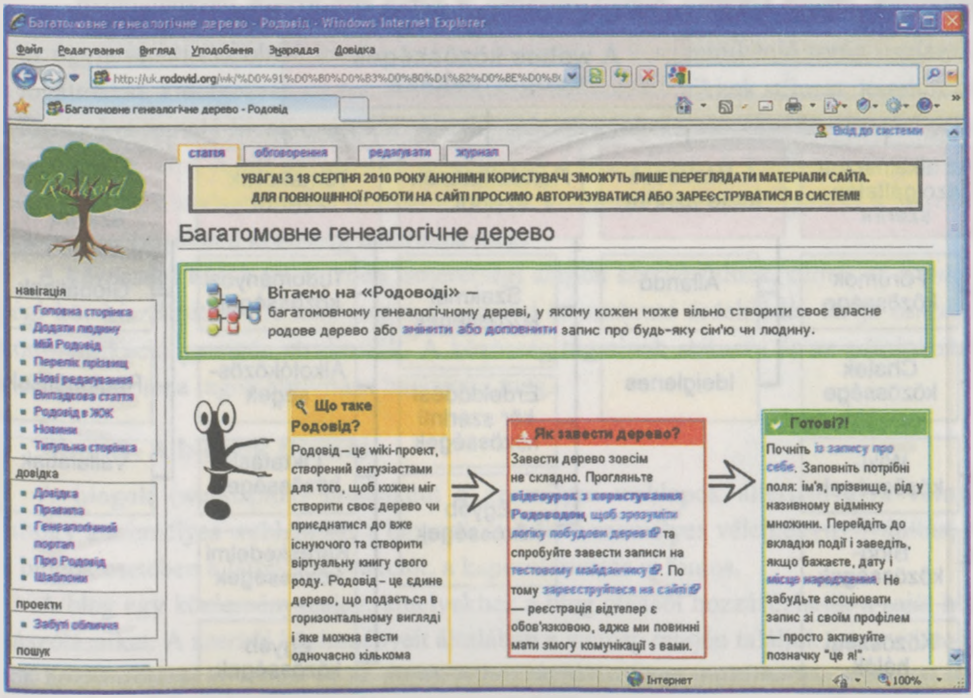
A **Web 2.0** szolgáltatások terjedése sajnálatos módon új problémák megjelenésével is jár. Mivel a szolgáltatások használata könnyen elsajátítható, az interneten nagy mértékben elterjedt a pontatlan, ellenőrizetlen információ, ami csökkentette az elektronikus források hitelét. A szerzői jog megsértése tömegjelenséggé vált. Az anonim véleménynyilvánítás lehetősége felelőtlen megnyilvánulásokra is lehetőséget biztosít. A regisztrációs adatok lehetőséget biztosítanak a személyes adatok törvénytelen használatára.

Wiki-projektek

A **Web 2.0** szolgáltatások között különleges helyet foglalnak el a wiki-projektek. Ezek rendeltetése információk gyűjtése és strukturált tárolása szerkesztési és megvitatási lehetőséget nyújtva a felhasználók meghatározott körének. A legismertebb wiki-projekt a **Wikipédia** webes enciklopédia, amelynek felhasználásával a 10. osztályban ismerkedtünk meg.

A wiki-projektek alapját az adatbázisok strukturált tárolását, webes megjelenítését, kereshetőségét biztosító wiki-technológia képezi, amely az oldalak közötti navigációt is biztosítja. A wiki-technológia a következő lehetőségeket biztosítja a felhasználóknak:

- szócikket bármelyik regisztrált felhasználó létrehozhat, a szerkesztés és olvasás joga korlátozható;
- minden szócikknek egyedi neve van a projekten belül, amely rá mutató hivatkozásul is szolgál;
- a szócikken egyidejűleg több felhasználó is dolgozhat;
- megengedett a szócikk többszöri szerkesztése, a szerkesztés története rögzítésre kerül.
- a változások a szócikk mentésével egyidejűleg jutnak érvényre;
- ha a szócikk tulajdonosának vagy a projekt vezetőjének véleménye szerint egy szerkesztési művelet rontott a cikk színvonalán, vissza lehet térni annak előző változatához;
- a felhasználó kényelmesen használható eszközöket kap, amellyel elvégezheti a betűformázást, táblázatokat, képeket, egyéb multimédiás objektumokat és hiperhivatkozásokat illeszthet be.
- lehetőség van a címkézés alkalmazására, valamint a projekt szócikkeinek kategóriákba sorolására a keresés megkönnyítése céljából;
- egy új szócikk létrehozása egy másik cikkből kiinduló hivatkozás létrehozásával kezdődik.



4.75. ábra. A **Családfa** wiki-projekt főoldala

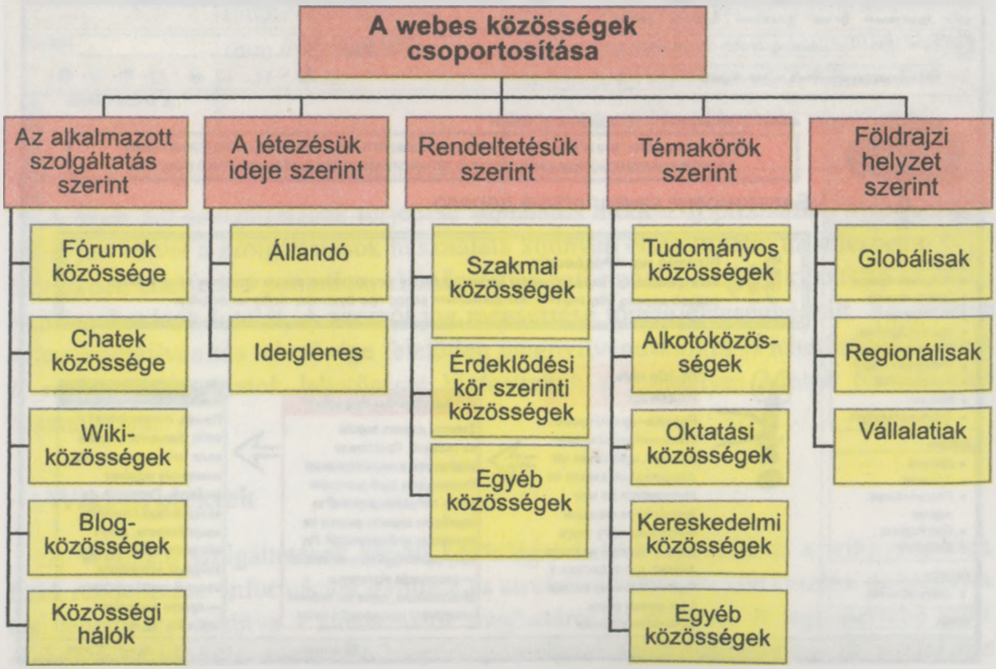
A wiki-technológia segítségével készült az **OpenWiki** (http://www.openwiki.org.ua) ukrán szabad enciklopédia, a **Майдан Вікі** (http://www.maidan.org.ua/wiki politikai enciklopédia, **Kijev web-encikopédiája** (http://wek.kiev.ua/uk), valamint néhány civil szervezet, politikai szerveződés, oktatási intézmény webhelye is. Ilyen a **ЗапоВікі** (http://wiki.ciit.zp.ua – a Zaporizzsjai terület oktatási intézményeinek webhelye, a **Родовід** (http://uk.rodovid.org családfa-projekt, a **Kirovohrádi Vinnicsenkó Állami Pedagógiai Intézet webhelye** (http://wiki.kspu.kr.ua).

Webes közösségek

A **Web 2.0** technológiák közösségszervező ereje azon mérhető le, mennyire aktívan vesznek részt a felhasználók egyes témák megvitatásában, mennyire aktívan kommentálják az eseményeket, kommunikálnak egymással, figyelik az érdeklődésüknek megfelelő újdonságok megjelenését. Ennek következményeként az internet-felhasználók különböző **webközösségeket** alkotnak. A közösség tagjai között nincs „hivatalos" kapcsolat, ezek az emberek nem feltétlenül ismerik egymást személyesen. A vita tárgyára irányuló szakmai vagy személyes érdeklődés köti össze őket. A webes közösség célja az információcsere, a közös tanulás, tapasztalatcsere, problémamegoldás.

A webes közösségeket különböző tulajdonságok alapján csoportosíthatjuk. A 4.76. ábra néhány lehetséges csoportosítást mutat be.

Minden webes közösségnek megvannak a maga jellegzetességei. A közösségek különböznek a tagok korában, foglalkozásában, a webes erőforrások használatában.



4.76. ábra. Webes közösségek

A **chates közösségek** a tetszőleges témában kommunikálni szerető fiatalok körében népszerű. A közösség összetétele gyorsan változik. A kommunikáció legtöbbször az aktuális események kommentálására korlátozódik. A közlemények jellemzően strukturálatlanok, tárolásuk rövid idejű. A felhasználónak nincs lehetősége kideríteni, hogy zajlott a megbeszélés az alatt az idő alatt, amíg nem volt jelen.

A **fórumok közössége** általában egy konkrét – általában szakmai – téma megvitatása céljából jön létre. A vita ideje alatt leginkább a téma felvetőjének és a hozzászólóknak a gondolataira reagálnak. A hozzászólások strukturáltak, hosszú ideig tárolódnak, áttekinthetőek. A közösségben működik az aktivitás értékelésének rendszere, a szerzők tájékozódhatnak a fórumozók új közleményeiről. A fórumnak van egy etikai szabályzata, amelynek betartását a moderátorok felügyelik. A fórumozók közössége csekély mértékben változik, jellemzően a témaváltásokhoz köthető.

A **blogok közössége** általában a blogger által felvetett, a blog nevében megnevezett téma megvitatására szerveződik. A kommunikáció során a közösség leginkább a blog szerzője által tett felvetésekre reagál. A vita legtöbbször lineáris szerkezetű. A hozzászólások hosszú ideig tárolódnak. A közösség tagjai tájékozódhatnak az új hozzászólásokról. A kommunikáció szabályait és a hozzáférési jogokat a blog szerzője határozza meg. Némely blogok trackback technológiával csatlakoznak egymáshoz. A blogszolgáltatók saját belső mechanizmusaik révén támogatják a blogok kapcsolódásait, ami elősegíti a blogos közösségek szerveződését. Az összekapcsolódó blogokat szokás blogoszférának nevezni.

