

INFORMATIKA

Tankönyv az általános oktatási rendszerű
tanintézetek 8. osztálya számára

Ajánlotta Ukrajna Oktatási és Tudományos Minisztériuma



Львів
Видавництво „Світ”
2016

УДК 004(075.3)
ББК 32.97я721
I-74

Авторський колектив:

РИВКІНД Й. Я., ЛИСЕНКО Т. І., ЧЕРНІКОВА Л. А., ШАКОТЬКО В. В.

Перекладено за виданням:

Інформатика : підруч. для 8-го кл. загальноосвіт. навч. закл. / Й. Я. Ривкінд
[та ін.]. – Київ : Генеза, 2016

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 10.05.2016 № 491)

Видано за рахунок державних коштів. Продаж заборонено

Експерти, які здійснили експертизу підручника під час проведення конкурсного відбору проектів підручників для учнів 8 класу загальноосвітніх навчальних закладів і зробили висновок про доцільність надання підручнику грифа „Рекомендовано Міністерством освіти і науки України”:

Жук Н. А., методист Гоцанського районного методичного кабінету Рівненської області;

Попова Л. М., учитель інформатики спеціалізованої загальноосвітньої школи I–III ступенів № 7 Світловодської міської ради Кіровоградської області.

Інформатика : підруч. для 8 кл. загальноосвіт. навч.
I-74 закладів з навч. угорською мовою / Й. Я. Ривкінд [та ін.] ;
пер. Г. Г. Семере. – Львів : Світ, 2016. — 288 с. : іл.
ISBN 978-966-914-008-1

Навчальний матеріал підручника поділено згідно з новою програмою на 8 розділів. У підручнику розділи „Текстовий процесор” і „Табличний процесор” подано для пакета **Microsoft Office**.

Кожний пункт підручника містить запитання для актуалізації знань, основний навчальний матеріал відповідно до програми, тренувальні вправи, запитання для самоконтролю та практичні завдання, які розподілено за рівнями навчальних досягнень. Підручник містить шістнадцять практичних робіт.

Для підвищення інтересу до вивчення предмета підручник, крім основного матеріалу, містить рубрики: „Для тих, хто хоче знати більше”, „Чи знаєте ви, що...”.

УДК 004(075.3)
ББК 32.97я721

© Ривкінд Й. Я., Лисенко Т. І.,
Чернікова Л. А., Шакотько В. В., 2016
© Видавництво „Генеза”, оригінал-
макет, 2016
© Семере Г. Г., переклад угорською
мовою, 2016

ISBN 978-966-914-008-1 (угор.)
ISBN 978-966-11-0692-4 (укр.)

KEDVES NYOLCADIKOSOK!

Ebben a tanévben folytatjuk az igen fontos és érdekes tantárgy, az **informatika** tanulmányozását.

Az előző osztályokban megismerkedtetek az operációs rendszer objektumaival és műveleteket végeztetek ezekkel, információs modelleket hoztatok létre, grafikai, szöveges, multimédiás és számadatokat dolgoztatok fel, megismerkedtetek az elektronikus levelezéssel és az internetes kereséssel, algoritmusokat állítottatok össze **Rőt Kandúr** számára, kompetencia- és projektfeladatokat oldottatok meg.

A nyolcadik osztályban megismerkedtek majd az adatok kódolásával, áttekintitek a számítógép hardverét és szoftverét, folytatjátok a szöveges valamint számadatok feldolgozását és a kompetenciafeladatok megoldását. Megtanultok videofájlok létrehozni, hangokat feldolgozni, programokat fejleszteni és végrehajtani egy programozási rendszerben. A megszerzett tudást tanulmányi projektek végrehajtása során kamatoztathatjátok majd. Ebben segít majd nektek ez a tankönyv.

A könyv fejezetekre van felosztva. Minden fejezet pontokra, azok alpontra bomlanak tovább. Minden pont ismétlő kérdésekkel kezdődik. Ezeket megválaszolva könnyebb lesz az új anyag elsajátítása. A kérdéseket




-vel jelöljük.



Olvasd a tankönyvet figyelmesen. Minden pont végén az **Összefoglalás**







rubrikában megtaláljátok annak rövid összefoglalását.



Az egyes pontok **Felelj a kérdésekre!** című résszel  zárulnak. Ajánljuk, hogy a pont elolvasása után válaszoljatok ezekre a kérdésekre! A kérdések melletti jel azt mutatja, hogy az adott kérdés megválaszolása:

- – alapszintű;
- – középszintű;
- * – vagy emelt szintű ismereteknek felel-e meg.

Ugyanilyen módon jelöltük a kérdéseket követő gyakorlati feladatok szintjeit. A házi feladatnak ajánlott feladatokat  -al jelöltük. A  -al jelzett feladatok elvégzése párokban vagy kiscsoportokban ajánlott.

A törzsanyagon kívül a könyvben a következő kiegészítéseket találjátok:

-  Ha többet szeretnél tudni
-  Tudtad-e, hogy...?
-  Érdekes történelmi tények
-  **Linux**-használóknak

-  **LibreOffice** felhasználóknak
-  **Microsoft Office 2010** felhasználóknak
- Magyarázó szótár (zöld háttérrel kiemelve)

A könyv végén **Szómagyarázót** találtok, ami a legfontosabb fogalmak definícióit tartalmazza.

A gyakorlati munkák elvégzéséhez szükséges fájlokat a <http://allinf.at.ua> webhelyen találjátok.

Sok sikert kívánunk a modern és érdekes tudomány, az informatika tanulmányozásához!

A szerzők

Ebben a fejezetben megismerkedtek:

az adatfeldolgozással,
mint információs
folyamattal

az üzenetek
kódolásával
és dekódolásával

a kódtáblázatokkal

a bináris
kódolással

a bináris kód
mértékegységeivel

1.1. AZ ADATFELDOLGOZÁS, MINT INFORMÁCIÓS FOLYAMAT. ÜZENETEK KÓDOLÁSA ÉS DEKÓDOLÁSA



1. Milyen információs folyamatokat ismertek? Mi az üzenetfeldolgozás folyamatának lényege?
2. Milyen módon reprezentálhatunk üzeneteket?
3. Mik az adatok? Hozz fel példákat!

ADATFELDOLGOZÁS, MINT INFORMÁCIÓS FOLYAMAT

A modern társadalomban nagyon sok szakma kapcsolódik a szavak, számok, grafikai ábrázolások formájában előállított adatok feldolgozásával. A könyvelők az elvégzett munkákat jellemző információk



1.1. ábra. Adatok feldolgozása különböző szakmákban

alapján végzik el a fizetések kiszámítását. Az orvosok a vizsgálati eredményekre támaszkodva határozzák meg a diagnózist. A meteorológusok a levegő mozgásáról összegyűjtött adatok alapján készítik el az előrejelzéseiket. A csillagászok a különböző kozmikus testek mozgáspályáját azok elhelyezkedése alapján számítják ki. A művész a könyv tartalma alapján készíti el annak illusztrációit (1.1. ábra). A tanulók is adatokat dolgoznak fel, amikor feladatokat oldanak meg, fogalmazásokat írnak, grafikonokat rajzolnak stb.

Ahogy már tudjátok, az **adatok feldolgozása** az a folyamat, amelynek során új adatokat hozunk létre a meglévők alapján.

Amikor például egy matematikafeladatot oldunk meg, akkor a feladat feltételéből ismert kiinduló adatokból új adatokat – a feladat eredménye – állítunk elő. A futóverseny eredményének meghatározása azt jelenti, hogy a versenyzők neveit az általuk elért időeredmények szerint növekvő sorrendbe helyezzük. A táblázatkezelőben egy diagram elkészítésekor a meglévő adatokat új, grafikai formában ábrázoljuk.

Az adatokat különféle eszközök – például számítógépek – segítségével is feldolgozhatjuk. Ez esetben az adatokat olyan formában kell előállítani, amilyenben a számítógép képes ezeket feldolgozni és meg kell határozni azokat a szabályokat, amelyek alapján a feldolgozást végre kell hajtani.



ÜZENETEK KÓDOLÁSA

Az üzenet tárolása, továbbítása, feldolgozása során gyakran szükség van arra, hogy megváltoztassuk az üzenet reprezentációjának (megjelenítésének) módját.

A szóbeli üzeneteket papírra rögzítik, ekkor az emberi hangot betűkre cseréljük. Ez leginkább az üzenet tárolása céljából történik. A telefonbeszélgetés során a szóbeli üzeneteket elektromos jelekre cserélik. Ennek célja az, hogy az üzenetet nagy távolságra lehessen továbbítani. A matematikaórán a számokat jelölő szavakat számjegyekké, a műveleteket jelölő szavakat pedig műveleti jelekké alakítjuk. Ebben a formában könnyebb az információ feldolgozása. Az üzenet tartalmának védelme céljából rejtjelezhetik azt, amelynek során a betűket más betűkre, számokra vagy jelekre cserélik.

Amikor tehát az üzenet tartalmát alkotó jeleket más jelekre cserélik, **rejtjelezzik az üzenetet**.

A kódolást gyakran alkalmazzák a mindennapokban is, hogy az üzeneteket ne csak az ember, de a gép is képes legyen feldolgozni. Az árukat például leggyakrabban vonalkóddal kódolják (1.2. ábra). A vonalkód számjegyei egy áru nevét, származási helyét, gyártóját kódolják. Ezt a számkódot az ember általi információ-feldolgozás és rendszerezett adattárolás céljából hozták létre. A számjegyeket a vonalak vastagsága és az azok közötti távolság kódolja. Ezt a grafikai kódot olvassa be a vonalkód-leolvasó, és dolgozza fel a későbbiek során a számítógép.

Az üzenetek kódolása céljából nemcsak az alkalmazott karaktereket kell ismerni, hanem a kód előállításának módját is.

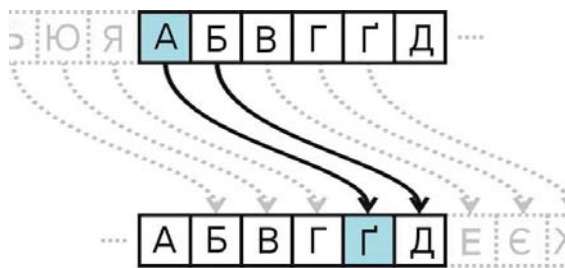
A történelemből jól ismert az a kódolási mód, amelyet Gaius Julius Caesar (i. e. 100–44) alkalmazott a tábormokaival folytatott levelezésében. A rejtjelezés lényege az, hogy minden egyes betűt az ábécében egy tőle meghatározott távolságra lévő betűvel kell helyettesíteni (1.3. ábra).

A *Hello* szóból 4 betűnyi eltolással így módon a *Khoos* szót kapjuk.

Rejtjel, kód a (*codex* – könyv szóból) – megegyezéssel jelölésrendszer, amelyet az üzenetek tárolása, továbbítása és feldolgozása céljából hoztak létre.



1.2. ábra. Egy árucikk vonalkódja



1.3. ábra. A 4 pozícióval eltoló Caesar-kód előállítása

AZ ÜZENETEK DEKÓDOLÁSA

A **dekódolás** az a folyamat, amelynek elvégzése után a kódból visszafejtjük az eredeti üzenetet.

Dekódolást végzünk például akkor, ha hangosan felolvasunk egy szöveget, zenét játszunk kottából, megfejtünk egy titkosírást vagy a vonalkód alapján azonosítunk egy árucikket stb.

Ha egy üzenetet 4 pozíciós jobbra tolt Caesar-kóddal kódoltak, akkor a dekódolás során a betűket 4 hellyel balra kell eltolni. A *mő rdtsü* üzenet dekódolva tehát *jó napot*-ra változik.

Az információ kódolása és dekódolása információ-feldolgozási feladatnak tekinthető.

! Összefoglalás

Az **adatifeldolgozás** – új adatok előállítása a meglévők alapján.

Az üzenetek **kódolása** – az üzenetet reprezentáló egyik szimbólumsorozat cseréje egy másik szimbólumsorozatra.

Az üzenetek kódolására azok tárolása, továbbítása, átalakítása és védelme céljából van szükség.

Az üzenetek kódolásakor meg kell adni az új kód előállítása során alkalmazandó szimbólumokat és az új kód képzésének szabályait.

Az üzenet dekódolása alatt az eredeti üzenet visszaállítását értjük a kódolt üzenet alapján.

Az információ kódolása és dekódolása információ-feldolgozási feladatnak tekinthető.

? Felelj a kérdésekre!

- 1°. Mit értünk információ-feldolgozás alatt?
- 2°. Milyen új adatokat állíthatunk elő a következő adatok alapján:
 - a) az üzletben levő árucikkek mennyisége és ára;
 - b) a tanulók magassága;
 - c) két település távolsága és a közöttük haladó vonat sebessége!
- 3°. Mit értünk az üzenetek kódolása alatt?
- 4°. Milyen célból végezhetünk kódolást?
- 5*. Milyen célból kódolják a zenét hangjegyekkel; a közlekedés szabályait közlekedési táblákkal?
- 6*. Milyen üzeneteket kódolhatunk színekkel; grafikai jelekkel?
- 7°. Mit szükséges meghatározni a kódolás megkezdése előtt?
- 8*. Hogyan kódolják a *Tengeri csata* játékban a hajók helyzetét? Miért?
- 9°. Mit értünk dekódolás alatt?



Végezd el a feladatokat!

1. Mondj példákat adatok feldolgozására!
2. Hozz fel példákat üzenetek kódolására, amelyeket a zene, fizika, kémia, földrajz stb. tanulmányozása során alkalmaztatok!
3. Kódold 4 betű eltolásával keletkező Caesar-kóddal a következő üzeneteket:

- a) *informatika*;
- b) *iskolánk*;
- c) *üzenetek kódolása és dekódolása!*



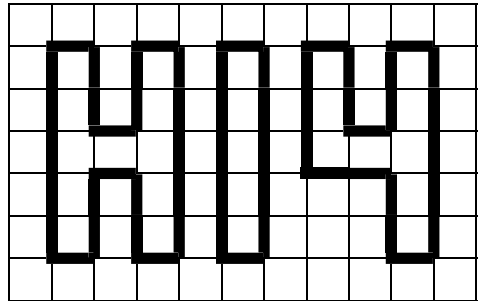
4. Az üzenetek 3 betű eltolásos Caesar-kóddal lettek kódolva. Végezd el a dekódolást:

- a) *úbgstüóegu*
- b) *mp kfr*
- c) *ógúngbfrfqüü c ücöy!*

5. Egy ceruzarajz a következőképpen van kódolva: a nyíl a ceruza mozgásának irányát mutatja, a szám pedig az elmozdulás mértékét kockákban. Hozd létre a képet:

- a) $\uparrow 5 \rightarrow 3 \downarrow 5 \leftarrow 1 \uparrow 4 \leftarrow 1 \downarrow 4 \leftarrow 1$;
- b) $\leftarrow 3 \downarrow 5 \rightarrow 3 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1$!

6. Kódold az ábrán látható képet az előző feladatban megismert módszer szerint!



- 7*. Egészítsd ki a megismert kódolási módszert oly módon, hogy meg lehessen adni a ceruza mozgását rajzolás nélkül is!

8. Az egyik legrégebbi rejtjelezési forma az i. e. III. századból származó Polübiosz-négyzet vagy Polübiosz-tábla. Itt minden betűt (vagy betűpárt) egy négyzetrácsban helyezünk el. Kódoláskor minden betűt egy oszlop és egy sor jelöl, amelyek metszésében a betű áll. Az ábra az ukrán ábécé egy lehetséges kódolását mutatja. Dekódold az alábbi üzeneteket:

	1	2	3	4	5	6
1	A	Б	B	Г/Г	Д	Е
2	Є	Ж	З	И/Й	І/Ї	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ь	Ю	Я

- a) 53 43 13 52 21 52 42;
- b) 62 43 51 34 31 11 33 33 65!

9. Készítsd el a magyar ábécé kódolására alkalmas Polübiosz-négyzetet, és kódold az alábbi szövegeket:

- a) *informatika*;
- b) *üzenet!*

Polübiosz-tábla

10*. Gondolj ki egy kódolási módszert, majd ennek segítségével kódold a következő mondatokat:

- a) *Ma szép nap van;*
- b) *Szeretem a focit!*



11*. Keress információkat a következő kódolási módszerekről:

- a) Morze-ábécé;
- b) Braille-írás;
- c) szemafor-nyelv;
- d) Selam vagy napkeleti virágnyelv!



1.2. KARAKTEREK KÓDOLÁSA



1. Mit értünk az üzenetek kódolása alatt? Mi célból végzik?
2. Mondj példákat üzenetek kódolására!
3. Mit értünk a dekódolás alatt?

A SZÖVEG KARAKTEREINEK KÓDOLÁSA

Ahhoz, hogy a szöveges adatokat számítógéppel dolgozhassunk fel, számok segítségével kell azt kódolnunk.

Teletype (a latin *tele* – távoli és az angol *type* – nyomtatás szavakból) – elektromechanikus írógép, amelyek vezetéseken továbbított információt nyomtattak.

A szöveg kódolása során **kódtáblázatokat** használnak, amelyekben minden egyes alkalmazandó karakterhez egy szám van rendelve. Az USA-ban 1963-ban kidolgoztak egy ilyen kódtáblázatot, amelyet telexgépekhez (*teletype*, távíró) fejlesztettek ki. Ez később

szabvánnyá vált a számítástechnikában és az **ASCII** (*American Standart Code for Information Interchange*) – Amerikai szabványos információcsere kód – nevet kapta.

Az **ASCII** kódtábla az angol abécé betűit, számjegyeket, elválasztójeleket, szerkesztési és formázási karaktereket tartalmaz, amelyek a 0 és a 127 közötti számokat feleltetnek meg (1.1. táblázat).

1.1. táblázat

Az ASCII kódtábla kódcsoportjai

Intervallum	Kódcsoport	Példa	Leírás
0-tól 31-ig és 127	Speciális karakterek	10	Új sor jel
		13	Sor elejére ugrás
		27	Az Esc billentyű lenyomásával egyenértékű



Az 1.1 táblázat folytatása

<i>Intervallum</i>	<i>Kódcsoport</i>	<i>Példa</i>	<i>Leírás</i>
32-től 64-ig, 91-től 96-ig, 123-tól 126-ig	Számjegyek és elválasztójelek	32	szóköz
		48	a 0 számjegy
		123	{
65-től 90-ig	Nagybetűk	65	<i>A</i>
		66	<i>B</i>
		90	<i>Z</i>
97-től 122-ig	Kisbetűk	97	<i>a</i>
		98	<i>b</i>
		122	<i>z</i>

Az **ASCII** kódtábla 32 és 127 közé eső kódjainak megfelelő karaktereket az 1. melléklet tartalmazza, amelyet az *allinf.at.ua* oldalon a nyolcadikos tankönyvnek megfelelő részben tekinthettek meg.

Az **ASCII** kód csak az angol betűket tartalmazza. Más nyelvek karaktereinek kódolásához ettől eltérő kódra volt szükség. A **KOI8-U** (**KOI** – код обміну інформацією (információcsere kód)) és a **Windows-1251** az **ASCII** karaktereken kívül még a cirill betűket is tartalmazzák. Ezeknek a karaktereknek a 128 és a 255 közötti kódok felelnek meg.

Cirill ábécé – a szláv nyelvek ábécéje. Ide tartozik az ukrán, orosz, belorusz, bolgár stb.

Az angol és egyes nyugat-európai nyelv **latin ábécét** használ.

A **Windows** operációs rendszerben a **Windows-1251** kódolás a szabványos. Itt az ukrán „*a*” karakter kódja 224, az „*i*” kódja 179, a „*I*” kódja pedig 180. A 128 és 255 közé eső kódoknak megfelelő karaktereket megtekinthetitek a 2. mellékletben az *allinf.at.ua* oldalon a nyolcadikos tankönyvnek megfelelő részben¹.

A 0 és 255 közötti kódok elégségesek arra, hogy az angol, a cirill és még néhány más nyelv betűit kódoljuk, de ezeken kívül még rengeteg karakter létezik (görög, arab, kínai stb.). Hogy ezeket a karaktereket is lehessen kódolni, kifejlesztették az **UNICODE** (Uniform Character Encoding – *univerzális karakterkészlet*) szabványt.

Az **UNICODE** táblázatban a kódok 17-től 65 536-ig terjedhetnek, ezek segítségével 1 114 112 karakter kódolható, amely kielégíti valamennyi nyelv igényeit. Ahogy minden kódrendszerben, az **UNICODE**-ban is a 0 és 127 közé eső kódok megegyeznek az **ASCII** kóddal. A kód további részében megtalálhatjuk a különféle nyelvek karaktereit. Itt az ukrán „*a*” karakter kódja 53 424, az „*i*” kódja 53 654, a „*I*” kódja pedig 53 905.

¹ A magyar karakterek kódolására az **ISO-8859-2**, illetve a **Windows-1250** táblák felelnek meg (a fordító megjegyzése).

1. fejezet

A legújabb operációs rendszerekben a **UNICODE**-ot használják.
Az 1.2. táblázatban összehasonlítottuk a különböző kódtáblázatok szerkezetét.



1.2. táblázat





Különböző kódtáblák szerkezetének összehasonlítása

Kódtábla	Karakterek száma	0 és 127 közötti kódok	128 és 255 közötti kódok	255-öt meghaladó kódok
ASCII	128	Angol betűk, számjegyek és elválasztójelek, speciális karakterek	Nincs	Nincs
KOI8-U	256	Ahogy az ASCII-ben	Cirill betűk, más nyelvek különleges betűi	Nincs
Windows-1251	256	Ahogy az ASCII-ben	Cirill betűk, más nyelvek különleges betűi	Nincs
UNICODE	1 114 112	Ahogy az ASCII-ben	Különböző nyelvek betűi, speciális karakterek	



Ha többet szeretnél tudni

Az **UNICODE** táblázat nemcsak betűket és számjegyeket tartalmaz, hanem egyéb szimbólumokat is: például bejegyzett védjegy-jelet (registered trademark), pénznevek jeleit, transzkripció jeleket, ideogramokat is. Az ukrán pénznem, a hrivnya ₴ jelének kódja például a 8 372, az *emberke* ideogrammé  10 080, a *nő* ideogrammé  10 081.

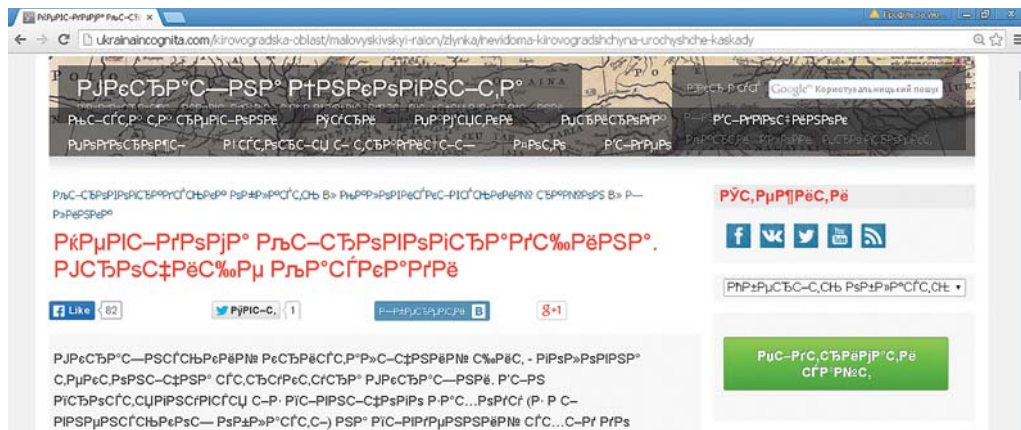
A speciális karaktereket az **UNICODE**-tábla megfelelő számértékeinek ismeretében könnyen beilleszthetjük a dokumentumainkba. E célból tartuk lenyomva az **Alt** billentyűt, és írjuk be a számbillentyűzet (a billentyűzet jobb oldali részén) a megfelelő számot. Ily módon beilleszthetjük például a hóember (kódja 9 731) , a virág alakú szív (kódja 10 087) , a hangjegy (kódja 9 834) , a sakkhuszár (kódja 9 822) , valamint egyéb jeleket. A karakterek kódjait megtalálhatjuk az **UNICODE**-tábla (unicode-table.com) oldalon. Ha azonban a kiválasztott jelet a számítógépünkön telepített betűtípusok nem tartalmazzák, az nem lesz látható a képernyőnkön.

Ideogramma (a görög *idéa* – és *gramma* – írásjel γράμμα) – olyan írásjel, amely a betűktől eltérően nem valamely hangot, hanem egy egész fogalmat jelöl.




WEBLAPOK KÓDOLÁSA

A karakterkódolások a weblapok tartalmára is hatással vannak. Előfordul, hogy amikor megnyitunk egy weblapot a böngészőben, a karakterkódolást a böngészőnk helytelenül választja ki. Ezekben az esetekben a weblap szövege értelmezhetetlenné válik (1.4. ábra).



1.4. ábra. Weblap helytelenül értelmezett karakterkódokkal

Minden böngészőben beállítható, hogy milyen karakterkódolásban szeretnénk megtekinteni a weblapot. A **Google Chrome** esetében ezt a következőképpen tehetjük meg:

1. Kattintsunk a **Chrome beállításai**  gombra!
2. Kattintsunk a **További eszközök** ⇒ **Karakterkódolás** pontra!
3. Válasszuk ki azt a karakterkódolást, amelyben a weblapunk értelmezhetően jelenik meg!

Az ukrán nyelvű weblapok esetében az leggyakrabban az **UNICODE (UTF-8)** lesz, de egyes esetekben a **Windows -1251** vagy a **KOI-8U** lehet a megfelelő választás¹.



Ha többet szeretnél tudni

Kódtáblázat a Windows rendszerben

A **Windows** segédprogramjai között megtalálhatjuk a **Karaktertábla** programot, amelynek segítségével olyan jeleket illeszthetünk be a dokumentumainkba, amelyeket nem találhatunk meg a billentyűzetben. A programot, amelynek ablaka az 1.5. ábrán látható, a **Start** ⇒ **Minden program** ⇒ **Kellékek** ⇒ **Rendszerezszközök** ⇒ **Karaktertábla** paranccsal nyithatjuk meg.

A program segítségével csoportosítva tekinthetjük át a beilleszthető karaktereket. Ezt a következőképpen tehetjük meg:

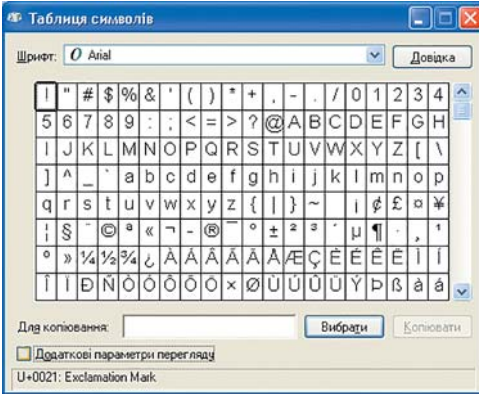
1. Kapcsoljuk be a **Speciális nézet** jelölőt!
2. A **Karakterkészlet** mezőben válasszuk ki az **Unicode**-ot!

¹ A magyar nyelvű weblapok esetében szintén beválhat az **UTF-8** vagy a **Windows-1250**, esetleg az **ISO-8859-2** (a fordító megjegyzése).

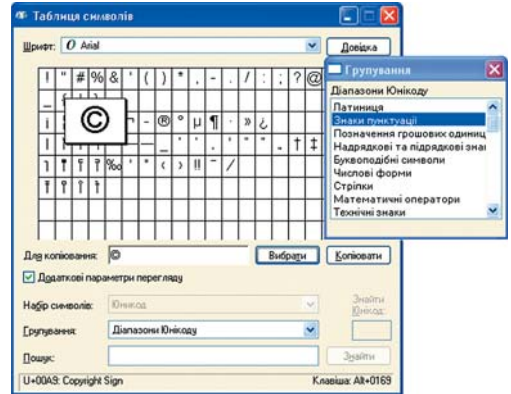


1. fejezet

3. A **Csoportosítás** menüben válasszuk ki az *Unicode alosztályt*, ekkor megnyílik az 1.6. ábrán látható párbeszédablak!



1.5. ábra. A **Karaktertábla** program ablaka



1.6. ábra. A **Karaktertábla** ablaka az **Általános központozás** csoport jeleivel

Ha kiválasztunk egy karaktert, a program státuszsorában megjelenik a megfelelő **Unicode** számkód és a karakter rövid angol nyelvű leírása, a jobb oldalon pedig a beillesztéshez szükséges billentyűkombináció. A szerzői jogra utaló copyright-jel © esetében ez *U+00A9 Copyright Sign* és az *Alt+0169* billentyűkombináció.

A kiválasztott karaktert különböző dokumentumokba – szövegbe, táblázatba, prezentációkba – illeszthetjük be a következőképpen:

1. Jelöljük ki a karaktert!
2. Hajtsuk végre a **Kijelölés** ⇒ **Másolás** parancsot!
3. Kattintsunk a dokumentumokban a beillesztés helyére!
4. Illesszük be a karaktert a **Vágólapra**!

Amikor a **Kiválasztás** gombra kattintunk, a karakter bekerül a **Másolandó** karakterek mezőbe. Ha még egy karaktert kiválasztunk, akkor két karakter lesz a mezőben. Ily módon egész karaktersorozatot választhatunk ki, valamennyi a vágólapra kerül majd.




Összefoglalás

Szöveges üzeneteket úgy dolgozhatunk fel számítógép segítségével, hogy azt előzőleg számok segítségével kódoljuk. E célból kódtáblázatokat használunk, például **ASCII**, **KOI8-U**, **Windows-1251**, **Unicode**.

Az **ASCII**-vel 128, a **KOI8-U** és **Windows-1251** segítségével 256, az **Unicode**-dal 1 114 112 különböző karaktert kódolhatunk. Valamennyi kódtábla első 128 karaktere megegyezik, ami megfelel az **ASCII**-nak.

Minden böngészőben lehetőségünk van a weblap karakterkódolásának beállítására.



 A **Karaktertábla** segédprogram rendeltetése, hogy csoportosítva mutassa meg a billentyűzetben nem szereplő karaktereket, és tegye lehetővé azok beillesztését különböző dokumentumokba.



Felelj a kérdésekre!

- 1• Hogyan végezzük a karakterek kódolását számítógépes szövegfeldolgozás során?
- 2° Milyen karaktertáblákat ismerünk?
- 3• Milyen karaktereket kódolhatunk **ASCII**, **KOI8-U**, **Windows-1251** és **Unicode** táblák segítségével?
- 4* Mi a közös, és mi a különbség az **ASCII**, **KOI8-U**, **Windows-1251** és **Unicode** táblákban?
- 5° Milyen esetekben van szükség a weblapok karakterkódolásának megváltoztatására?
- 6• Hogyan választjuk ki az alkalmazandó karakterkódolást **Google Chrome** böngésző esetében?
- 7° Mi a **Karaktertábla** segédprogram rendeltetése?
- 8° Hogy indítjuk el a **Karaktertábla** programot?
- 9• Hogy illesztünk be karaktereket a **Karaktertáblából** a dokumentumainkba?



Végezd el a feladatokat!

- 1• Az alábbi üzenetet **ASCII** tábla segítségével kódoltuk. Végezd el a dekódolást (használd az 1. mellékletet¹):
 - a) 83 117 110;
 - b) 104 111 109 101!
- 2• Kódolt az előbbi feladatban dekódolt üzenetet **Windows-1251** tábla segítségével (használd a 2. mellékletet):
 - a) Підручник «інформатика»
 - b) a nevedet!
- 3* Az alábbi közlemény egyes szavai az **Unicode** tábla szerint vannak kódolva. Keresd meg az **Unicode** táblában (unicode-table.com) a megfelelő karaktereket, és egészítsd ki a szöveget!

Ma reggel az idő 9925, a meteorológusok 9928-at ígérnek. 127979-be induláskor ne felejtssd magaddal vinni 9730-at.
- 4• Látogasd meg az ua.uacomputing.com weboldalt, és tekints meg annak tartalmát **Windows-1251** és **KOI8-U** kódolásban! Határozd meg, melyik kódolásban értelmezhető a szöveg!
- 5• Helyezd el egy dokumentumban valamennyi pénznem jelét a **Karaktertáblából**. Mentsd el a dokumentumot **1.2.5.feladat** néven a szövegszerkesztő alapértelmezett formátumában a mappádba!

¹ Valamennyi mellékletet az allinf.at.ua weboldalon, a **Nyolcadikos tankönyv** fejezetben találtok meg.

1. fejezet



- 6*. Helyezd el egy dokumentumban valamennyi betűszerű szimbólum jelét a **Karaktertáblából!** Mentsd el a dokumentumot **1.2.6.feladat** néven a szövegszerkesztő alapértelmezett formátumában a mappádba!
- 7*. Határozd meg, milyen karaktereket lehet beilleszteni az **Alt + kód** billentyűkombinációkkal a szövegedbe, ha a kód értéke 1234, 3754, 8986, 9708, 9415, 10001, 10239!



1.3. BINÁRIS KÓD



1. Mit értünk az üzenetek kódolása és dekódolása alatt?
2. Milyen kódtáblákat használunk szövegek kódolása során?
3. Mit jelentenek a *kilo*, *mega*, *giga* prefixumok?

BINÁRIS KÓD. A BIT ÉS A BYTE



1.7. ábra. Samuel Morse

Ahogy arról már beszéltünk, adatok kódolásakor különböző jelrendszereket használhatunk. A nyelv hangjainak kódolása betűk segítségével történik. Az ukrán nyelvben 33, az angolban 26 betűt alkalmaznak. A számok írásakor 10 számjegyre van szükségünk. A gyalogátkelőn a forgalom szabályozását három szín segítségével oldják meg. A morzeábécé csak kétféle jelet használ, a *pontot* és a *vonást*.

A Samuel Morse (1791–1872) amerikai művész által 1844-ben a távirón történő adatátvitel céljából megalkotott, később a tiszteletére elnevezett ábécében a betűket pontokból és vonalakkal álló sorozatokkal kódolták. Néhány magyar betű kódját az 1.3. táblázatban láthatjátok.

1.3. táblázat

Néhány magyar betű morzekódja

Betű	Morzekód
a	· –
d	– – –
p	· – – ·
r	· – ·
t	–



A *pad* szó morzekódja tehát $\cdot - - \cdot \cdot -$ lesz.