A **wiki-közösségek** általában egy információs erőforrás minőségi feltöltése céljából – az úgynevezett tudásháló létrehozására – jön létre. A kommunikáció során tisztázzák a fogalmakat, kidolgozzák az enciklopédikus és tájékoztató cikkek stílusát, létrehozzák a forrásokra mutató hivatkozásokat. A wiki-oldalak gyakran oktatási, távoktatási céllal jönnek létre, ami segíti az oktatási közösségek kialakulását. A közösség anyagai jól szervezettek, hivatkozásokkal összekötöttek. A kommunikáció szabályait és a hozzáférési jogosultságokat a közösség határozza meg.

A **közösségi hálók** jellemzően ismeretségi alapon szerveződnek, céljuk az ismeretségi kör kiszélesítése. A közösséget az egyéni és szakmai érdeklődés tartja össze. A kommunikáció gyengén strukturált. A közösség tagjainak státusza és az adminisztráció összetételét a tagok aktivitása határozza meg.

Blogok. A blogok típusai

A **blogok** (webnaplók) lassanként a személyes weblapok alternatíváivá válnak. Ahogy a személyes weblapnak, a blognak is célja a személyes véleménynyilvánítás, de a blogok esetében a látogatók reakciói, a kapcsolattartás is fontos.

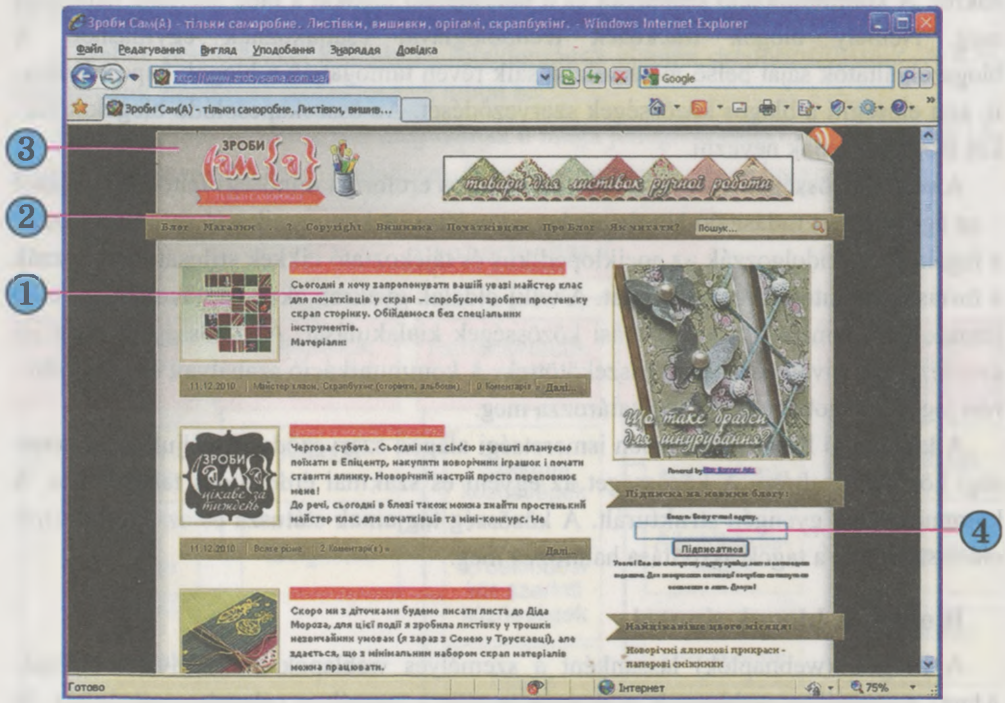
A blog egy közleményszalag, amelyekhez a blog olvasói hozzáfűzhetik a saját hozzászólásaikat. A szerzői közleményeit általában a weblap tetején találjuk, amit a látogatók hozzászólásai követnek (4.77. ábra). A hozzászólásokban multimédiás adatokat is el lehet helyezni. Minden hozzászólásnak egyedi URL-je van.

A blog jellemzői:

- a hozzászólások rövidek;
- az információtartalom megújítása rendszeres;
- a szerző és az olvasók közötti párbeszéd;
- hozzáférhetők az új témában tett hozzászólások;
- egy, az összes hozzászólást átható gondolat megléte, amire a blog címe utal.

A **létrehozás céljától** függően a blogok lehetnek:

- **személyesek,** amelyek gyakorlatilag a személyes napló webes változatai, az aktuális eseményeket és a szerző véleményét tükrözik;
- **tematikusak,** amelyek egy adott téma köré szerveződnek;
- **kereskedelmiek,** amelyek célja egy termék vagy szolgáltatás terjesztése, általában reklámot, termékismertetőket is tartalmaz;
- **közösségiek** – kollégák, szervezetek, szerveződések tagjai közötti kommunikáció bonyolítása céljából hozzák létre;
- **oktatásiak** – az oktatási folyamat tagjai közötti kommunikáció céljából hozzák létre;



1. Közlemények
2. Navigációs sáv
3. Cím
4. Feliratkozás

4.77. ábra. Blogoldal

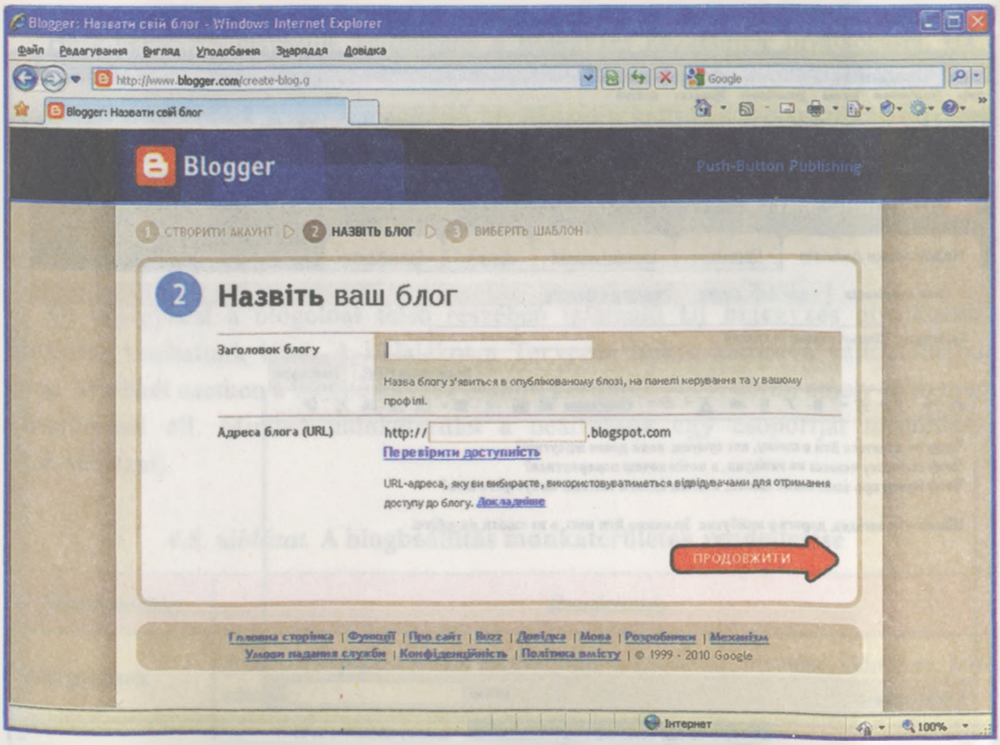
A **közlemények típusa** szerint megkülönbözetünk:

- **szöveges blogokat,** amelyeken többnyire szöveges hozzászólások jelennek meg;
- **fotoblogokat** – amelyek fotógyűjteményeket tartalmaznak;
- **zenei blogokat** – amelyek tartalma főleg hanganyag;
- **blogcastokat** – amelyeken a jellemző tartalom a blogger által elmondott hanganyagot tartalmazó hangfájl;
- **videoblogokat** – amelyek jellemző tartalma a videofájl;
- **Blogokat a Blog.net.ua** (http://blog.net.ua), (http://blogs.mail.ru), (http://blogger.com) és még sok más szolgáltató oldalain hozhatunk létre.

Blogok létrehozása. A bejegyzések publikálása

Vizsgáljuk meg a blog létrehozásának folyamatát a **blogger.com** szolgáltatás példáján. A szolgáltatást a **Google** fiókkal rendelkező felhasználók vehetik igénybe. A blog létrehozása a következőképpen történik:

1. Nyissuk mega http://blogger.com oldalt.
2. Jelentkezzünk be a **Google** felhasználói adatokkal.
3. Kattintsunk a **Létrehozás** linkre.
4. Adjuk meg a **blog címét** (4.78. ábra).

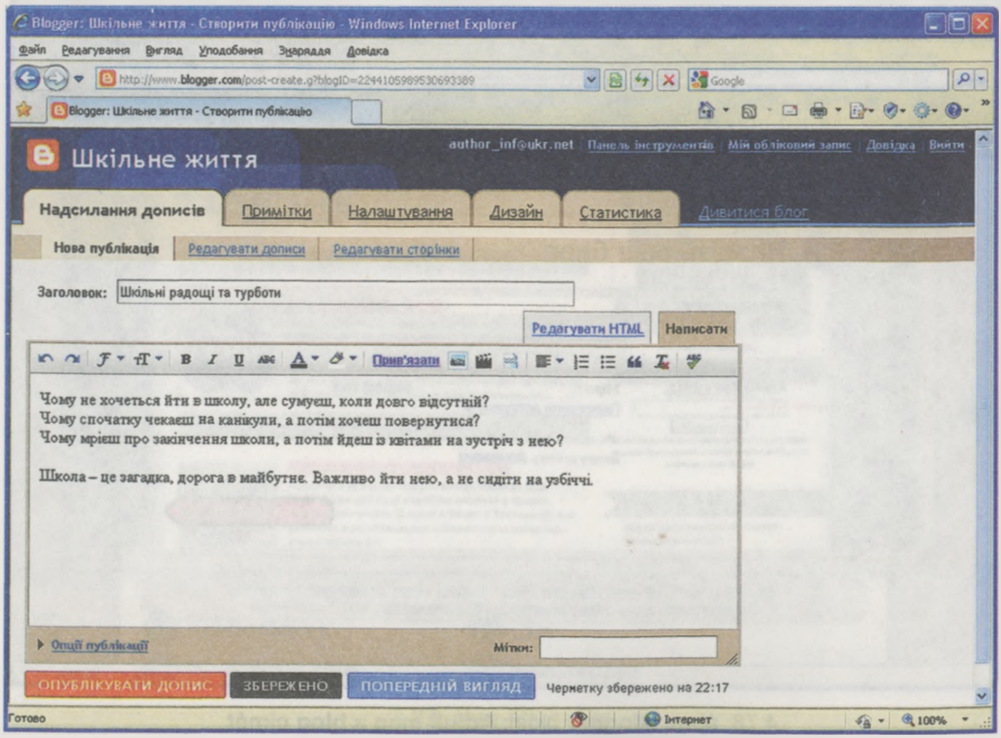


4.78. ábra. **Blogger** oldal: **Adjuk meg a blog címét**

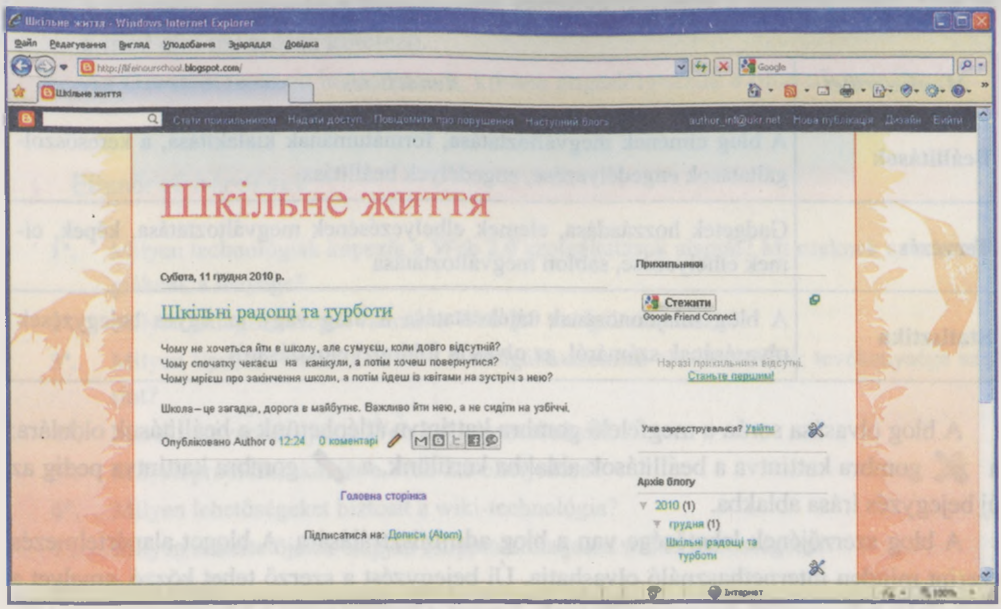
5. Adjuk meg a **blog URL-jét** az elérhetőség ellenőrzése céljából. Az URL http://*blognev*.blogspot.com, ahol a *blognev*-et a felhasználó választja ki.
6. Kattintsunk az **Elérhetőség ellenőrzése** hivatkozásra az URL egyediségének ellenőrzése céljából. Szükség esetén változtassuk meg a blognev-et.
7. Kattintsunk a **Tovább** linkre.
8. Válasszunk egyet a felajánlott sablonok közül és kattintsunk a **Tovább** linkre.
9. Válasszunk egyet a felajánlott lehetőségek közül:
   - **bejegyzések írása** – hozzáláthatunk az első blogbejegyzés létrehozásához. Ugyanez történik, ha a **Blogolás megkezdése** hivatkozásra kattintunk.
   - **blog személyre szabása** – megváltoztathatjuk a sablont, a tördelést, a színbeállításokat, betűtípusokat stb.
   - **speciális beállítások** – egyéni domaint hozhatunk létre a bloghoz vagy importálhatunk egy létező blogot.

Ha az első lehetőséget választjuk, akkor megnyílik a ***Blog neve*** – **Új bejegyzés ablak**, ahol a blog neve a felhasználó által a létrehozás alatt megadott név (4.79. ábra). Bejegyzés létrehozása céljából töltsük ki a bejegyzés címe mezőt, majd írjuk meg a bejegyzés szövegét. Az ablakban található eszközök segítségével a szöveget formázhatjuk is.

A munkaterület alatt találjuk a **Címkék** mezőt, ahová a bejegyzés tartalmára utaló kulcsszavakat írhatunk, amelyek alapján a keresőgépek megtalálhatják és indexelhetik a bejegyzést.



4.79. ábra. Blogbejegyzés létrehozása



4.80. ábra. A blog egy oldala

Ha a bejegyzés elkészült, kattintsunk a **Bejegyzés közzététele** hivatkozásra. Ha a **Mentés** linkre kattintunk, akkor a bejegyzés mentődik, de még nem kerül publikálásra. A szerző a későbbiek során a megfelelő hivatkozásra kattintva szerkesztheti, bővítheti azt. Ha az **Előnézet** linkre kattintunk, akkor megtekinthetjük a blogbejegyzés előnézeti képét (4.80. ábra).

Blog beállítása és adminisztrálása

Új bejegyzést a blogoldal felső részében található **Új bejegyzés** hivatkozásra kattintva hozhatunk létre. A külalakot a **Tervezés** linkre kattintva változtathatjuk meg. Mindkét esetben a blogbejegyzés oldalra kerülünk (4.79. ábra), amely több munkaterületből áll. Minden munkaterület a beállítások egy csoportját tartalmazza (4.8. táblázat).

4.8. táblázat. **A blogbeállítás munkaterületek rendeltetése**

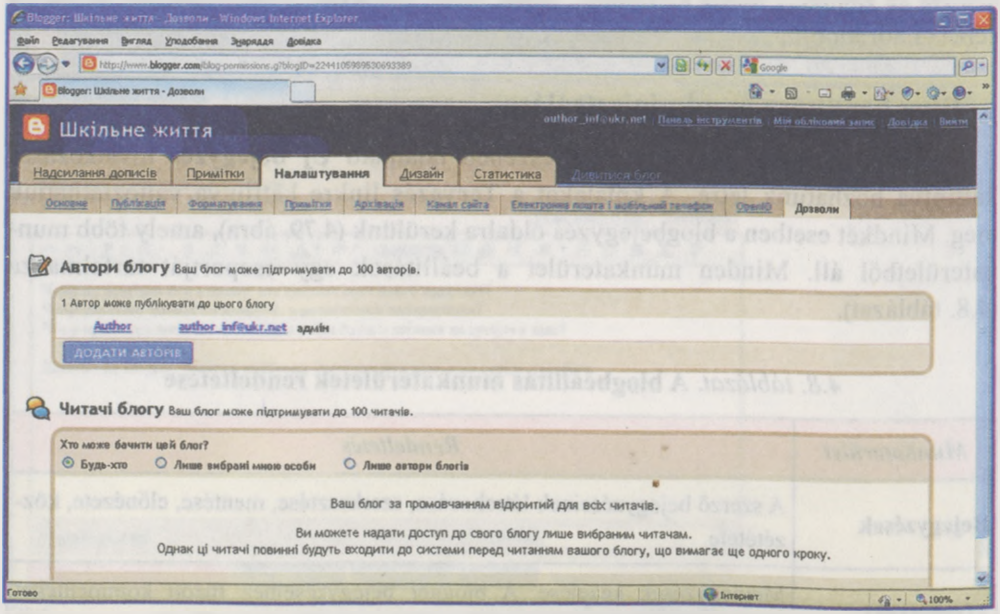
Munkaterület	*Rendeltetés*
Bejegyzések	A szerző bejegyzéseinek létrehozása, szerkesztése, mentése, előnézete, közzététele
Megjegyzés	Megjegyzések kezelése. A blogger bejegyzéseihez fűzött kommenteket megjelölhetjük spamként, részlegesen vagy teljesen törölhetjük

Munkaterület	*Rendeltetés*
Beállítások	A blog címének megváltoztatása, formátumának kialakítása, a keresőszolgáltatások engedélyezése, engedélyek beállítása
Tervezés	Gadgetek hozzáadása, elemek elhelyezésének megváltoztatása, képek, címek elhelyezése, sablon megváltoztatása
Statisztika	A blog tulajdonosának tájékoztatása a blog vagy az egyes bejegyzések olvasásának számáról, az olvasók földrajzi eloszlásáról

A blog olvasása során a megfelelő gombra kattintva átléphetünk a beállítások oldalára: a gombra kattintva a beállítások ablakba kerülünk, a gombra kattintva pedig az új bejegyzés írása ablakba.

A blog szerzőjének lehetősége van a blog adminisztrálására. A blogot alapértelmezés szerint minden internethasználó olvashatja. Új bejegyzést a szerző tehet közzé, amelyet a regisztrált felhasználók kommentálhatnak. Ezeket a beállításokat a következőképpen változtathatjuk meg:

1. Nyissuk meg a beállítások oldalt.
2. Kattintsunk a **Beállítások** ablakban a **Megjegyzés** fülre.
3. Állítsuk be, hogy a felhasználók csoportjai közül a **Kik írhatnak bejegyzést.**
4. Kattintsunk az **Engedélyek** fülre (4.81. ábra).



4.81. ábra. Az **Engedélyek** munkaterület

5. A **Szerzők hozzáadása** hivatkozásra kattintva bővítsük a szerzők körét. Név és e-mail cím megadása kötelező.
6. Az **Olvasók** részben beállíthatjuk, kiknek engedélyezzük a blog olvasását.

Ellenőrző kérdések

1°. Milyen technológiák képezik a **Web 2.0** szolgáltatások alapját? Mi ezeknek a technológiáknak a lényege?
2°. Milyen jellegzetességei vannak a **Web 2.0** szolgáltatásoknak?
3°. Milyen típusai vannak a **Web 2.0** szolgáltatásoknak a felhasználók tevékenysége szerint?
4*. Mivel magyarázhatjuk a **Web 2.0** népszerűségét?
5°. Milyen problémákkal jár a **Web 2.0** elterjedése?
6°. Milyen lehetőségeket biztosít a wiki-technológia?
7°. Milyen ismertetőjelek alapján csoportosíthatjuk a webes közösségeket?
8°. Mi a blog?
9*. Mi az oka a blogok népszerűségének? Miért tekinthetjük a blogot a személyes weblap alternatívájának?
10°. Hogyan csoportosíthatjuk a blogokat? Mik az egyes csoportok jellemzői?
11°. Milyen webhelyeken hozhatunk létre blogokat?
12°. Ismertessük a blog létrehozásának módját a **blogger.com** oldalon.
13°. Hogy hozunk létre új bejegyzést?
14°. Milyen beállításai vannak a blognak?
15°. Milyen engedélyeket adhat a felhasználóknak a blog adminisztrátora?

Végezzük el a feladatokat!