Amikor az üzenetet két jel segítségével kódoljuk, **bináris kódolásról** beszélünk. Az ilyen kódolás eredményeként előálló kódot **binárisnak** nevezzük.

A számítógépekben is bináris kódolást használunk. A számítógépekben valamennyi információt **0** és **1** számjegyek sorozataként kell előállítani.

A 0 és 1 számjegyeket a bináris kódban biteknek (az angol *binary digit* – bináris szám szavakból) nevezzük.

Az 1.3. táblázatból láthatjuk, hogy a *t* betű kódolásához egyetlen jel szükséges, egy vonal, az *a* betű kódolásához kettő, a *p* kódolásához pedig 4. A távirókban az egyes betűket hosszabb szünetekkel választották el egymástól. A számítógépekben ez a módszer nem igazán használható jól. Kényelmesebb minden betű kódolásához ugyanannyi jelet használni.

Ha egy üzenet kódolására egyetlen bitet (amelynek értéke 0 és 1 lehet) használunk, akkor a következő üzeneteket áll módunkban kódolni:

- egy állítás *igaz* (1) vagy *hamis* (0);
- egy ember neme *férfi* (1) vagy *nő* (0);
- egy kapcsoló *bekapcsolt* (1) vagy *kikapcsolt* (0) állapotban van stb.

Két bitből $2^2=4$ kódot készíthetünk (00, 01, 10, 11). Ezekkel kódolhatjuk például az égtájakat: 00 – észak, 01 – kelet, 10 – dél, 11 – nyugat.

Ha három bitünk van, akkor a kódok száma 8 (2^3) (000, 001, 010, 011, 100, 101, 110, 111), ezek egy sakktábla sorának vagy oszlopának kódolására lehetnek alkalmasak.

4 bitből 16 (2^4), ötből pedig 32 (2^5) kód állítható elő és így tovább.

Nyolc bitből $2^8=256$ kód állítható elő, ami elég az angol és az ukrán (magyar stb.) betűk, továbbá a számjegyek és néhány különleges karakter kódolásához. Pontosán ennyi kódot tartalmaz a **Windows-1251** tábla.

A nyolc bitet tartalmazó sorozatot **bájtnak** (byte-nak régi írásmód) nevezzük.

1 bájtnak = 8 bit.

A BINÁRIS KÓD HOSSZA

Ha egy üzenet minden karakterét 8 biten kódoljuk, akkor az egy karakter kódolásához 8 bitet, tehát 1 bájtnak használunk.

Egy **üzenet bináris kódjának hossza** az üzenet kódolásához szükséges bájtnak száma.

Az *informatika érdekes* üzenet például a szóközökkel együtt 17 karaktert tartalmaz. Ha minden karaktert egy bájtnak kódolunk, akkor a hossza 17 bájtnak lesz.

Azt már tudjuk, hogy a **Windows-1251** táblázatban minden karakternek egy 0 és 255 közötti szám van megfeleltetve. Ezeket a számokat egy bájtnak kódolhatjuk. Az 1.4. táblázat ilyen kódolásokat mutat be.

Bináris kódok

<i>Karakter</i>	<i>Szám kód</i>	<i>Bináris kód</i>
!	33	00100001
@	64	01000000
W	103	01100111
ю	254	11111110
я	255	11111111

A **Windows-1251** táblában tehát minden karaktert egy bájtton ábrázolunk.

Ha tehát egy szöveg bináris kódját szeretnénk előállítani, minden karakter kódját ki kell cserélni annak bináris megfelelőjére. Ily módon kódolja a karaktereket a **Jegyzettömb**, amely a mentéskor **ANSI** (*American National Standards Institute*) kódolást alkalmaz. Az ily módon mentett fájlban a bájtok száma megegyezik a betűk számával. Fontos megjegyezni, hogy az **Enter** billentyű leütése két bájtnyi vezérlő kódot eredményez, az egyik a sor elejére ugrás, a másik pedig az új sor jele.

A számítógéppel feldolgozott grafikai, audio- és videoinformációt szintén binárisan kell kódolni.

Az üzenet bináris kódjának a hosszát **adatmennyiségnek** is nevezik.

A BINÁRIS KÓD HOSSZÁNAK TÖBBSZÖRÖS MÉRTÉKEGYSÉGEI

A bináris kód hosszának kifejezésére a *kilo*, *mega*, *giga* és *tera* prefixumok segítségével többszörös mértékegységeket alkottak. Ezeket a prefixumokat megtalálhatjuk a SI mértékegységrendszerben. Ugyanitt megtalálhatjuk a megfelelő szorzószámokat is. Történelmileg úgy alakult, hogy ezek a prefixumok az informatikában kicsit mást jelölnek, mint a fizikában. Hatodik osztályban megtanultátok, hogy:

$$1 \text{ kbájt (kilobájt)} = 2^{10} \text{ bájt} = 1024 \text{ bájt};$$

$$1 \text{ Mbájt (megabájt)} = 2^{10} \text{ kbájt} = 2^{20} \text{ bájt} = 1\,084\,576 \text{ bájt};$$

$$1 \text{ Gbájt (gigabájt)} = 2^{10} \text{ Mbájt} = 2^{20} \text{ kbájt} = 2^{30} \text{ bájt};$$

$$1 \text{ Tbájt (terabájt)} = 2^{10} \text{ Gbájt} = 2^{20} \text{ Mbájt} = 2^{30} \text{ kbájt} = 2^{40} \text{ bájt}.$$

Ez azért alakult így, mert a számítógépekben bináris kódokat használunk, amelyben kényelmesebb a 2 hatványaival számolni, mint a 10 hatványaival. Mivel pedig $2^{10} = 1024$ elég közel van az 1000-hez, ezért a 2^{10} bájtot nevezték el **kilobájt**nak. Hasonlóképpen a 2^{10} kilobájtból lett az 1 **megabájt**.



ÁTVÁLTÁS A BINÁRIS KÓD HOSSZÁNAK MÉRTÉKEGYSÉGEI KÖZÖTT

A bináris kód hosszát bitekben, bájtokban és azok többszöröseiben fejezhetjük ki. Lássunk néhány feladatot, amelyben megvizsgáljuk, hogyan történik az átváltás ezek között a mértékegységek között.

1. feladat. A bináris kód hossza 4,5 Mbájt. Fejezd ki ezt bájtokban!

Megoldás. Az átalakítás során előbb alakítsuk át a kód hosszát kilobájtokra. Mivel 1 Mbájt = 1024 kbájt:

$$4,5 \text{ Mbájt} \cdot 1024 = 4608 \text{ kbájt.}$$

Figyelembe véve, hogy 1 kbájt = 1024 bájt:

$$4608 \text{ kbájt} \cdot 1024 = 4\,718\,592 \text{ bájt.}$$

Felelet: 4,5 Mbájt = 4 718 592 bájt.

2. feladat. A bináris kód hossza 4 194 304 000 bit. Fejezd ki a kódot egész számmal a lehető legnagyobb mértékegységben!

Megoldás. Fejezzük ki a kód hosszát bájtokban, figyelembe véve, hogy 1 bájt = 8 bit:

$$4\,194\,304\,000 \text{ bit} : 8 = 524\,288\,000 \text{ bájt.}$$

Az eredményünk meghaladja az 1024-et, amivel a bájtokat kilobájtokká alakíthatjuk, ezért áttérhetünk kilobájtokra:

$$524\,288\,000 \text{ bájt} : 1024 = 512\,000 \text{ kbájt.}$$

Az előzőhöz hasonló gondolatmenet mentén áttérhetünk megabájtokra:

$$512\,000 \text{ kbájt} : 1024 = 500 \text{ Mbájt.}$$

Mivel a kapott szám kevesebb 1024-bél, magasabb mértékegységre már nem térhetünk át oly módon, hogy az eredmény egész szám legyen.

Felelet: 4 194 304 000 bit = 500 Mbájt.

3. feladat. Mérd fel az informatikakönyv kódjának közelítő mértékét! Az illusztrációk kódjának hosszát hagyd figyelmen kívül!

Megoldás. A könyv szövegét **Windows-1251**-ben kódoltuk, ahol egy karakter kódjának hossza 1 bájt.

A szöveg egy sora átlagosan 60 karaktert tartalmaz, vagyis egy sor bináris kódja 60 bájt. Oldalanként átlagosan 50 sort találunk, tehát egy oldalnyi szöveg $50 \cdot 60 = 3000$ bájt. A könyv nagyjából 250 oldalt tartalmaz, az ehhez tartozó kód hossza nagyjából $3000 \cdot 250 = 750\,000$ bájt. Váltuk ezt át egy nagyobb mértékegységre:

$$750\,000 \text{ bájt} : 1024 \approx 732,4 \text{ kbájt.}$$

Felelet: a könyv szövegének kódja 732,4 kbájt.

Összefoglalás

A kódolásnak azt a módját, amikor csak kétféle jelet használunk, **binárisnak** nevezzük. Az így keletkezett kód a **bináris kód**.

A bináris kód 0 vagy 1 jegye az 1 **bit**.

A 8 bitből álló kódsorozat az 1 **bájt**.

1 bájt = 8 bit.

Az üzenet kódjának hossza alatt a kódja bájtokban kifejezett hosszát értjük.

A **Windows-1251** kódtáblában minden karakternek 1 bájtnyi kód felel meg.

1 kbájt (kilobájt) = 2^{10} bájt = 1024 bájt;

1 Mbájt (megabájt) = 2^{10} kbájt = 2^{20} bájt = 1084576 bájt;

1 Gbájt (gigabájt) = 2^{10} Mbájt = 2^{20} kbájt = 2^{30} bájt;

1 Tbájt (terabájt) = 2^{10} Gbájt = 2^{20} Mbájt = 2^{30} kbájt = 2^{40} bájt

Felelj a kérdésekre!


- 1°. Mit értünk bináris kódolás alatt?
- 2°. Mi a bit?
- 3°. Mi a bájt?
- 4°. Mit értünk az üzenet kódjának hossza alatt?
- 5°. Milyen hosszú egy karakter kódja a **Windows-1251** táblában?
- 6°. Mekkora lesz a **Jegyzet**mben létrehozott szöveg bináris kódja, ha az **ANSI** kódban mentettük? És ha **Unicode**ban?
- 7*. Mennyiben fog különbözni annak a **Jegyzet**mben létrehozott fájlnek a tartalma, amely a nevedet tartalmazza, attól a fájlától, amely a neved betűit soronkénti tördelésben tartalmazza?
- 8°. Mi az 1 kbájt, 1 Mbájt, 1 Gbájt és 1 Tbájt?
- 9°. Milyen matematikai műveletet kell végrehajtani a következő átalakítások során:
 - a) biteket bájtokká alakítunk;
 - b) bájtokat kilobájtokká alakítunk;
 - c) megabájtokat kilobájtokká alakítunk?

Végezd el a feladatokat!

- 1°. Hozz fel példát a mindennapi életben alkalmazott bináris kódolásra!
- 2°. Kódold a morzeábécé segítségével az *adat*, *part*, *radar* szavakat! Milyen hosszúságúak lesznek a keletkező kódok?
(Megjegyzés: Használd a morzetáblázatot!)
- 3°. Határozd meg a következő üzenetek bináris kódjának hosszát abból a feltételezésből kiindulva, hogy **Windows-1251** kódlappal történt a kódolás:




- a) a *Windows-1251* kódtábla;
 b) a morzeábécé két szimbólumot használ, a pontot és a vonalat.

 4*. Határozd meg a következő mondat kódjának hosszát, ha azt **Jegyzet**ombben állítottuk elő! Vizsgáld meg a különböző kódolásokat! *Az ember barátok nélkül olyan, mint a fa gyökerek nélkül.*

5*. Töltsd ki az üres helyeket:

- a) 5 bájt = _____ bit;
 b) 4096 bájt = _____ kbájt;
 c) 10 kbájt = _____ bit;
 d) 3 Mbájt = _____ bájt!

 6*. Töltsd ki az üres helyeket:

- a) 2 bájt = _____ bit;
 b) 8192 bájt = _____ kbájt;
 c) 2 kbájt = _____ bit;
 d) 100 Mbájt = _____ kbájt!

7*. A szöveg egy sora 60 karaktert tartalmaz, egy oldalon 40 sor fér el. Hány oldalnyi lehet a szöveg, ha a kód hossza nem haladhatja meg a 8 Mbájtot feltételezve, hogy **Windows-1251** kódtáblát használtunk?

1. SZ. GYAKORLATI MUNKA

Feladatok megoldása szöveges kód hosszának meghatározására

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Határozd meg a szöveg bináris kódjának a hosszát, ha **Windows-1251** kódolást alkalmaztunk, a szöveg 32 oldalas, oldalanként 48 sornyi szöveget tartalmaz, egy sorban átlagosan 56 karakter helyezkedik el. A kód hosszát a legnagyobb, még egész számokkal kifejezhető mértékegységben határozd meg!

2. A **Jegyzet**omb segítségével határozd meg, mennyi az *1. számú gyakorlati munka* üzenet kódjának hossza, ha:

- a) **Windows-1250**;
 b) **Unicode**
 kódolást alkalmaztál!

A fájlokat **gyakorlati_1_Windows.txt** és **gyakorlati_1_unicode.txt** néven mentsd el!



1. fejezet

3. Töltsd ki a táblázatkezelőben a következő táblázatot!

<i>Bit</i>	<i>Bájt</i>	<i>kbájt</i>	<i>Mbájt</i>
614 400			
	524 288		
		256	
			4

A fájlt mentsd el **gyakorlati 1.xlsx** néven!

Ebben a fejezetben megismerkedtek:

a számítógép architektúrájával

az üzenet-feldolgozás történetével

a számítógépes szoftverek osztályozásával és főbb jellemzőivel

az operációs rendszerrel és annak segédprogramjaival

az operációs rendszerek osztályozásával és rendeltetésével

az adatok archiválásával és tömörítésével

a számítógép hardverelemeinek rendeltetésével és főbb jellemzőivel:

a processzorral

a belső és külső memóriával

a monitorokkal

a nyomtatókkal

a multimédiás eszközökkel





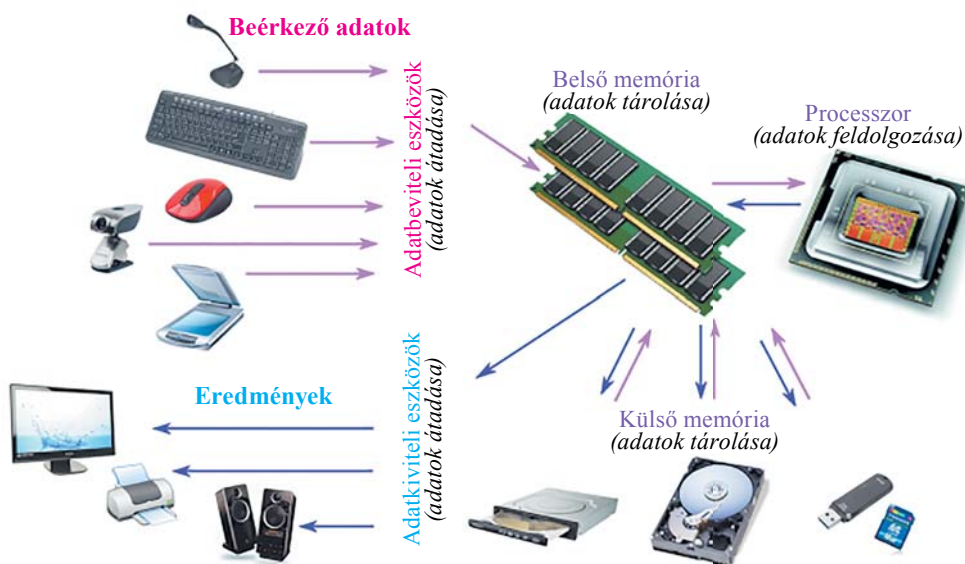
2.1. A SZÁMÍTÓGÉP ARCHITEKTÚRÁJA



1. Mik a számítógép főbb részei? Mi ezek rendeltetése?
2. A számítógép melyik eleme végzi az adatok feldolgozását? Hol található meg ezt az elemet a számítógépben?
3. Nevezd meg néhány információs folyamatot! A számítógép melyik eleme végzi az adatok tárolását?

A SZÁMÍTÓGÉP ARCHITEKTÚRÁJA

Az ötödik osztályban már megismerkedtünk a számítógép főbb részeivel. Most ennél részletesebben megismerkedünk a számítógépben lezajló információs folyamatokkal, és a számítógép részeinek ebben játszott szerepével (2.1. ábra). Az adatok a beviteli eszközökről (billentyűzet, egér, mikrofon, szkennel stb.) vagy a külső memóriából



2.1. ábra. A számítógépben zajló információs folyamatok vázlata

(merevlemez, optikai tároló, pendrive, memóriakártya stb.) kerülnek a belső memóriába. A memóriából az adatok a processzorba kerülnek, innen a feldolgozás után visszajutnak a belső memóriába, majd a külső memóriába mentődnek, vagy a kiviteli eszközökön (monitor, nyomtató, hangfalak stb.) kerülnek megjelenítésre. Mindezen műveletek egy-egy információs folyamattal kapcsolhatók össze, amelyeket a megfelelő szoftverek vezérelnek.

Architektúra – görög szó, jelentése *épület, ötlet, terv.*

A 2.1. ábrán látható vázlat egy átlagos számítógépre vonatkozik. A **számítógép architektúrája** egy modell, amely leírja a gép részeinek és szoftverének együttműködését, amely lehetővé teszi a számítógépben lezajló információs folyamatok helyes végrehajtását.



A PROCESSZOR ÉS ANNAK RENDELTETÉSE

Azt már tudjátok, hogy a számítógépben a processzor (2.2. ábra) végzi az adatok feldolgozását. Ez az eszköz biztosítja a programok futását a számítógépben. A processzor, mint minden algoritmus-végrehajtó, saját utasításkészlettel rendelkezik.

A processzor a számítógép egyik legfontosabb része. Tartalmazza a programok értelmezését és végrehajtását lehetővé tévő **vezérlőegységet**, valamint a matematikai műveletek elvégzéséhez elengedhetetlen **aritmetikai-logikai egységet**. A mai processzorok már saját belső memóriával – **cache-memória** – is rendelkeznek. Ennek rendeltetésére később még visszatérünk.

A processzor legfőbb jellemzője a **műveleti sebesség**, amely az **órajel frekvenciájától**, a **magok számától**, a **sínszélességtől** és a **cache-memória méretétől** függ (2.1. táblázat).



2.2. ábra. Tízmagos Helio X20 processzor

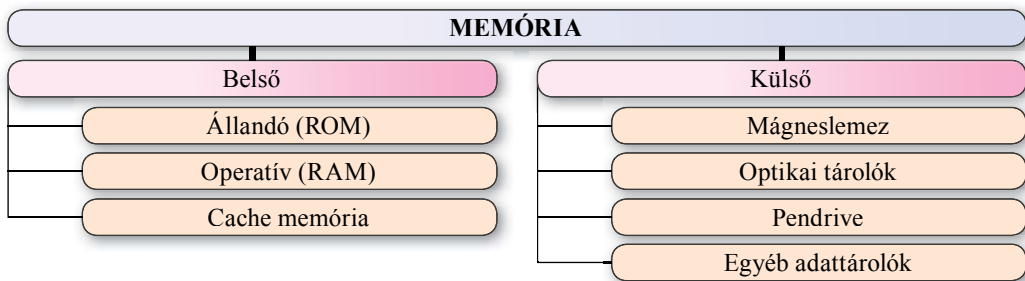
2.1. táblázat

Az asztali gép és a táblagép processzorainak néhány jellemzője

Tulajdonság	Mit jellemez	Jellemző érték	
		asztali gép	táblagép
Modell	A gyártó által adott megnevezés	Intel Core i5-6600	Qualcomm Snapdragon MSM8939
Órajel frekvenciája	A processzor részeit összehangoló vezérlőjelek frekvenciája	3,3 GHz	1,8 GHz
Magok száma	Az egy eszközbe integrált azonos szerkezetű processzorok száma	4	8
Sínszélesség	A processzor által egyidejűleg feldolgozott bináris kód hossza	64 bit	64 bit

A SZÁMÍTÓGÉP MEMÓRIÁJA. BELSŐ MEMÓRIA

A memória rendeltetése az adatok tárolása. A memóriát **belső** és **külső** memóriára oszthatjuk (2.3. ábra). A memória legfőbb jellemzője a kapacitás, amit bájtokban, kilobájtokban, megabájtokban, terabájtokban fejezünk ki.



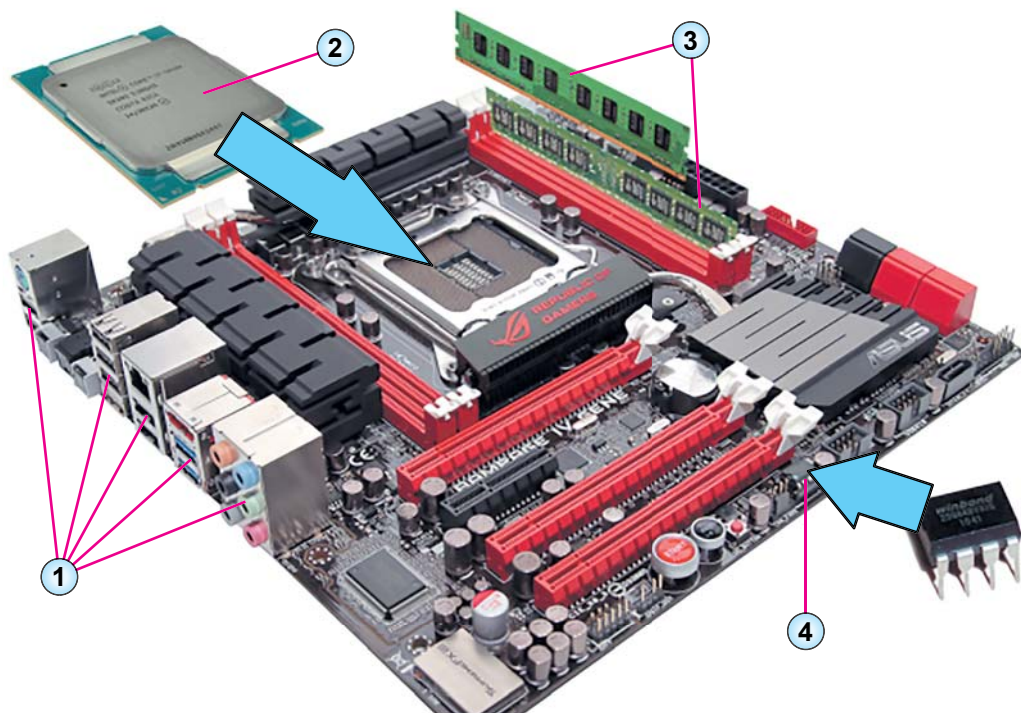
2.3. ábra. A memória típusai

Alaplap (*motherboard, main-board*) – a modern személyi számítógép elengedhetetlen része, biztosítja az adatátvitelt a számítógép részei között.

A belső memória biztosítja a számítógép működését. A processzorhoz hasonlóan az alaplapon helyezkedik el (2.4. ábra).

Az **operatív tár** (RAM) a modern számítógép elengedhetetlen része, itt tárolódnak a processzor által feldolgozandó adatok. A processzor csak az operatív tárból képes utasításokat és adatokat

fogadni. A processzor által feldolgozott adatok ugyanide íródnak vissza. Ahhoz tehát, hogy



- | | |
|--|--------|
| 1. Adatbeviteli és kiviteli eszközök csatlakozói | 3. RAM |
| 2. Processzor | 4. ROM |

2.4. ábra. A számítógép alaplapja

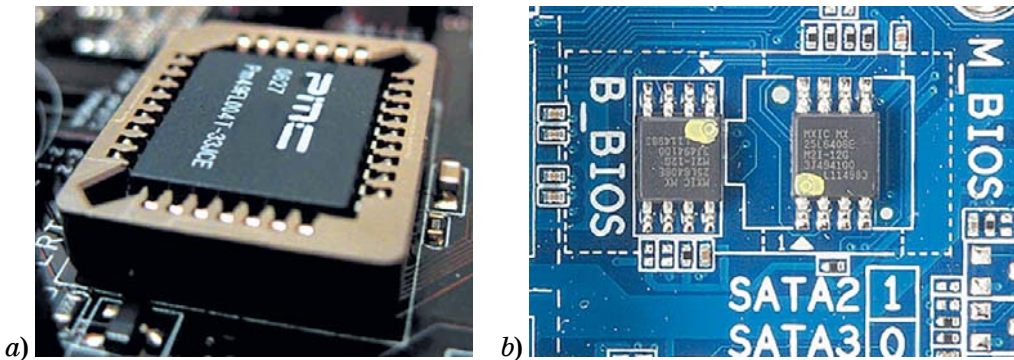


a processzor feldolgozhassa az adatokat, azokat át kell adni (be kell tölteni) az operatív tárba az adatbeviteli eszközökről vagy a külső memóriából.

Az operatív tár modulokba szervezett integrált áramkörökből áll, amelyeket az alaplapon speciális illesztőhelyekre (slot – angolul *rés, nyílás*) építünk be (2.4. ábra). Egy számítógép általában több ilyen modult tartalmaz.

Ahhoz, hogy a számítógépet működésbe hozzassuk, szükség van **állandó memóriára** (ROM) is. A számítógép bekapcsolása után az indításhoz elengedhetetlen programok betöltődnek a RAM-ból a ROM-ba, majd a processzor végrehajtja azokat.

Az állandó memória egy vagy néhány IC-ből áll, amelyeket az alaplapon speciális csatlakozóiba (socket – angolul *foglalat, csatlakozó aljzat*) (2.5.a ábra) vagy közvetlenül az alaplapon (2.5.b ábra) illesztünk.



2.5. ábra. RAM IC az alaplapon

Ahogy már említettük, a mai számítógépeknek része a **cache-memória** is (cache – angolul *rejtekhelyet, titkos raktárt* jelent). Rendeltetése, hogy meggyorsítsa az adatátvitelt a RAM és a processzor között. A cache-nek három szintjét különböztetik meg. Az első szintű cache kapacitása a legtöbb processzornál 128 kb-ot. A második és harmadik szintű cache kapacitása az egyes processzortípusok esetében jelentősen eltérhet egymástól. 2016-ban az asztali gépek esetében a második szintű cache kapacitása 1–8 Mb-ot, a harmadik szintűé pedig 2–20 Mb-ot. A cache kapacitásának növelése általában a processzor sebességének növekedésével jár.

KÜLSŐ TÁROLÓESZKÖZÖK

Ha a belső memória az alaplapon helyezkedik el, akkor a külső tárolóeszközöket különböző csatlakozókkal kötjük az alaplaphoz. A belső tárolóktól (ez alól a ROM kivétel) eltérően a külső tárolóeszközök rendeltetése az adatok tartós tárolása. A számítógép kikapcsolása után a külső tárolók tartalma nem vesz el.



- 1. Lemezfelület
- 2. Elektromágneses író- és olvasófej

2.6. ábra. A merevlemez felépítése

A külső tárolóeszközökhöz soroljuk a merevlemezeket, az optikai tárolóeszközöket, a pendrive-okat stb.

A **merevlemez** egy vagy néhány mágneses anyaggal bevont fémlemezről áll (2.6. ábra). A lemezre egy elektromágneses írófej segítségével íródnak fel az adatok, amely felmágnesezi a lemez egyes részeit. Ugyanez az eszköz olvassa le az adatokat a lemeztől.

A 2.2. táblázatban a mai merevlemezek legfontosabb paramétereit és azok jellemző értékeit foglaltuk össze.

2.2. táblázat

A merevlemez jellemzői

Tulajdonság	Jellemző érték	
	asztali gépek	notebookok
Kapacitás	6 Tbájt	2000 Gbájt
Átmérő	3,5 inch	2,5 inch
Fordulatszám	5400–7200 ford/perc	5400 ford/perc

Az **optikai adattárolókban** az adatok rögzítése és olvasása lézerek segítségével történik. Ezek a lemezek kapacitásukban és újrairthatóságukban különböznek egymástól. A 2.3. táblázatban az optikai lemezek típusait és azok tulajdonságait foglaltuk össze.

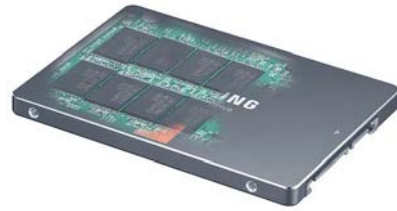
2.3. táblázat

Az optikai lemezek típusai

Rendeltetése			Kapacitás
csak olvasható	egyszer írható	többször írható	
CD-ROM	CD-R	CD-RW	640–800 Mbájt
DVD-ROM	DVD-R, DVD+R	DVD-RW DVD+RW	9,4 Gbájt
BD-ROM	BD-R	BD-RE	100 Gbájt



2.7. ábra. Különböző típusú flash-memóriák



2.8. ábra. SSD-lemez

A szilárdtest vagy **flash-memóriák** népszerűsége az utóbbi években egyre nő. Most már nemcsak mint információ -eszközt vehetjük figyelembe, hanem a merevlemez alternatívájaként, mint elsődleges külső tárolóeszközt is, egyelőre inkább netbookokban, táblagépekben, mobiltelefonokban, fotó- és videokamerákban (2.7. ábra).

A kapacitásuk növekedése oda vezetett, hogy ezeket a memóriákat egyre gyakrabban alkalmazzák a merevlemezek helyett. Ezeket az eszközöket **SSD-lemezeknek** nevezik.

SSD – (*Solid-Statdisk Drive* – szilárdtest (félvezető) alapú meghajtó.

A mai SSD-memóriák (2.8. ábra) kapacitása már meghaladja a 2 Tb-át, ami eléri a merevlemezekét, sebességük azonban már meghaladja a merevlemezek sebességét.

Összefoglalás

A **számítógép architektúrája** az a modell, amely megmutatja, hogyan működnek együtt a számítógép egyes részei a számítógépben lezajló információs folyamatok végrehajtásának céljából.

A **processzor** az az eszköz, amely az adatok feldolgozását végzi. Tartalmaz egy **vezérlőegységet**, amely biztosítja az utasítások értelmezését és végrehajtását, valamint egy **aritmetikai-logikai egységet** a számítások elvégzése céljából. A processzor legfontosabb tulajdonsága a **műveleti sebesség**, amely az **órajel frekvenciájától**, a **magok számától**, a **sínszélességtől** és a **cache kapacitásától** függ.

Az adatok tárolása a memória feladata, amit **belső** és **külső memóriára** osztunk.

A **belső adattárolók** (operatív tár, RAM, cache) az alaplapon helyezkedik el. Ezekből a memóriákból az adatok a számítógép kikapcsolásakor törlődnek, az állandó memóriákból viszont nem.

A **külső adattárolók** (memóriák) közé soroljuk a merevlemezt, az optikai lemezeket, a flash-memóriát. Rendeltetésük az adatok tartós tárolása. Ezekben a tárolókban az adatok a számítógép kikapcsolása után is megmaradnak.

A memória legfontosabb tulajdonsága a kapacitás, amelyet kb-ájtokban, Mb-ájtokban, Gb-ájtokban és Tb-ájtokban stb. fejezhetünk ki.



Felelj a kérdésekre!

- 1°. Mit értünk a számítógép architektúrája alatt?
- 2°. Foglald össze a számítógépben zajló információs folyamatokat a 2.1. ábra alapján!
- 3°. Mi a processzor rendeltetése? Milyen részei vannak a processzornak?
- 4°. Milyen memóriákat találhatunk egy modern számítógépben?
- 5°. Nevezd meg a processzor legfontosabb paramétereit és azok jellemző értékeit!
- 6°. Ismertesd a memóriák típusait! Mi a memória legfontosabb jellemzője?
- 7°. Miben méri a memória kapacitását?
- 8°. Mi a RAM rendeltetése? Hol helyezkedik el a RAM?
- 9°. Mi az állandó memória (ROM) rendeltetése? Miben különbözik ez a RAM-tól?
- 10°. Mi a cache rendeltetése?
- 11°. Nevezd meg, milyen eszközök tartoznak a külső memóriához! Sorold fel ezek legfontosabb paramétereit és azok jellemző értékeit!
- 12°. Miben különbözik egymástól a belső és külső memória?
- 13*. Mi a közös bennük, és miben különböznek egymástól a külső memóriák típusai?



Végezd el a feladatokat!

- 1°. Keresd meg az interneten (például a **hotline.ua** oldalon) a modern számítógépek jellemzőit, válassz ki három olyan processzortípust, amelyek az utóbbi időben kerültek forgalomba Ukrajnában, majd töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
Modell			
Órajel frekvenciája			
Magok száma			
Sínszélesség			



- 2*. Keresd meg az interneten (például a **hotline.ua** oldalon) a modern számítógépek jellemzőit, majd határozd meg, hogy maximálisan mekkora RAM-ot ajánlanak a notebookokhoz! Töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
Modell			
Gyártó			
Kapacitása			
Ár			



- 3*. Sorold fel az általad ismert adathordozókat kapacitásuk csökkenő sorrendjében!
- 4*. Az internet segítségével keresd meg a környezeted számítógép-szaküzleteinek katalógusait, és határozd meg azoknak a processzoroknak a paramétereit, amelyeket ezekben forgalmaznak!
- 5*. Az internet és a szaküzletek katalógusai alapján hasonlítsd össze az azonos magyszámú és órajelű processzorok másodsztintű cache-memóriáinak kapacitárait!
- 6*. Készíts rövid ismertetőt a ROM-ok cseréjének lehetőségeiről!



2.2. ADATBEVITELI ÉS KIVITELI ESZKÖZÖK



1. Milyen adatbeviteli eszközök kapcsolhatók egy számítógéphez? Milyen típusú adatok bevitelére szolgálnak ezek az eszközök?
2. Milyen kiviteli eszközök kapcsolhatók egy számítógéphez? Milyen típusú adatok kivitelére szolgálnak ezek?
3. Milyen beviteli és kiviteli eszközök alkalmasak multimédiás adatok be- és kivitelére? Mondj példákat!

BEVITELI ESZKÖZÖK

Ahogy azt már tudjátok, a számítógépnek vannak az adatok bevitelére, illetve kivitelére szolgáló eszközei.