1*. Ismerkedjünk meg a **Wikipédia** (http://uk.wikipedia.org), a **Flickr** (http://www.flickr.com), a **Google Earth** (http://earth.google.com) oldalakkal! Határozzuk meg, melyik oldalra milyen technológia jellemző! Töltsük ki a táblázatot!

Technológia	*Wikipédia*	*Flickr*	*Google Earth*
Web mint platform			
Mash-up			
Kollektív agy			
Folkszonómia			
Web-hírügynökség			
Közösségi háló			

2*. Állapítsuk meg, milyen összefüggés áll fenn a **Web 2.0** szolgáltatások és az emberi tevékenységek között!

	Tevékenységek
1	Adatkeresés
2	Közös tartalomfejlesztés
3	Dokumentumok tárolása
4	Kollektív hivatkozáslista kezelése
5	Helymeghatározás
6	Kommunikáció
7	Közös projektek

	Web 2.0 szolgáltatások
A	Közösségi értékelés
B	Projektmenedzserek
C	Közösségi keresőszolgáltatások
D	Közösségi médiatárak
E	Wiki-projektek
F	Geoszolgáltatások
G	Blogok

3*. A 4.7. táblázat adatainak alkalmazásával hozzuk létre a **Web 2.0** szolgáltatások hierarchia-diagramját!

4. Hasonlítsuk össze a **Családfa** (http://uk.rodovid.org) és a **Kijev web-enciklopédiája** (http://wek.kiev.ua/uk) wiki-projekteket! Töltsük ki a 4.9. táblázatot!

4.9. táblázat

Kérdés	*1. projekt*	*2. projekt*
Milyen anyagok vannak bemutatva?		
Milyen eszközökkel van a téma bemutatva?		
Milyen eszközök szolgálnak az oldalak közötti navigációra?		
Milyen segítséget kapunk a webhely használatához?		
Milyen eszközökkel szerkeszthetjük a szócikkeket?		
Kell-e regisztrálni az oldal szerkesztőinek?		
Hogy tekinthető meg a kezelés története?		
Milyen eszközökkel hozhatunk létre új szócikket?		
Hogy van megszervezve a szócikkek megvitatása?		
Szükséges-e regisztrálni a megvitatás résztvevőinek?		
Mikor történt a webhely utolsó frissítése?		

5°. Hasonlítsuk össze az **ЕкоАгроВікі** (http://agrowiki.nubip.edu.ua) és **Фармацевтична енциклопедія** (http://www.pharmencyclopedia.com.ua) weboldalakat! Töltsük ki a 4.9 táblázatot!

6*. Töltsük ki webes közösségeket összehasonlító táblázatot.

Tulajdonság	*Közösség*				
	Chat	*Fórum*	*Wiki*	*Blog*	*Közösségi háló*
Létrehozás célja					
Az anyagok tárolásának ideje					
Az anyagok strukturáltsága					
A tagok aktivitásának értékelése					
Ki állítja fel a kommunikáció szabályait					
Hozzáférés korlátozása					
Tájékozódási lehetőség a változásokról					
Keresési lehetőségek					
A közösség minden tagja szerkesztheti az információs anyagokat					

7°. Hozzuk létre a blogok osztályozásának hierarchia-diagramját!

8°. Vizsgáljuk meg a (http://blogosphere.com.ua) webhelyet! Értékeljük a blog arculatát, szerkezetét, a bejegyzéseket, a kommenteket! Határozzuk meg, milyen lehetőségeket használnánk szívesen a saját blogunkban!

9°. Hozzunk létre saját blogot a http://blogger.com webhelyen! A blog témája legyen az otthoni állattartás! Alkalmazzuk a *Vizjel* sablont! Helyezzünk el bejegyzést azokról az állatokról, amelyeket szeretnénk otthonunkban tartani! Közöljük a blog címét a tanárunkkal!

10°. Írjunk a blogra egy bejegyzést a barátunk kedvenc állatáról!

11°. Változtassuk meg a blog beállításait: a címben ismertessük a blog célját, és helyezzük ott el egy háziállat fotóját a (**Тема 4\Завдання 4.10\Тварини**) mappából! Adjunk egy gadgetet a statisztikához! Változtassuk meg a háttérszínt és a gadgeteket helyezzük el a közleményektől balra!

12°. Változtassuk meg a hozzáférést a blogunkhoz! Kommentár elhelyezését engedélyezzük mindenkinek, a névtelen felhasználóknak is, engedélyezzük bejegyzés írását a tanárunknak. Küldjünk tanárunknak erre felkérést e-mail-ben!

13°. Olvassuk el az osztálytársunk blogját, és írjunk abba egy kommentet!

19. számú gyakorlati munka **Saját blog létrehozása és működtetése**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

Hozzunk létre blogot a http://blogger.com webhelyen!:

1. A blog témája legyen a leendő szakmánk és ezt fejezze ki a blog címe is!
2. Alkalmazzunk egy tetszőleges sablont!
3. Helyezzünk el a blogon egy bejegyzést a leendő szakmánkról!
4. Változtassuk meg a blog beállításait:
   a) a címben írjuk le a blog rendeltetését;
   b) helyezzük el a blogon a szakma attribútumait a **Тема 4\Практична 19\Професії;** mappából;
   c) helyezzünk el a cím alatt egy **Kereső** gadgetet;
   d) változtassuk meg a blog szerkezetét. Helyezzük el a gadgeteket két oszlopba a bejegyzésektől jobbra, illetve balra.
5. Írjunk a blogra egy bejegyzést szüleink, ismerőseink szakmájáról!
6. Változtassuk meg a felhasználók jogosultságait:
   a) a blogra csak a **Google** felhasználók helyezhetnek el bejegyzéseket;
   b) a blogot csak az általunk kiválasztott felhasználók olvashassák. Kérjük fel erre néhány osztálytársunkat;
   c) kérjük fel a tanárunkat, hogy legyen a blog egyik szerzője.

4.11. Dokumentumok közös kezelése

1. Milyen **Web 2.0** szolgáltatásokat ismerünk? Ezek közül melyek szolgálnak a felhasználók közötti kapcsolattartás megszervezésére?
2. Mire szolgálnak az online office-ok? Milyen lehetőségeket nyújtanak?
3. Milyen **Google** szolgáltatásokat ismerünk? Jellemezzük ezeket!
4. Mi a regisztrációs bejegyzés? Milyen szolgáltatások alkalmazzák?
5. Milyen szabályokhoz kell tartsuk magunkat a chat-en és a fórumon?

A kollektív együttműködés eszközei

Az internet elterjedése az együttműködő csoport fogalmának kardinális változását hozta magával. Különböző országokban élő, más-más szervezetekben tevékenykedő emberek dolgozhatnak közös projekteken, hozhatnak létre közös dokumentumokat. Ma már nem számít különlegesnek az olyan alkalmazott, aki nem jelenik meg a munkahelyén, hanem otthon végzi el a munkáját. Vannak, akik utazás közben szemináriumokon, konferenciákon vesznek részt.

Ma már nem ritkaság, hogy különböző irodákban, épületekben dolgozó munkatársak közös projekten dolgoznak. Ilyen esetekben a projekt résztvevői *virtuális irodát* hoznak létre, ahol a közös használatú dokumentumokat tartják. Az ilyen szervezet létrehozása során az egyik legnagyobb probléma a közös használatba kerülő dokumentumok elérésének és a projekt résztvevői közötti kommunikációnak megszervezése. Erre a célra a már megismert eszközök – elektronikus posta, chat, üzenetküldő szolgáltatások, **Web 2.0** szolgáltatások – használhatók.

Léteznek speciálisan a kollektív munka megszervezésére szolgáló szoftvereszközök, például a **Google Wave,** a **12Sprints,** a **Sharepoint,** a **Novell Pulse,** az **IPI.Manager,** a **Feng Office,** a **Microsoft Office Groove 2007.** Ezeknek az eszközöknek még nincs közismertségnek örvendő neve, leginkább a kollektív együttműködés eszközeiként emlegetik ezeket. Ezek az eszközök a világhálón és a helyi hálózatban egyaránt használhatók.

A kollektív együttműködés eszközeinek rendeltetése a következő:

- a csoport közös erőforrásai elérhetőségének biztosítása;
- a csoport tagjai közötti kommunikáció biztosítása;
- a munka koordinálása és dokumentálása.

A közös munka megszervezése a Microsoft Office Groove 2007 segítéségével

A kollektív munka és a virtuális iroda megszervezésének egyik eszköze a **Microsoft Office Groove 2007,** ami a **Microsoft Office 2007** része.

Segítségével online találkozókat szervezhetünk meg, közös projektek megbeszélését bonyolíthatjuk le, fájlokat továbbíthatunk közvetlenül az egyik számítógépről a másikra, dokumentumokat tarthatunk közös tárhelyen, amelyek automatikusan frissülnek a csoport tagjainak munkája során. A program segítségével tájékozódhatunk a résztvevők online jelenlétéről, tájékozódhatunk a résztvevőkről, üzeneteket küldhetünk nekik. A program azonnali üzenetküldést, csevegést és közös dokumentumkezelést is lehetővé tesz. A résztvevők együttműködése az úgynevezett munkaterületen történik, a közös dokumentumok egy zónában koncentrálódnak. A program működéséhez elengedhetetlen az internet.

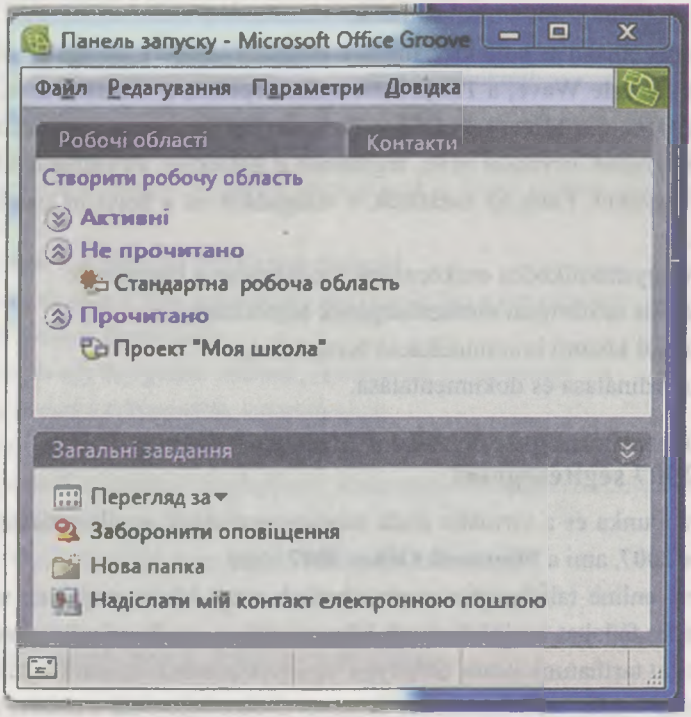
A programot a ***Start*** ⇒ ***Minden program*** ⇒ ***Microsoft Office*** ⇒ ***Microsoft Office Groove 2007*** paranccsal történik.