2.9. ábra. Gamepad



2.10. ábra. Keypad



2.11. ábra. Dancepad

A beviteli eszközöket a bevitt adatok típusa szerint csoportosíthatjuk. Ennek alapján a beviteli eszközök szolgálhatnak szöveges adatok bevitelére (**billentyűzet**), grafikai adatok bevitelére (**szkenner**, **fényképezőgép**, **digitális rajztábla**), audio-adatok bevitelére (**mikrofon**),

video-adatok bevitelére (**videokamera**, **webkamera**, **TV-tuner**). A beviteli eszközök egy másik csoportja különféle programok vezérlésére szolgál. Ezek közé tartozik az **egér**, a **touchpad**, az **elektronikus multimédiás tábla**, az **érintőképernyő**, a **joystick**, a **gamepad** (2.9. ábra), a **keypad** (2.10. ábra), a **kormány**, a **pedálok**, a **dancepad** (2.11. ábra).

Gamepad – játéktér, játékvezérlő eszköz.
Keypad – kiegészítő billentyűzet.
Dancepad – tánctér, táncszőnyeg.

A tudományos célra alkalmazott számítógépekben különleges beviteli eszközöket is találhatunk, például hőérzékelőket, légnedvesség-mérőket, különböző szennyeződések szenzorait, sebességmérőket stb.

Tekintsük át a beviteli eszközök főbb paramétereit és azok jellemző értékeit.

A **billentyűzet** rendeltetése szöveges adatok és utasítások bevitele. Attól függetlenül, hogy a billentyűzetek nem sokat változtak az elmúlt néhány évtizedben, mégis csoportosíthatjuk ezeket néhány szempont szerint:

- rendeltetés szerint: *szabványos* (asztali gépekhez), *kompakt*, *multimédiás billentyűzet*, *játékvezérlőbillentyűzet*;
- a csatlakozó típusa szerint *vezetékes* és *vezeték nélküli* (rádió, *bluetooth*, *wifi*).

A hordozható gépeket vetíthető billentyűzettel is elláthatják (2.12. ábra).

Az **egereket** is több szempont szerint csoportosíthatjuk:

- rendeltetésük szerint lehetnek: *asztali géphez*, *notebookhoz* vagy *játékgéphez* készült egerek;
- a csatlakozó típusa szerint: *vezetékes* és *vezeték nélküli* (rádió, *bluetooth*, *wifi*);
- az érzékelőjük típusa szerint: *optikai* és *lézeres*;
- a gombok száma szerint: *2*, *3–5*, *6–9*, *10* és *több gombos*;
- kialakítás szerint: *szimmetrikus*, *balkezes szimmetrikus*, *ergonomikus*.

A 2.13. ábrán egy vezetéklessé lézeres ergonomikus egeret láthatunk, amelyen 19 gombot és egy görgőt találhatunk.

Gyakorlatilag valamennyi notebook alapfelszerelése a **touchpad** (*touch* – hozzáérni, *pad* – lapocska, párna, terület), de ennek az eszköznek is van asztali géphez kapcsolható változata (2.14. ábra). Ha az ujjunkat húzogatjuk a lapkán, az mozgásba hozza az egérmutatót. Ha egyszer megnyomjuk a lapkát, az a bal gombbal történt egyszeri kattintással egyenértékű, ha kétszer, akkor a dupla kattintással. A touchpad gombjai az eger gombjainak funkcióival bírnak.



2.12. ábra. Vetített billentyűzet



2.13. ábra. Eger



2.14. ábra. Külső touchpad



A **multimédiás digitális táblát** különféle prezentációk bemutatására használják, többnyire oktatási intézményekben. Ezek különböző méretben (70–79, 80–89, 90 feletti *inch*) elérhetők, továbbá az érzékelés módjában is különböznek.

Érintőképernyőket leginkább táblagépekben, okostelefonokban, tájékoztató rendszerekben (2.15. ábra) alkalmaznak. Az adatbevitelt az ujjainkkal vagy speciális tollal végezzük, amelyekkel a képernyő megfelelő részeihez kell hozzáérnünk. A képernyő érzékeli az érintést, és továbbítja a megfelelő információkat a számítógéphez. A legfontosabb tulajdonságuk a képernyőméret (3 és 70 *inch* között), valamint az érzékelés módja.

Az érintőképernyőhöz sokban hasonlít a **digitalizáló tábla**, amelyet rajzolásra, grafikai információk bevitelére használnak. A felhasználó egy speciális tollal hozza létre az ábrát. Előfordul, hogy maga a kép nem jelenik meg a táblán, csak a számítógép képernyőjén. Más esetekben a kép a tábla felületén jelenik meg (2.16. ábra).

A digitalizáló táblák a munkafelületük mérete, felbontóképességük (2000 és 4000 dpi – képpont inchenként – között) szerint különböznek egymástól. Sokféleségük abban is megnyilvánul, milyen nyomásszintet tudnak megkülönböztetni az érintőceruzák (512-től 2048-ig).

A **szkenner** szintén grafikai információk bevitelére szolgál. Megkülönböztetünk *kéziszkennereket*, *síkgyas szkennereket*, *szkennerkamerákat* és *rollszennereket* (2.17. ábra). A szkennernek főbb jellemzői a dokumentum mérete, a felbontóképesség (600 és 6100 dpi–képpont inchenként – között) és a szkennelés sebessége (5–20 másodperc).

A fotó és videokamerákkal, webkamerákkal, mikrofonokkal és egyéb multimédiás eszközökkel már a hatodik osztályban megismerkedtek. Itt jegyezzük meg, hogy a kamerák legfontosabb paramétere a felbontóképesség, ami meghatározza a kép minőségét. **Videokamerák** esetében megkülönböztetjük az **SD** (*Standard Definition*, 720 x 576 képpont), a **HD** (*High Definition*, 1280 x 720 képpont), a **Full HD** (*Full High Definition*, 1920 x 1080 képpont) és **Ultra HD** (*Ultra High Definition*, 3840 x 2160 képpont) felbontóképességet.



2.15. ábra. Vasúti érintőképernyős tájékoztató tábla



2.16. ábra. Digitalizáló tábla



síkgyas



rollszkenner



kameraszkenner

2.17. ábra. Szkennerek

A **fényképezőgépek** esetében a képminőséget az objektív minősége és a képpontok száma határozza meg (10 és 24 megapixel között). A **webkamerák** felbontóképessége 320 x 240 és 2560 x 2048 közé esik.

A **TV-tuner** teszi lehetővé, hogy a számítógépünkkel tévéadásokat is nézhessünk.

KIVITELI ESZKÖZÖK

A kiviteli eszközöket a beviteli eszközökhöz hasonlóan feloszthatjuk a feldolgozott adattípusok szerint. A szöveges és grafikus információk megjelenítésére szolgálnak a **monitor**, **printer**, **plotter (rajzgép)**, a hangadatokat a **fülhallgató** és a **hangfalak** jelenítik meg, a videoadatokat pedig a **projektorok**. Vannak eszközök, amelyek több különböző adattípus megjelenítésére alkalmasak, például a monitor, a multimédiás projektor, megjelenítő fal.

A **monitor** a személyi számítógép legfontosabb megjelenítő eszköze. Működési elvük alapján megkülönböztethetjük a **folyadékkristályos (LCD – Liquid Crystall Display)**, a **plazma**, az **OLED (Organic Light Emitting Diode – szerves LED)**, az **elektronikus tinta (e-ink)** kijelzőket. A legelterjedtebb manapság a folyadékkristályos technológia.

Az **LCD**-monitorok között megkülönböztethető a **TN + film**, a **PLS**, az **IPS** és a **VA** technológiák. A képátló mérete 19 és 30 inch közötti, a felbontás 1280 x 1024-től 5120 x 2880 pixelig terjed, a reakcióidő (ennyi idő alatt változik meg a képpont színe) pedig 1 és 8 ms közötti.

Mielőtt az adatok megjelenhetnének a képernyőn, azokat a számítógép egy speciális eszköze, a videokártya dolgozza fel. A videokártyák lehetnek alaplapra integráltak vagy külsők. Az alaplapra integrált videokártya egy specializált, alaplapba beépített chip. Ezt a megoldást olyan számítógépeknél alkalmazzák, amelyeken nem dolgoznak fel nagy mennyiségű videóinformációt. Amennyiben nagy mennyiségű videót kell feldolgozni,



egy külső, az alaplap egyik bővítőhelyébe illesztett videokártyát célszerű alkalmazni (2.18. ábra). A videokártya egy speciális processzort és kiegészítő RAM-ot (videó-RAM is) tartalmaz. A videokártyákat különböző gyártók állítják elő (például *Radeon R7 370*, *GeForce GTX 950*), különbözhetnek a videomemória kapacitásában (1 és 6 Gb-ot között) és az alaplapi csatlakozójukban.



2.18. ábra. Videokártya

Az információ papír vagy fényképlemez alapú megjelenítésére printereket és plottereket használunk.

A nyomtatók között lehetnek *mátrixnyomtatók*, *lézernyomtatók*, *hőnyomtatók*, *tintasnyomtatók* és *3D-nyomtatók*. A nyomtatók főbb tulajdonságai:

- a nyomtatás mérete;
- színek használata (*színes*, *fekete-fehér*);
- felbontás (*600 x 1200-tól 5760 x 1440 - dpi-ig*);
- nyomtatási sebesség (*1 és 50 oldal/perc között*);
- csatlakozó típusa (*vezetékes*, *vezeték nélküli*).



2.19. ábra. 3D-nyomtató

Az utóbbi években egyre jobban nő a *3D-nyomtatók* népszerűsége, mivel ezek térbeli nyomtatásra képesek (2.19. ábra).

MULTIMÉDIÁS ESZKÖZÖK

Ahogy már említettük, a számítógépes eszközök között ki lehet emelni azokat, amelyek multimédiás adatok feldolgozására alkalmasak. Ezek a **fényképező**, a **digitalizáló tábla**, a **mikrofon**, a **videokamera**, a **webkamera**, a **digitális tábla**, a **hangfal**, a **projektor**, a **videofal** stb. A képfeldolgozó eszközök legfontosabb jellemzője a felbontás, a hangfeldolgozóké pedig az a frekvenciatartomány, amivel az adott eszköz képes dolgozni.

Összefoglalás

Az **adatbeviteli eszközökhöz** tartozik a szövegbevitelre szolgáló **billentyűzet**, a grafikai adatok bevitelére alkalmazható **szkenner**, **fényképező**, **digitalizáló tábla**, a hang digitalizálására szolgáló **mikrofon** és a videók digitalizálását lehetővé tévő **videokamera**, **webkamera**, **Tv-tuner**, valamint a számítógép vezérléséhez szükséges **egér**, **touchpad**, **elektronikus tábla**, **érintőképernyő**, **joystick**, **keypad**, **gamepad**, **kormány**, **pedál**, **dancepad**.

A **kiviteli eszközökhöz** tartoznak a szöveges és grafikai adatokat megjelenítő **monitor** és **printer**, a hang lejátszásához szükséges **hangfalak** és **fülhallgatók**, a videólejátszáshoz pedig a **multimédiás projektor** és **videofal**.

A felsorolt eszközöket különböző szempontok szerint csoportosíthatjuk.

Felelj a kérdésekre!

- 1°. Milyen beviteli eszközöket ismerünk?
- 2°. Milyen kiviteli eszközöket ismerünk?
- 3°. Milyen szempontok szerint csoportosíthatjuk a billentyűzeteket? Hozzával példákat!
- 4°. Milyen típusú egerekkel van felszerelve a számítógépes termékek? Nevezd meg az egerek paramétereit!
- 5°. Milyen típusú szkennerekkel van felszerelve iskolák számítógépes terme?
- 6°. Nevezd meg a monitorok főbb tulajdonságait és azok jellemző értékeit!
- 7°. Ismertesd a monitor és a videokártya rendeltetését! Milyen paraméterekkel rendelkeznek a korszerű videokártyák?
- 8°. Ismertesd a printerek csoportosítását! Milyen nyomtatókkal van felszerelve az iskolák?
- 9°. Véleményed szerint milyen paraméterekkel kell rendelkezni egy otthoni használatra szánt nyomtatónak?
- 10°. Milyen célt szolgálnak a multimédiás felszereléshez tartozó eszközök?
- 11°. Hasonlítsd össze a digitalizáló táblát és a táblagépet! Mire szolgálnak ezek az eszközök?

Végezd el a feladatokat!

- 1°. A tankönyv alapján készítsd el a beviteli eszközök csoportosítását!
- 2°. A tankönyv adatai alapján készíts egy diát, amely bemutatja a beviteli eszközök osztályozását! Mentsd el a munkádat **2.2.2. feladat** néven a mappádba!



- 3*. Internetes források (például a **hotline.ua**) alapján keresd ki három olyan egér jellemzőit, amelyek a közelmúltban kerültek forgalomba! Töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
Modell			
Csatlakozó			
Mozgásérzékelő típusa			
Gombok száma			
Kialakítás			



- 4*. Internetes források (például a **hotline.ua**) alapján keresd ki három olyan monitor jellemzőit, amelyeket Ukrajnában forgalmaznak! Töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
Modell			
Gyártó			
Képtárló, <i>inch</i>			
Típus			
Felbontás, <i>pixel</i>			
Válaszidő, <i>ms</i>			

- 5*. A tankönyv adatai alapján készítsd el a nyomtatók csoportosítását valamely általad ismert program segítségével! Mentsd a dokumentumot **2.2.5. feladat** néven a mappádba!



- 6*. Határozd meg az otthoni számítógépek paramétereit, és töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
Monitor típusa			
Képtárló, <i>inch</i>			
Mátrix típusa			
Felbontás, <i>pixel</i>			
Válaszidő, <i>ms</i>			

- 7*. Internetes adatok vagy a szaküzletek katalógusai alapján határozd meg három különböző gyártótól származó, Ukrajnában forgalmazott nyomtató paramétereit! Töltsd ki a táblázatot!

A táblázat folytatása

Tulajdonság	Értékek		
Modell			
Gyártó			
Működési elv			
Felbontás, <i>pixel</i>			
Nyomatási sebesség, <i>oldal/perc</i>			



- 8*. Készíts rövid ismertetést a **keypadről**! Mutasd be a felépítését, működését és alkalmazását!



- 9*. Készíts rövid ismertetést az egyik legújabb beviteli eszközről, a **dancepadról**! Mutasd be a felépítését, működését és alkalmazását!



2.3. AZ INFORMÁCIÓS FOLYAMATOK MEGVALÓSÍTÁSÁNAK TÖRTÉNETE. A MAI SZÁMÍTÓGÉPEK TÍPUSAI



- Milyen számítógép-típusokat ismertek? Miben különbözik ezek rendeltetése?
- Milyen alkalmazásai vannak a számítógépeknek az emberek különféle tevékenységi területein?
- Nevezd meg olyan ukrán tudósokat, akik jelentősen hozzájárultak a számítástechnika fejlődéséhez!

AZ INFORMÁCIÓS FOLYAMATOK MEGVALÓSÍTÁSÁNAK SZAKASZAI

Az emberiség fejlődése szorosan kapcsolódik az üzenetek továbbításának, feldolgozásának és tárolásának fejlődéséhez. Ez a fejlődés több szakaszra bontható (2.4. táblázat).

2.4. táblázat

Az információs folyamatok megvalósításának fejlődése

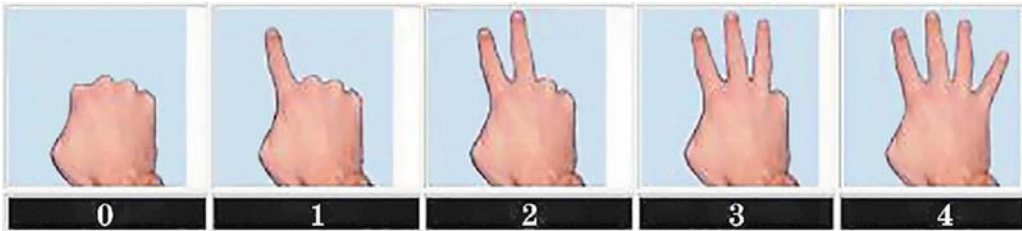
A szakasz neve	Történelmi kor	Az adatok továbbításának, feldolgozásának, tárolásának eszközei
A kézi eszközök kora	Az ókortól a XV. század közepéig	<i>Adathordozók</i> – agyagtáblák, papirusztekercsek, rovásbetűk, képek. <i>Adatok továbbítása</i> – jelzötüzek, dobok, küldöncök, postagalambok, lovasfutárok. <i>Adatok feldolgozása</i> – emberi agy és ujjak, kézi számolóeszközök (abakusz, szorobán stb.).



A 2.4. táblázat folytatása

A szakasz neve	Történelmi kor	Az adatok továbbításának, feldolgozásának, tárolásának eszközei
A mechanikus eszközök kora	A XV. század közepétől a XIX. sz. közepéig	<i>Adathordozók</i> – könyvek, újságok, folyóiratok, fotólemezek. <i>Adatok továbbítása</i> – postaszolgálat. <i>Adatok feldolgozása</i> – nyomdagépek, fényképezők, aritmóméterek, kézi számológépek.
Az elektromos eszközök kora	A XIX. sz. közepétől a XX. sz. 40-es éveikig	<i>Adathordozók</i> – lemezek, filmek, mágnesszalagok. <i>Adatok továbbítása</i> – telefon, táviró, rádió. <i>Adatok feldolgozása</i> – fonográf, gramfon, elektromos írógépek, elektromos aritmóméterek, magnók, nyomdagépek.
Az elektronikus eszközök kora	A XX. sz. 40-es éveitől napjainkig	<i>Adathordozók</i> – mágneses és optikai lemezek, mikroszippek. <i>Adatok továbbítása</i> – televíziós műsorszórás, számítógépes hálózatok, mobilszolgáltatás. <i>Adatok feldolgozása</i> – zsebszámológépek, számítógépek.

AZ ÜZENETFELDOLGOZÁS ESZKÖZEINEK FEJLŐDÉSE



2.20. ábra. Számolás ujjakon

Vizsgáljuk meg részletesebben az üzenetfeldolgozás eszközeinek, jelesül a számolás eszközeinek fejlődését. A számolás első eszközei az emberi ujjak voltak (2.20. ábra). Erről tanúskodnak például a római számok (I, V, X). A régi szláv számlálásban is ujjakat használtak, ezt bizonyítja az öt (pjaty) szó, ami egy kézfejet jelentett.

Jelentős lépés volt a számolás fejlődésében az abakusz feltalálása az i. e. V. században az ókori Görögországban. Az ötletet a görögök a babiloniaktól vették át. A 2.21. ábrán az abakusz egy későbbi változatát, a római abakuszt láthatjuk. Az abakusz tehát egy függőleges bevágásokkal ellátott deszka, amelynek vágataiba apró tárgyakat, jellemzően kőveket helyeztek el.

Hasonló eszközöket más országokban, Kínában, Japánban (**szorobán**), Oroszországban (**szcsoti** – számológép). Ezek az eszközök a XX. század végéig használatban voltak.



2.21. ábra. Abakusz



2.22. ábra. Számológép

Ezekkel az egyszerű és széles körben elterjedt eszközökkel egyidejűleg az ókori Görögországban és Rómában különböző fogaskerekekből összeállított számolóeszközöket is használtak, melyeket egy elsüllyedt ókori hajón talált eszközök tanúsítják. Ezt az eszközt *antiküthérai szerkezetnek* nevezték el, mivel a roncsot Antiküthéra mellett találták a szivacshalászok 1902-ben. A tudósoknak csak a XX. század második felében sikerült megfejteni a szerkezeti elemek rendeltetését és restaurálni azt (2.23. ábra). Az i. e. 100–150 körül létrehozott eszköz naptárként is funkcionált, alkalmas volt az égitestek – Nap, Hold, Merkúr, Vénusz, Mars, Jupiter, Szaturnusz – helyzetének meghatározására.



2.23. ábra. Antiküthérai mechanizmus: a megtalált maradványok és a számítógépes modell

Sajnos, az antiküthérai mechanizmus létrehozásához elengedhetetlen tudás hosszú időre feledésbe merült. A hasonló eszközök létrehozására irányuló próbálkozások csak a XVII. század végén éledtek újra, mivel az ipar, a kereskedelem és a haditechnika fejlődése ekkor ezt egyre erősebben igényelte.

1642-ben **Blaise Pascal** (1623–1662) francia matematikus, fizikus, mérnök a királyi tanács elé terjesztette egy mechanikus számolóeszköz tervét. A későbbiek során több tíz ilyen eszközt (2.24. ábra) készített. Az eszközöket összeadásra és kivonásra használták.

A későbbiekben több matematikus és mérnök fejlesztette tovább Pascal ötleteit. **Gottfried Leibnitz** (1646–1716) német tudós hozta létre az első olyan számológépet, amellyel mind a 4 alpműveletet el lehetett végezni (2.25. ábra).

Ezeket az eszközöket idővel tovább tökéletesítették és a XX. század közepéig használták is, amíg fel nem váltották őket az elektronikus kalkulátorok.



2.24. ábra. Pascal számológépe



2.25. ábra. Leibnitz számológépe

A számológépek tökéletesítése során a tudósok arra törekedtek, hogy a gép képes legyen automatikusan számításokat végezni egy előre kidolgozott program alapján. Az ilyen gép létrehozását először **Charles Babbage** (1792–1871) angol matematikus és tervező (2.26. ábra) tűzte ki céljául. **Analiticalengine**-nek (**analitikus gép**) nevezte el az eszközt.

A tervek szerint Babbage eszköze a következő részekből állt:

1. **Raktár** – mai szóhasználattal **memória** a számok tárolására.
2. **Malom** – a matematikai műveletek végrehajtására, mai nyelven **aritmetikai egység**.
3. A matematikai műveletek sorrendjét (**vezérlő egység**).
4. A kezdőértékeket beállító egység.
5. Kiviteli egység.

Ada Lovelace (2.27. ábra) (1815 – 1852), George Byron angol költő lánya, Babbage társa volt az analitikai gép tervezésében és kivitelezésében. Ő dolgozta ki a programtervezés lépéseit. Ezért Ada Lovelace-t tekintik az első programozónak, tiszteletére nevezték el **Ada**-nak az egyik programozási nyelvet.

A XIX. sz. végéig a számológépek kézi meghajtású mechanikus gépek voltak (2.28. ábra). **Herman Hollerith** (1860–1929) amerikai tudós csak a XIX. század végén fejlesztette ki az első elektromos árammal hajtott eszközt, a **tabulatort** (2.29. ábra). A gépet a népszámlálás adatainak feldolgozásához használták. Az emberek adatait nem papírra írták, hanem egy speciális, lyukakkal ellátott kártyára vitték fel.



2.26. ábra. Charles Babbage



2.27. ábra. Ada Lovelace



2.28. ábra. Müller számológépe



2.29. ábra. Hollerith tabulatora

Hollerith 1896-ban alapította a tabulatorok gyártására szakosodott vállalatát, amelyből néhány tulajdonosváltás és átszervezés után 1924. február 14-én megalakult az **IBM Corporation**. Ma ez a vállalat a mainframe számítógépek egyik legnagyobb gyártója.

A XX. század közepéig sokat javítottak a mechanikus és elektromechanikus gépek szerkezetén, gyorsítottak a műveletek elvégzésén.

A XX. század 30-as – 40-es éveiben a tudósok új típusú, elektromos eszközöket gyártó számítógépeken dolgoztak. 1941-ben Németország-

ban **Konrad Zuse** (1910–1995) létrehozta az elektromágneses reléken működő **Z3**-at. Ez a számítógép már kettes számrendszert használt. Ugyanúgy Zuse fejlesztette ki 1950-ben az elektroncsöveket alkalmazó **Z4**-et.

Ugyancsak 1941-ben az USA-ban **John Vincent Atanasoff** (1903–1995) és **Clifford Barry** (1918–1963) létrehozták az első kettes számrendszert alkalmazó elektroncsöves gépet, az **ABC**-t (*Atanasoff-Berry Computer*). Valamivel később (1939–1944), ugyancsak az USA-ban **Howard Aiken** (1900–1973) az IBM-nel együttműködve létrehozta az automatikus vezérlésű elektromechanikus gépet, a **Mark-1**-et.



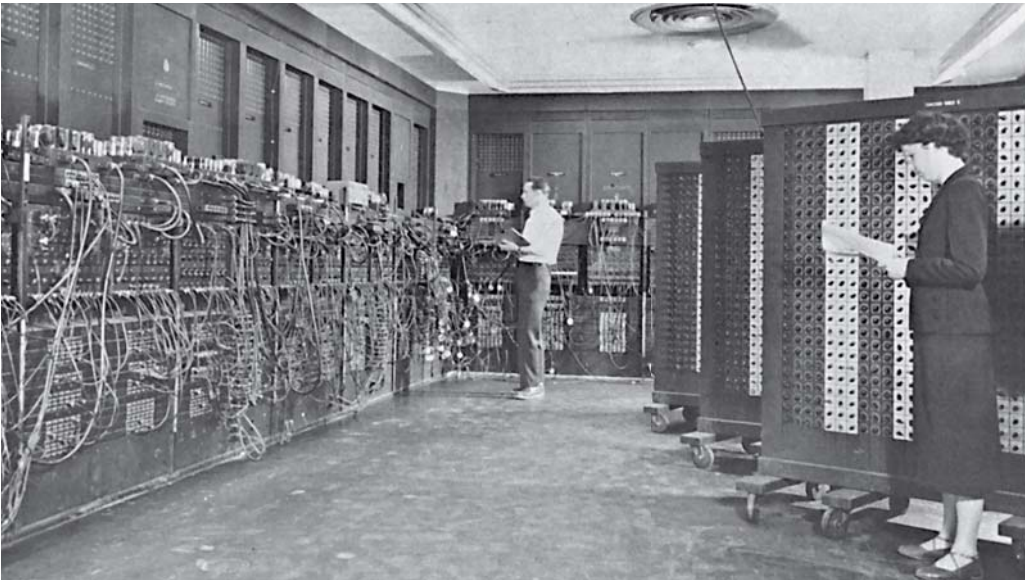
2.30. ábra. Konrad Zuse

Brit tudósok 1943–1944-ben a legnagyobb titokban a német rádiótávíratok rejtjelezésének megfejtésére létrehozták a **Colossust**. Ez abban különbözött a többi számítógéptől, hogy a vezérlőprogramját a memóriájában tárolta.

1943 és 1946 között az USA-ban **John Mauchly** (1907–1980) és **Presper Eckert** (1919–1995) megalkották az **ENIAC**-ot (*Electronic Numerical Integrator and Calculator*), a 18 000 elektroncsövből álló 30 t súlyú számítógépet, amely ötezer műveletet volt képes elvégezni másodpercenként¹ (2.31. ábra). A projekt befejezése után

a fejlesztők azonnal hozzáálltak az **EDVAC** (*Electronic Discrete Variable Automatic Computer*) katonai célú számítógép létrehozásához. A munkába bekapcsolódott **Neumann János** (1903–1957) magyar származású matematikus, aki a munka kapcsán megírta a *First*

¹Csak összeadást, kivonást, a többi műveletet sokkal lassabban végezte (*a ford. megjegyzése*).



2.31. ábra. ENIAC számítógép

Draft of a Report on the Edvac című munkáját, amelyben lefektette az univerzális számítógép működésének elveit. Bár a munkában Eckert és Mauchly is részt vettek, ezeket az elveket a világ *Neumann-elvekként* ismerte meg.

A számítógépek a nagy mennyiségű és viszonylag olcsó **minigépek** megjelenésével kezdtek elterjedni, melyeket később személyi számítógépeknek neveztek el. Kezdetben ezeket alkatrészekből álló készletekként forgalmazták, amelyeket a felhasználónak kellett összeszerelni és a tévékészülékeket használták monitorként. Az első személyi számítógépek a MITS (*Micro Instrumentation and Telemetry Systems*) által gyártott **Altair 8800** (1975), az **Apple Computer Company** által gyártott **Apple** (1976) és az **IBM** által készített **IBM PC** (1981) voltak. A számítógépek elterjedése során egyre több feladat megoldására lettek ezek a gépek alkalmasak.

A számítástechnika elterjedésének történeti kronológiáját a 3. mellékletben találjátok meg az allinf.at.ua oldalon a nyolcadikos tankönyvhöz tartozó résznél.



2.32. ábra. Altair 8800



2.33. ábra. Apple I



2.34. ábra. IBM PC

A SZÁMÍTÁSTECHNIKA FEJLŐDÉSE UKRAJNÁBAN

1951-ben Kijevben üzembe helyezték a Szerhij Olekszijovics Lebegyev (1902–1973) vezetésével létrehozott első olyan számítógépet – **MEOM** (kis elektronikus számítógép) –, amely a programot a memóriájában tárolta. Ez volt az első ilyen számítógép az akkori Szovjetunióban. Paraméterei a következők voltak:

- az elektroncsövek száma 6000 körül;
- kettes számrendszert használt;
- sínszélesség – 16 számjegy és egy előjelbit;
- memória – 31 szám és 63 utasítás;
- műveleti sebesség – 3000 művelet percenként.

A számítógép első programját **K. L. Juscsenko** (1919–2001) írta.

Kijevben a számítógépek fejlesztését a kibernetikai kutatóintézetben folytatták

Pionír (az angol *pioneer* – első teleges, kedvezményező, kutató) olyan ember, aki egy ismeretlen tevékenységi területen elsőként tapossa ki az utat.

V. M. Hluskov (1923–1982) vezetésével.

Itt hozták létre a **Kijev** számítógépet (1959, fejlesztők V. B. Hnedenko, L. M. Dasevszkij, K. L. Juscsenko), a **Dnyipro** számítógép-sorozatot (1961, a fejlesztés vezetője B. M. Malinovszkij), a mérnöki számítások elvégzésére szánt **Prominy** számítógépet, a mérnöki számítások elvégzésére tervezett **MIR** sorozatot a 60-as években és számos katonai célú számítógépet. Az IEEE (Computer Society) nemzetközi egyesület V. M. Hluskovot 1996-ban *A számítástechnika pionírja* éremmel (2.35. ábra) tüntette ki.



2.35. ábra. A számítástechnika pionírja érem



2.36. ábra. Zsibszámológép

A számítástechnika rakétatechnikai alkalmazásainak fejlesztésében Ukrajna több városának fejlesztői vettek részt. A szeverodonecki *Impilsz* tervezőiroda munkatársai fejlesztették ki az M6000–M7000 automatikus termelésvezérlési rendszereket, az *Iszkra* billentyűzetes számítógépeket, a ballisztikus rakéták számítógépes vezérlőrendszereit. A Szovjetunióban elsőként Svitlovodszkban gyártották a kijevi *Krisztal* tervezőiroda által fejlesztett **Elektronika** zsebszámológépeket (2.36. ábra).



A 70–90-es években a Harkovi *Hartron* által fejlesztett rakétavezérlési rendszereket a kijevi rádiógyárban állították elő.

A 70–80-as években a kijevi Rádiótechnikai Intézetben fejlesztették és a *Burevisznik* gyárban állították elő a *Karat* hajó- és tengeralattjáró vezérlési rendszert.

1975-ben az Ukrán TA kibernetikai intézetében M. M. Amoszov vezetésével fejlesztették ki az első szállító robotot, a TAIR-t (2.37. ábra), amely képes volt a természeti akadályok kikerülésére.

Ukrán tudósok és mérnökök a 60–80-as években egy sor számítógépet terveztek különböző rakétarendszerek irányítására. A világ legnagyobb ballisztikus rakétájának, a dnyipropetrovszki déli gépgyárban előállított P-36M2 *Szatanának* a harkivi *Elektroprilad* és a kijevi rádiógyár munkatársai készítették a fellövést és repülést vezérlő számítógépeket (2.38. ábra).

Ma ezt a rakétát alkalmazzák a *Dnyipro* nemzetközi program kereteiben felbocsátott műholdak pályára állítása során.



2.37. ábra. A tervezők a TAIR robot beállításán dolgoznak



2.38. ábra. A P-36M2 fellövése

A MODERN SZÁMÍTÓGÉPEK TÍPUSAI ÉS ALKALMAZÁSA

Már tudjátok, hogy manapság a leginkább elterjedt az úgynevezett **személyi számítógép**, amelyek között **asztali gépeket** és **mobil számítógépeket** különböztetünk meg. Az asztali gépek között megkülönböztetjük az **irodai**, az **otthoni** és a **játékra tervezett** gépeket. A mobil gépek közé soroljuk a **notebookokat**, **netbookokat**, **táblagépeket**, **okostelefonokat**.

A nagy számítási igényű összetett feladatokat nagy teljesítményű **szuperszámítógépek** segítségével oldják meg. 2015-ben



2.39. ábra. A Tianhe-2 szuperszámítógép



a legnagyobb számítási teljesítménnyel a kínai Haditechnológiai Egyetem **Tianhe-2** (2.39. ábra) gépe rendelkezett Guangzhouban, amelybe *3 millió 120 ezer* Intel Xeon mag van beépítve.

Manapság a számítógépet széles körben alkalmazzák a tudományban, az ipari termelésben, a hivatalokban, a bankrendszerben, az oktatásban, a gyógyításban, a közlekedésben, a kommunikációban, a mezőgazdaságban, a szociális ellátó rendszerben és a gazdaság más területein.

A tudományban a számítógépet többek között számítógépes kísérletek elvégzésére használják. A tudományos kísérletek egy része meglehetősen sok anyagi, műszaki, energetikai problémát vet fel. Ha például az atommagok hasadása során lejátszódó folyamatokat szeretnénk tanulmányozni, magreaktorokra van szükség, amelyek megépítése költséges, bonyolult, veszélyes és sokáig tart. Egyes esetekben a kísérleti feltételek biztosítása teljesen lehetetlen, nem tudunk változtatni például a távoli galaxisokban lejárló folyamatokon. Ilyen esetekben azonban lehetséges a jelenség **informatikai modelljének** tanulmányozása, amennyiben rendelkezésünkre állnak a megfelelő számítógépek és szoftvertermékek.

Fontos szerepet játszik a számítógép a termelésben. A különböző termékek modellezése és számítógépes tervezése csökkenti a tervezés idejét, javítja a minőséget és a termelés hatékonyságát, csökkenti az árakat. A számítógép alkalmazása előtt egy új autó tervezésétől a gyártásig 5–6 év telt el, ami manapság 1 év alá csökkent.

A szolgáltatások területén a számítógépeket különféle adatok – szövegek, táblázatok, adatbázisok, képek, fotók, multimédiás adatok – tárolására és feldolgozására használják. Nehéz lenne ma már olyan munkahelyet találni, ahol ne használnának számítógépeket. A könyvtárak és kórházak kartotékjait adatbázisok váltották ki, amelyek kezelése sokkal kényelmesebb a hagyományos kartotékoknál. A könyvelő a számítógépe segítségével néhány perc alatt elvégzi azt a munkát, amire régebben órákra vagy napokra volt szükség. A bankár anélkül követheti a tőzsdei árfolyamokat, hogy elmozdulna az asztala mellől és néhány egérgattintással intézi az átutalásokat.

A számítástechnika főbb alkalmazási területeit így foglalhatjuk össze:

- nagy mennyiségű és nagy pontosságú számítások elvégzése;
- számítógépes modellezés és számítógépes kísérletek;
- automatizált vezérlőrendszerek működtetése;
- nagy mennyiségű adat tárolása és feldolgozása;
- gyors információcsere;
- ipari, háztartási és katonai eszközök irányítása;
- a tanulás támogatása;
- diákok, szakemberek távoktatásának megszervezése, a mozgáskorlátozott emberek tanulásának támogatása.



A SZÁMÍTÓGÉP IGÉNY SZERINTI KONFIGURÁLÁSA

A számítógép konfigurációját rendeltetésének megfelelően kell kialakítani. A következő paraméterekre kell figyelmet fordítani:

- **a processzor számítási teljesítménye**, amit az órajel frekvenciája, a magok száma, a második és harmadik szintű cache mérete határoz meg. Minél nagyobbak ezek az értékek, annál nagyobb a processzor teljesítménye;
- **a RAM kapacitása** – minél nagyobb, annál nagyobb a gép teljesítőképessége;
- **a merevlemez kapacitása** – attól függ, mekkora adatmennyiséget kell a felhasználónak feldolgozni, audio- és videofeldolgozáshoz lényegesen több szükséges, mint szövegszerkesztéshez;
- **külső videokártya** – erősen befolyásolja a videók feldolgozásának sebességét. A videó RAM mérete és a grafikus processzor teljesítménye a döntő fontosságú két tényező;
- **a monitor képminősége**, ami a méretétől, a felbontásától és a válaszidejétől függ.

A számítógép kiválasztására nyilván egyéb tulajdonságok is befolyással vannak, például optikai meghajtó szükségessége, az alaplap sajátosságai, a hálózati tápellátás nélküli üzemidő (mobil eszközöknél), csatlakoztathatóság különböző hálózatokhoz. A konfiguráció kialakítására természetesen annak az ára is lényegesen kihat.

Az internetáruházak nagy része lehetővé teszi a számítógép architektúrájának kialakítását intelligens szűrők alkalmazása által. A **Hotline** (<http://hotline.ua/computer>) rendeltetéstől függően a következő csoportokat különbözteti meg: *kezdő számítógép, munkagép, munkaállomás, optimális alapkonfiguráció, univerzális optimális konfiguráció, haladó optimális konfiguráció, nagy teljesítményű játékgép*. A **Rozetka** (<http://rozetka.com.ua/computers-notebooks>) három csoportot különböztet meg: *kezdő, munka és tanulás, játék*.

A rendszeregységek árai 2,5 ezertől 120 ezer hrivnya között változik. A 2.5. táblázatban néhány, a fenti csoportosításnak megfelelő gép paramétereit foglaltuk össze.

2.5. táblázat

Különböző rendeltetésű számítógépek paramétereit

Tulajdonság	Érték		
	Irodai	Otthoni	Játék
Típus	asztali	asztali	asztali
Processzor	Intel Celeron Dual-Core E3400	AMD FX-8320	Intel Core –i7 4790K
Órajel frekvenciája, GHz	2,6	3,5	4,0

Tulajdonság	Érték		
	Irodai	Otthoni	Játék
RAM, GB	2	8	16
Merevlemez, Gb	250	1000	1000
Videókártya	integrált Intel HD Graphics	AMD Radeon R9 270X	Nvidia GeForce GTX 980 Ti
VideoRam, Gb	nincs	2	6
Hangkártya	integrált	CreativeX-Fi Xtreme Audio PCI Express	integrált HD Audio 7.1
Monitor	LG 19M45A	Philips 233V5LSB	ASUS VX238H
Képtároló, inch	18,5	23	23
Felbontás	1366 × 768	1920 × 1080	1920 × 1080
Válaszidő, ms	5	4	1

! Összefoglalás

Az első számolóeszközöket több ezer évvel ezelőtt hozta létre az emberiség. A számítástechnika fejlesztésében jelentős szerepet játszottak B. Pascal, G. Leibnitz, Ch. Babbage, A. Lovelace, H. Hollerith. Az első elektronikus számítógépeket K. Zuse (**Z4**), J. Atanasoff és C. Barry (**ABC**), H. Aiken (**Mark-1**), **Colossus** angol tudóscsoport, J. Mauchly és P. Eckert (**ENIAC**) valamint a Sz. Lebegyev vezette szovjet tudóscsoport (**MEOM**) hozták létre a XX. század 40–50-es éveiben.

A Szovjetunióban Ukrajnában hozták létre az első számítógépeket. A számítástechnika gazdasági, vezetési és katonai célú hasznosításába Ukrajnában több tíz tervező és kivitelező vállalat volt bevonva. Vezető szerepe az Ukrán TA V. Hluskov vezette kibernetikai kutatóintézetének volt.

A számítógépet rendeltetésük és számítási teljesítményük alapján **szuperszámítógépekre** és **személyi számítógépekre** oszthatjuk. Az utóbbiak között megkülönböztetünk **asztali** és **mobil** számítógépeket. Az asztali gépek között vannak **irodai**, **otthoni** és **játék-gépek**. A mobil eszközök közé a **notebookok**, **netbookok**, **táblagépek** és **okostelefonok** tartoznak.

**Felelj a kérdésekre!**

- 1°. Nevezd meg a számítástechnika fejlődésének korszakait!
- 2°. Foglald össze, hogy változtak az adatok feldolgozásának eszközei a különböző korokban!
- 3°. Ismertesd, mi újat jelentett Pascal munkássága az ember által alkotott számolóeszközök fejlődésében!
- 4°. Miben különbözik Leibnitz számológépe a Pascalétól?
- 5°. Milyen részei voltak a Babbage által tervezett univerzális számítógépnek?
- 6°. Kit tekinthetünk a világ első programozójának, és miért?
- 7°. Ismertesd a számítástechnika területén a XX. század 40–50-es évei során végbement fejlődést! Milyen tudósok játszottak kiemelkedő szerepet ebben?
- 8°. Nevezd meg az első személyi számítógépeket! Kik, és mikor hozták létre ezeket?
- 9°. Mutasd be a számítástechnika fejlődését Ukrajnában!
- 10*. Magyarázd el, milyen tulajdonságokban különböznek egymástól a mai számítógépek! Mutasd be a számítógépek osztályozásának szempontjait!
- 11°. Milyen paraméterek határozzák meg a számítógép kiválasztását egy adott feladat céljaira?

**Végezd el a feladatokat!**

- 1°. Készítsd el a számítógépek osztályozásának vázlatát egy tetszőleges, általad ismert szoftver segítségével!
- 2°. Az internet vagy nyomtatott sajtótermékek alapján keress anyagokat az első számítógépekről, és töltsd ki a táblázatot!

<i>Tulajdonság</i>	<i>Értékek</i>		
	<i>Z4</i>	<i>Colossus Mark2</i>	<i>ABC</i>
Fejlesztők			
Elektroncsövek száma			
Kettes számrendszer alkalmazása			
A program tárolása a memóriában			



- 3*. Vesd össze az első elektroncsöves számítógépek, az amerikai **ENIAC** és az ukrainai **MEOM** tulajdonságait! Az összehasonlítás eredményét jegyezd le a füzetedbe!



- 4*. Készíts rövid beszámolót arról, milyen fejlesztéseket hajtottak végre az ukrán tudósok az atomjégtörők vezérlése céljából!

- 5°. Állíts össze egy házi multimédiás központnak alkalmas számítógép-konfigurációt! Válaszaidat indokold meg!



2. SZ. GYAKORLATI MUNKA Számítógép-konfiguráció kialakítása igény szerint

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Állíts össze számítógépes konfigurációt a következő szakmákban dolgozók számára:
 - egy írónak, aki a dolgozószobájában írja a műveit, végzi az elektronikus levelezését, és anyagokat keres az interneten;
 - egy diáknak, aki a feladatait készíti a számítógépen, létrehozza és karbantartja a saját fonotékáját, oktatófilmeket hoz létre és szerkeszti azokat;
 - egy internetes kiadvány tudósítójának, aki a világ legkülönbözőbb helyéről rendszeresen küldi tudósításait!
2. Az előbbi feladat alapján töltsd ki a következő táblázatot!

Tulajdonság	Érték		
	író számítógépe	diák számítógépe	tudósító számítógépe
Típus			
Processzor			
Órajel frekvenciája, GHz			
RAM kapacitás, Gb			
Merevlemez kapacitása, Gb			
Videokártya típusa			
Video RAM, Gb			
Hangkártya típusa			
Monitor			
Képpátló, inch			
Felbontás			
Válaszidő, ms			

3. Készíts a szövegszerkesztőben egy fájlt, amelyben megindokolod a választásaidat!
4. A fájlt küldd el elektronikus levélben a tanárodnak!



2.4. A SZÁMÍTÓGÉPES SZOFTVEREK CSOPORTOSÍTÁSA ÉS ÁLTALÁNOS JELLEMZŐI

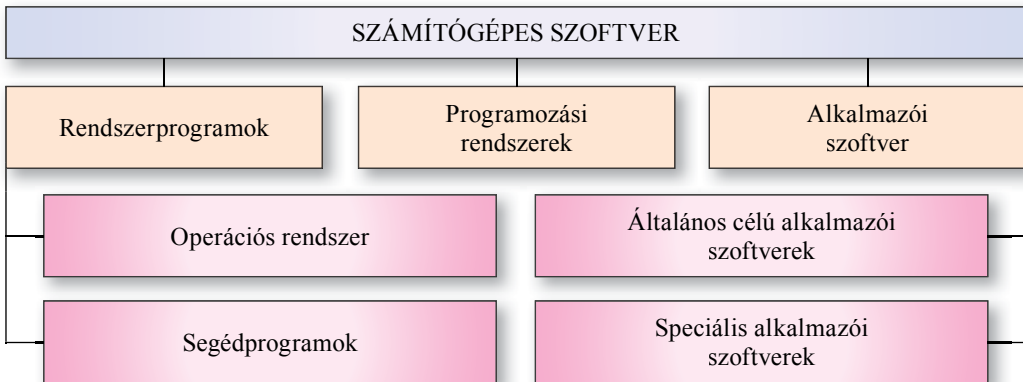


1. Milyen számítógépes programokat ismertek? Mi ezek rendeltetése?
2. Mi az operációs rendszer? Mi a rendeltetése?
3. Ismertesd a programok indításának módjait!

A SZÁMÍTÓGÉPES SZOFTVEREK CSOPORTOSÍTÁSA ÉS ÁLTALÁNOS JELLEMZŐI

Már jó néhány számítógépes szoftvert megismertetek, például a szövegszerkesztőt, a képszerkesztőt, a prezentációkészítőt, a printer, a szkennert, a monitor vezérlőprogramjait stb. A számítógépes programok összességét **számítógépes szoftvernek** nevezzük. A számítógép csak a megfelelő szoftver segítségével képes az adatok feldolgozását elvégezni.

Manapság a számítógépes szoftvereket a következő csoportokba sorolhatjuk (2.40. ábra).



2.40. ábra. A szoftverek csoportosítása

A **rendszerprogramok** a számítógép egyes részeinek vezérlését és az azok közötti információcserét végzik, kijavítják a számítógép működésében keletkező hibákat, automatizálják az adatfeldolgozást, vezérlik a felhasználó és a számítógép közötti kommunikációt. A rendszerprogramokhoz az operációs rendszer és a segédprogramok tartoznak.

A **programozási rendszerek** új programok létrehozását teszik lehetővé, valamennyi alapját egy programozási nyelv alkotja, ilyenek a **Scratch 1.4**, a **Free Pascal 2.6**, a **DEV-C++ 5.11**, a **Microsoft Visual Studio 2013 Professional**, az **Android Studio 1.4.0**, a **Lazarus 1.4.4**, a **Python 2.6.1** stb.

Az **alkalmazói szoftverek** valamilyen konkrét feladat megoldására szolgálnak, amelyekkel a felhasználók a mindennapi életük során találkozhatnak. Ezeket általános célú és speciális szoftverekre oszthatjuk.

Az *általános célú szoftverek* közé azok a programok tartoznak, amelyeket különféle emberi tevékenységi területeken egyaránt használni lehet. Ilyenek a szövegszerkesztők, a képszerkesztők, a multimédia-szerkesztők, elektronikus táblázatkezelők, prezentáció-készítők stb.

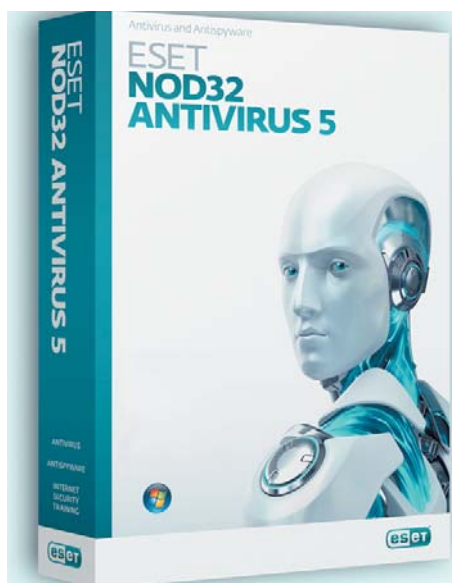
A *speciális szoftvereket* egy konkrét tevékenységi területen, szervezetben, cégben lehet alkalmazni. Ilyenek a speciális videoeffektusok szerkesztői, amelyeket a mozifilmek gyártásánál használnak, a tervezőirodákban használatos CAD programok, az orvosi diagnosztikai programok, az iskolai órarendkészítő programok stb.

A SZOFTVEREK JOGI STÁTUSZA

A számítógépes szoftvertermékekre is kiterjed Ukrajna szerzői jogi törvénye, ezért ezek használata csak a törvény által támasztott követelmények teljesítése, valamint a megfelelő felhasználói- és licencszerződések megléte mellett lehetséges.

A szoftvereknek néhány különböző jogállása lehet:

- **kereskedelmi szoftver**, amelynek alkalmazása akkor jogos, ha a felhasználó megvásárolta a megfelelő mennyiségű licencet. Ezek között megkülönböztetjük a:
 - *dobozos verziót*, amely a szoftver egy példányát tartalmazza valamilyen hordozón, továbbá a megfelelő telepítési utasításokat és telepítőkulcsokat (2.41. ábra);
 - *OEM (Original Equipment Manufacturer – Eredeti felszerelés gyártó) verziót*, amikor a szoftver egy példánya a megfelelő hardvereszközzel kerül forgalomba és csak azon képes működni. Erről tanúskodik egy notebookon látható hologramos matrica (2.42. ábra);
 - *vállalati licenc*, amely a szoftver több példányának használatára jogosít, például egy iskola valamennyi számítógépén;
- **próba (trial) licenc** – kereskedelmi szoftverek ideiglenes használatára (általában 1 hónap) jogosít fel. Előfordulhat, hogy a kipróbálható szoftver nem bír teljes funkcionalitással.



2.41. ábra. A vírusirtó dobozos verziója



2.42. ábra. OEM szoftver hologramos matricája



- **szabad felhasználású (freeware)** licenc, ami a program ingyenes használatát biztosítja, ugyanakkor a programkód megváltoztatására nem jogosít fel;
- **a szabad szoftver (FSF, free software)** biztosítja a szoftver ingyenes használatának és megváltoztatásának jogát.

AZ OPERÁCIÓS RENDSZEREK CSOPORTOSÍTÁSA ÉS FŐBB FELADATAI

A hatodik osztályban már tanultatok, hogy sokféle operációs rendszer létezik. Megkülönböztetünk **asztali gépeken futtatható (Windows 10, OS X)**, **mobil számítógépeken futtatható (Windows Mobile 6.0, iOS 9.0, Android 6.0)**, valamint **szervereken és szuperszámítógépeken futtatható (Unix, Linux, Windows Server 2012, Solaris 11)** operációs rendszereket.

Az operációs rendszereket feloszthatjuk a licenc típusa alapján is. A **Windows** operációs rendszerek például a szabadon nem terjeszthető kereskedelmi szoftverek, amelyek egyes verzióihoz tartozhat *trial*-licenc is. Vannak **free software** licenc alatt terjesztett operációs rendszerek, például a **Linux** különböző változatai.

Az operációs rendszer feladata a számítógép részeinek vezérlése, az adatok tárolása a RAM-ban és a külső hordozókon, a programok futtatása, a számítógép és a felhasználó közötti kommunikáció biztosítása.

SZOFTVEREK TELEPÍTÉSE ÉS ELTÁVOLÍTÁSA

Ahhoz, hogy a számítógépünkön legyen operációs rendszer, el kell végezni annak **telepítését**.

Az installációhoz rendelkezniünk kell az operációs rendszer **telepítő-készletével** egy optikai adathordozón vagy egy flashmemórián. Ha a számítógépet erről az adathordozóról indítjuk, elindul az operációs rendszer telepítője, a felhasználó engedélyét kéri ahhoz, hogy megkezdje a telepítést, majd egy sor kiegészítő kérdést tesz fel annak tisztázására, mi legyen a telepítés helye, milyen a telepítés nyelve, milyen dátum- és időbeállítást alkalmazzon, milyenek legyenek a regionális beállítások stb. Az **Ubuntu Linux** telepítését a **Munkaasztalon** látható parancsikon vagy az **Indítópanel gombja** segítségével kezdeményezhetjük (2.43. ábra). Léteznek a telepítésnek a vázolttól eltérő módjai is, például a helyi hálózatról vagy az internetről.

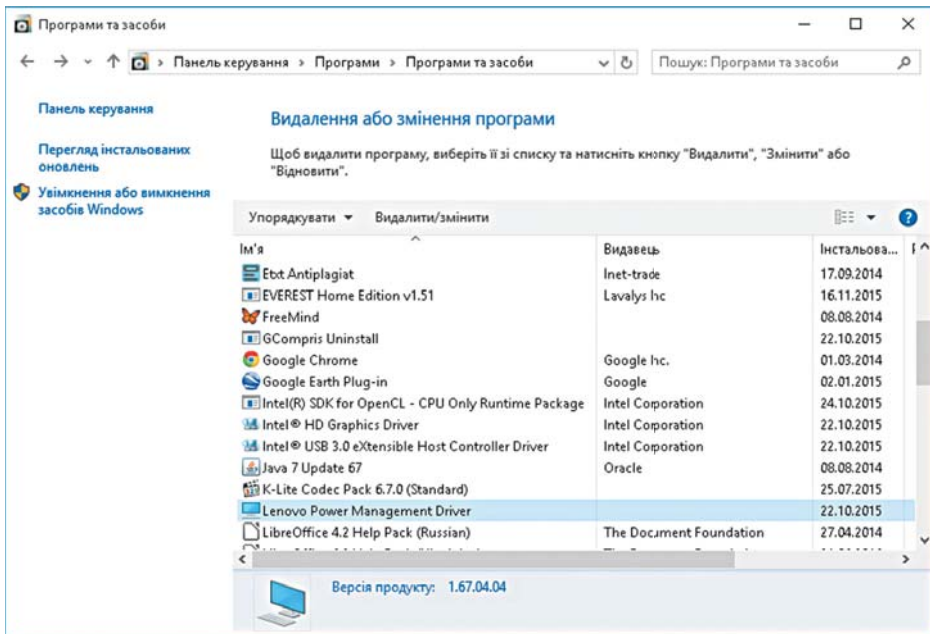


2.43. ábra. Az **Ubuntu Linux** munkaasztala a telepítők parancsikonjaival

A telepítés során az operációs rendszer komponensei felmásolódnak a kiválasztott lemezre, összehangolódnak a számítógép hardvereszközei. Azt a lemezt, ahová az operációs rendszert telepítettük, **rendszerlemeznek** nevezzük.

Minden, a számítógépen működő szoftver az operációs rendszer irányítása alatt végzi a feladatait. Minden szoftvert egy adott operációs rendszerhez terveznek, ezért az csak annak vezérlése alatt képes működni. Egy program indítása előtt végre kell hajtani az adott program és az operációs rendszer összehangolását. Ezt a folyamatot is **telepítésnek** nevezzük. A telepítést speciális programok végzik, ezek leggyakrabban *setup.exe*, *install.exe* nevet viselik (**Windows**) vagy *INSTALL.sh*, illetve *deb* kiterjesztést (**Linux**).

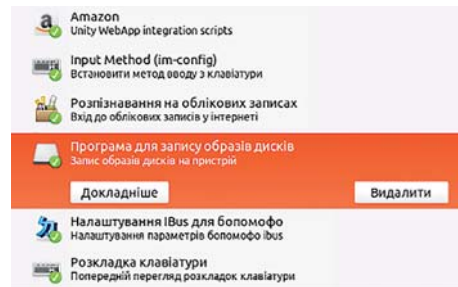
A telepítés elindítása után már csak követni kell a program utasításait. Sikeres telepítés után a program indítása a **Start** menüből (általában a **Programok** részből) lesz lehetséges és a program bekerül a telepített programok listájába (**Windows**); vagy a **Start** panelből történik és a **Szoftverközpontból** lesz lehetséges (**Ubuntu Linux**).



2.44. ábra. Programok és eszközök ablak


Amennyiben a programot el szeretnénk távolítani, használjuk a:

- **Windows** esetében a **Start** ⇒ **Minden program** ⇒ **Rendszer** ⇒ **Windows** ⇒ **Vezérlőpult** ⇒ **Windows 10 programok** vagy **Start** ⇒ **Vezérlőpult** ⇒ **Windows 7 programok** parancsot, válasszuk ki az eltávolítandó programot, majd kattintsunk a **Törlés/Módosítás** gombra (2.45. ábra);



2.45. ábra. Szoftverközpont részlete



- **Ubuntu Linux** esetében használjuk az *Indítópanel* ⇒ *Szoftverközpont*  parancsot, válasszuk ki a törlendő programot, és kattintsunk a **Törlés** gombra. (2.45. ábra).
Ezzel a művelettel a program valamennyi komponensét eltávolíthatjuk a számítógépről.

SZOFTVERKOMPATIBILITÁSI PROBLÉMÁK

A számítógépes programokat egy adott operációs rendszer alá készítik azoknak a szabványoknak megfelelően, amit az operációs rendszer fejlesztői ehhez megadnak. Egy adott típusú operációs rendszer alá írt programok általában összeegyeztethetetlenek egy másik operációs rendszerrel, nem is lehet telepíteni ezeket a másik rendszer alá. A **Windows** alá írt programok tehát **Linux** rendszeren nem működnek. Ahhoz, hogy a program működőképes legyen, meg kell azt változtatni a megfelelő operációs rendszer szabványainak megfelelően. Ezért létezik például **Chrome** operációs rendszer **Windows**, **Linux**, **Android** rendszer alá (2.46. ábra).

Увага:

У вас є можливість скачати Google Chrome для linux, скачати Google Chrome для mac, скачати Google Chrome для android, а також скачати Google Chrome для ios (посилання для завантаження приведено нижче). Скачати Google Chrome

Ліцензія:	Freeware
Автор:	Google
Оновлено:	4-02-2015, 18:23
Платформа:	    
Українська мова:	
Російська мова:	

2.46. ábra. A **Google Chrome** böngésző weblapjának részlete

Megesik, hogy egy adott program nem működik helyesen egy operációs rendszer család mindegyik tagján. A **Windows 10** alá írt programok nem futnak **Windows XP** rendszeren.

SEGÉDPROGRAMOK

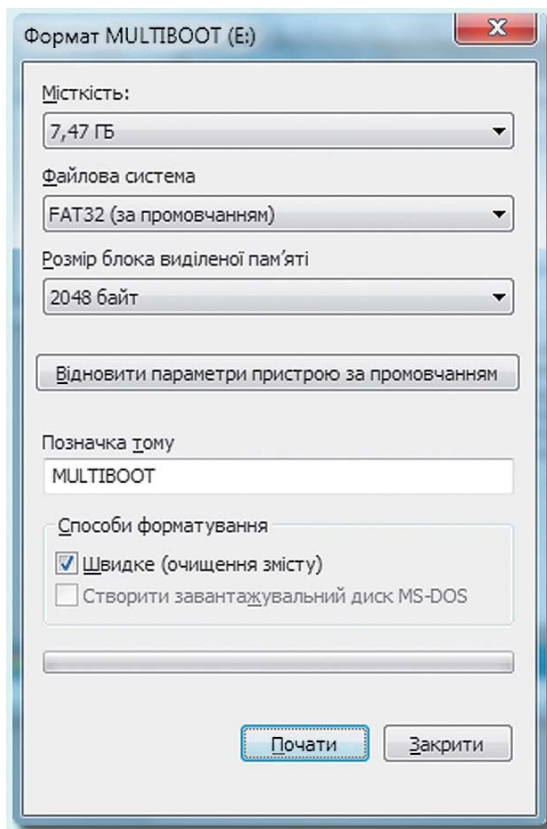
A **segédprogramok** rendeltetése a számítógép hardver és szoftverelemeinek tesztelése, a rendszer lehetőségeinek kibővítése. Szükség esetén ilyen programok végezhetik a rendszer optimalizálását. Ezeket a programokat **segédprogramoknak** (*utility*) nevezik. A segédprogramok egy része az operációs rendszerhez tartozik és a rendszer telepítése során telepítésre is kerül. Ilyenek a **rendszer-visszaállítás**, a **töredezettség-mentés**, a **lemezkarbantartás**, **lemezkezelés**, az **adatforrás-figyelő**, a **rendszerinformáció**, a **feladattervezés**, a **vezérlőpult programok**.

Hasonló rendeltetésű programok természetesen **Linux** rendszerben is léteznek. Az **Ubuntu Linux** alatt például ilyen az **archívum manager**, a **rendszerapló**, a **lemezfelhasználás-elemző**, a **rendszermonitor**, a **system testing**, a **lemez**ek stb.

Lehetőség van arra is, hogy a fejlesztők weboldalairól további segédprogramokat töltsünk le és installáljunk.

LEMEZFORMÁZÁS

A segédprogramok között találunk olyanokat, amelyek a külső tárolóeszközöket szolgálják ki. Ilyenek a particionálást, formázást, töredezettség-mentesítést, törlést végző programok.



2.47. ábra. A **Formázás** program ablaka

Particionálás után minden logikai lemezt **formázni** kell. A formázás azt jelenti, hogy a lemezt **cellákra** osztjuk fel, ezeket **szektorok**-nak nevezzük. Minden szektorban maximum 512 bájt adatmennyiséget tárolhatunk. Nagy kapacitású lemezekben a bizonyos számú szektorokat **klaszterekbe** egyesítenek. A klaszter néhány szektort egyesít, ha például négyet, akkor a klaszterméret $4 \times 512 = 2048$ bájt lesz.

A formázás során létrejön még a fájlok elhelyezkedését mutató táblázat is.

Formázásra külső adathordozók – például flashmemóriák – esetében is szükség van. Ezt a következőképpen tehetjük meg:

Az ilyen programok alkalmazását az adatok tartós és biztonságos tárolásának, gyors és pontos hozzáféréseinek igénye teszi szükségessé.

Már tudjátok, hogy az adatokhoz akkor férhetünk hozzá gyorsan, ha azokat rendszerezve tároljuk. Ennek biztosítása érdekében a merevlemezeket felhasználás előtt particionálni – **logikai lemezekre** bontani – kell. A logikai lemezek számát a felhasználó adja meg. A merevlemez kapacitásától függően 1–3 logikai lemezt használunk. Ezeket a lemezeket az operációs rendszer külön objektumokként kezeli. Az első logikai lemeze telepítjük az operációs rendszert, a többi a felhasználó adatainak tárolására használjuk.

Particionálás után minden logikai lemezt **formázni** kell. A formázás azt jelenti, hogy a lemezt **cellákra** osztjuk fel, ezeket **szektorok**-nak nevezzük. Minden szektorban



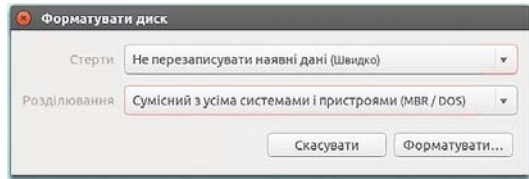
1. A **Start** ⇒ **Számítógép** parancssal nyissuk meg az **Intézőt!**
2. Nyissuk meg a flashmemória helyi menüjét!
3. Válasszuk ki a **Formázás** parancsot!
4. A megnyíló ablakban (2.47. ábra) állítsuk be:
 - a fájlrendszer típusát a **Fájlrendszer** legördülő menüben (ha a memóriát több különböző számítógéphez kívánjuk csatlakoztatni, ajánlott az alapértelmezett FAT32);
 - a klaszterméretet a **Lemezfogalási egység mérete** mezőben;
 - az eszköz nevét a **Kötetcímke** mezőben;
 - a formázás módját a **Gyorsformázás** jelölő bekapcsolásával!
5. Kattintsunk az **Indítás** gombra!



Linux-használóknak

Ubuntu Linux alatt a következőképpen végezzük a formázást:

1. Nyissuk meg a **Lemezek (Indítópanel) ⇒ Programok ⇒ Lemezek** ablakot!
2. Válasszuk ki a hordozót az ablak bal oldali részében!
3. Kattintsunk a **További lehetőségek** gombra !
4. Válasszuk ki a **Formázást!**
5. Állítsuk be a formázás paramétereit a **Lemez formázása** ablakban (lásd az ábrán):
 - a formázás módját (gyors vagy sem);
 - a fájlrendszer típusát!
6. Kattintsunk a **Formázás** gombra!
7. Erősítsük meg a formázási szándékunkat a megnyíló párbeszédablakban!

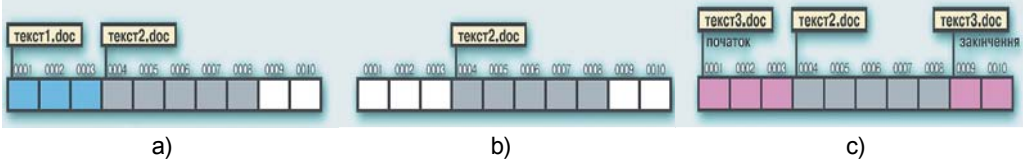


Ha többet szeretnél tudni

A mai operációs rendszerek automatikusan ellenőrzik a külső memóriák állapotát. Ez fokozottan igaz a személyi számítógépek merevlemezei esetében. Gyakran van szükség a lemez fizikai állapotának és logikai szerkezetének ellenőrzésére. A felhasználók ezt szabványos segédprogramok segítségével oldhatják meg, ha a lemezegység helyi menüjéből kiválasztják a **Tulajdonságok** pontot, majd az **Eszközök** fület választják, ott pedig az **Ellenőrzés** gombra kattintanak.

Ha itt bekapcsolják a **Fájlrendszer hibáinak automatikus javítását**, akkor az ellenőrzés során megtörténik a fájlok és mappák hibáinak automatikus javítása is. Ellenkező esetben az ellenőrzés csak informálni fog a hibákról.

A lemez alaposabb ellenőrzése céljából kapcsoljuk be a **Szektorhibák keresése és kísérlet javításukra** jelölőt. Ilyenkor a rendszer kísérletet tesz a lemez fizikai hibáinak javítására, ezért a folyamat sokáig eltarthat.



2.48. ábra. Fájlok mentésének és törlésének folyamata

A számítógép használata során a fájlok a lemez szabad klasztereibe íródnak be, ezért azok töredezetten lesznek jelen a lemezen. Ha csak felírnánk a fájlokat a lemeze, azok az egymás utáni klaszterekben helyezkednének el. De mivel a fájlokat töröljük és módosítjuk is, a szabad klaszterek soha nem egymás után találhatóak a lemezen. Ez a **lemez töredezettségéhez** vezet. Tegyük fel, hogy a lemezünkre írtuk a három klasztert elfoglaló **text1.doc** fájlt, majd az 5 klaszternyi **text2.doc**-ot (2.48.a ábra). Idővel a **text1**-et töröltük, ezáltal felszabadítottuk az első 3 klasztert (2.48.b ábra), majd felírtuk a **text3.doc**-ot, amely 5 klasztert foglal el. Ez a fájl már két részre töredezetten kerül fel a lemeze: az első 3 klaszter a lemez elejére (a törölt első fájl helyére), a többi a végére kerül (2.48.c ábra). Minél gyakrabban hajtunk végre törlési és felírási műveleteket, annál töredezettebb lesz a lemezünk.

Hogy azt a situációt elkerüljük, időről időre **töredezettség-mentesíteni** kell a lemezt. A folyamat során a rendszer megpróbálja egymás utáni szektorokba elhelyezni a fájlokat. Indítása az **Eszközök** fülön a **Töredezettség-mentesítés** gombbal történik.

Összefoglalás

A számítógépes programok összessége alkotja a **számítógépes szoftvert**. A szoftveren belül megkülönböztetünk *rendszer-szoftvert*, *programozási rendszereket* és *alkalmazói szoftvert*.

A szoftvereket különböző licenceknek megfelelően forgalmazzák: *kereskedelmi szoftver* (dobozos és OEM), *próba* (trial), *ingyenes* (freeware) és *szabad szoftver* (fsf, free software).

Megkülönböztetünk *asztali számítógépekre*, *mobil számítógépekre* és *szerverekre* vagy *szuperszámítógépekre tervezett operációs rendszereket*. Az operációs rendszerek is különböző *licencek keretein belül kerülnek forgalmazásra*.

A programok a **telepítési eljárás** során kerülnek a számítógépünkre. Egy adott operációs rendszer alá kifejlesztett programok nem fognak működni más operációs rendszereken, nem is installálhatjuk arra.