Ha a programnakt először indítjuk, elindul a **Fiókkonfiguráció varázsló,** amelynek ablakában a következő lépéseket kell megtenni:

1. Ki kell választani, hogy új **Groove** fiókot kívánunk-e létrehozni, vagy egy már létezőt kívánunk használni!
2. Meg kell válaszolni azt a kérdést, hogy rendelkezünk-e **Groove** konfigurációs kóddal! Amennyiben nem, válasszuk a **Nem rendelkezem Groove fiók konfigurálására szolgáló kóddal** beállítást.
3. Adjuk meg a felhasználó nevét, e-mail címét és jelszavát.
4. Adjuk meg a hozzáférés módját! Ha lehetséges, engedélyezzük a hozzáférést másoknak, ez megkönnyíti a ***Groove*** használatát.

Ha nem először indítottuk a programot, akkor a program indítása után meg kell adni a felhasználó fiókjának jelszavát.

Belépés után a megnyílik a 4.82. ábrán látható ablak, amelyben a **Munkaterületek, Kapcsolatok** és **Szokásos feladatok** területeket találjuk.



4.82. ábra. A **Microsoft Office Groove 2007** indulóablaka

A ***munkaterület*** a felhasználók egyike által létrehozott hely, amelyet a partnerek a közös dokumentumok elhelyezésére és a munka során felmerülő kérdések megvitatására használnak. Sokban hasonlít egy chat-szobára. A **Groove 2007**-ben háromféle munkaterület hozható létre, amelyek az elérhető dokumentumok típusaiban különböznek egymástól:

- **Normál** – a tagok virtuális kommunikációs közege, amely közös használatú fájlok elhelyezését is lehetővé teszi;
- **Fájlmegosztás** – mappa a tagok egyikének számítógépén, amelyet a csoport közösen használ;
- **Sablon** – olyan munkaterület, amelyet a csoport tagjai a következő céllal használnak:
  - *Rajztábla* – közös rajzterület;
  - *Jegyzettömb* – szövegszerkesztő, amelyet a csoport minden tagja használhat;

- *Naptár* – fontos dátumok megjelölése, ütemtervek készítése;
- *Vitafórum* – részletes megbeszélés a munkaterületek többi tagjával;
- *Képek* – grafikus elemek és képek megosztása;
- *Értekezletek* – napirendek és teendők kiosztása, jegyzőkönyvek rögzítése.

A munkaterületet létrehozó tag a munkaterület *Vezetője*. Ő hívja meg vagy zárja ki a csoport tagjait, csatol vagy töröl munkaeszközöket, változtatja meg a tagok szerepkörét, irányítja a munkaterület működését. A többi felhasználó vagy *Résztvevő* vagy *Vendég* lehet. A résztvevőnek joga van új munkaeszközöket csatolni és új felhasználókat meghívni. A vendég részt vehet a vitákban és dolgozhat a közös dokumentumokon.

A munkaterület létrehozása a következőképpen történik:

1. A **Kezdőpanelen** kattintsunk az **Új munkaterület** linkre.
2. Válasszuk ki a munkaterület típusát (*Normál, Fájlmegosztás, Sablon*).
3. Adjunk nevet a munkaterületnek.
4. Kattintsunk az **OK** gombra.

A létrehozott munkaterületet a nevére duplán kattintva nyithatjuk meg. A 4.83. ábrán egy normál munkaterületet láthatunk, ahol a felhasználóknak két eszköz áll rendelkezésre: a **Fájl** és a **Vitafórum**, amelyek ikonjait az ablak alján találjuk (10, 9).

Az ablak felső részében találjuk a menüt (2), az eszköztárat (3), a jobb oldalán a munkaterület tárolási szerkezetét (4), a közepén pedig a mappák taralmát (5). A munkaterület eszközeinek segítségével új fájlokat helyezhetnek el a közös területen.

Az ablak jobb oldalán láthatjuk a csoport tagjait, akik három kategóriához tartozhatnak:

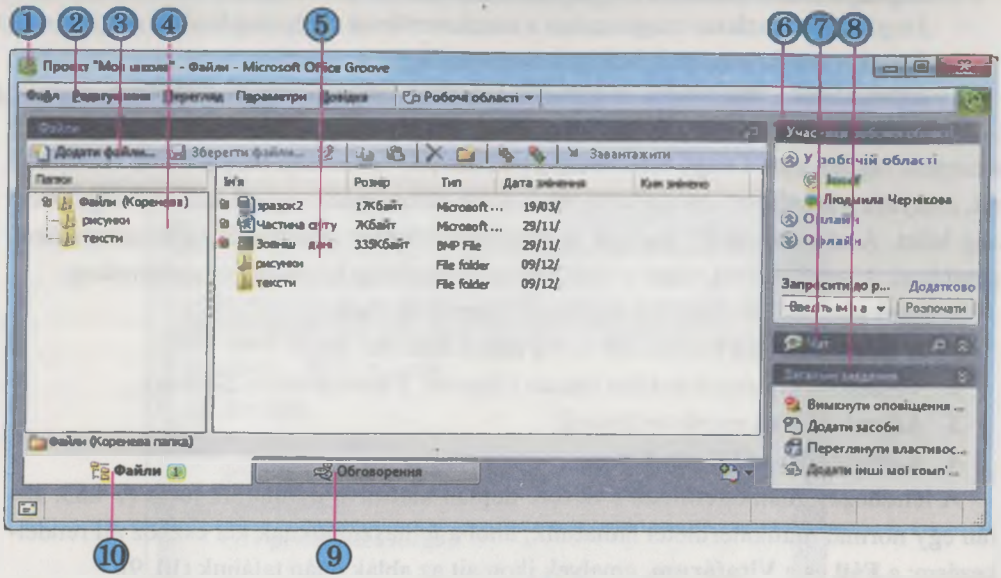
- *A munkaterületen aktívak* – akik éppen dolgoznak a munkaterületen;
- *Online* – akik az interneten tevékenykednek, de a munkaterületen aktuálisan nem aktívak;
- *Offline* – azok a csoporttagok, akik jelenleg nem aktívak az interneten.

A **Chat** (7) panel a szöveges kommunikációt teszi lehetővé, az **Általános feladatok** (8) pedig azokat műveleteket tartalmazza, amelyek a csoport számára elérhetőek.

A munkaterület létrehozása után meghívókat kell szétküldeni a csoport megszervezése céljából. Tag csak az lehet, akinek a számítógépén telepítve van a **Groove**, regisztrálva van és természetesen van internet-kapcsolata.

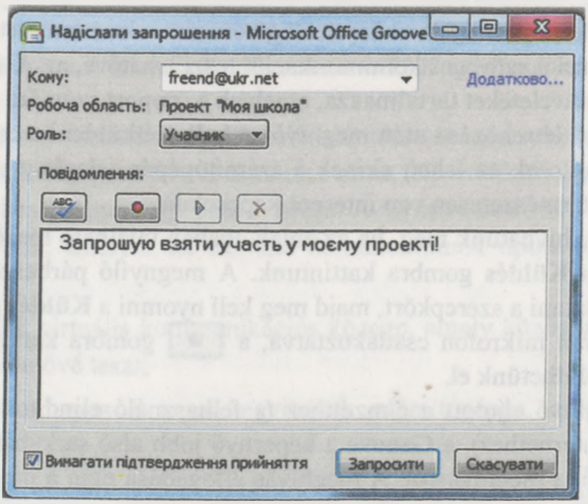
Új tagokat úgy hívhatunk meg, ha az ablak aljában található mezőbe beírjuk annak e-mail-címét és a **Küldés** gombra kattintunk. A megnyíló párbeszédablakban (4.84. ábra) ki kell választani a szerepkört, majd meg kell nyomni a **Küldés** gombot. Ha a számítógépünkhöz van mikrofon csatlakoztatva, a [●] gombra kattintva hangüzenetet vehetünk fel és küldhetünk el.

Ahogy a meghívó eljutott a címzetthez (a felhasználó elindította a **Groove**-ot és csatlakozott az internethez), a **Groove** a képernyő jobb alsó sarkában megnyíló ablakban tájékoztatja őt a meghívásról. A meghívás elfogadása után a program a küldőt hasonló módon tájékoztatja. Az új felhasználó számítógépén megnyílik a **Groove** ablaka az új munkaterület tartalmával. A meghívást elfogadó felhasználók nevét a **Munkacsoport tagjai** listán találjuk.



1. A munkaterület neve
2. A munkaterület menüje
3. Eszköztárak
4. **Mappa** ablak
5. A mappák tartalma ablak
6. A **munkacsoport tagjai** panel
7. **Chat** panel
8. **Általános feladatok** panel
9. **Vitafórum** ikonja
10. **Fájlok** ikonja

4.83. ábra. A normál munkaterület ablaka



4.84. ábra. A **Meghívó küldése** ablak

A továbbiakban a munkacsoport tagjai szöveges üzeneteket válthatnak, közös fájlokat hozhatnak létre és szerkeszthetnek, megváltoztathatják a munkaterület felületét. A változások automatikusan mentődnek a munkaterületen és azokat a tagok is megkapják. Ez a folyamat a **Groove** *mappák szinkronizálása.* Ha a csoport valamely tagjának nincs online, a változásokat azonnal megkapja, amint online lesz.

Vizsgáljuk meg tüzetesebben a normál munkaterületen dolgozó csoport munkáját.