A **segédprogramok** (utility) olyan programok, amelyek a hardverelemek tesztelésére, karbantartására és az operációs rendszer lehetőségeinek kiterjesztésére szolgálnak.

Az adathordozókat a formázás során készítjük elő arra, hogy rá adatokat írassunk, illetve visszaolvassunk. Formázáskor a lemezen 512 bájtos szektorok jönnek létre, amelyeket



klaszterekbe egyesítenek. A formázás során létrejön a fájlok elhelyezését leíró struktúra (FAT), ennek alapján ismeri fel a rendszer, hogy egy adott fájl mely klaszterekben került tárolásra.



Felelj a kérdésekre!

- 1°. Mi a szoftver?
- 2°. Ismertesd a szoftverek csoportosítását!
- 3°. Mik a rendszerprogramok? Milyen programok tartoznak ebbe a csoportba?
- 4°. Mik tartoznak a programozási rendszerekhez? Mondj példákat!
- 5°. Milyen általad ismert programok tartoznak az alkalmazói szoftverhez? Ezek közül melyek általános célúak?
- 6°. Milyen programok tartoznak a speciális célú alkalmazói szoftverek körébe? Miben különböznek ezek más programoktól? Mondj példákat!
- 7°. Milyen licenckel forgalmazzák a programokat? Miben különböznek az egyes licenccelési módok?
- 8°. Ismertesd a merevlemez előkészítésének folyamatát!
- 9°. Mi a klaszter? Milyen lehet egy klaszter mérete?
- 10°. Ismertesd a lemez formázásának folyamatát!



Végezd el a feladatokat!



- 1°. Ismertesd a szoftverek licenccelési módjait!
- 2°. A lemez egy klasztere 1024 bájt. A lemezre egy 2750 bájtos és egy 324 kbájtos fájl irtak fel. Hány klasztert foglalnak el ezek a fájlok?

3°. Indítsd el a **Lemezkarbantartót** (*Start* ⇒ *Minden program* ⇒ *Kellékek* ⇒ *Rendszereszközök* ⇒ *Lemezkarbantartó*)! Határozd meg, mit jelent a lemez karbantartása! Milyen objektumcsoportokban történik a fájlok keresése, milyen fájlokat találhat a rendszer fölöslegesnek? Mekkora helyet ajánl a program felszabadításra?



- 4°. Végezd el a tanárod által kijelölt hordozó formázását! A formázás során alkalmazd az alábbi beállításokat:
 - fájlrendszer – maradjon a rendszer által felajánlott;
 - klaszterméret – 512 bájt;
 - kötetcímké – *a neved*;
 - formázás módja – *gyorsformázás*!
- 5*. Figyeld meg, hogy a fájlok mérete különbözik azok helyfoglalásától. Vizsgáld meg a tanárad által megadott fájlokat, és töltsd ki a táblázatot!



Fájl neve	Fájl mérete	Helyfoglalás	Klaszterméret



- 6*. Készítsd el az operációs rendszerek csoportosítását a tankönyv alapján! Internetes források alapján egészítsd ki a csoportosítást egy újabb szempont szerint!



2.5. ADATOK TÖMÖRÍTÉSE ÉS ARCHIVÁLÁSA



1. Milyen számítógépes programokat tudsz már használni? Mi ezek rendeltetése?
2. Mik a segédprogramok, és mi ezek rendeltetése?
3. Mi az archívum? Milyen célokra használják?

ADATTÖMÖRÍTÉS, TÖMÖRÍTÉSI MÓDOK

Gyakran előfordul, hogy szükség van a számítógépen tárolt fájlok méretének csökkentésére. Erre speciális módszereket, **tömörítési algoritmusokat** fejlesztettek ki. Vannak olyan fájlformátumok – például a **TIFF**, **JPEG**, **PNG** képfájlok, **MPEG3**, **WMA** hangfájlok –, amelyek eleve tömörítettek.

Az adatok tömörítése az adatok olyan kódolását jelenti, melynek eredményeképpen csökken a fájlok mérete.

Vannak tömörítési eljárások, amelyek **nem járnak adatvesztéssel** és olyanok is, amelyek megengedik a **részleges adatvesztést**.

Veszteséges tömörítést akkor alkalmazunk, ha az adatok integritása nem feltétlenül fontos szempont. A kép-, hang- és videofájlok tömörítetlenül olyan információkat tartalmaznak, amelyeket az emberi érzékszervek nem képesek érzékelni. Ilyen esetekben a minőségbeli romlás nem annyira lényeges, ha a fájlokat egy optikai adathordozóra szeretnénk felírni, vagy az interneten elhelyezni. Ugyanakkor szövegek és számok tömörítésére a veszteséges eljárások alkalmatlanok.



Ha többet szeretnél tudni

Tömörítési módok

A **Huffman** és **Shannon-Fano algoritmusok** egyaránt adatok kódolására alkalmasak.

Az eljárások lényege, hogy a gyakrabban előforduló karakterekhez rövidebb, a ritkábban előforduló kódokhoz pedig hosszabb kódot rendelünk. Az előző mondat hossza

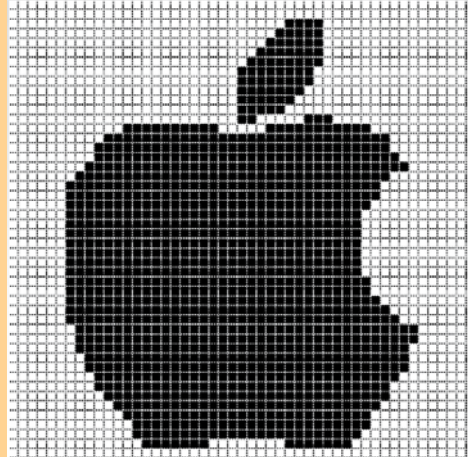


például a szóközökkel együtt 139 betű, ami **Windows-1250** kódban 139 bájt. A mondatban 10 a, 4 z, 12 e, 7 l, 1 j, 3 á, 9 r, 3 s, 8 o, 10 k, 1 é, 5 n, 3 y, 4 g, 4 h, 8 b, 2 ő, 2 f, 7 d, 2 u, 4 ó, 3 t, 1 ö, 1 v, 3 i, és 16 szóköz van. Ha a szóközt egyetlen bittel – legyen az értéke 0 – jelöljük, az a-t pedig 1-essel, a következő két leggyakoribb jelhez pedig kétbites kódot – a k 00, az r pedig 01 lehetne. E mentén a logika mentén a még ritkábban előforduló karakterekhez 4 bites vagy hosszabb kódot rendelhetünk, ekkor a mondat kódja 100 bit körülire lesz, ami valamivel több, mint 12 bájt. A mondat kódja tehát körülbelül tizedére csökkent.

A fent ismertetett módszer eléggé le van egyszerűsítve és ne felejtjük el, hogy a tömörített fájlba el kell helyezni a hozzá alkalmazott kódtáblát és egyéb, „szolgálati” információkat is. Ezért a módszer alkalmazása során a kis fájlok mérete nem csökken, ellenkezőleg, még meg is nő.

Ha azonban a fájlban nagy mennyiségű szöveget és számokat tartalmazó információt tárolunk, akkor a módszer elég jó tömörítési arányt ad.

Az **RLE (Run-Lenght Encoding, azaz ismétlődések kódolása)** inkább képi információk tömörítésére alkalmas. A módszer lényege, hogy ismétlődő hasonló adatsorozatokhoz rendel kódot. A 2.49. ábrán az *Apple* logóját láthatjuk fekete-fehérben. Ha a fekete színhez 1-est, a fehérhez pedig nullát rendelünk, az első sorban 48 darab nullát kellene leírni (48 bit). Ezt azonban lecserélhetjük egy rövidebb beírással: 0 48. A mennyiséget leírhatjuk 9 biten, egy bit pedig kell a 0-ra, a többi pedig az ismétlések számára. A második sor kódja ugyanez lesz, a harmadiké 0 29 1 4 0 15 (összesen 23 bit). A módszer nagy egyszínű területeket tartalmazó színes képek esetében is jó tömörítést ad.



2.49. ábra. Az Apple logója

ADATOK ARCHIVÁLÁSA

Bár a számítógépek és az adathordozók megbízhatósága sokat javult, adataink még sincsenek teljes biztonságban. Az adatvesztésnek súlyos következményei lehetnek. Az ügyfelek adatainak sérülése vagy elvesztése egy bank számára megengedhetetlen. Ha sérülnek a jegyeladási információk, problémák jelentkeznek az utasok szállításában. A kísérleti eredmények elvesztése sokévi kutatómunkát tehet tönkre. De ha csak a barátaink telefonszámait veszítjük el, az is sok kellemetlenséget tud okozni. Ez pedig azt jelenti, hogy adatainkról másolatokat kell létrehozunk. A legfontosabb adatainkat másik merevlemezre vagy optikai tárolókra mentjük. A másolatok elkészítése során célszerű az adatokat tömöríteni. Az ilyen másolatokat nevezzük **archívumoknak**, a benne tárolt fájlokat pedig **archivált fájloknak**.

Az adatokról tömörített másolatokat speciális programok, **tömörítők** segítségével készítjük.

Az operációs rendszerek többsége rendelkezik tömörítőprogrammal. Ezek a programok készítik el az operációs rendszer visszaállításához szükséges információkat, de adattömöritésre is használhatóak.

A fájlok archiválását a következőképpen végezzük:

1. Nyissuk meg a biztonsági másolat készítésére szolgáló segédprogramot (**Start** ⇒ **Minden program** ⇒ **Kellékek** ⇒ **Rendszerezszközök** ⇒ **Biztonsági másolat**)!

Для яких елементів створювати резервні копії?

Установіть прапорці поруч із елементами, які потрібно включати до резервних копій.



2.50. ábra. Biztonsági másolat beállításai ablak

2. Kattintsunk a **Biztonsági másolat beállításai** linkre (első indításkor vagy a **Paraméterek megváltoztatására**)!
3. Állítsuk be, melyik meghajtóra készüljön el a másolat!
4. Adjuk meg azokat a mappákat és fájlokat, amelyekről biztonsági másolatot szeretnénk készíteni (2.50. ábra)!
5. Állítsuk be a biztonsági másolat készítésének időzítését!
6. Indítsuk el a biztonsági másolat elkészítését!

Az újabb operációs rendszerekben lehetőségünk van úgynevezett **rendszerkép** készítésére is, amelyek a felhasználó fájljain kívül az operációs rendszer beállításait és a telepített programokat is tartalmazzák. Ha egy ilyen rendszerképről állítjuk vissza az adatokat, nem lesz szükség a rendszer és a programok újbóli telepítésére.

Windows 7 operációs rendszerben a következőképpen hozzuk létre a rendszerképet:

1. Válasszuk a **Start** ⇒ **Vezérlőpult** ⇒ **Rendszer és biztonság** ⇒ **Biztonsági másolat és helyreállítás** parancsot!
2. Az ablak bal oldalában válasszuk a **Rendszerkép létrehozása** lehetőséget!



- Adjuk meg, mely meghajtóra készüljön a rendszerkép, amennyiben nem az alapértelmezett rendszermeghajtót választjuk!
- Kattintsunk a **Rendszerkép létrehozásának megkezdése** gombra!

A művelet eredményeképpen létrejön a kiválasztott meghajtón egy **iso** kiterjesztésű fájl. A beállított időpontokban a rendszer **automatikusan frissíti majd az archívum tartalmát**: hozzáadja a felhasználó által megadott mappák tartalmát, és frissíti a megváltozott régebbi tartalmakat is.

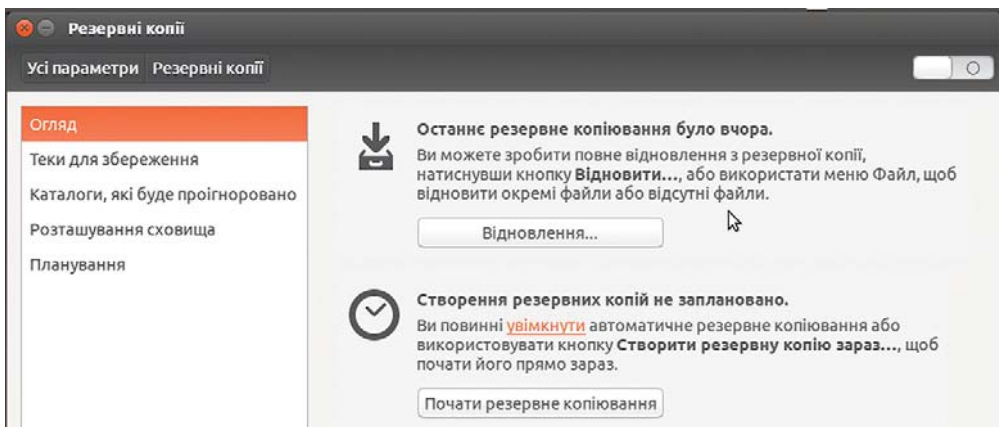
A rendszerképről a **rendszerkép készítése és visszaállítása** ablakban válasszuk ki a visszaállítás lehetőséget, majd adjuk meg, hogy a **saját adatokat, más felhasználók adatait** szeretnénk-e visszaállítani, de **választhatunk más rendszerképet is**. Az első esetben csak az aktív felhasználó adatai kerülnek visszaállításra, a másikban pedig valamennyi felhasználóé. A harmadik esetben meg kell adni a másik rendszerkép helyét.



Linux-használóknak

A rendszerparaméterek között (**Indítópanel** ⇒ **Rendszerbeállítás**) a **Rendszer** csoportban keressük meg a **Biztonsági másolat** programot (2.51. ábra). A program ablakában beállíthatjuk, mely mappákról szeretnénk másolatot készíteni, hol helyezzük el az archívumot, elvégezhetjük az egyéb beállításokat. A **Visszaállítás** gombra kattintva visszaállíthatjuk az adatainkat a biztonsági másolatokból.

Ubuntu Linux alatt az optikai meghajtókat kezelő **Brasero** vagy **K3b** programokat is használhatjuk rendszerkép készítésére. Ezek a programok általában a rendszerrel együtt kerülnek telepítésre, de szükség esetén letölthetők az **Ubuntu** letöltőközpont oldaláról.



2.51. ábra. A Biztonsági másolatot készítő program ablaka

ARCHIVÁTOROK ÉS ARCHÍVUM-TÍPUSOK

Az operációs rendszer eszközein kívül léteznek más tömörítőprogramok is. Az archívumokat kezelő programokat **archivátoroknak** nevezzük.

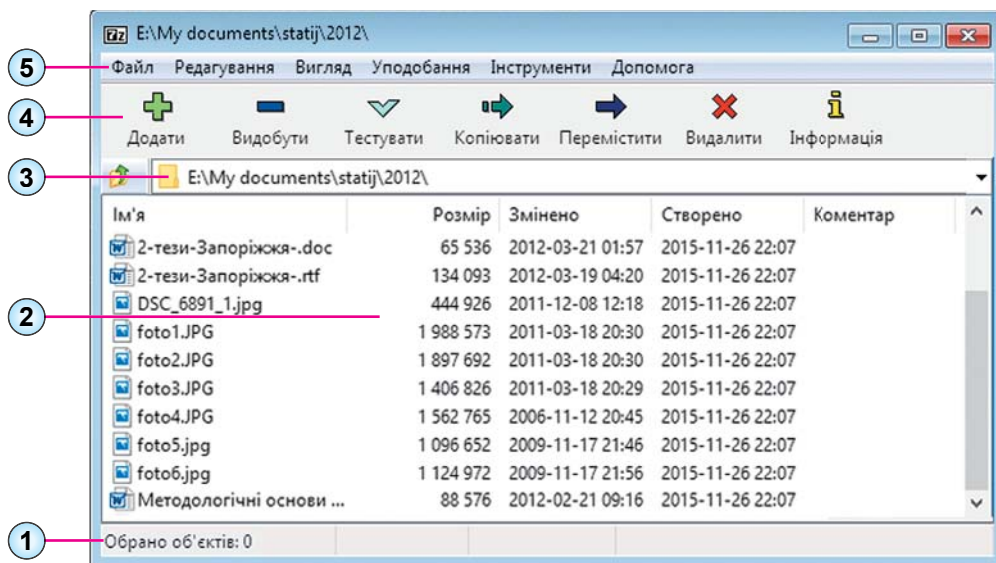
Ilyen programok a **WinZIP**, a **WinRAR**, a **7-zip**, a **PowerArchiver**, a **WinAce**, az **Ark**, a **Btar**, az **AndroZip**, a **FreeArc** stb. A program működése eredményeképpen *archivált fájl* jön létre, amely tömörített állapotban tartalmazza a fájlokat. A tömörítés során lehetőségünk van jelszóval védett tömörített állományok létrehozására is.

A tömörítési eljárás függvényében a tömörített állomány kiterjesztése **ZIP**, **RAR**, **TER**, **ARJ**, **CAB**, **LZH**, **ACE**, **7z** stb. lehet. Leggyakrabban, elsősorban a világhálón, **ZIP**-pel archivált fájlokat használnak.

ARCHÍVUMOKKAL VÉGEZHETŐ MŰVELETEK

Archívumokkal a következő műveleteket végezhetjük:

- archívum létrehozása, fájlok és mappák archívumba helyezése;
- új mappák és fájlok hozzáadása az archívumhoz;
- archívum tartalmának megtekintése;
- fájlok és mappák cseréje és frissítése az archívumban;
- fájlok és mappák kitömörítése az archívumból;
- az archívum teljes tartalmának kitömörítése;
- többkötetes archívum (több tömörített állomány keletkezik) létrehozása; a kötetméretet a felhasználó adhatja meg;
- egy- vagy többkötetes önkítő (SFX–*SelFeXtracting*) archívumok létrehozása; ezek kitömörítéséhez nem szükséges tömörítőprogram;
- archívum integritásának ellenőrzése;
- archivált adatok és fájlnevek kódolása.



- | | |
|-----------------|--------------|
| 1. Статусзор | 4. Eszköztár |
| 2. Munkaterület | 5. Menüsor |
| 3. Címsor | |



2.52. ábra. A 7-zip ablaka

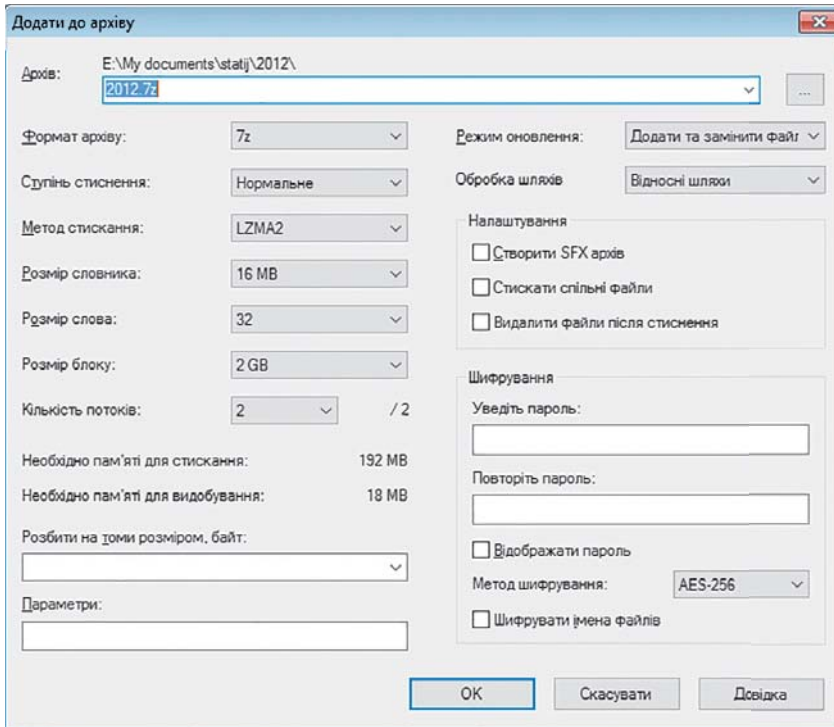


Vizsgáljuk meg, hogy végezzük ezeket a műveleteket a **7-zip** tömörítőprogram segítségével. A program 7z, ZIP, CAB, Z, RAR, ARJ, BZIP2, TAR, LZH stb. fájltypusokat képes kezelni. A program **freeware** licenclésű, azaz a fejlesztők honlapjáról bárki ingyen letöltheti és telepítheti. Létezik **Windows** és **Linux** operációs rendszerekhez készült változatban is.

A program ablakát a 2.52. ábra mutatja.

Archívum **létrehozása** a következőképpen történik:

1. Indítsd el a **7zip**-et (**Start** ⇒ **Minden program** ⇒ **7zip**!)
2. Nyisd meg a tömörítendő fájlokat tartalmazó mappát!
3. Jelöld ki a tömörítendő fájlokat, majd kattints a **Hozzáadás**  gombra!
4. A **Hozzáadás** ablakban állítsd be az archívum tulajdonságait (2.53. ábra):
 - *Név* – a név mezőben;
 - *Mentés helye* – válaszd ki a **Tallózás**  gomb használatával;
 - *Formátum* – 7z, tar, wim vagy zip;
 - *Tömörítés foka*: *tömörítés nélkül*, *leggyorsabb*, *gyors*, *normál*, *maximális*, *ultra*;
 - A tömörítés módját;
 - Szükség esetén – ha korlátozott kapacitású az archívum tárhelye – a többkötetes módot;
 - Szükség esetén – ha a kitömörítésnél nem áll rendelkezésre a tömörítőprogram – állítsd be az SFX opciót;
 - Írj be jelszót, ha védett archívumot kívánsz létrehozni!



2.53. ábra. Hozzáadás az archívumhoz ablak




2. fejezet

5. Kattints a z **OK** gombra!


Amennyiben a fájlokat egy **már létező archívumhoz adjuk hozzá**, akkor az algoritmus negyedik lépésében meg kell adni a már létező archívum nevét és elérési útját.

Az archívum **tartalmának megtekintéséhez** meg kell azt nyitni a program vagy az intéző ablakában.

Az archívum **valamennyi fájljának kibontása** a következőképpen történik:

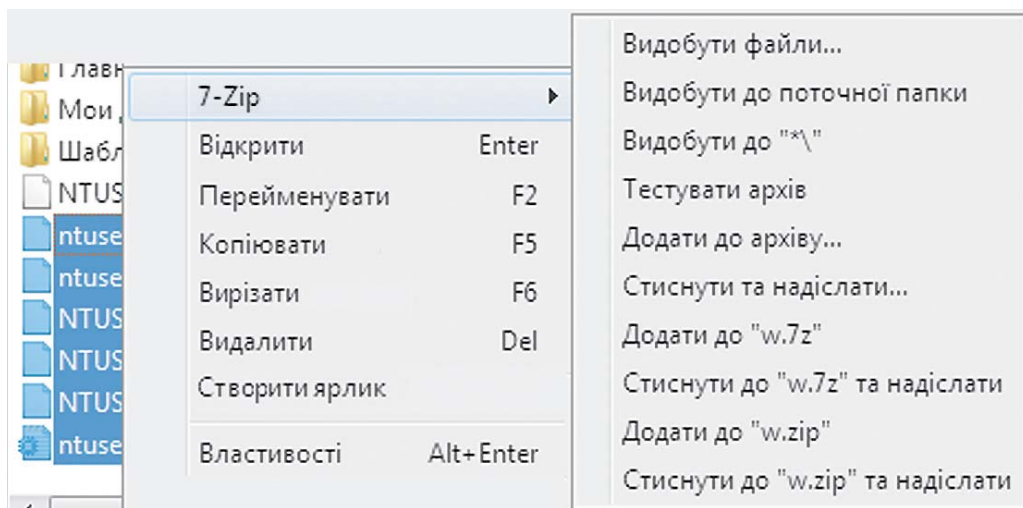
1. Indítsd el a **7-zip-et!**
2. Nyisd meg az archívumot!
3. Kattints a **Kitömörítés**  gombra!
4. A **Kitömörítés** ablakban jelöld ki azt a mappát, ahová az archívumot ki szeretnéd tömöríteni!
5. Kattints az **OK** gombra!

Ha csak **részben szeretnéd kitömöríteni** az archívumot:

1. Indítsd el a **7-zip-et!**
2. Nyisd meg az archívumot!
3. Jelöld ki a kitömörítendő fájlokat!
4. Kattints a **Kitömörítés**  gombra!
5. A **Kitömörítés** ablakban jelöld ki azt a mappát, ahová az archívumot ki szeretnéd tömöríteni!
6. Kattints az **OK** gombra!

Amennyiben fájlokat és mappákat szeretnél **eltávolítani az archívumból**, megnyitás után jelöld ki ezeket, majd kattints a **Törlés**  gombra!


A program korrekt telepítése után a mappák és fájlok helyi menüjéhez (2.54. ábra) hozzáadódnak a tömörítés parancsai. A tömörítőt ezek segítségével is el lehet indítani.

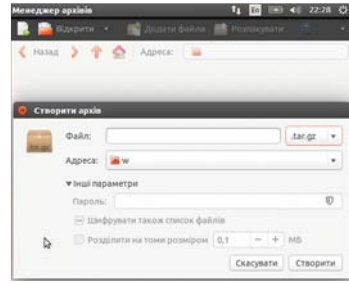


2.54. ábra. Fájlcsoport helyi menüje



Linux-használóknak

Az **Ubuntu Linux** operációs rendszerben a tömörített fájlok kezelésére a **7-zip**-en kívül rendelkezésünkre áll még az **Archívum manager**  is. Tömörítéskor több mint 10 formátum közül választhatunk, lehetőségünk van többkötetes és védett archívum létrehozására is.



2.55. ábra. Az Archívum manager ablaka

Összefoglalás

A fájlok méretének csökkentésére **adattömörítést** használunk, amelyek speciális **tömörítési eljárások** szerint működnek. A **tömörítés** tehát nem más, mint adatok újrakódolása a fájl méret csökkentése céljából.

A tömörítési eljárások lehetnek **veszteségesek** és **veszteség nélküliek**.

Az adatok olyan másolása, amely tömörítőprogramok segítségével történik, az **archiválás**. A **tartalékmásolatok** és **rendszerképek** létrehozására az operációs rendszer segédprogramjait használjuk.

A tömörítőprogramokat szokás még **archivátoroknak** is nevezni. Archívumokon a következő műveleteket végezhetjük: *létrehozás* (egykötetes, többkötetes, SFX), *tartalom megtekintése*, új fájlok *hozzáadása*, régiek *cseréje*, objektumok *eltávolítása*, archívum részleges vagy teljes *kitömörítése*.



Felelj a kérdésekre!

- 1°. Mire használhatjuk az adattömörítést?
- 2°. Milyen esetekben megengedett a veszteséges tömörítés, és mikor nem?
- 3°. Mire szolgálnak az archívumok?
- 4*. Mit értünk az archiváláson, és mit a tömörítésen? Hogyan kapcsolódnak ezek egymáshoz és mi közöttük a különbség?
- 5°. Hogy nevezzük az archiváló programokat? Milyen műveletek végezhetnek ezek az archívumokkal?
- 6°. Nevezz meg tömörített formátumokat!
- 7°. Ismertesd több fájl egyidejű tömörítésének lépéseit!
- 8°. Miben különbözik néhány fájl kicsomagolásának folyamata a teljes archívum kitömörítésétől?
- 9°. Hogy törölünk fájlokat az archívumból?
- 10°. Hogy hozunk létre néhány fájlból archívumot?
- 11°. A helyi menü segítségével?



Végezd el a feladatokat!

- 1°. Indítsd el a tömörítőprogramot! Tömörítsd a **Rozdil 2** mappa **Archív** mappájának első 5 **doc** kiterjesztésű fájlját! A tömörített állományt helyezd el a saját mappádba!
- 2°. Csomagold ki a **Rozdil 2\pukt 2.5\Archív** mappában található **Archív 1.7z** fájlt a saját mappád **Archívumom** mappájába!
- 3°. Add hozzá a **Rozdil 2\Archív** mappájában található **Archív 2.zip** fájlhoz ugyanezen mappa két utolsó fájlját! Mentsd el a tömörített állományt a saját mappádba!



- 4*. Indítsd el a tömörítőprogramot, majd:
 - a) tömörítsd a dokumentumaid közül az első 15 **doc** kiterjesztésű fájlt (vagy a **Rozdil 2\Archív** mappa első 15 **doc** kiterjesztésű fájlját);
 - b) a tömörítést a következő paraméterekkel végezd: név – *dokumentumok*, mappa – *Munkasztal*, formátum – *ZIP*, módszer – *tömörítés nélkül*;
 - c) a helyi menü alkalmazásával tömörítsd ki a létrehozott állományt a **Dokumentumok\Doc** mappába!
- 5*. Derítsd ki, milyen mértékben tömöríti a **7z** program az adatokat a program különböző tömörítési módszereivel! A kísérlethez használd a **Rozdil 2\pukt 2.5\képek** mappa **bmp** kiterjesztésű fájljait! Töltsd ki a táblázatot! *Megjegyzés: a tömörítési arány a tömörített és a tömörítetlen fájl méretének aránya.*

Sorszám	Archívum neve	Tömörítési mód	Fájl mérete		Tömörítési arány
			tömörítés előtt	tömörítés után	
1	kep1.7z				
2	kep2.7z				
3	kep3.7z				
4	kep4.7z				
5	kep5.7z				
6	kep6.7z				



- 6*. Indítsd el a tömörítőt, majd:
 - a) tömörítsd a **mintazene** mappa első két fájlját; az archívum neve legyen *Archív02*, formátuma pedig *7z*, *SFX-archív*!
 - b) töröld az előző feladatban létrehozott archívumból az utolsó fájlt;
 - c) add hozzá az archívumhoz a képek mappa harmadik fájlját;
 - d) csomagold ki az archívumot a dokumentumaid között létrehozott **zene** mappába!



3. SZ. GYAKORLATI MUNKA

Fájlok tömörítése és visszaállítása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Indítsd el a **7-zip**-et!
2. Tömörítsd a dokumentumaid közül az első 16 **doc** kiterjesztésű fájlt (vagy a **Rozdil 2\Archív** mappa első 16 **doc** kiterjesztésű fájlját); a tömörítéshez használd a következő paramétereket:
 - formátum – **7z**;
 - tömörítés foka – *gyors tömörítés*;
 - *SFX-archívum* létrehozása!
3. Végezd el a következő műveleteket:
 - teszteld az archívumot;
 - másold az archívumot a saját mappád **Archívum másolata** mappájába, majd töröld belőle az utolsó öt fájlt!
4. Határozd meg, és jegyezd le a füzetedbe:
 - a fájlok összméretét tömörítés előtt;
 - a fájlok összméretét tömörítés után;
 - a tömörítési arányt!
5. A helyi menü segítségével tömörítsd ki az archívumot a mappád **DOC** mappájába!
6. A tömörítőprogram segítségével tömörítsd ki az első 10 fájlt a mappád **1_10** mappájába!
7. Zárj be valamennyi, a megoldás során használt ablakot!

Ebből a fejezetből megtudhatod:

hogyan illessz a szövegbe olyan karaktereket, amelyeket nem találsz a billentyűzeten

hogyan használd a stílusokat

hogyan használhatsz számozást, felsorolást, többhasábos tördelést

hogyan illessz hivatkozásokat a szöveges dokumentumokba

hogyan illessz a szövegbe táblázatot, képletet, képeket, és hogyan kell feldolgozni ezeket

hogyan ellenőrizd a dokumentum szerkezetét és hogyan kell létrehozni tartalomjegyzéket

hogyan használd a sablonokat

hogyan állíthatod be az oldal tulajdonságait és hogyan kell létrehozni fejléceket, lábléceket

hogyan dolgozz egyszerre több dokumentummal



3.1. KARAKTEREK, LISTÁK ÉS HASÁBOK BESZÚRÁSA SZÖVEGES DOKUMENTUMBA




1. Milyen szövegkijelölés módokat ismertek?
2. Milyen szerkesztési műveleteket végezhetünk betűkkel és bekezdésekkel? Hogy végezzük ezeket?
3. Milyen formázási műveleteket végezhetünk betűkkel és bekezdésekkel? Hogy végezzük ezeket?
4. Mik a vízszintes vonalzon található jelek rendeltetése?

SPECIÁLIS KARAKTEREK ELHELYEZÉSE A SZÖVEGBEN

Már megtanultad, hogyan kell szöveget létrehozni és szerkeszteni **Word 2007**-ben: megismerkedtél a beírás, szerkesztés, formázás fortélyaival, a karakterek és a bekezdések tulajdonságaival. A szövegszerkesztőben azonban rengeteg egyéb művelet is végezhető. Most ezekkel foglalkozunk majd.

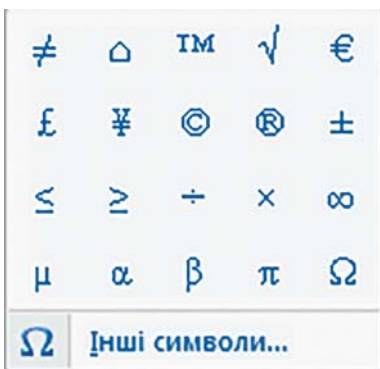
A **Word 2007**-ben a szövegben olyan szimbólumokat is elhelyezhetünk, amelyek nincsenek a billentyűzeten: más nyelvek betűit, matematikai jeleket, egyezményes jeleket stb. Ilyenek például a © ≤ β ∑ € ∞ ≈ ≠ ∅ ∞ ∫ § ¶ X ЙЖ.

Az ilyen jeleket a következő algoritmus szerint helyezzük el a szövegben:

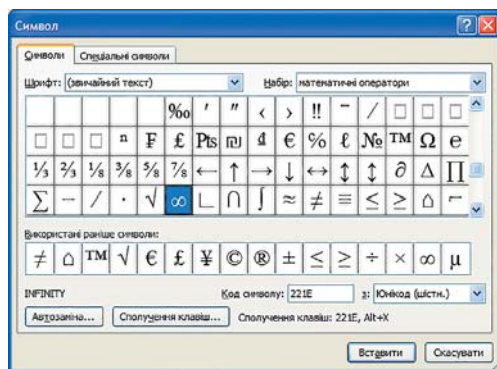
1. Helyezzük a kurzort a szövegben a beszúrás helyére!
2. Kattintsunk a **Beszúrás** fülön a **Szimbólum** csoport **Szimbólum**  gombjára!
3. Jelöljük ki a megnyíló párbeszédablakban (3.1. ábra) a megfelelő szimbólumot!