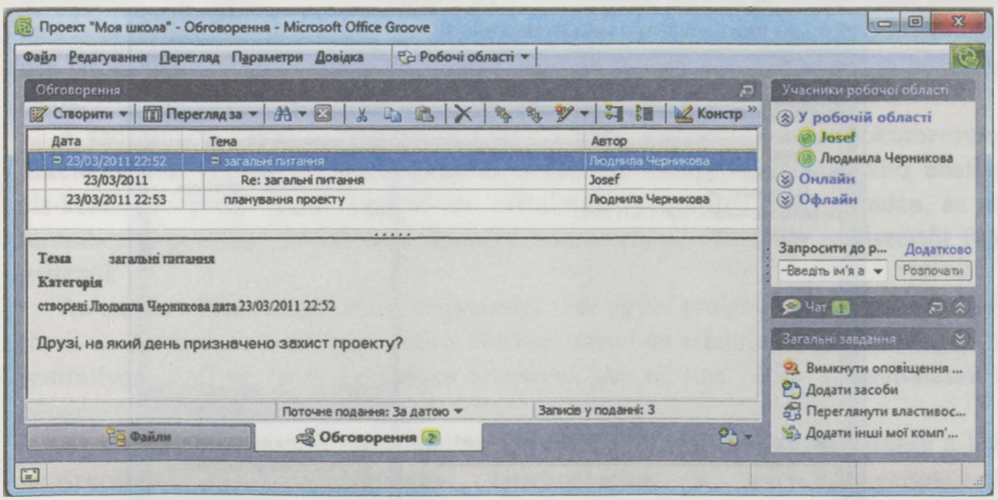
A **Fájlok** fül eszközeivel mappákat hozhatunk létre, objektumokat másolhatunk, törölhetünk és helyezhetünk át a munkaterületen. Ehhez csak a csoport vezetőjének van joga.

A többi felhasználónak csak dokumentumok elhelyezésére van joga. Ezt a következőképpen teheti meg:

1. Kattintsunk a **Fájl** fülre.
2. Nyissuk meg a szükséges mappát.
3. Kattintsunk a **Fájl hozzáadása** eszközre.
4. A megnyíló párbeszédablakban válasszuk ki a helyi gép egy fájlját.
5. Kattintsunk a **Megnyitás** gombra.

A fájl megjelenik a munkaterületen. Automatikusan megtörténik az adatok szinkronizálása. A tagok megnyithatják és szerkeszthetik a fájlt. Szerkesztés után a megnyíló párbeszédablakban jóvá kell hagyni a változtatások mentését, és amennyiben szükséges, meg kell adni az új fájl nevét. Ha egyidejűleg több tag is dolgozik a fájlon, a változások eltérő sorozatszámú verziószámú változatokba lesznek mentve.

A fájlnevek mellett látható jel arról informálja a felhasználót, hogy még nem tekintette meg a dokumentum legújabb változatát.



4.85. ábra. A **Vitafórum** ablak

A normál munkaterületen a közös munka másik eszköze a **Vitafórum** (4.85. ábra). A panel eszközei lehetővé teszik a hozzászólások létrehozását, törlését és rendezését. Az eszköz használata, a hozzászólások kezelése ugyanolyan, mint a fórumokon.

Az új téma létrehozása a következőképpen történik:

1. Nyissuk meg a **Vitafórum** fület.
2. Kattintsunk a **Létrehozás** gombra, majd válasszuk a **Téma** parancsot.
3. Írjuk be a téma nevét és a hozzászólás szövegét.
4. Kattintsunk a **Mentés** gombra.

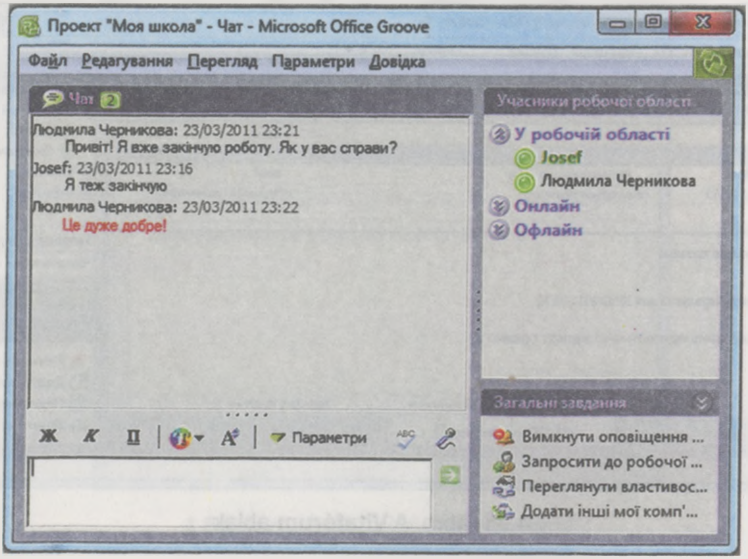
Választ a következőképpen küldhetünk:

1. Nyissuk meg a hozzászólást.
2. Kattintsunk a **Létrehozás** gombra, majd a **Válasz** parancsra.
3. Írjuk be a válasz szövegét.
4. Kattintsunk a **Mentés** gombra.

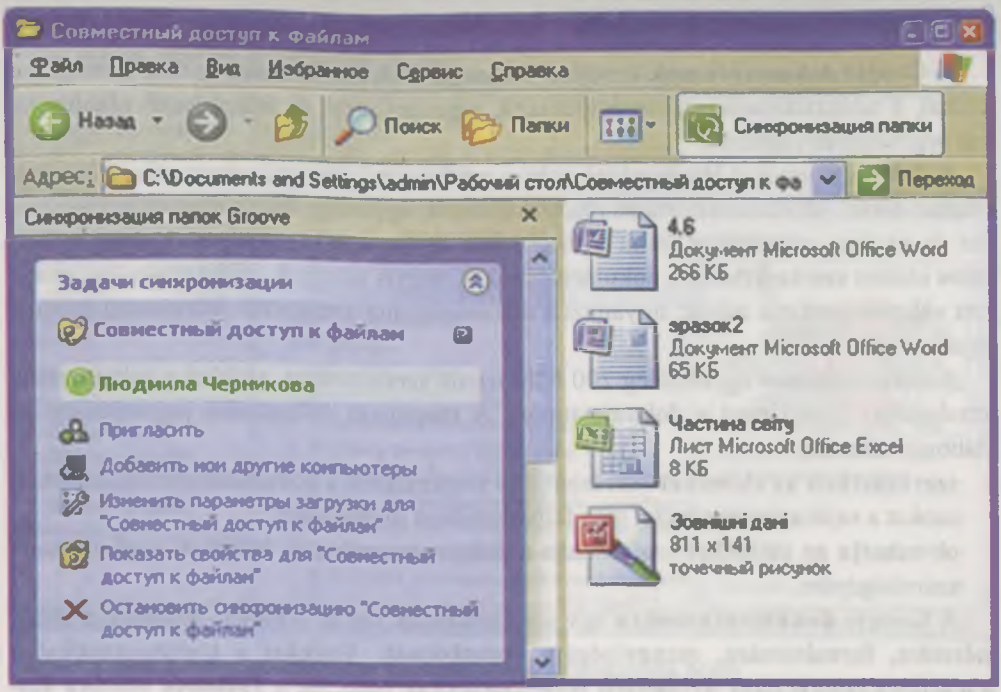
A csoport tagjai cseveghetnek is egymással. E célból nyissuk meg a munkaterület **Chat** ablakát. A **Kibontás** gombra kattintva egy külön chatablakot nyithatunk (4.86. ábra). A chat használatának szabályait már ismerjük.

A munkaterületen az **Eszköz hozzáadása** gombra kattintva új eszközöket (**Kép, Naptár, Rajzeszközök**) helyezhetünk el.

Ha a csoport tagjának számítógépén *Fájlmegosztás* munkaterületet hozunk létre, akkor annak kezelése az **Intéző** segítségével történik. A **Groove** a munkaterületen elhelyezi az **Intéző** ablakát és a **Mappák szinkronizálása** gombot. Ez utóbbira kattintva megnyílik a **Feladatok szinkronizálása** panel (4.87. ábra). A panel segítségével megnézhetjük, kik dolgoznak a munkaterületen, új felhasználókat hívhatunk meg, új számítógépeket adhatunk a csoporthoz, amelyeket szinkronizálhatunk a munkaterülettel.



4.86. ábra. A kibontott **Chatablak**



4.87. ábra. A **Feladatok szinkronizálása** panel

Az online dokumentumcsere eszközei

A csoport dokumentumainak közös kezelését az *Online Office* néven ismert on-line szolgáltatás segítségével is végrehajthatjuk. A technológia minden felhasználónak kényelmes hozzáférést biztosít a közös dokumentumokhoz, amelyek nem egy helyi számítógépen tárolódnak, hanem egy hálózati gépen, ezért azok bármikor és bárhonnan hozzáférhetők.

A szolgáltatás minden, a hagyományos office programoktól elvárt dokumentumkezelő eszközt biztosít, szövegszerkesztő, táblázatkezelő, prezentációkészítő, adatbázis-kezelő, szervező- és naptárszoftvert tartalmaz. Az on-line office minden, az internethez kapcsolódó felhasználó számára hozzáférhető, operációs rendszertől függetlenül.

Az online office szolgáltatásai ingyenesek, bár egyes szolgáltatások lehetnek fizetősek is. A **Microsoft Office Groove 2007**-tel szemben a felhasználó számítógépére semmilyen szoftvert nem szükséges telepíteni. Az on-line office szolgáltatásait a böngésző segítségével használhatjuk.

Az ingyenes online office szolgáltatások közül a **Google dokumentumok** a legnépszerűbbek. Dokumentumainkat létrehozhatjuk a **www.docs.google.com** webhelyen, de feltölthetjük a helyi számítógépről is. A dokumentumokat a **Google** online dokumentumtárába mentve, bármely internet-felhasználó hozzájuthat, akinek ezt engedélyezzük.

Google dokumentumok közös kezelése

A **Google dokumentumok** kezelését a **Google Dokumentumok** on-line szövegszerkesztő, a táblázatkezelő, prezentációkészítő, rajzszerkesztő és űrlapkészítő alkalmazások segítségével végezzük.

A dokumentumokat létrehozásuk után a rendszer néhány másodpercenként automatikusan menti, minden változtatás rögzítésre kerül, ugyanúgy használhatjuk a visszavonás és az újra műveleteket, mint a megszokott szövegszerkesztőkben. A felhasználók valós időben szerkeszthetik a dokumentumokat, vagyis az egyik felhasználó által elvégzett változtatásokat a másik, ugyanazt a dokumentumot szerkesztő felhasználó azonnal látja.

A dokumentumot egyidejűleg 200 felhasználó szerkesztheti, akikkel a dokumentum létrehozója osztja meg a dokumentumot. A meghívott felhasználó jogosultságai az alábbiak lehetnek:

- **szerkesztheti az elemeket** –olvashatja és szerkesztheti a dokumentumokat, letöltheti azokat a saját számítógépére, más felhasználókat hívhat meg;
- **olvashatja az elemeket** – olvashatja a dokumentumokat és letöltheti azokat a saját számítógépére.