Amennyiben a szükséges szimbólumot nem találjuk a listán:

1. Kattintsunk a **További szimbólumok** utasításra!
2. Tekintsük meg a megnyíló párbeszédablakban (3.2. ábra) a **Szimbólumok** és a **Különleges karakterek** fülek tartalmát! Szükség esetén kiválaszthatjuk a **Betűtípust**, ez megváltoztatja az elérhető szimbólumok listáját.
3. Válasszuk ki a szimbólumot!
4. Kattintsunk a **Beszúrás** gombra!



3.1. ábra. A szimbólumok listája




3.2. ábra. A Szimbólum ablak

Ha egy szimbólum beillesztésére gyakran van szükség, a **Szimbólumok** ablakban külön **Billentyű-kombinációt** rendelhetünk hozzá (a **Szimbólum** ablak **Billentyűparancs** gombjának lenyomása után ki kell tölteni a párbeszédablak mezőit). Ezután a billentyűparancs segítségével gyorsan beilleszthető a szimbólum.

A **Word 2007**-ben rajzjellegű szimbólumokat is beilleszthetünk a szövegbe, például ilyeneket:



Ezeket a szimbólumokat az MS Outlook, Webdings, Wingdings stb. betűtípusokban találjuk, de a **Szimbólum** ablak segítségével is beilleszthetők.

A szöveg szerkesztése során, különösen ha azt az internetről mentettük le, hasznos a **formázási szimbólum megjelenítése**, amit a **Kezdőlap Bekezdés** csoportjának **Minden látszik**  gombjával kapcsolhatunk be.

A **formázási jelek** közé azokat a jeleket soroljuk, amelyeket a felhasználó visz be a beírás során, de nyomtatásban nem jelennek meg. Ha azonban bekapcsoljuk ezek megjelenítését, akkor speciális jelekként megjelennek a képernyőn (3.1. táblázat).

3.1. táblázat

Formázási jelek

Szimbólum	Billentyű	Jel	Szimbólum	Billentyű	Jel
Szóköz	Szóköz	.	Nem törölhető szóköz	Ctrl + Shift + szóköz	◦
Bekezdés vége	Enter	¶	Sortörés	Shift + Enter	↵
Tabulátor	Tab	→	Oldaltörés	Ctrl + Enter	...Oldaltörés...

Ha megjelenítjük a formázási jeleket is, könnyebben megérthetjük a dokumentum szerkezetét. Ha látjuk a **Bekezdés vége** ¶ jeleket, pontosan tudjuk, hol végződik a bekezdés és hol kezdődik a következő. Ha a kurzort elhelyezzük a bekezdésben, majd **Entert** ütünk, a bekezdést két bekezdésre tagoljuk annál a helynél, ahol a kurzor a leütéskor volt. Ahol leütöttük a szóközt, ott a szövegben egy rejtett jel jelenik meg, amelynek törlése után a szavak „összeragadnak”.

A szöveg szerkesztése és a karakterek formázása a már régebben megismert módszerekkel történik.



Ha többet szeretnél tudni

Vannak olyan szimbólumsorozatok, amelyeket a bevitel során a **Word 2007** automatikusan egy másik szimbólumra cserél. Ezekre láthatunk néhány példát a 3.2. táblázatban.



Speciális szimbólumok beville

Szimbólumsorozat	Helyettesítő jel	Szimbólumsorozat	Helyettesítő jel
szó, szóköz, mínusz, szóköz, szó	– kötőjel	--> mínusz, mínusz, nagyobb-jel	→
:)	☺	:	☹

A szimbólumsorozatok automatikus cseréje a **Word 2007**-ben alapértelmezett. A felhasználó a **Szimbólum** párbeszédablak **Automatikus javítás** gombjával saját elemeket adhat ehhez a listához.

LibreOffice-használóknak

A billentyűzetten nem található karaktereket a **LibreOffice Writer**-ben a **Standard** eszköztár **Speciális szimbólumok beville** gombjára kattintva szűrhatunk be.

A rejtett karaktereket a **Standard** eszköztár **Nem nyomtatható karakterek** gombjára kattintva jeleníthetjük meg.

LISTÁK ÉS AZOK BEILLESZTÉSE A SZÖVEGBE

A **listák** lényegében speciálisan formázott bekezdések. Listákkal általában objektumok vagy teendők felsorolását formázzuk meg. Ilyenek például az osztálynévsor, egy felhasználói utasítás, egy dokumentum kialakításának szabályai, a gyógyszerek listája egy patikában, egy étel elkészítésének lépései stb.

A **Word 2007**-ben háromféle listát alkalmazhatunk:

- **Felsorolás**, amikor minden egyes bekezdés egy speciális karakterrel, **felsorolási jellel** kezdődik. Például:

Évszakok: ▪ Tél ▪ Tavasz ▪ Nyár ▪ Ősz	Évszakok: ➤ Tél ➤ Tavasz ➤ Nyár ➤ Ősz	Évszakok: ❖ Tél ❖ Tavasz ❖ Nyár ❖ Ősz
---	---	---




- **Számozott**, amikor minden egyes bekezdésnek saját sorszáma van. A sorszámot helyettesíthetjük betűvel, egyéb sorszámozott jellel. Például:

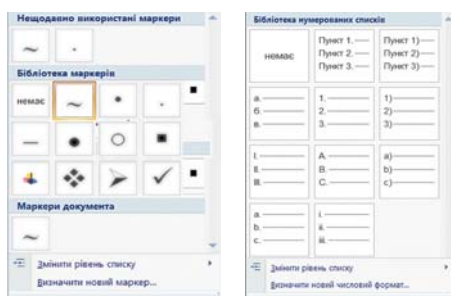
Városok: 1. Zaporizzsja 2. Kijev 3. Kremencuk 4. Csernyivci	Dokumentum objektumai: a) szimbólum b) bekezdés c) oldal d) dokumentum	Tartalomjegyzék: 1. pont. Szövegszerkesztő 2. pont. Táblázatkezelő 3. pont. Képszerkesztő 4. pont. Multimédiás prezentáció
---	--	--

- **Többszintű**, amikor a listák egyes elemei alá új listák vannak rendelve. A szövegszerkesztőben maximum 9 szintű lista hozható létre. Például:

<p><i>Évszakok:</i></p> <ol style="list-style-type: none"> 1. Tél <ul style="list-style-type: none"> • December • Január • Február 2. Tavasz <ul style="list-style-type: none"> • Március • Április • Május 3. Nyár 4. Ősz 	<p><i>Évszakok:</i></p> <ol style="list-style-type: none"> 1. Tél <ol style="list-style-type: none"> a) December b) Január c) Február 2. Tavasz <ol style="list-style-type: none"> a) Március b) Április c) Május 3. Nyár 4. Ősz 	<p><i>Évszakok:</i></p> <ol style="list-style-type: none"> 1. Tél <ol style="list-style-type: none"> 1.1. December 1.2. Január 1.3. Február 2. Tavasz <ol style="list-style-type: none"> 2.1. Március 2.2. Április 2.3. Május 3. Nyár 4. Ősz
--	--	--

Egy szövegrészt többféleképpen listává alakíthatunk.

I. mód s z e r. A leggyakrabban alkalmazott. A kurzort a leendő lista elejére állítjuk, majd bekapcsoljuk a **Kezdőlap Bekezdés** csoportján a **Felsorolás**  vagy **számozás**  gombot. Ekkor a kurzor helyén megjelenik a legutóbb használt lista jele vagy sorszáma. Ha listajelt vagy sorszámozási stílust szeretnénk váltani, használjuk a megfelelő gombok mellett található  legördülő menüt, és válasszuk ki a megfelelőt (3.3. ábra). Ezután begépeljük a lista első elemét.



3.3. ábra. Felsorolás és számozás stílusok

1. Jelöljük ki ezeket a bekezdéseket!

2. Kattintsunk a **Szalagon a Felsorolás**  vagy a **Sorszámozás**  gombra!

3. Szükség esetén válasszuk ki a megfelelő stílust a legördülő listából!

Ezután minden bekezdés első sora előtt megjelenik a megfelelő listajel vagy sorszám.

Az első elem begépelése után üssünk **Entert**, a következő sor elején megjelenik a listajel vagy a következő sorszám. Ha a lista utolsó elemét is begépeztük, kattintsunk a **Menüszalagon** ugyanarra a gombra, amellyel a listát elkezdtük, vagy üssünk két **Entert**, vagy üssük le a **Backspace**-et.

II. mód s z e r. Bekezdések átalakítása listává. Ha már kész bekezdéseket szeretnénk listává alakítani, akkor:

Ha többet szeretnél tudni



III. mód s z e r. Automatikus listaképzés. Helyezd a kurzort a dokumentum megfelelő jelére, írd be a listajelt vagy sorszámot, majd üss **Szökőzt** (3.3. táblázat). A sor elején megjelenik a megfelelő jel vagy sorszám. Megkezdheted az első elem begépelését.



Automatikus listajelek kialakítása

Szimbólum	Listajel	Szimbólum	Listajel
<i>Felsorolás</i>		<i>Felsorolás</i>	
csillag *	•	o betű o	◦
mínusz -	–	nagyobb jel >	➤
<i>Számozás</i>		<i>Számozás</i>	
1 és pont	1.	1 és zárójel	1)
1 és nagyobbjel >	1>	a betű és zárójel	a)

LibreOffice-használóknak

A LibreOffice Writer-ben egy szövegrész listává alakítására a **Formázás** eszköztár **Felsorolás**  vagy **Számozás**  eszközét kell használni. Ezeket a **Formátum** menü **Felsorolás és számozás** parancsával is elérhetjük. A gombok lenyomása után megnyíló eszköztárak segítségével választhatjuk ki a lista stílusát.

LISTÁK SZERKESZTÉSE

A **Word 2007** listáit igen könnyű szerkeszteni.

Ha szeretnénk beszúrni egy új listaelemet, elég a kurzort a lista megfelelő helyére vinni, majd ott **Entert** ütni. Ekkor beszúrásra kerül egy új sorszámozott vagy megjelölt bekezdés, ráadásul a többi listaelem sorszámozása is átalakul.

Ha egy listaelemet kijelölünk, majd a **Delete** vagy **Backspace** billentyűvel töröljük, a sorszámozás automatikusan megváltozik.

Előfordulhat, hogy a sorszámozott bekezdések között el kell helyezni egy nem sorszámozottat. Ezt úgy tehetjük meg, hogy a sorszámot a **Backspace**-el töröljük, vagy ha megnyomjuk a **MenüszalagSorszámozás** gombját.

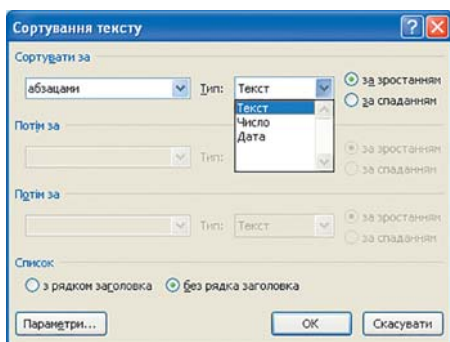
Sorszámozott lista másolása vagy áthelyezése során megesik, hogy a számozás nem felel meg az elvárásainknak. A sorszámozást a következőképpen változtathatjuk meg:

1. Vigyük az egérmutatót az újraszámozandó bekezdésre! Amennyiben a sor elejére helyeztük a kurzort, a sorszám ki lesz jelölve.
2. Nyissuk meg a helyi menüt (3.4. ábra)!
3. Válasszuk ki a megfelelő lehetőséget!
 - **Újraszámozás a következőnél:1**, ekkor értelemszerűen az aktuális bekezdés sorszáma egy lesz.



3.4. ábra. Sorszámozás a helyi menüben

- **Számozás folytatása** – a kijelölt elem sorszáma az előző lista utolsó elemét egy-egyel fogja meghaladni.
- **Számozás értékének beállítása** – ekkor a kijelölt bekezdés sorszáma a felhasználó által megadott lesz.



3.5. ábra. A Szövegrendezés ablak

- **Rendezés alapja** – bekezdés.
- **Típusa** – szöveg, sorszám, dátum.
- **Emelkedő vagy csökkenő sorrend!**

4. Kattintsunk az **OK** gombra!

A listaelemeket rendezhetjük sorszám szerint vagy a tartalom szerint ABC-rendbe, esetleg dátum szerint emelkedő vagy csökkenő sorrendbe. Rendezéskor a sorszámok megmaradnak, a bekezdések sorrendje változik meg a megadott szempontok szerint.

A rendezést az alábbi algoritmus szerint végezzük:

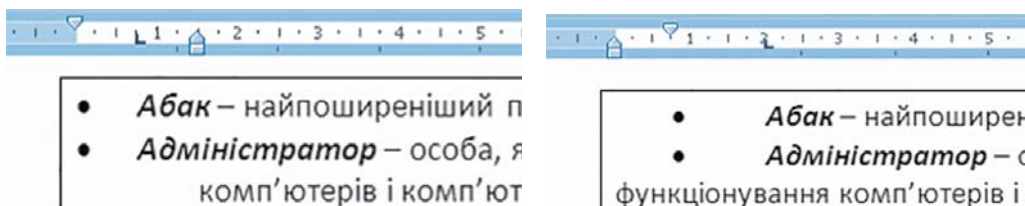
1. Jelöljük ki a rendezendő bekezdéseket!
2. Kattintsunk a **Kezdőlap** **Bekezdések** csoportjának **Rendezés** gombjára!
3. A megnyíló párbeszédablakban (3.5. ábra) állítsuk be a rendezés paramétereit:

LISTÁK FORMÁZÁSA

Szükség esetén a felhasználónak lehetősége van a lista formázására is: megváltoztathatja a listajelet, a számozás stílusát, a listajel és a szöveg behúzását stb.

E célból ki kell jelölni a formázandó elemeket, majd a **Kezdőlap** vezérlőelemeinek, párbeszédablakoknak és a helyi menük alkalmazásával el kell végezni a megfelelő beállításokat. A bekezdés formázására például jól használhatjuk a vízszintes vonalzó vezérlőelemeit (3.6. ábra).

A szöveg és a listajel távolságát a **tabulátorjel** segítségével szabályozzuk, amelynek segítségével a bekezdés elemeit szigorúan meghatározott pozíciókhoz (**tabulátorpozíciók**) köthetjük. A vonalzón a tabulátorok helyét alapértelmezetten a **L** jel mutatja.



3.6. ábra. A vonalzó vezérlőelemei



Ahogy a 3.6. ábrán láthatjuk, a tabulátor az egyik esetben 0,75 cm, a másikban 2 cm pozícióban van. A szöveg ezért ezekben a pozíciókban kezdődik.

Tabulátort úgy helyezünk el a vonalzón, hogy a vonalzó megfelelő helyére kattintunk. Ha meg szeretnénk változtatni a tabulátorpozíciót, húzzuk a tabulátorjelet a megfelelő helyre. Ha el akarjuk távolítani a tabulátort, húzzuk le a vonalzóról.

A tabulátorok helyzetét és tulajdonságait a **Tabulátor** párbeszédablakban is szabályozhatjuk, amit a **Kezdőlap-Bekezdés** csoportjának **Tabulátor** eszközével nyithatunk meg.

A listák formázásának egyik sajátossága, hogy a szöveg és a listajel formátuma eltérő lehet. Ha kijelölünk egy listajelet vagy sorszámot, ki lesznek jelölve a lista hasonló objektumai (például az azonos szint összes sorszáma). E kijelölés után a jeleket a szövegtől függetlenül formázhatjuk (3.7. ábra).

1.	Понеділок	1.	Понеділок
2.	Вівторок	2.	Вівторок
3.	Середа	3.	Середа
4.	Четвер	4.	Четвер
5.	П'ятниця	5.	П'ятниця
6.	Субота	6.	Субота
7.	Неділя	7.	Неділя


3.7. ábra.
Lista formázása



A felhasználónak lehetősége van saját listajel vagy számformátum létrehozására, ha a **Felsorolás** vagy **Számozás** legördülő menü megfelelő elemére kattint (lásd a 3.3. ábrát). A megnyíló párbeszédablakban be kell állítani a kiválasztott elemet vagy formátumot.



Ha többet szeretnél tudni

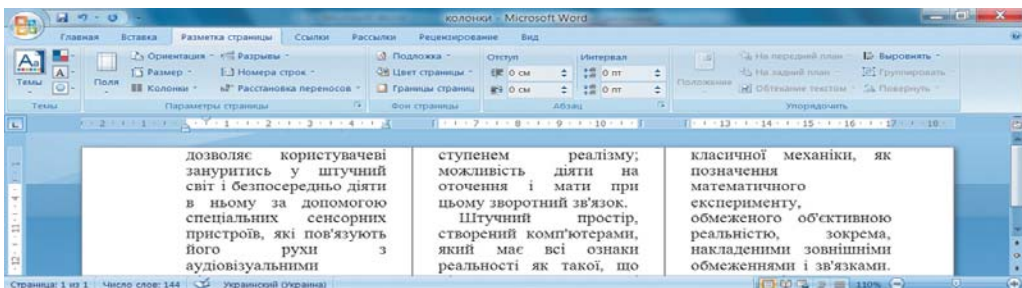
Többszintű listák

Többszintű listát a **Bekezdés** csoport **Többszintű**  gombjára kattintva hozhatunk létre. Ha legördítjük a gombhoz tartozó legördülő menüt, választhatunk a **Listastílusok** közül.

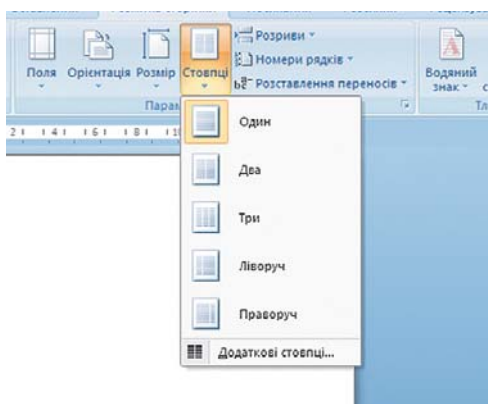
Többszintű listát egyszintű listából is létrehozhatunk, sorszámozottat és felsoroltat egyaránt. Ha kijelöljük az egyszintű lista valamelyik elemét, majd a **Behúzás növelése**  gombra kattintunk, a kijelölt elem szintje eggyel csökken. A listaszint növelését a **Behúzás csökkentése**  gombra kattintva érhetjük el.

TÖBBHASÁBOS TÖRDELÉS

Word 2007-ben lehetőségünk van arra, hogy a szövegünket újságszerűen tördeljük (3.8. ábra).



3.8. ábra. Három hasábra tördelt szöveg



3.9. ábra. Hasábok legördülő menü

A szöveget a *Lap elrendezése* ⇒ *Oldalbeállítás* ⇒ *Hasábok* parancs végrehajtása után tördelhetjük több hasábra. A legördülő menüből (3.9. ábra) kell kiválasztani a hasábok számát és elrendezését. A szöveg automatikusan a kiválasztott hasábokba lesz törelve.

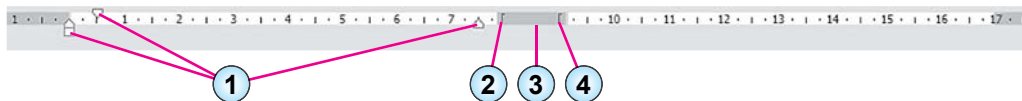
Ha vissza át szeretnénk térni az egyhasábos tördeléshez, jelöljük ki a szöveget, majd a legördülő menüben válasszuk ki az egyhasábos tördelést. A művelet után az elő sor előtt és az utolsó sor után megmarad a **hasábtörés rejtett** formázókarakter, amit ajánlatos eltávolítani.

LibreOffice-használóknak

A LibreOffice Writer-ben a **Formátum** menü **Hasábok** parancsával végezzük a többhasábos tördelést. A hasábok számát a megnyíló párbeszédablakban kell megadni.

HASÁBOK SZERKESZTÉSE ÉS FORMÁZÁSA

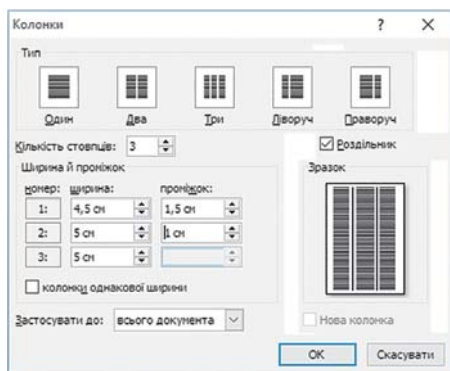
Ha a kurzor egy több hasábra tördelt szövegben van, akkor a vízszintes vonalzó a bekezdésbehozásokon és tabulátorjeleken kívül *hasábelválasztó jeleket* is találunk (3.10. ábra). Ezek mozgásával megváltoztathatjuk a hasábszélességeket és a hasábok távolságát.



1. Bekezdésbehozások
2. Első hasáb jobb széle

3. Hasábelválasztó
4. Második hasáb jobb széle

3.10. ábra. Markerek a vízszintes vonalzó




3.11. ábra. Hasábok párbeszédablak



A hasábok formázása a **Hasábok** párbeszédablakban (3.11. ábra) is elvégezhető, amit a *Lap elrendezése* ⇒ *Oldalbeállítás* ⇒ *Hasábok* ⇒ *További hasábok* parancssal nyithatunk meg. Itt is beállíthatjuk a hasábok számát és a távolságot, elválasztóvonalat (függőleges vonal) helyezhetünk el a hasábok között. Ha elkészültünk a beállításokkal, nyomjuk meg az **OK** gombot!



Összefoglalás

A **Word 2007** szövegszerkesztőben lehetőségünk van olyan szimbólumok elhelyezésére a szövegben, amelyek nincsenek a billentyűzeten. Ezeket a szimbólumokat a **BeszúrásSzimbólum** eszközzel illeszthetjük a szövegbe. A szövegbe különböző rajzszerű jeleket is beilleszthetünk, ezeket az MS Outlook, a Wingdings és a Webdings betűtípusok rejtik.

A **rejtett szimbólumok** alatt azokat a jeleket értjük, amelyek nyomtatásban nem jelennek meg, de amelyeket a felhasználó ír be a szövegbe, és amelyek hatással vannak a szöveg formázására. Ilyenek a szóköz, a bekezdésvégjel, a tabulátor stb. Ezeket a jeleket a **Kezdőlap** ⇒ **Bekezdés** ⇒ **Minden látszik**  eszközére kattintva tehetjük láthatóvá.




A bekezdéseket felsorolássá vagy sorszámozott listává alakíthatjuk, így tehetjük jobban áttekinthetővé a szövegeket. Ezt úgy tehetjük meg, hogy kijelöljük a megfelelő szövegrészt, majd a **Felsorolás**  vagy a **Sorszámozás**  gombra kattintunk. Ha szükséges, a gombok legördülő menüjéből kiválaszthatjuk a megfelelő formátumot.

Szerkesztéskor új elemeket adhatunk a listához, elemeket távolíthatunk el belőle, növekvő vagy csökkenő sorrendbe rendezhetjük a sorszám vagy a tartalom szerint.

A szöveget több oszlopba tördelhetjük. Ezt a **Lap elrendezése** ⇒ **Oldalbeállítás** ⇒ **Hasábok** ⇒ **További hasábok** parancs hatására megnyíló párbeszédablakban állíthatjuk be, ahol a hasábok számát, méretét és elválasztóját szabályozhatjuk.



Felelj a kérdésekre!

- 1°. Hogy illeszthetünk a szövegbe a billentyűzeten nem található szimbólumokat (matematikai jeleket, grafikai jeleket stb.)? Hogy szerkeszthetjük és formázhatjuk ezeket?
- 2°. Mik azok a rejtett (nyomtatásban nem megjelenő) jelek? Hozz fel példákat! Mi ezeknek a jeleknek a rendeltetése?
- 3°. Milyen listatípusok léteznek a **Word 2007**-ben?
- 4°. Mi a **Bekezdés** csoport következő , ,  gombjainak neve?
- 5°. Mít értünk a lista automatikus sorszámozása alatt? Hogy változtathatjuk meg egy elem sorszámát?
- 6°. Hogy változtathatjuk meg egy lista sorszámozását a szerkesztés befejezése után?
- 7°. Az osztálynévsorból kimaradt néhány név. Hogy illeszthetjük be ezeket a névsorba?
- 8°. Az osztálynévsort tetszőleges sorrendbe írtuk be. Hogy rendezhetjük a névsort?
- 9°. Hogy tördelhetjük a szöveget hasábokba? A hasábok milyen tulajdonságait változtathatjuk meg?

- 10•. Hogy változtathatjuk meg a hasábok és az elválasztók tulajdonságait valamint a hasábok számát?



Végezd el a feladatokat!



- 1°. Nyisd meg a **Word 2007**-et és helyezz el a szövegben 5 MS Outlook, Webdings vagy Wingdings típusú grafikai jelet! MentSD a fájlt **feladat 3.1.1.docx** néven a mappádba!
- 2°. Nyisd meg a tanárod által meghatározott fájlt (például a **Rozdil 3\punkt 3.1\zrazok 3.1.2.docx**-et! A szöveg első három bekezdését alakítsd felsorolássá, a negyedik–hatodik bekezdéseket pedig számozott listává! A szöveg címénél szüntesd meg a számozást! MentSD a fájlt **feladat 3.1.2.docx** néven!
- 3°. Hozz létre új dokumentumot, írd be az osztálytársaid neveit számozott listaként! Rendezd a listát ábécé sorrendbe! MentSD el a fájlt **feladat 3.1.3-2.docx** néven a mappádba!



- 4°. Hozz létre új dokumentumot, írd be Ukrajna hét csodáját (megtalálod az interneten) felsorolásként! A felsorolásjel legyen a szövegtükör bal szélétől *2 cm*-re, a listaelemek pedig további *1 cm*-re a felsorolásjelektől! MentSD el a fájlt **feladat 3.14.docx** néven a mappádba!
- 5°. Nyisd meg a tanárod által meghatározott fájlt (például a **Rozdil 3\punkt 3.1\zrazok 3.1.5.docx**-et! Formázd meg a dokumentumot a benne található minta szerint! MentSD el a fájlt **feladat 3.15.docx** néven a mappádba!

- 6*. A **Súgó** segítségével derítsd ki, mikor nem láthatók a listajelek a szövegben, és hogy lehet ezt a problémát orvosolni!



- 7*. Keresd a **Microsoft Office Online** oldalon grafikai felsorolásjeleket, és a **Súgó** segítségével határozd meg, hogy lehet ezeket használni a dokumentumokban! A jelek segítségével készítsd el Ukrajna területeinek listáját! MentSD el a fájlt **feladat 3.1.7.docx** néven a mappádba!

- 8°. Nyisd meg a tanárod által meghatározott fájlt (például a **Rozdil 3\punkt 3.1\zrazok 3.1.8.docx**-et! A második bekezdést tördeld két egyforma szélességű oszlopba, melyek között nincs elválasztó, a negyediket pedig három különböző szélességű, vonallal elválasztott hasádba! MentSD el a fájlt **feladat 3.1.8.docx** néven a mappádba!

**3.2. TÁBLÁZATOK SZÖVEGES DOKUMENTUMOKBAN**









1. Mondj példákat, hol találkozhatunk táblázatokkal az iskolai életben és a mindennapokban!
2. Hogy mozgatjuk a kurzort a táblázatkezelőben?
3. Milyen tulajdonságokkal bírnak a grafikus elemek egy szöveges dokumentumban? Hogy változtathatjuk meg ezek értékeit?

TÁBLÁZATOK ÉS AZOK TULAJDONSÁGAI

A táblázatok a szöveges dokumentumokban az adatok rendezett és szemléletes bemutatására szolgálnak. A táblázatba rendezett adatok kompaktak és könnyen átláthatók. Figyeljük meg az alábbi példát (3.4. táblázat):

3.4. táblázat

A 8. osztály órarendje

Sorszám	Hétfő	Kedd	Szerda	Csütörtök	Péntek
1.	Informatika, I. csoport	Testnevelés		Mértan	Angol
	Informatika, II. csoport				Ukraina története
2.	Fizika	Ukrán nyelv	Média		
3.	Testnevelés	Ukrán irodalom	Algebra	Világtörténelem	Biológia
10:45–11:15	 Dinamikus szünet 				
4.	Kémia	Biológia	Kémia	Földrajz	Ukrán irodalom
				Egészségtan	
5.	Algebra	Angol	Fizika	Világirodalom	Ukrán nyelv
6.	Ukraina története	Földrajz	Informatika, I. csoport	Munka	Testnevelés
			Informatika, II. csoport		
14:00 Szakörök					

A táblázat *sorokból* és *oszlopokból* áll, ezek metszéspontjában találjuk a *cellát*. A sorok, oszlopok és cellák a táblázat objektumai. A **Word 2007** táblázata 63 oszlopot tartalmazhat, a sorok száma nincs korlátozva.

Ahogy a 3.4. táblázatban is láthatjuk, a sorok magassága és az oszlopok szélessége eltérőek lehetnek. A cellákat egyesíthetjük, de egyes cellákat fel is oszthatunk. A szövegrány lehet függőleges és vízszintes. A celláknak lehet kerete és kitöltése.

A táblázatok a következő tulajdonságokkal bírnak:

- **szélesség**, amit *centiméterekben* vagy a *lapszélesség százalékában* fejezhetünk ki;
- **oszlopszélesség, sormagasság, cellaméret**, amit centiméterekben vagy a lapszélesség százalékában fejezhetünk ki;
- a **szöveg körbefuttatása**, amit *engedélyezhetünk* vagy *megtíthatunk*;
- **keret**, amelynek *színe, vonalstílusa* és *szélessége* van, objektumonként eltérő lehet;
- **kitöltés**, amihez *szín* és *mintázat* tartozik.


TÁBLÁZAT LÉTREHOZÁSA

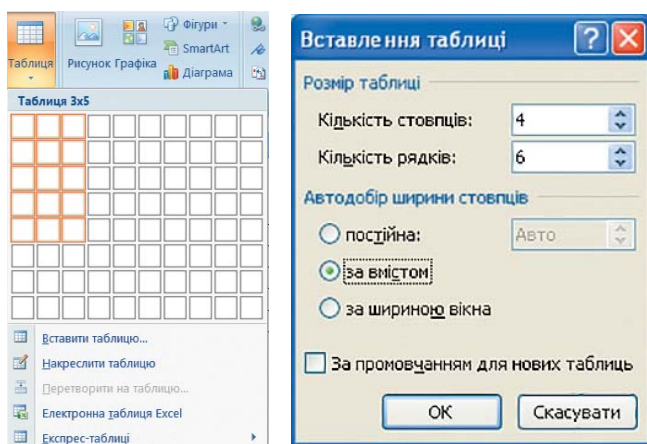
Word 2007-ben többféleképpen is létrehozhatunk táblázatokat:

1. Egyszerű szerkezetű táblázat beillesztése.
2. Tetszőleges szerkezetű táblázat megrajzolása.
3. Szöveg táblázattá alakítása.
4. Táblázat beillesztése sablonból.

Az összes lehetőséget a **Beszűrés** menü **Táblázat** csoportjának legördülő menüjében találjuk. Vizsgáljuk meg a két leggyakrabban alkalmazott módszert.

I. módszer. Egyszerű szerkezetű táblázatot az alábbi módon illeszthetünk a szövegbe:

1. Helyezzük a kurzort a szöveg megfelelő helyére!
 2. Kattintsunk a **Beszűrés** lap **Táblázat**  gombjára!
 3. Jelöljük ki a megfelelő számú sort és oszlopot, majd kattintsunk a bal egérgombbal!
- A 3.12. ábrán 3 oszlopot és 4 sort jelöltünk ki.



3.12. ábra. Táblázat beszűrése

Ezzel a módszerrel maximum 10 oszlopot és 8 sort illeszthetünk be. Amennyiben ettől nagyobb táblázatra van szükség, használjuk a **Beszűrés** ⇒ **Táblázatok** ⇒ **Táblázat** ⇒ **Táblázat beszűrése** parancsot. A megnyíló párbeszédablakban (3.12. ábra, jobb oldal) a beviteli mezőkbe beírhatjuk, hány sora, illetve oszlopa legyen a létrehozandó táblázatnak. Ha elkészültünk, kattintsunk az **OK** gombra!

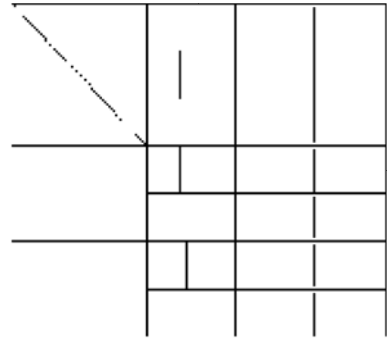


II. módszer. Ha a táblázat szerkezete bonyolult (lásd a 3.13. ábrát), célszerű azt **berajzolni** a dokumentumba. Ezt a következőképpen tehetjük meg:

1. Hajtsuk végre a **Beszűrés** ⇒ **Táblázatok** ⇒ **Táblázat** ⇒ **Táblázat rajzolása** parancsot!
2. A ceruzaalakot öltött egérmutatóval húzzuk meg a táblázat kontúrjait!
3. Húzzuk meg az oszlopokat és sorokat elválasztó vonalakat!

Ha a rajzolás során fölösleges vonalakat húztunk volna, azokat a **Radír** eszközzel könnyedén eltávolíthatjuk.

Ezt az eszközt a **TáblázateszközökTervezés** lapján találjuk (használatakor megváltozik az egérmutató alakja). Ha végeztünk a radirozással, kattintsunk újból a **Radír** eszközre.



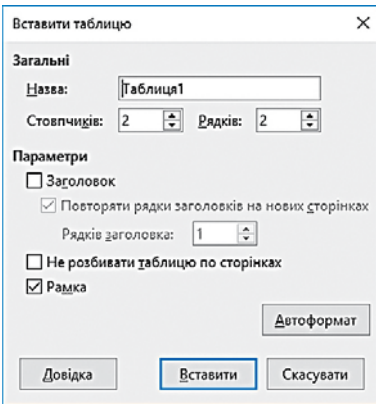
3.13. ábra. Összetett táblázat szerkezete




Ha többet szeretnél tudni

III. módszer. A **Beszűrés** ⇒ **Táblázatok** ⇒ **Táblázat** ⇒ **Szövegből táblázat** parancsával a kijelölt szövegrészt táblázattá alakíthatjuk. A megnyíló párbeszédablakban meg kell adni, hány oszlopra, vagy sorra szeretnénk felosztani a szöveget, illetve milyen jelet tekintünk cellahatárolónak (szóköz, tabulátor, bekezdés, pontosvessző, egyéb jel).