A **Google dokumentumokra** egy sor korlátozás van érvényben, amelyek a fájlok méretére, formátumára, mennyiségére vonatkoznak. Ezekkel a korlátozásokkal a **www.docs.google.com** webhelyen ismerkedhetünk meg, ha a **Feltöltés** gombra kattintunk.

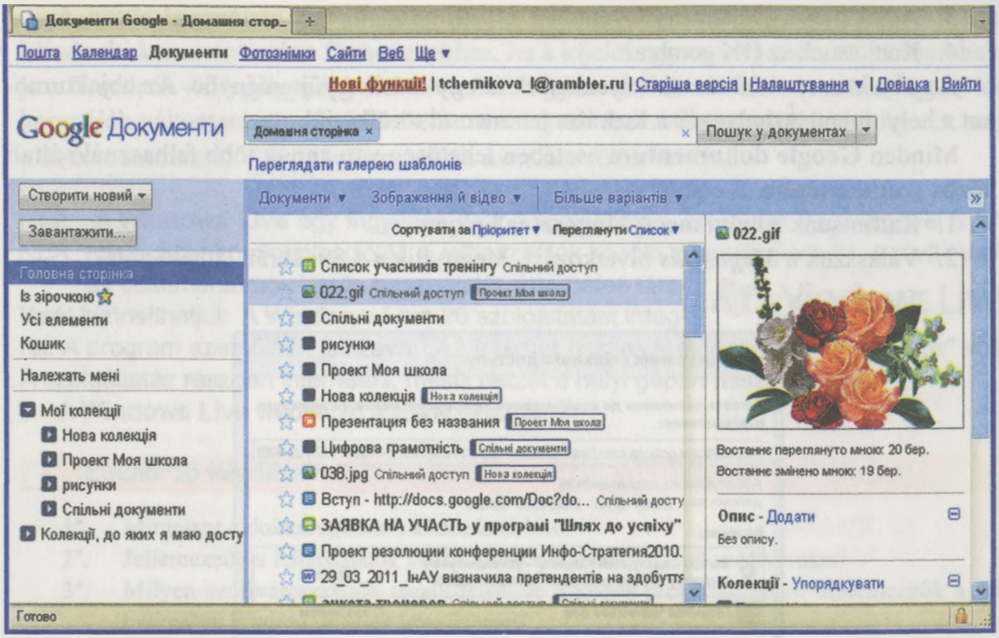
A **Google dokumentumokat** a **Google** fiókkal rendelkezők használhatják. A fiók létrehozásának módjával fentebb már megismerkedtünk.

A **Google** dokumentumokhoz a következőképpen férhetünk hozzá:

1. Nyissuk meg a **Google webhelyet** (www.google.com.ua).
2. Kattintsunk a képernyő jobb felső sarkában található **Belépés** gombra.
3. Adjuk meg az azonosítónkat és a jelszavunkat.
4. Kattintsunk a **Továbbiak** gombra.
5. Kattintsunk a **Dokumentumok** linkre. Megnyílik a **Google** dokmentumok főoldala.

A **Google dokumentumok** főoldala (4.88. ábra) három részre van felosztva. Az ablak bal oldala tartalmazza a navigációs elemeket és a megosztott dokumentumok listáját. A navigációs sáv felett találjuk az **Új létrehozása** gombot (erre kattintva a beépített szoftvereszközök segítségével hozhatunk létre új dokumentumot) és a **Feltöltés** gombot (amely a helyi gépen tárolt dokumentumok feltöltésére és konvertálására ad lehetőséget). Az ablak középső részét az aktuális tároló tartalomjegyzéke foglalja el. E lista felett találjuk meg a szűrő és rendező eszközöket, valamint a **Dokumentumok, Képek és videók** és a **További beállítások** gombokat. Jobboldalt a kiválasztott elem tulajdonságait, valamint az azok megváltoztatására szolgáló eszközöket találjuk.

Belépés után a **Főoldalra** kerülünk, amely a tárolón található összes elem rendezett listáját tartalmazza. Ha jobb gombbal kattintunk az elem nevére, a megnyíló helyi menü segítségével törölhetjük, átnevezhetjük az elemet, beállíthatjuk a hozzáférési jogosultságokat.



4.88. ábra. A **Google dokumentumok** ablak

Új **Google dokumentumot** a következőképpen hozunk létre:

1. Kattintsunk az **Új létrehozása** gombra.
2. Válasszunk dokumentumtípust: *dokumentum, táblázat, prezentáció* stb.
3. Hozzuk létre a tartalmat.

A létrehozott fájlt menteni nem kell, amiről a **Mentve** feliratú gombra tekintve is meggyőződhetünk.

A helyi számítógép egy fájlja feltöltésének módja:

1. Kattintsunk a **Feltöltés** gombra.
2. Kattintsunk a **Fájlok** linkre.
3. Válasszuk ki a feltöltendő fájlt.
4. Szükség esetén kattintsunk a **További fájlok feltöltése** gombra.
5. Állítsuk be a konverzió típusát.
6. Állítsuk be, melyik gyűjteményben szeretnénk elhelyezni a fájlt.
7. Állítsuk be a felhasználói jogosultságokat.
8. Kattintsunk a **Feltöltés** gombra.
9. Szükség esetén kattintsunk a **További dokumentumok feltöltése** linkre.

Bizonyos idő elteltével a fájl felkerül a feltöltött dokumentumok listájára.

A felhasználó a dokumentumok rendezett tárolása céljából gyűjteményekbe sorolhatja azokat. Új gyűjteményt így hozhatunk létre:

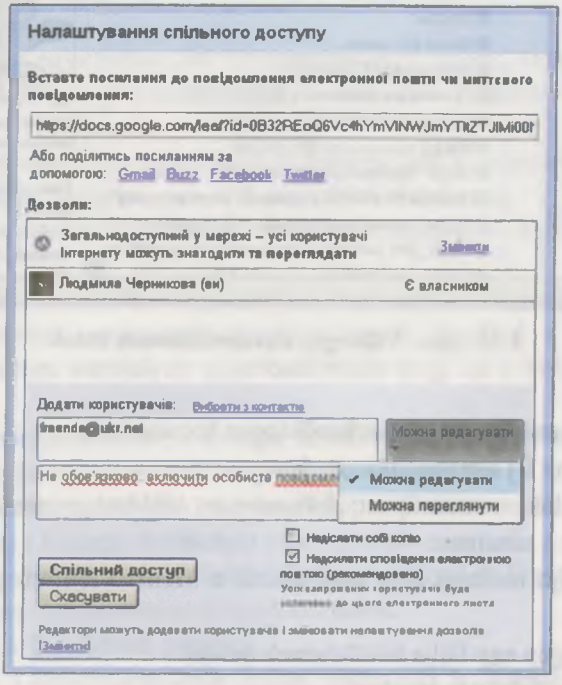
1. Kattintsunk az **Új létrehozása** gombra.
2. Válasszuk ki a *Gyűjtemény* objektumot.

3. Adjuk meg a gyűjtemény nevét.
4. Kattintsunk az **OK** gombra.

Az objektumot áthúzással helyezhetjük át egy másik gyűjteménybe. Az objektumokat a helyi menü **Áthelyezés a kukába** paranccsal törölhetjük.

Minden **Google dokumentum** esetében lehetőség van annak több felhasználó általi közös szerkesztésére. E célból:

1. Kattintsunk a helyi menü **Megosztás** linkjére.
2. Válasszuk a ***Megosztás*** hivatkozást. Megnyílik a 4.89. ábrán látható ablak.



4.89. ábra. A **Megosztás beállítása** párbeszédablak

3. A **Kinek van hozzáférése** területen láthatjuk azoknak a felhasználóknak a listáját, akiknek már van hozzáférése a fájlhoz.
4. A **Módosítás** hivatkozásra kattintva beállíthatjuk, hogy a fájl:
   1) *Nyilvános legyen az interneten; a link birtokában bárki hozzáférhessen; maradjon privát használatban.*
   2) Szükség esetén szabályozzuk a szerkesztési hozzáféréseket.
   3) Kattintsunk a **Mentés** gombra.
5. Írjuk be a **Személyek hozzáadása** mezőbe az új felhasználók e-mail-címeit.
6. Kattintsunk a **Kész** gombra.

A hozzáférést úgy is engedélyezhetjük, hogy tudatjuk velük a fájlra mutató linket (az ablak felső részében láthatjuk).

A közös használatú dokumentumokat a felhasználók valós időben szerkeszthetik. Mindenki kommentárokat fűzhet ezekhez, ha a kijelöl egy szövegrészt, majd végrehajtja a ***Beszúrás – Megjegyzés*** parancsot. Szükség esetén vissza lehet vonni az egyes felhasználók változtatásait. A dokumentum szerkesztése során láthatjuk, ki dolgozik még a dokumentummal. Ezt az információt a menüsor fölött találjuk.

A **Windows Live** egy ingyenes online szolgáltatás, amelynek segítségével a felhasználók kommunikációját, a felhasználók közötti információcserét, a multimédiás dokumentumok közös használatát, internetes tárolását biztosíthatjuk. A **Windows Live** 20 szolgáltatást integrál. A program szerzőinek jelszava: „Az internet összes lehetőségét ide gyűjtöttük" volt. A szolgáltatás részben internetes, másik részét a helyi gépen használhatjuk.

Windows Live

A **Windows Live** webhelye a **www.live.com.**

Ellenőrző kérdések

1°. Mit jelent a dokumentumok közös kezelése?
2°. Jellemezzük a felhasználók közös munkáját lehetővé tevő szoftvereket!
3°. Milyen szoftvereszközök teszik lehetővé a közös projektkezelést? Jellemezzük a felhasználók kommunikációs lehetőségeit!
4°. Mi a virtuális iroda? Mi a rendeltetése?
5°. Milyen szolgáltatásokat biztosít a **Microsoft Office Groove 2007**?
6°. Ismertessük a **Google dokumentumok** szolgáltatást!
7°. Miben különbözik a **Google dokumentumok** használata a **Microsoft Office Groove 2007**-től?
8°. Mi a **Microsoft Office Groove 2007** munkaterület? Milyen lehetőségeket biztosít?
9°. Milyen lehet a felhasználó hozzáférése a **Microsoft Office Groove 2007** közös dokumentumokhoz? Milyen lehetőségeket biztosít?
10°. Milyen korlátozások vonatkoznak a **Google** dokumentumokra?
11°. Milyen fájlokat hozhatunk létre a **Google** dokumemtumokban?
12°. Mit jelent a közös projektkezelés?

Végezzük el a feladatokat!

1°. Indítsuk el a **Microsoft Office Groove 2007**-et:

1) Hozzunk létre saját fiókot!
2) Hozzunk létre egy **Hobbim** nevű **Normál** munkaterületet!
3) Keressünk az interneten hasonló témájú képeket, és mentsük a munkaterületre!
4) Hozzunk létre egy dokumentumban hivatkozáslistát azokra az erőforrásokra, ahonnan a képek származnak!
5) Helyezzük el a dokumentumot a munkaterületre!
6) Hívjuk meg néhány osztálytársunkat a munkaterületre!
7) Küldjünk üzenetet a felhasználóknak, amelyben kérjük őket, hogy vizsgálják meg a munkaterület anyagait, és nyilvánítsanak arról véleményt a **Vitafórumon**!
8) Fogadjuk el osztálytársaink meghívását a közös munkára!
9) Nyissuk meg az osztálytársak munkaterületét, és ismerkedjünk meg anyagaikkal!

10) Írjuk meg véleményünket a **Vitafórumra**!
11) Tekintsük meg az osztálytársaink véleményét a vitafórumon!
12) Keressünk közöttük olyanokat, akiknek a hobbija hasonló a miénkkel!
13) Hozzunk létre velük egy közös prezentációt a hobbinkról!
14) Készítsünk a prezentációhoz 3 diát, és mentsük azt a munkaterületre!
15) Küldjünk üzenetet a résztvevőknek azzal a kéréssel, hogy bővítsék a prezentációt a saját diáikkal!
16) Írjuk a saját linkjeinket a csoporttag hivatkozáslistájához, és mentsük azt a munkaterületre!
17) Tekintsük meg a közösen készített prezentációt!
18) Küldjünk a munkatársainknak köszönetet a munkájukért!
19) Zárjuk be a **Microsoft Office Groove 2007**-ot.

2*. Nyissuk meg a **Google** webhelyet:
1) Keressük meg a **Google Dokumentumok** szolgáltatást!
2) Hozzunk létre **Otthonom** mappát, és engedélyezzük minden felhasználónak a hozzáférést!
3) Keressünk az interneten képeket vidékünk ismert helyeiről és mentsük a mappánkba!
4) Hozzunk létre egy dokumentumot a képek forrásaira mutató hivatkozásokkal és mentsük azt a mappánkba!
5) Készítsünk a prezentációhoz 3 diát, és mentsük azt a munkaterületre!
6) Küldjünk a csoporttagoknak e-mailben meghívást a közös prezentáció készítésére!
7) Fogadjuk el osztálytársaink hasonló meghívásait.
8) Nyissuk meg osztálytársaink prezentációit és helyezzük el azokban a saját diáinkat!
9) Írjuk hozzá az osztálytársaink hivatkozásaihoz a sajátjainkat!
10) Nyissuk meg a közösen készített prezentációt!
11) Küldjünk a csoporttársaknak köszönetet a munkájukért!
12) Zárjuk be a **Google** fiókunkat!

20. számú gyakorlati munka **Közös projekt létrehozása**

Figyelem! *A számítógéppel végzett munka során tartsuk be a munkavédelmi és higiéniai szabályokat!*

Készítsünk közös projektet a következő témák egyikében: **„Iskolánk", „11. osztály", „Tanáraink", „Leendő szakmám", „Hol tanuljunk tovább".**

A projekt lebonyolítása céljából:
1. Alkossunk 3–4 fős csoportokat!
2. Beszéljük meg a csoportban a választott témát, osszuk szét a feladatköröket!
3. Készítsük el a projekt összetevőit: plakátokat, határidőnaplókat, hírleveleket, videókat, hanganyagokat, publikációkat stb. a rendelkezésre álló szoftvereszközök segítségével!
4. Hozzunk létre a **Microsoft Office Groove 2007** vagy a **Google dokumentumok** segítségével közös erőforrást a projekthez!
5. Vizsgáljuk meg a létrehozott dokumentumokat, végezzük el változtatásainkat! Beszéljük meg a változásokat és készüljünk fel a projekt közös megvédésére!
6. Mentsük a projektet a közös mappába!
7. Mutassuk be munkánkat az osztálytársainknak!

Tartalom

Kedves tizenegyedikesek! 3

1. fejezet. **Információs technológiák az oktatásban** 4
1.1. Oktatási szoftverek, matematikai programcsomagok 4
1. sz. gyakorlati munka. Számítások automatizálása 12
1.2. A **GRAN1** alkalmazása függvénygrafikonok ábrázolásakor 13
2. számú gyakorlati munka. Függvények grafikonjainak szerkesztése és elemzése 18
3. számú gyakorlati munka. Egyenletek és egyenletrendszerek grafikus megoldása 18
1.3. A fizika, kémia és biológia tanítását segítő oktatási szoftverek 19
4. számú gyakorlati munka. Oktatási szoftverek alkalmazása a fizika, kémia és biológia tanulása során 23

2. fejezet. **Algoritmizálás és programozás** 24
2.1. Modellek és modellezés. A modellek típusai 24
2.2. Az algoritmusok és azok tulajdonságai. Az algoritmusok megadásának módjai 29
2.3. Számítógépes programok és programozási nyelvek. A számítógépes feladatmegoldás lépései 35
2.4. A **Delphi** programozási nyelv és a **Turbo Delphi 2006** fejlesztőkörnyezet 44
5. számú gyakorlati munka. Projekt létrehozása és futtatása. Vezérlőelemek alkalmazása 55
2.5. Események és azok feldolgozása 56
6. számú gyakorlati munka. Adatbevitel és kivitel. Változókezelés 67
2.6. Hibakeresés 68
7. számú gyakorlati munka. Hibakeresés 72
2.7. A matematikai logika alapjai 73
2.8. Elágazások az algoritmusokban és programokban 78
8. számú gyakorlati munka. Elágazásokat tartalmazó programok létrehozása 88
2.9. Ciklikus algoritmusok és programok 89
9. számú gyakorlati munka. Ciklusokat tartalmazó programok létrehozása 99
2.10. Felhasználó által definiált és beépített eljárások és függvények **Delphi**-ben 99
10. számú gyakorlati munka. Alprogramok alkalmazása 105
2.11. Egydimenziós tömbök 105
11. számú gyakorlati munka. Egydimenziós tömbök kezelése 116

3. fejezet. **Adatbázisok. Adatbázis-kezelő rendszerek** 117
3.1. Adatbázisok Az egyed-kapcsolat modell 117
3.2. Adatmodellek. A relációs adatmodell alapfogalmai 122

3.3. Az adatbázis létrehozásának lépései. Az adatbázis-kezelő rendszer fogalma ... 128
3.4. **Microsoft Access 2007** adatbázis-kezelő rendszer ... 132
3.5. Az egyed-kapcsolat modell leképezése az adatbázisra. Táblakezelés **Access 2007**-ben ... 141
12. számú gyakorlati munka. Az adott alkalmazási terület egyed-kapcsolat modelljének létrehozása. Adatbázis létrehozása az adatbázis-kezelőben ... 152
3.6. Űrlapok létrehozása, szerkesztése és formázása ... 152
3.7. Adatok rendezése, szűrése és keresése adatbázisokban ... 162
13. számú gyakorlati munka. Űrlapok. Adatok felöltése. Adatok rendezése, szűrése és keresése ... 168
3.8. Lekérdezések **Access 2007**-ben ... 168
3.9. Jelentések. Jelentések létrehozása **Access 2007**-ben ... 174
14. számú gyakorlati munka. Lekérdezések és jelentések létrehozása ... 182

4. fejezet. **A személyes és csoportos kommunikáció információs technológiái** ... 184
4.1. A multimédiás adatok fogalma és feldolgozása ... 184
4.2. Hang és videó hozzáadása az oldalalapú prezentációkhoz ... 196
4.3. Videofilmek (időalapú prezentációk) készítése és szerkesztése ... 206
15. számú gyakorlati munka. Videofilm (időalapú prezentáció) készítése ... 217
4.4. A számítógépes kiadványszerkesztés alapjai. **A Microsoft Office Publisher 2007** ... 217
4.5. Szöveges és grafikai objektumok kezelése **Publisher 2007**-ben. Információs bulletin és füzet készítése ... 227
16. sz. gyakorlati munka. Információs bulletin és füzet létrehozása ... 234
4.6. Az elektronikus dokumentumok integrált kezelésének eszközei ... 236
17. számú gyakorlati munka. Adatok kezelése több szoftvereszközzel ... 247
4.7. Weblaptípusok. A weblapok osztályozása ... 247
4.8. A weblapok létrehozásának lépései. A webdizájn alapjai ... 254
4.9. A webfejlesztés automatizálásának eszközei ... 260
18. számú gyakorlati munka. Weblap automatizált létrehozása ... 271
4.10. A **Web 2.0** technológia. Webes közösségek. Blogok létrehozása ... 272
19. sz. gyakorlati munka. Saját blog létrehozása és működtetése ... 288
4.11. Dokumentumok közös kezelése ... 288
20. számú gyakorlati munka. Közös projekt létrehozása ... 300

Навчальне видання

РИВКІНД Йосиф Якович
ЛИСЕНКО Тетяна Іванівна
ЧЕРНІКОВА Людмила Антонівна
ШАКОТЬКО Віктор Васильович

ІНФОРМАТИКА

Підручник для 11 класу загальноосвітніх навчальних закладів
із навчанням угорською мовою

Академічний рівень, профільний рівень

Рекомендовано Міністерством освіти і науки України

**Видано за рахунок державних коштів.
Продаж заборонено**

Переклад з української
Перекладач *Гавриїл Гаврилович Семере*

Угорською мовою

Зав. редакцією *А. А. Варга*
Редактор *Б. Б. Ковач*
Обкладинка, художнє оформлення
та технічні малюнки *В. Марущинця*
Коректор *І. О. Петро*

Формат 70 × 100/16. Папір офс. Друк офс.
Умовн. друк арк. 24,7. Обл.-вид. арк. 23,56.
Тираж 1100 пр. Зам. № 220-11.

Державне підприємство "Всеукраїнське спеціалізоване видавництво «Світ»"
79008 Львів, вул. Галицька, 21
Свідоцтво суб'єкта видавничої справи ДК № 2980 від 19.09.2007 р.

Друк на ПРАТ "Львівська книжкова фабрика "Атлас"
79005 м. Львів, вул. Зелена, 20
Свідоцтво про реєстрацію серія ДК № 1110 від 08.11.2002 р.