LibreOffice Writert használóknak



3.14. ábra. Táblázat beszűrése ablak

Egyszerű táblázatot a **Standard** eszköztár **Táblázat beszűrése**  eszközével illeszthetünk be. A megnyíló panelen jelöljük ki a sorokat és oszlopokat, majd kattintsunk a **Beszűrés** gombra. Ezzel a módszerrel 10 oszlopot és 15 sort illeszthetünk be.

Ha ennél nagyobb táblázatot szeretnénk beilleszteni, használjuk a **Táblázat** ⇒ **Táblázat beszűrése** parancsát, majd a megnyíló párbeszédablakban (3.14. ábra) adjuk meg a sorok és oszlopok számát, aztán kattintsunk a **Beszűrés** gombra.

ADATOK BEÍRÁSA A TÁBLÁZAT CELLÁIBA. MOZGÁS A TÁBLÁZATBAN

Ha létrehoztuk a táblázatot, hozzáláthatunk az adatok beírásához. A szöveget mindig a táblázat aktuális cellájába írhatjuk a már megismert szövegbeviteli szabályok betartásával.

A beírás során a sormagasság és oszlopszélesség automatikusan megváltozik, ez a program alapértelmezett beállítása.

Ahhoz, hogy a kurzort a megfelelő cellába helyezzük, kattintsunk a cellába az egérrel, vagy használjuk a kurzormozgató billentyűket (3.5. táblázat).

3.5. táblázat






Kurzor mozgatása a táblázatban

<i>Billentyű</i>	<i>Rendeltetés</i>	<i>Billentyű</i>	<i>Rendeltetés</i>
Tab	Következő cella	Shift + Tab	Előző cella
→	Következő karakter vagy következő cella	←	Előző karakter vagy előző cella
↓	A szöveg következő sora vagy következő cella	↑	A szöveg előző sora vagy előző cella

A TÁBLÁZAT OBJEKTUMAINAK KIJELÖLÉSE

A táblázatot, annak objektumait és a cellák tartalmát szerkeszthetjük és formázhatjuk is. A műveletek mindig az aktuális vagy a kijelölt objektumra vonatkoznak. A kijelölést a következő *két* módszer egyikével ejthetjük meg:

1. Az egér segítségével:

- *Egy cella kijelöléséhez* húzzuk az egeret bal felől a cella irányába, és amikor a kurzor -ra változik, kattintsunk a bal egérgombbal!
- *Egy sor kijelöléséhez* húzzuk az egeret bal felől a cella irányába, és amikor a kurzor -ra változik, kattintsunk a bal egérgombbal!
- *Egy oszlop kijelöléséhez* húzzuk az egeret felülről a cella irányába, és amikor a kurzor -ra változik, kattintsunk a bal egérgombbal!
- *Az egész táblázat kijelöléséhez* kattintsunk a cella bal felső sarkában található  jelre akkor, amikor az egérkurzor -ra vált!
- *A táblázat néhány szomszédos elemének kijelöléséhez* jelöljük ki egérrel azt a területet, ahol ezek az objektumok elhelyezkednek!
- *A táblázat néhány nem szomszédos elemének kijelöléséhez* jelöljük ki az egyik objektumot, majd lenyomott **Ctrl** billentyű mellett a többi objektumot!

2. A **Menüszalag** alkalmazásával: tedd aktuálissá a táblázat objektumát, vagy hajtsd végre a **Táblázateszközök** ⇒ **Elrendezés** ⇒ **Kijelölés** parancsot, és a legördülő menüből válaszd ki az objektumot (**Sor**, **Oszlop**, **Táblázat**, **Cella**)!

A kijelölés megszüntetése céljából kattints a kijelölt objektum területén kívülre.

Bármely objektumát jelöljük ki egy táblázatnak, a **Menüszalagon** megjelenik a **Táblázateszközök**, azon belül pedig a **Tervezés** és az **Elrendezés** ideiglenes eszköztár.



TÁBLÁZAT SZERKESZTÉSE

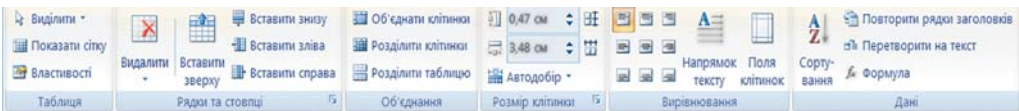
A táblázatot a **Word**-ben megszokott módokon szerkeszthetjük. A táblázat vagy valamely objektuma tartalmának gyors törléséhez ki kell jelölni az objektumot, majd pedig leütöni a **Delete** gombot. A táblázat ekkor üresen marad.

A táblázat szerkesztése azt is jelenti, hogy új objektumot helyezünk el a táblázatban, vagy objektumokat törölünk, cellákat egyesítünk vagy osztunk fel.

A **Tab** billentyű lenyomásával például új sort illeszthetünk a táblázatba. Ehhez a billentyű leütése előtt a kurzort a táblázat utolsó cellájába kell elhelyezni. Ha a táblázat közepére szeretnél új sort elhelyezni, akkor helyezd a kurzort a táblázat sorának végére a táblázaton kívülre, majd üss **Entert**.

A táblázat kijelölt sorát vagy oszlopát a **Backspace** billentyűvel törölhetjük.

A táblázaton valamennyi művelet elvégezhetünk az **Elrendezés** ideiglenes eszköztár eszközeinek segítségével (3.15. ábra). A **Sorok és oszlopok** csoportban található a megfelelő objektumok beszúrására és törlésére szolgáló gombokat. A műveletek mindig a táblázat aktuális elemeire vonatkoznak.



3.15. ábra. **Elrendezés** ideiglenes eszköztár

Használhatjuk a kijelölt táblázatelemek helyi menüjét is.

A táblázatot, mint a szöveg elemét, másolhatjuk és áthelyezhetjük. E célból valamelyik ismertetett módszerrel ki kell jelölnünk a teljes táblázatot, majd billentyűparancsokkal, áthúzással vagy a **Menüszalag** parancsaival el kell végezni a műveletet.



Ha többet szeretnél tudni


Ha a táblázat szomszédos celláit egyetlen cellává szeretnéd átalakítani, használd az **Egyesítés** csoport **Cellák egyesítése** parancsát! Ekkor a cellák mintegy „összeragadnak”, tartalmuk azonban nemvész el, hanem az egyesített cellába kerül.


Ha egy cellát részekre szeretnéd bontani:

1. Jelöld ki a cellát!
2. Hajts végre a **Cellák felosztása** parancsot!
3. A párbeszédablakban add meg, hány oszlopra, illetve sorra szeretnéd a cellát felosztani!
4. Add meg a szöveg elhelyezésének módját a megfelelő jelölő bekapcsolásával!

A táblázatot a **Rajzolás** és a **Radír** eszközökkel is fel lehet osztani, illetve egyesíteni lehet a táblázat celláit.

Ha a táblázatot ketté szeretnéd osztani oly módon, hogy a két rész közé szöveget helyeznél el, használd az **Egyesítés** csoport **Táblázat felosztása** eszközét! A kurzort természetesen a felosztás helyén kell elhelyezni előzőleg.

Amennyiben a táblázat nagy és több oldalt foglal el, megtehetjük, hogy a táblázat címsora az oszlopok neveivel minden oldalon ismétlődjön. Erre szolgál az **Elrendezés** eszköztár **Adatok** csoportjának **Címsorismétlés**  eszköze.

Ha a táblázatot szeretnénk a szövegből eltávolítani, de a tartalmát szöveggént megtartani, használjuk az **Adatok** csoport **Szöveggé alakítás**  eszközét, majd a megnyíló párbeszédablakban jelöljük ki, milyen szimbólumot szeretnénk a cellák után elhelyezni.





TÁBLÁZAT FORMÁZÁSA

A formázás alatt a következő műveleteket értjük: az oszlopszélesség és sormagasság megadása, a cellák igazításának megadása, a szövegirány megváltoztatása, a kitöltés és a keretezés beállítása.



A formázást a **Tervezés** és **Elrendezés** ideiglenes eszköztárak eszközeivel végezhetjük (3.6. táblázat). A szöveget a cellákban a **Word**-ben megszokott módokon formázhatjuk.

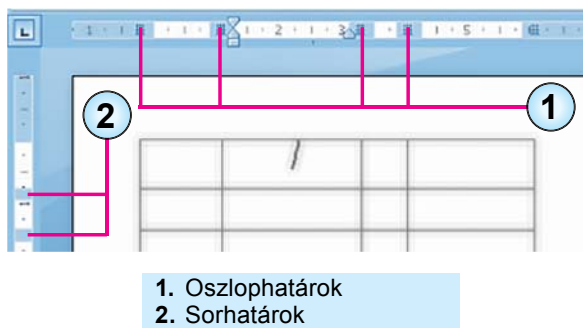
3.6. táblázat

Az Elrendezés eszköztár eszközei

<i>Eszköz</i>	<i>Rendeltetés</i>
Igazítás  lista	A szöveg elhelyezése a cellában (<i>balra, jobbra, középre, alulra, felülre</i> stb.)
Szövegirány  gomb	A szöveg <i>függőleges</i> vagy <i>vízszintes</i> elhelyezése a cellában
Cellamargók  gomb	A cella tartalmát a keretétől elválasztó <i>távolság</i> megadása
Cellamagasság és oszlopszélesség  számláló	A táblázat egyes objektumainak pontos méretét állíthatjuk be

A táblázat egyes objektumainak átméretezését másféleképpen is elvégezhetjük:

- **Az objektum határvonalainak mozgatásával.** Vigyük az egeret a sorok vagy oszlopok elválasztóvonalára fölé: itt az egérmutató  vagy  alakot ölt. A mozgatás során megjelenő szaggatott vonal mutatja a határvonal leendő helyzetét.

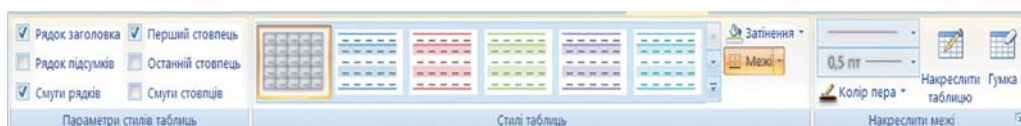


3.16. ábra. Oszlopok és sorok határvonalainak markerei a vonalzókon



- **A határolómarkerek mozgásával.** Amikor a kurzor a táblázatban van, a vonalzókon megjelennek a *sorok és oszlopok határvonalainak markerei* (3.16. ábra). Ha ezeket mozzgatjuk, akkor megváltoznak a megfelelő cellaméretetek is.

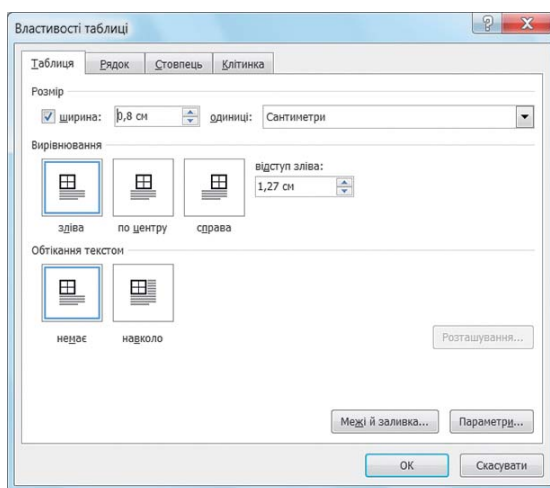
A táblázat és a táblázatobjektumok határvonalainak megváltoztathatjuk a színét, stílusát és vastagságát. A táblázat objektumait kitölthetjük színnel, különböző kitöltési effektusokat alkalmazva. A **Word 2007**-ben a táblázatok alapértelmezetten 0,5 pt vastagságú folytonos vonallal keretezettek. A kereteket a **Tervezés** ideiglenes eszköztár eszközeivel változtathatjuk meg (3.17. ábra).



3.17. ábra. Tervezés ideiglenes eszköztár

A vonalstílust, szint és vastagságot a **Szegélyek rajzolása** csoport elemeivel végezzük. A **Táblázat stílusa** csoport eszközeivel kitöltést (**Kitöltés** gomb) rendelhetünk az objektumokhoz, bekapcsolhatjuk, illetve kikapcsolhatjuk a kereteket (**Keretek** gomb). Ezeket a beállításokat **Szegély és mintázat** párbeszédablakban is elvégezhetjük.

A fenti tulajdonságokon kívül a táblázat elhelyezkedését, a körbefuttatás stílusát is szabályozhatjuk a **Táblázat tulajdonságai** ablakban (3.18. ábra). Ezt az ablakot az **Elrendezés** eszköztár **Tulajdonságok** gombjával, vagy a **Táblázatobjektumok** helyi menüjével érhetjük el.



3.18. ábra. Táblázat tulajdonságai párbeszédablak



Ha többet szeretnék tudni

A táblázatot gyors és könnyed formázását szolgálják az előre definiált táblázatstílusok, amelyeket a **Tervezés** eszköztárban találunk. A **Stílus** kiválasztása automatikusan megváltoztatja a táblázat külalakját.

A **Táblázatstílus módosítása** eszközzel – ezt szintén a **Tervezés** eszköztár **Táblázatstílusok** csoportjában találjuk – megváltoztathatjuk az alkalmazott stílus beállításait (címsor, első és utolsó sor, összesítés).

LibreOffice-használóknak

A táblázat tulajdonságait a **Táblázat** menü (3.19. ábra) vagy a **Táblázat** eszköztár eszközeinek segítségével szabályozhatjuk.



3.29. ábra. Táblázat eszköztár

Összefoglalás

A jobb áttekinthetőség és rendszerezettség céljából a szövegekben táblázatokat helyezhetünk el. A táblázat sorokból és oszlopokból áll, amelyek metszéspontjaiban cellák helyezkednek el. A sorok, oszlopok és cellák a táblázat objektumai. A cellák szöveget, képeket, számokat, képleteket és táblázatokat tartalmazhatnak.

A táblázatnak, mint a szöveges dokumentum objektumának, a következő tulajdonságai vannak: méret, oszlopszélesség, sormagasság, igazítás, szöveg körbefuttatása, keretek, kitöltés.

A dokumentumban többféleképpen helyezhetünk el táblázatokat. A leginkább általános a **Beszúrás** ⇒ **Táblázatok** ⇒ **Táblázat** ⇒ **Táblázat beszúrása** parancs alkalmazása, majd a megnyíló párbeszédablak mezőinek kitöltése (sorok, oszlopok száma, az oszlopok közötti távolság stb.), ami után az **OK** gombot lenyomva a táblázat bekerül a szövegbe.


A táblázat szerkesztése azt jelenti, hogy új objektumokat helyezünk a táblázatba, meglévőket törölünk, vonunk össze vagy osztunk fel. A táblázat formázása alatt a következőket értjük: oszlopszélesség és sormagasság megadása, igazítás megváltoztatása, a táblázat elhelyezése a lapon, a szövegirány, a keretek és a kitöltés megváltoztatása.

A **Táblázat** ideiglenes eszköztár **Tervezés és elrendezés** lapjain található eszközök segítségével valamennyi szerkesztési és formázási műveletet elvégezhetünk.

Felelj a kérdésekre!

- 1°. Miért használunk táblázatokat a szöveges dokumentumokban? Miből áll a táblázat? Milyen objektumokat tartalmaz?
- 2°. Milyen tulajdonságai lehetnek a táblázatnak, és ezek milyen értékeket vehetnek fel?
- 3°. Hogyan helyezhetünk el táblázatot a szöveges dokumentumban?
- 4°. Mit értünk az „összetett szerkezetű táblázat” kifejezés alatt? Hogy a legkényelmesebb ilyen táblázatot létrehozni?
- 5°. Hogy jelölhetjük ki a táblázat egy objektumát?
- 6°. Hogy szúrhatunk be új sort vagy oszlopot a táblázatba?
- 7*. Miben különbözik a **Delete** és a **Backspace** billentyűk használata kijelölt táblázat-rész esetében?



- 8*. A táblázat végére új sort szeretnénk beszúrni. Hogy tehetjük ezt meg gyorsan? Hogy illeszthetünk új sort a táblázat közepébe?
- 9*. Hogy változtathatjuk meg a sorok és oszlopok méretét?
- 10*. Milyen irányba lehet a cellában elhelyezett szövegnek? Hogy változtathatjuk ezt meg?
- 11*. Hogy helyezkedik el a cellában a szöveg a  gomb lenyomása után?
- 12*. Mikor lehet szükség a cellák egyesítésére? Mikor lehet szükség a cellák felosztására, és hogy tehetjük ezt meg?
- 13*. Hol található a táblázat vezérlőelemei?

**Végezd el a feladatokat!**

1. Hozd létre az alábbi táblázatot! A dokumentumot mentsd **feladat 3.2.1.docx** néven a mappádba!

НАРАХУВАННЯ ЗАРОБІТНОЇ ПЛАТИ

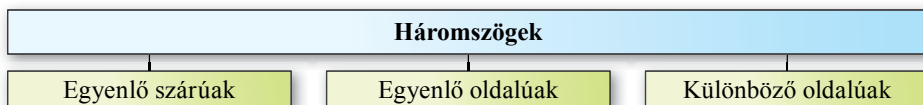
Прізвище	1 квартал	2 квартал	3 квартал	4 квартал
Іванов	2200.35	2200.35	2200.35	2200.35
Петров	3378.89	3378.89	3378.89	3378.89
Сидоров	5456.90	5456.90	5456.90	5456.90
Козлов	1456.67	1456.67	1456.67	1456.67
Андреев	4589.00	4589.00	4589.00	4589.00



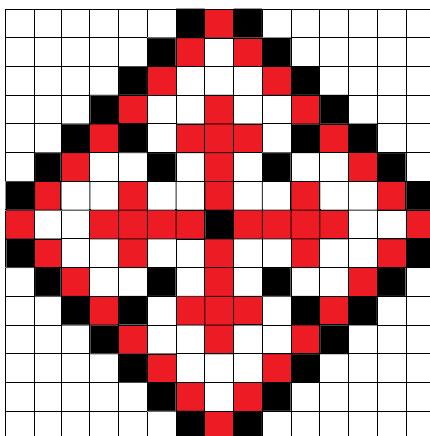
- 2*. Készítsd el egy szöveges dokumentumban az osztályod órarendjét a 3.4. táblázatban látható minta alapján! Mentse a dokumentumot **feladat 3.2.2.docx** néven a mappádba, és küldd el e-mailben a tanárodnak!
- 3*. Nyisd meg a tanárod által kijelölt fájlt (például a **zrazok 3.2.3.docx**-et a **Rozdil 3\pukt 3.2** mappából)! Töröld a táblázat utolsó sorát, illetve oszlopát! Illessz be a táblázat első oszlopa elé egy új oszlopot, és sorszámozd azt! Illessz be a táblázat első sora elé egy új sort! Egyesítsd a sor celláit, és írd be bele a táblázat címét! Mentse a dokumentumot **feladat 3.2.3.docx** néven a mappádba!
- 4*. Nyisd meg a tanárod által kijelölt fájlt (például a **zrazok 3.2.4.docx**-et a **Rozdil 3\pukt 3.2** mappából)! Formázd a táblázatot az alábbiakban megadottak alapján! Mentse a dokumentumot **feladat 3.2.4.docx** néven a mappádba!

Objektum	Betűtípus	Betűméret	Betűszín	Stílus	Igazítás
Első sor	Comic Sans MS	14	Sötétzöld	Félkövér	Középre
Negyedik sor	Arial	12	Sötétkék	Dólt	Balra
Első oszlop	Times New Roman	14	Fekete	Normál	Középre

- 5*. Hozd létre egy dokumentumban az alábbi, a háromszögek felosztását bemutató táblázatot! Mentsd a dokumentumot **feladat 3.2.5.docx** néven a mappádba!



- 6*. Hozd létre egy dokumentumban az alábbi minta alapján a színes négyzetekből álló táblázatot! Mentsd a dokumentumot **feladat 3.2.6.docx** néven a mappádba!



- 7*. Hozd létre az aktuális hónap naptárát! Mentsd a dokumentumot **feladat 3.2.7.docx** néven a mappádba!
- 8*. Az előző feladatban olyan táblázatot hoztál létre, ami a szöveg elején helyezkedik el. A táblázatról sajnos lemaradt a neve. A **Súgó** segítségével állapítsd meg, hogy illeszthetünk szöveget a táblázat elé!

3.3. GRAFIKAI ELEMELÉTREHOZÁSA, SZERKESZTÉSE ÉS FORMÁZÁSA A SZÖVEGES DOKUMENTUMBAN



1. Nevezd meg a grafikai elemek tulajdonságait **Word 2007**-ben! Milyen értékei lehetnek ezeknek?
2. Milyen szerkesztési műveleteket végezhetünk a **Word 2007** grafikai objektumain? Hogyan hajtódnak végre?
3. Milyen formázási műveleteket végezhetünk a **Word 2007** grafikai objektumain? Hogyan hajtódnak végre?

A SZÖVEGES DOKUMENTUM GRAFIKAI ELEMELÉS AZOK TULAJDONSÁGAI

A 6. osztályban már megismertették, hogyan illeszthetők grafikai elemek egy szövegbe, továbbá ezen elemek néhány tulajdonságával is. Ezek rajzok, vázlatok, diagramok voltak.

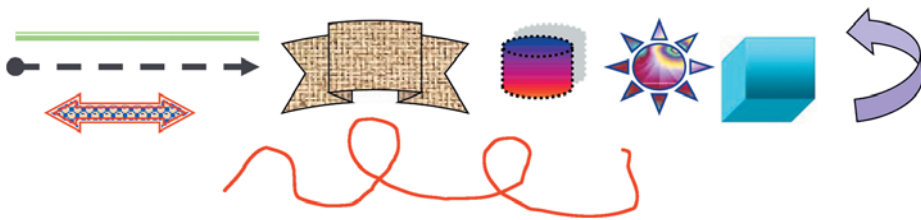


Az ilyen elemek leginkább illusztrációként szolgáltak. Azt is tudjátok, hogy a szövegbe illesztett grafikai elemek bizonyos tulajdonságokkal bírnak, ilyenek a *méret*, az *elhelyezkedés*, a *szöveg körbefuttatása*, a *fényerő*, a *kontraszt*, a *keretek*, a *kitöltés*.

A szöveges dokumentumba egyéb képi elemeket is beilleszthetünk. Az erre szolgáló eszközök a **Beszúrás** lap **Ábrák** csoportjában találhatók:

- **Kép** – tetszőleges hordozón található képfájl beillesztésére.
- **ClipArt** – a számítógépre telepített vagy az **Office Online** webhelyéről származó kollekciók képeinek beillesztésére.
- **Alakzatok** – vonalas ábrák beillesztésére.
- **SmartArt** – organogramok, vázlatok, szerkezeti diagramok beillesztésére.
- **Diagramok** – hisztogramok, kördiagramok, grafikonok beillesztésére.

A felsorolt típusok közül eddig a **Clipart** és a **SmartArt** objektumok, internetről vagy a személyes fotóalbumból származó képek beillesztésével foglalkoztatok. Ebben a paragrafusban olyan grafikai elemek beszúrásával foglalkozunk, amelyeket a **Word** grafikai elemeiből állíthatunk össze (3.20. ábra).



3.20. ábra. Különböző tulajdonságokkal rendelkező alakzatok példái

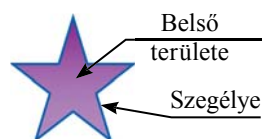
Az alakzatoknak két alapvető típusa létezik: a **vonala** és a **síkidom**. A **vonalas** elemeknek *alakja*, *vastagsága*, *típusa*, *színe*, *szaggatása* lehet. A 3.6. táblázatban ezek lehetséges értékeit foglaltuk össze.

3.6. táblázat

Vonalak tulajdonságai és azok értékei

Alak	Vastagság	Típus	Szaggatás	Szín
Egyenes	0,25 pt 0,5 pt 0,75 pt			
Görbe	1 pt 1,5 pt 2,25 pt			
Töröttvonal	3 pt 4,5 pt 6 pt			

A **síkidomokat** zárt töröttvonalak és az azok által határolt terület (az **alakzat belső területe**) alkotja (3.21. ábra). A síkidom lapjának lehet kitöltése. Ez a kitöltés lehet *egyszínű, színátmenetes, anyagmintával* vagy *képpel kitöltött* (3.7. táblázat).



3.21. ábra. Síkidom szegélye és belső területe

3.7. táblázat

Síkidomok kitöltése

Egyszínű kitöltés	Színátmenet	Minta	Anyagminta	Kép
A paletta tetszőleges színe	Néhány szín és folytonos átmenet ezek között	Mozaikszerű minta	A telepített anyagszerű minták egyike	Tetszőleges kép

KÉP KÉSZÍTÉSE A WORD GRAFIKAI ELEMEIBŐL



3.22. ábra. Alakzatok beillesztésére szolgáló eszközök

A **Word 2007**-ben használható alakzatok csoportokba vannak foglalva (3.22. ábra), mint például vonalak, téglalapok, egyszerű alakzatok, nyilak, csillagok és szalagok, folyamatábra stb.

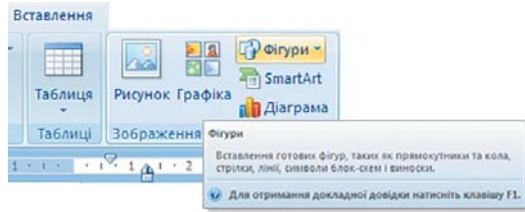
A **Word 2007**-ben több vonaltípust használhatunk, például egyenest, görbét, összekötő vonalakat, firkát, szabadkézi sokszöget. Ezek elkészítésére szolgálnak a következő eszközök:

- **Egyenes** , **Nyíl** , **Kettős nyíl** – egyenes vonalak beillesztésére. Létrehozásuk ugyanúgy történik, mint a **Paint** képszerkesztőben.
- **Firka** – tetszőleges, szabad kézzel rajzolt vonal beillesztésére. Elkészítése hasonlít ahhoz, ahogy vonalat rajzolunk a **Paint**-ben.
- **Görbe** – hajlított vonal elkészítésére.
- **Szabadkézi sokszög** – töröttvonalak beillesztésére.



Lássuk, hogyan illesztünk be töröttvonalat (hasonlóképpen történik a görbe beillesztése is):

1. Kattintsunk a **Beszűrés** lap **Ábrák** csoportjában az **Alakzatok** gombra (3.23. ábra)!
2. Kattintsunk a **Szabadjézi sokszög** eszköze!
3. Jelöljük ki a lapon a vonal kezdőpontját!
4. Kattintunk a lapon a töröttvonal csúcsainak helyére!
5. Dupla kattintással jelöljük ki a vonal végét!



3.23. ábra. Alakzat beillesztése

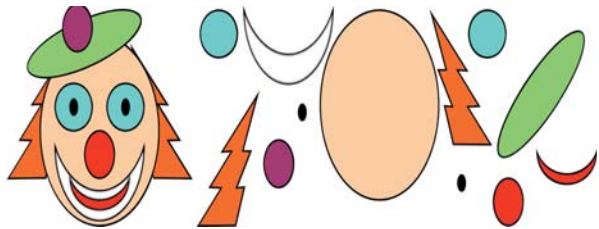
Zárt töröttvonalat úgy hozhatunk létre, hogy a kezdőpontot és a végpontot nagyon közel helyezzük el. A program automatikusan zárja a vonalat és kitölti a zárt területet.

Az **Alakzatok** csoport többi eleme adott formájú geometriai alakzat beillesztésére szolgál. Ilyenek a rombusz, a háromszög, a kettősnyíl, az ív, a mosoly-jel, a Hold stb. Ezek beillesztésének algoritmusai a következők:

1. Kattintsunk a **Beszűrés** lap **Ábrák** csoportjában az **Alakzatok** gombra (3.23. ábra)!
2. Válasszuk ki a megfelelő elemet!
3. Vigyük az egérmutatót a dokumentum munkaterületére!
4. Jelöljük ki a beillesztés helyét az egérrel!
5. Nyomjuk le a bal egérgombot a beillesztendő elem sarkába, és lenyomott gombbal húzzuk át az átellenes sarkába!
6. Engedjük el a bal egérgombot!

Megjegyzés:

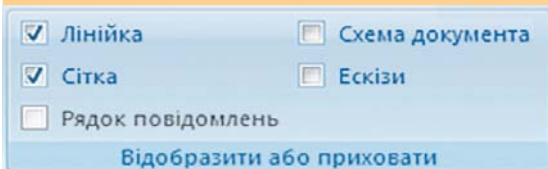
- ha ellipszis vagy téglalap rajzolásakor lenyomva tartjuk a **Shift** billentyűt, akkor szabályos alakzatot (kör, négyzet) kapunk;
- vonalak rajzolásakor a lenyomott **Shift** billentyű annak dőlésszögét 15 fokos lépésekben engedi megváltoztatni;
- ha az alakzatot lenyomott **Ctrl** billentyű mellett rajzoljuk, az alakzat nem a felső sarkától, hanem a középpontjától fog kirajzolódni;
- ha egy alakzatot ismételtelen meg szeretnénk rajzolni, a bal helyett a jobb gombbal válasszuk ki; az üzemmódból az **Esc** billentyű segítségével léphetünk ki;
- a pontosabb rajzolás céljából célszerű a dokumentumot rajzolás előtt nagyítani.



3.24. ábra. Alakzatokból összeállított képek

A dokumentumban minden rajzolt objektum külön jön létre, majd ezek másolásával, mozgatásával, egymásra helyezésével hozzuk létre a képet (3.24. ábra).

Ha többet szeretnél tudni









3.25. ábra. Rácsvonalak bekapcsolása

A rácsvonalakat csak nyomtatási elrendezés nézetben láthatjuk, nyomtatáskor nem. A jelölő kikapcsolása után a rácsvonalak nem lesznek láthatóak.

A rajzolást, méretezést, igazítást Word 2007-ben megkönnyíthetjük, ha bekapcsoljuk a **rácsvonalakat**. Ezt a **Nézet** lap **Megjelenítés** csoportjában tehetjük meg, ha bejelöljük a **Rácsvonalak** négyzetet (3.25. ábra). A rácsvonalak távolságát a **Lap elrendezése** ⇒ **Elrendezés** ⇒ **Igazítás** ⇒ **Rácsbeállításai** paranccsal szabályozhatjuk.


LibreOffice felhasználóknak

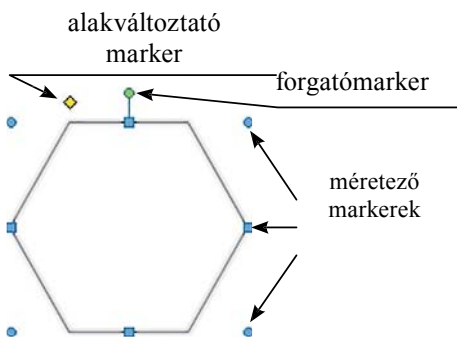
A grafikai elemeket a **Beszúrás: Kép, Média, Alakzat** parancsokkal vagy a **Grafika** eszköztár (3.26. ábra) segítségével illeszthetünk a dokumentumba. A legördülő menügombok az **Alapalakzatok** , **Szimbólumok** , **Nyilak** , **Csillagok** , **Képaláírások** , **Sémák**  csoportjait rejtik, ahonnan kiválaszthatjuk a megfelelő elemet.



3.26. ábra. Grafika eszköztár

GRAFIKAI ELEMELK SZERKESZTÉSEWORD 2007-BEN

Grafikai elemek önálló objektumok, amelyek a többi objektumtól függetlenül szerkeszthetők. Természetesen most is minden művelet a kijelölt objektumra vonatkozik. Kijelölni bal egérgombbal lehet egy objektumot. Amikor az egeret az objektum fölé visszük, a mutató alakja -ra változik.




3.27. ábra. A grafikai elem markerei

A kijelölt objektum körül megjelennek az egyes elemek fölött a *forgatómarkerek* is. Ezek négyzet és kör alakú jelek, amelyeket az elem körül láthatunk (3.27. ábra).

Ha az egeret a méretező markerek fülé viszszük, a mutató alakja kétirányú nyillá változik, amelyek a mozgatás lehetséges irányait jelzik. Ha a markert ezekben az irányokban mozgatjuk, megváltozik az objektum mérete.

A forgatómarker segítségével az objektumot tetszőleges mértékben elforgathatjuk. Eh-



hez csak meg kell fogni a markert (az egérmutató  alakot ölt), majd az egeret lenyomott bal gombbal a kör mentén mozgatni.

Vannak olyan elemek, amelyek *alakváltoztató markerrel* is bírnak, amit kis sárga rombusz jelöl. Az ilyen markerrel változtathatjuk meg például a nyíl hegyének hosszát (3.28. ábra). Itt minden elem az első nyíl módosításával jött létre.




3.28. ábra. Alakzat módosítása az alakváltoztató markerrel



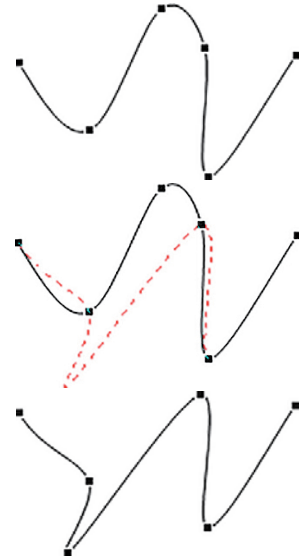
Ha többet szeretnél tudni

A vonalak alakját a csomópontok szerkesztésével (3.29. ábra) is megváltoztathatjuk. E célból:

1. Kattintsunk a **Rajz** ideiglenes eszköztár **Formátum** lapján az **Alakzatok beszurása** csoport **Alakzat szerkesztése**  gombjára!
2. Válasszuk a **Csomópontok szerkesztése** lehetőséget!
3. Mozgassuk a csomópontokat (az egérmutató kis fekete négyzetre vált) a megfelelő helyre!

Az alakzatok áthelyezése, másolása és törlése a már ismert módokon, a **Vágólap** alkalmazásával, áthúzással vagy billentyűkombinációkkal történhet.

Amennyiben az alakzat helyét csak kicsit kell megváltoztatni, a kurzormozgató billentyűket is használhatjuk. A mozgatás lépéshossza megegyezik az alapértelmezett beállításokkal. Ha csökkenteni szeretnénk a lépéshosszt, tartsuk lenyomva a mozgatás során a **Ctrl** billentyűt.



3.29. ábra. Görbe módosítása csomópontok szerkesztésével

GRAFIKAI ELEMOK FORMÁZÁSA WORD 2007-BEN

A létrehozott objektumnak megváltoztathatjuk a tulajdonságait, azaz formázhatjuk. A formázás alatt az alakváltoztatást, a vonal színének, stílusának módosítását, a kitöltés megváltoztatását értjük.

A formázást a **Rajz** ideiglenes eszköztár **Formátum** lapjának (3.30. ábra) eszközeivel végezhetjük, amely bármely grafikai objektum kijelölése után válik elérhetővé a **Menüsávon**.











3.30. ábra. A Formátum lap

Ezeknek az elemeknek a rendeltetését a 3.8. táblázatban foglaltuk össze. A formázás során jellemzően a művelet során azonnal láthatjuk annak eredményét.

3.8. táblázat

A Formátum lap eszközeinek rendeltetése





<i>Eszköz</i>	<i>Név</i>	<i>Rendeltetés</i>
Stílusok csoport		
	Gyorsstílusok	Objektum formázása megadott minta alapján
	Kitöltés	A síkidom területének kitöltése megadott színnel. A szín mellett kitöltési effektusok is rendelkezésre állnak
	Körvonal	A keret színének, a vonal vastagságának, stílusának megváltoztatása
	Alakzat cseréje	Alakzat cseréje a formázás megtartásával
Effektusok csoport		
	Árnyékok	Alakzat árnyékának testreszabása
	Térhatás	Térhatások testreszabása
Méret csoport		
	Magasság	Az alakzat magasságának beállítása
	Szélesség	Az alakzat szélességének beállítása

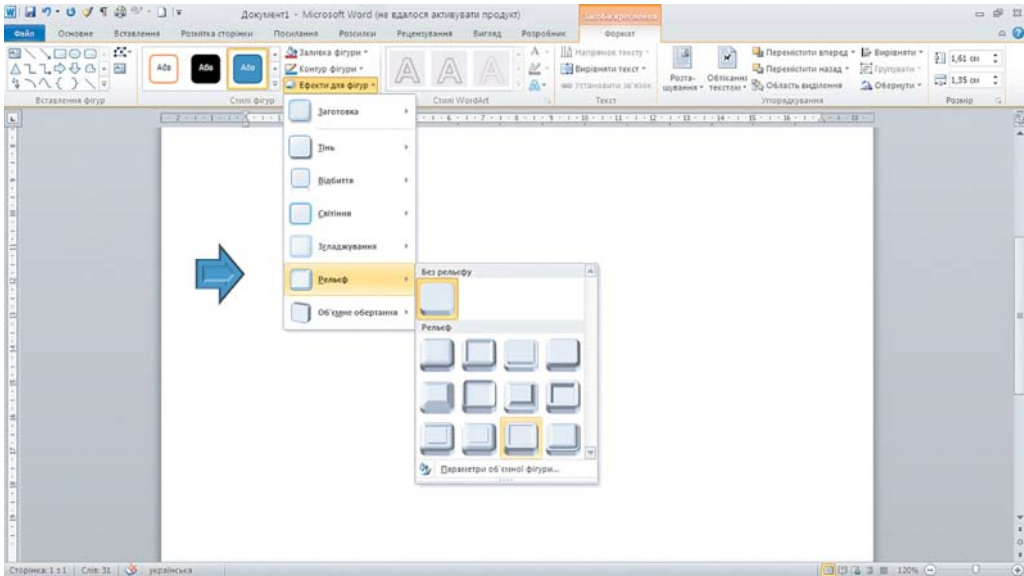
Az objektumok tulajdonságait az objektum helyi menüjében elérhető **Alakzat formázása** parancs segítségével is elvégezhetjük. A megnyíló ablakban a több fülön csoportosítva állnak rendelkezésre a formázási eszközök.

A szövegrészekhez hasonlóan grafikai elemekre is alkalmazhatjuk a minta alapján történő formázást, ami egy meglévő objektum formátumát örökíti át egy másik elemre.



Microsoft Office 2010-et használóknak

Az **Alakzat cseréje**  eszközt az **Alakzat szerkesztése**  legördülő menüben találjuk. Az **Árnyékok**  és **Térhatás**  eszközök nem léteznek, ezeket az **Effektusok** legördülő eszköztár helyettesíti (3.31. ábra), amely kész stílustárat bocsát rendelkezésünkre, ahol a fénytörést, visszaverődést, árnyékokat, térbeli forgatást állíthatjuk be.



3.31. ábra. Effektusok eszköztár



LibreOffice Writert használóknak

A grafikai elemek tulajdonságait az **Alakzat tulajdonságai** (3.32. ábra) eszköztár eszközeivel szabályozhatjuk.




3.32. ábra. Az Alakzat tulajdonságai eszköztár

GRAFIKAI OBJEKTUMOK CSOPORTJAIN VÉGEZHETŐ MŰVELETEK

Ha egy formázási műveletet nem egyetlen objektumon, hanem néhányon szeretnénk elvégezni, célszerű ezeket az objektumokat csoportba foglalni, és így elvégezni a műveleteket.

A **Word 2007**-ben az objektumok csoportjának kijelölését az egérrel végezhetjük el, ha közben a **Ctrl** billentyűt lenyomva tartjuk, ahogy azt a **Windows** operációs rendszer használata során a fájlokkal és mappákkal végzett műveletek során már megszokhattuk. Ezenkívül egy derékszögű területen elhelyezkedő valamennyi objektumot kijelölhetjük a

KezdőlapSzerkesztés csoportjában található **Kijelölés**  eszköz használatával. Ha erre a gombra kattintunk, majd körberajzolunk egy téglalap alakú területet, akkor az ott elhelyezkedő valamennyi grafikai objektum ki lesz jelölve.

Figyelmeztetés: az eszköz használata során a program grafikai objektumok kijelölése módba vált, a szöveg feldolgozása ebben a módban nem lehetséges. Az üzemmódból a gomb újbóli megnyomása segítségével léphetünk ki.

Ha azt szeretnénk, hogy az objektumok egyetlen objektumként viselkedjenek, kijelölés után *csoportba kell ezeket foglalni*. Ezután a szerkesztési és formázási műveletek az egyesített objektumra vonatkoznak. Ezt a csoportosítást akkor célszerű elvégezni, ha az objektumok szerkesztését befejeztük.

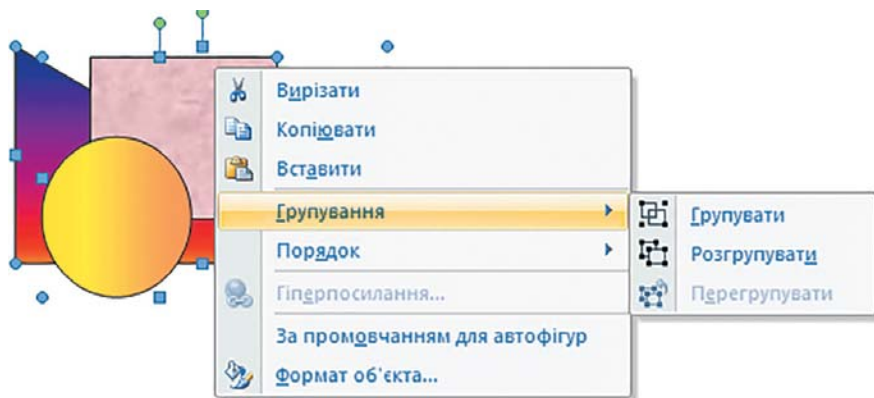
A csoportosítást a következőképpen végezhetjük:

1. Jelöljük ki a csoportosítandó objektumokat!
2. A **Formátum** lap **Rendezés** csoportjában válasszuk ki a **Csoportosítás** eszközt!
3. Hajtsuk végre a csoportba foglalást!

A csoportba foglalás után megmarad az egyes objektumok szerkeszthetőségének lehetősége is.

Szükség esetén a **Csoportosítás** eszköz alkalmazásával a csoportot *felbonthatjuk, újra-csoportosíthatjuk*.

A csoportosítás és csoportbontás műveleteit a kijelölt objektumok helyi menüjének **Csoportosítás** eszközével is elvégezhetjük (3.33. ábra).



3.33. ábra. A helyi menü **Csoportosítás** parancsa

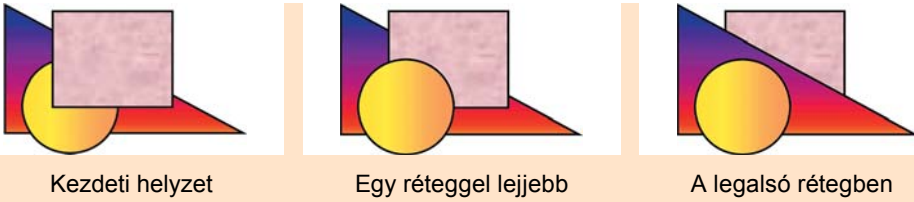


Ha többet szeretnél tudni

Grafikai objektumok rendezése

A grafikai objektumokból alkotott kép egy **többrétegű szerkezetet** alkot, egyes alakzatok mások fölött helyezkednek el. Az objektumok sorrendjét ebben a szerkezetben megváltoztathatjuk (3.34. ábra), egyes elemeket elmozdíthatunk a többihez képest.

Az objektumon és csoportokon egyéb műveleteket is végrehajthatunk, például megváltoztathatjuk a *helyzetüket, csoportosíthatjuk, forgathatjuk és tükrözhetjük* ezeket.



Kezdeti helyzet

Egy réteggel lejjebb

A legalsó rétegben

3.34. ábra. A négyzet mozgatása az alsóbb rétegek felé

Az objektumcsoportot elforgathatjuk 90° -kal balra vagy jobbra, vízszintesen vagy függőlegesen tükrözhetjük (3.35. ábra). A műveletet a **Formátum** lap **Rendezés** csoportjának **Forgatás** eszközével végezzük, miután kijelöltük az adott objektumot.



3.35. ábra. Grafikai objektumok csoportjának forgatása

A csoporton végzett valamennyi műveletet a **Formátum** lap **Rendezés** csoportjának (3.9. táblázat) eszközeivel, illetve a helyi menü parancsaival kezdeményezhetjük.

3.9. táblázat

A Formátum lap Rendezés csoportja eszközeinek rendeltetése

Eszköz	Név	Rendeltetés
	Elhelyezés a szövegben	Az objektum szöveghez viszonyított helyzetének megadása
	Előrehozás	Az objektum felemelése eggyel fentebbi rétegbe
	Hátraküldés	Az objektum leeresztése egy réteggel lejjebb
	Szöveg körbefuttatása	A szöveg körbefuttatásának magadása
	Igazítás	A csoport elemeinek egymáshoz viszonyított elhelyezése
	Csoportba foglalás	Objektumok csoportosítása, csoportosítás feloldása
	Forgatás	Objektum orientációjának megváltoztatása

Összefoglalás

A szövegben különböző képi elemeket – rajzokat, képeket, ábrákat stb. – helyezhetünk el. A képi elemek egy csoportját alkotják az alakzatok: vonalakból és az azok által bezárt területekből felépített geometriai elemek.

A **vonal** objektum tulajdonságai az *alak*, a *vastagság*, a *stílus*, a *típus*, a *szín*. A síkidomokat zárt vonalak alkotják. Ezek a zárt vonalak alkotják a keretet, ami az idom területét határolja. A körülhatárolt területnek kitöltése lehet.

A **Word 2007** alakzatai csoportokat – vonalak, nyilak, téglalapok, képalírások, csillagok, szalagok stb. – alkotnak. Az alakzatok kezelésére szolgáló eszközöket a **Rajz** ideiglenes eszköztár **Formátum** és **Elrendezés** lapjain találjuk.

Minden grafikai elemet külön illesztünk a szövegbe, ezek mozgatása, másolása, egymásra helyezése által keletkezik a kép.

A grafikai elemeken kijelölés után különböző műveleteket hajthatunk végre. Kijelölés után az objektum körül megjelennek a méretező markerek, a forgatómarkerek, egyes elemek esetében az átalakító markerek is. Ezekkel szerkeszthetjük az objektumokat. A másolás, áthelyezés és törlés a **Vágólap**, a billentyűkombinációk és az egér segítségével történhet.




A grafikai elemeket formázhatjuk is: megváltoztathatjuk a keretét, a kitöltését, különböző effektusokat rendelhetünk hozzájuk. Ezt a **SzalagFormátum** eszköztárának elemeivel végezhetjük, ami a grafikai objektum kijelölése után lesz elérhető.



Felelj a kérdésekre!

- 1°. Milyen alakzatokból állhat a **Word 2007**-ben létrehozott kép?
- 2°. Milyen tulajdonságai vannak a vonalnak? Milyen értékei lehetnek ezeknek?
- 3°. Milyen tulajdonságai lehetnek egy síkidomnak? Milyen értékei lehetnek ezeknek?
- 4°. Milyen vonalakat húzhatunk a **Word 2007**-ben? Milyen eszközök állnak rendelkezésünkre ehhez?
- 5°. Mi a **Shift** és a **Ctrl** billentyűk rendeltetése rajzoláskor?
- 6°. Mi a méretező, forgató és átalakító markerek rendeltetése?
- 7°. Milyen kitöltési effektusokat alkalmazhatunk a **Word 2007**-ben?
- 8°. Hogy jelölhetünk ki a szöveges dokumentumban grafikai objektumot, illetve objektumcsoportot?
- 9°. Milyen szerkesztési és formázási műveleteket végezhetünk az alakzatokon?
- 10°. Milyen célból csoportosítunk objektumokat a **Word 2007**-ben? Hogy tehetjük ezt meg?

**Végezd el a feladatokat!**

- 1°. A tankönyvben ismertetett módon rajzolj egy töröttvonalat! Készíts belőle néhány másolatot, és formázd azokat különbözőképpen!
- 2°. Rajzolj egy csillagot! Készíts belőle 5 másolatot, és helyezd el ezeket egy kör mentén!
-  3°. Rajzolj trapézt! Készíts belőle néhány másolatot! Rendelj a másolatokhoz különböző árnyékot, térbeli elforgatást, kitöltést!
- 4°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\Punkt 3.3\prava 3.3.4.docx**-et)! Helyezd el az objektumokat 3–1–2 sorrendben! Méretezd át az objektumokat: a kicsiket növeld, a nagyokat csökkentsd!
- 5°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\Punkt 3.3\prava 3.3.5.docx**-et)! Töröld belőle a nyilakat! Változtasd meg az objektumokat az alakváltoztató marker segítségével!
-  6°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\Punkt 3.3\prava 3.3.6.docx**-et)! Készíts az objektumról három másolatot! Forgasd el ezeket különböző irányokban!
- 7°. Rajzolj egy csillagot és egy kört, majd csoportosítsd ezeket!
-  8*. Keress a számítógépen WMF formátumú fájlokat! Illeszd a fájl tartalmát a szövegbe! Bontsd fel az objektumcsoportot, ami a képet alkotta, majd formázd ezeket az objektumokat különbözőképpen!

**3.4. SPECIÁLIS GRAFIKAI OBJEKTUMOK LÉTREHOZÁSA, SZERKESZTÉSE ÉS FORMÁZÁSA**

1. Hogy írhatunk a szövegbe olyan matematikai jeleket, mint például x_3 , x^3 , $\frac{3}{8}$, α ?
2. Milyen grafikai objektumokat helyezhetünk el a szövegben?
3. Milyen szerkesztési és formázási műveleteket végezhetünk el a **Word 2007** grafikai objektumain? Milyen eszközökkel tehetjük ezt meg?

KÉPLETEK BEILLESZTÉSE SZÖVEGES DOKUMENTUMOKBA

Ha egy referátumot írunk, vagy a diákkonferencián tartandó előadásunkhoz készülünk, szükség lehet arra, hogy a szövegbe képleteket, egyenleteket, tételek bizonyításait illesszük be, amihez speciális matematikai, fizikai, kémiai jelek szükségesek.

Az egyszerű matematikai kifejezések lineáris alakúak, vagyis az elemeit egy sorba írhatjuk. Az ilyen kifejezés nem tartalmaz gyökjeleket, törtet és egyéb nemlineáris struktúrákat. Például:

- a) $c_2m_2(t_2 - \theta) = c_1m_1(\theta - t_1)$;
- b) $ax^2 + bx + c \geq 0$;
- c) $(x^3 + 2x^2y - 5xy^2 - 3y^3)(5x - 4y)$;
- d) $\text{Na}_2\text{CO}_3 + 2\text{HCl} \Leftrightarrow 2\text{NaCl} + \text{CO}_2 + \text{H}_2\text{O}$.

3. fejezet

Az ilyen kifejezések beírása során szükség lehet speciális szimbólumokra, amelyeket a **Beszűrés** ⇒ **Szimbólumok** ⇒ **Szimbólum** Ω ⇒ **Egyéb szimbólumok** eszközzel illeszthetünk be.

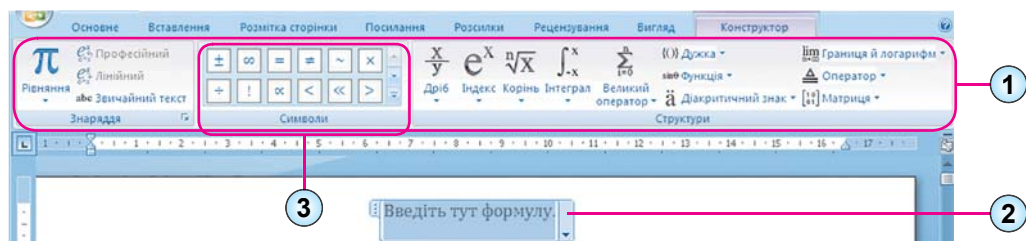
Ezenkívül használhatjuk még a **Kezdőlap Betűtípus** csoportjának **Alsó index** x_2 és **Felső index** x^2 eszközt.

Előfordulnak azonban nemlineáris többszintes struktúrák is, például:

$$\begin{aligned} \text{a) } \frac{h_1}{h_2} &= \frac{\rho_2}{\rho_1}; & \text{b) } E_k &= \frac{mv^2}{2}; & \text{c) } \eta &= \frac{A_1}{A_2} \cdot 100\%; & \text{d) } \frac{3x+7}{4} - \frac{x-3}{2} &= \frac{5x+2}{8}; \\ \text{e) } y &= \begin{cases} -3x^3 - 2, & \text{ha } x < 0, \\ \frac{7x}{4} - 2, & \text{ha } x \geq 0; \end{cases} & \text{f) } \frac{5a^4 - 20b^2}{2a^2 + b^2} &: \frac{30(a-4b)^2}{9a^4 - b^4}. \end{aligned}$$

Ilyen képleteket az **Egyenletszerkesztő** segítségével készíthetünk, amely tartalmazza az ilyen struktúrák beillesztéséhez szükséges sablonokat.

Az **Egyenletszerkesztőt** a **Beszűrés** ⇒ **Szimbólumok** ⇒ **Egyenlet** π paranccsal indíthatjuk. Indítása után a **Menüszalagon** megnyílik az **Egyenleteszközök** ideiglenes eszköztár (3.36.1. ábra), a dokumentumba pedig bekerül egy speciális beviteli mező, ahová a képlet beilleszthetjük (3.36.2. ábra).



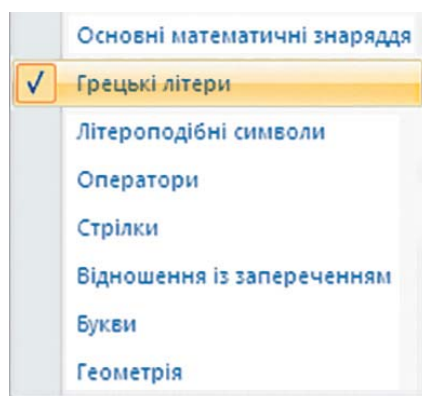
- | | |
|-----------------------------------|---|
| 1. Az Egyenleteszközök lap | 3. Szimbólumok beillesztésére szolgáló gombok |
| 2. Képlet beírásának helye | |

3.36. ábra. Az **Egyenleteszközök** lap

A képlet beírása során a betűket, műveleti jeleket a billentyűzettel, a speciális jeleket a **Szimbólum beillesztése** eszközzel írjuk az egyenletbe.

A **Menüszalag** csak azokat a szimbólumokat tartalmazza, amelyeket legutóbb használtunk (3.36.3. ábra). Ha másik szimbólumcsoportra van szükségünk, használjuk a \uparrow és a \downarrow jeleket, vagy a \square jel segítségével nyissuk meg a **Szimbólumcsoportok** listáját. A teljes listát akkor láthatjuk, ha az aktuális csoport címére kattintunk.

A beilleszthető szimbólumok nyolc csoportot alkotnak (3.37. ábra), amelyeket blokkokra bontottak (3.38. ábra).



3.37. ábra. Szimbólumcsoportok



A szimbólumokon kívül rendelkezésünkre állnak különböző matematikai struktúrák – törtek, függvények, zárójelek, gyökök stb. – sablonjai is. Ezeket a **Struktúrák** csoportban találjuk.

A struktúrák típusonként csoportokba vannak foglalva, amelyeket a megfelelő legördülő menükre (3.39. ábra) kattintva tekinthetünk meg. Ha a megfelelő sablonra kattintunk, az bekerül a szerkesztett kifejezésbe.

A sablont természetesen fel kell tölteni adatokkal. A beillesztés helyei szaggatott vonallal vannak kijelölve (3.40. ábra).

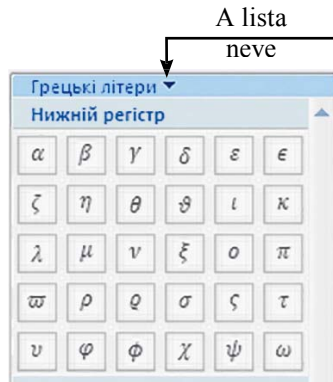
A mezőkbe nemcsak számokat és betűket illeszthetünk, hanem sablonokat is.

A képleten belül az egérmutatóval vagy a kurzormozgató billentyűkkel mozoghatunk. A képletszerkesztőből akkor lépünk ki, ha a képlet területén kívülre kattintunk.

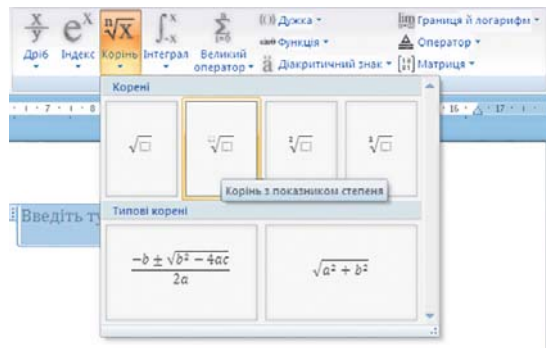
A képlet szerkesztése – elemek beillesztése, törlése, áthelyezése, megváltoztatása – a **Word 2007**-ben megszokott módokon történik. A formázás során megváltoztathatjuk az egyes szimbólumok tulajdonságait, beállíthatjuk a sortávot, igazítást, a képlet elhelyezkedését a szövegben. Ezt a képlet helyi menüjének (3.41. ábra) eszközeivel tehetjük meg.

A megszerkesztett képletet a felhasználó hozzáadhatja a beépített képletekhez. E célból!

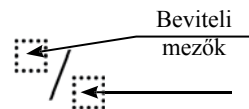
1. Jelöljük ki a képletet!
2. Hajtsuk végre a **Tervezés** ⇒ **Eszközők** ⇒ **Egyenlet** parancsot!
3. Kattintsunk a **Kijelöltek mentése az egyenletek gyűjteményébe** linkre!
4. A megnyíló párbeszédablakba írjuk be az egyenlet nevét, és adjuk meg az egyéb paraméterek értékeit!
5. Kattintsunk az **OK** gombra!



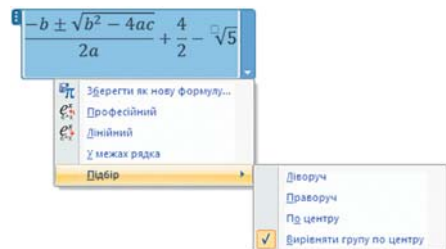
3.38. ábra. Szimbólumok listája



3.39. ábra. A **Struktúrák** csoport és **Gyökök** sablonok



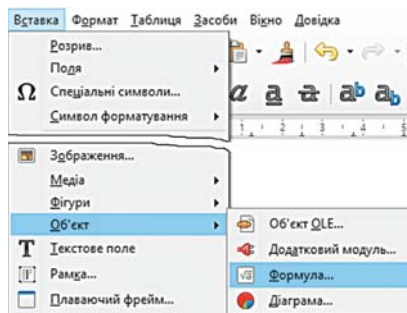
3.40. ábra. Tört sablonja



3.41. ábra. A képlet helyi menüje

LibreOffice-használóknak

Képletet a **Beszúrás** ⇒ **Objektum** ⇒ **Képlet** (3.42. ábra) paranccsal illeszthetünk a szövegbe. A továbbiakban az ablak bal oldalában megnyíló panelen található sablonok közül választhatunk. A lineáris szerkezetű képlet beírására a program ablakának alsó részében van lehetőségünk, ekkor a képlet a szövegben jelenik meg.




3.42. ábra

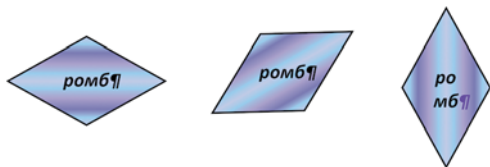
Ha többet szeretnél tudni

Feliratok a szöveges dokumentumban

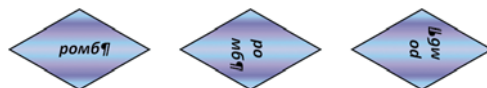
A szövegben különböző feliratokat is elhelyezhetünk, ha például a szöveget hozzáadjuk egy alakzathoz. A **Word**-nek azonban van egy speciális objektuma is erre a célra, a **WordArt**.

Az alakzatok többségének területén elhelyezhetünk szöveget is, erre az alakzat helyi menüjének **Szöveg hozzáadása** menüpontjának segítségével nyílik lehetőség. Ha az alakzatot forgatjuk vagy tükrözzük, a művelet a szövegre is vonatkozik majd (3.43. ábra).


Az alakzathoz rendelt szöveget a megszokott módokon szerkeszthetjük. A szöveg iránya vízszintes vagy függőleges lehet, ezek között a  gomb lenyomásával választhatunk. A gombot a **Formátum** lap **Szöveg** csoportjában találjuk. A szöveg elhelyezésének lehetőségeit a 3.44. ábra szemlélteti.



3.43. ábra. Szöveg elhelyezése alakzatokban rombusz



3.44. ábra. Szövegirányok az alakzatokban

Ha szöveges feliratot grafikai objektumként szeretnénk elhelyezni, használjuk a **Beszúrás** lap **Ábrák** csoportjának **Szövegdoboz**  eszközét.

A szöveg szerkesztése a szövegdobozban az ismert módszerekkel történik, a szövegdoboz szerkesztése pedig ugyanúgy, mint valamennyi grafikai objektumé.

A szövegben **WordArt** objektumokat (3.45. ábra) is elhelyezhetünk, ezek a szöveg művészi formázását teszik lehetővé.


Бажаємо успіху!

3.45. ábra. WordArt objektum






A **WordArt** objektumoknak kétféle – *szöveges* (betűtípus, méret, szín, stílus) és *grafikai* (méret, elhelyezkedés, kitöltés, vonalszín, árnyék stb.) – tulajdonságai vannak.

A **WordArt** beillesztése a következőképpen történik:

1. Hajtsuk végre a **Beszűrés** ⇒ **Szöveg** ⇒ **WordArt** parancsot .
2. Válasszunk stílust!
3. Írjuk be a szöveget!
4. Kattintsunk a mezőn kívülre!

Az objektum a szövegkurzor helyére kerül majd. Az objektum formázása és szerkesztése a **Formátum** ideiglenes eszköztár segítségével történik!

! Összefoglalás

Egyszerű képletek beírása során szükség lehet a speciális szimbólumok beillesztésére. Ezt a **Beszűrés** ⇒ **Szimbólumok** ⇒ **Szimbólum**  ⇒ **Egyéb szimbólumok** paranccsal tehetjük meg. Ezenkívül használhatjuk még a **Kezdőlap Betűtípus** csoportjának **Alsó index**  és **Felső index**  eszköztét.

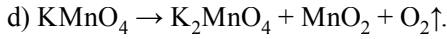
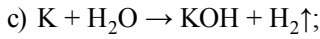
Az összetett, többszintű képletek beillesztése a **Képletszerkesztő** segítségével történik. A beillesztett képletek szimbólumokból és speciális sablonokból kialakított struktúrákból állnak, amelyeket ki kell tölteni tartalommal.

? Felelj a kérdésekre!

- 1°. Hogyan írhatunk be vegyjeleket tartalmazó képleteket?
- 2°. Hogy illeszthetjük a szövegbe a \pm \geq \neq \approx matematikai jeleket?
- 3°. Milyen eszközökkel írhatunk képletet a szöveges dokumentumba?
- 4°. Milyen matematikai szimbólumokat használhatunk a képletekben? Hogy illeszthetjük be ezeket?
- 5°. Milyen struktúrákat tartalmazhat egy képlet? Nevezd meg néhányat ezek közül!
- 6°. Mi a sablon? Milyen elemekből áll, és hogy használjuk?
- 7°. Milyen szerkesztései műveleteket végezhetünk a képlettel? Hogyan végezzük ezeket?
- 8°. Milyen formázási műveleteket végezhetünk a képlettel? Hogyan végezzük ezeket?

Végezd el a feladatokat!

- 1°. Hozz létre egy szöveges dokumentumot! Írd be az alábbi szöveget, majd mentsd a fájlt **feladat 3.4.1.docx** néven a mappádba! „A Coca-Cola összetevői: H₂O víz, CO₂ szén-dioxid, H₂CO₃ szénsav, H₃PO₄ foszforsav, C₁₂H₂₂O₁₁ szacharóz, C₈H₁₀N₄O₂ koffein stb.”.
- 2°. Hozz létre egy szöveges dokumentumot! Írd be az alábbi kémiai képleteket, majd mentsd a fájlt **feladat 3.4.2.docx** néven a mappádba!
 - a) $O_2 + 2H_2 \rightarrow 2H_2O$;
 - b) $AgCl \rightarrow Ag + Cl_2$;



- 3• Hozz létre egy szöveges dokumentumot! Írd be az alábbi fizikai képleteket, majd mentsd a fájlt **feladat 3.4.3.docx** néven a mappádba!

a) a forgás periódusa: $T = \frac{t}{N}$;

b) a közlekedő edények egyenlete: $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$,

c) a kinetikus energia: $E_k = \frac{mv^2}{2}$;

d) a hatásfok $\eta = \frac{A_1}{A_2} \cdot 100\%$.

- 4• Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.4.4.docx** néven a mappádba!

a) $\frac{3}{4}(12 - x) = \frac{3}{8}$;



c) $\frac{7}{8}(2 - x) = \frac{1}{8}$;

b) $\frac{3}{7}(1 - 2x) + \frac{1}{7} = -2$;



d) $\frac{2}{3}(x - 4) + \frac{1}{3} = 5$

- 5• Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.4.5.docx** néven a mappádba!

a) $y = \begin{cases} x^2 + 3, & \text{ha } x \leq 0, \\ \frac{x}{2} + 3, & \text{ha } x > 0; \end{cases}$



c) $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$

b) $y = \begin{cases} -3x^3 - 2, & \text{ha } x < 0, \\ \frac{7x}{4} - 2, & \text{ha } x \geq 0; \end{cases}$



d) $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6 \end{cases}$

- 6• Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.4.6.docx** néven a mappádba!

a) $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}$;



c) $\frac{a + \frac{25}{a+10}}{\frac{25}{25-a} - a}$,

b) $\frac{x}{7a^2b^3} \cdot \frac{6a^3c^2}{b} = \frac{y}{4c}$;



d) $1 - \frac{1}{1 - \frac{a}{1 - \frac{1}{a+1}}}$



4. SZÁMÚ GYAKORLATI MUNKA

Különböző objektumokat tartalmazó szöveg szerkesztése

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Nyisd meg a **zrazok pr4.docx** fájlt a **Rozdil 3** prakticsna 4 mappából! Formázd a dokumentumot az alábbiak szerint:

Дорогі діти!

Юнацька туристична фірма «Топ-топ» пропонує провести незабутні веселі канікули на легендарному острові Хортиця!

На вас чекають:


- ❖ подорожі,
- ❖ екскурсії,
- ❖ дискотеки,
- ❖ туристичні естафети,
- ❖ спортивні конкурси,
- ❖ козацькі розваги,
- ❖ прогулянки по річці,
- ❖ дивовижні схили,
- ❖ піщані пляжі,
- ❖ історичні місця.

Для проживання можна вибрати:

1. Табори пластунів (7 днів, до 20 дітей)
2. Стаціонарні корпуси (20 днів, до 500 дітей)
3. Дерев'яні будинки (10 днів, до 100 дітей)

Вартість путівок:

Термін	Ціна (за 1 місце в день) у грн.		
			
Березень – травень	20	50	70
Червень – серпень	30	70	100
Вересень – жовтень	20	50	70

 Заповіти путівки можна:
на сайті www.top-top.zp.ua або
електронною поштою admin@top-top.zp.ua

2. Hozd létre a dokumentumban a *Top-top* turisztikai vállalkozás logóját az alábbi minta szerint!



3. Egészítsd ki a szöveget a következőkkel!

a) a szilárd test sűrűsége: $\rho = \rho_0 \frac{P}{P - P_1}$;



c) $\frac{3x + 7}{4} - \frac{x - 3}{2} = \frac{5x + 2}{8}$.

4. Mentsd a dokumentumot **prakticsna robota 4.docx** néven a mappádba!



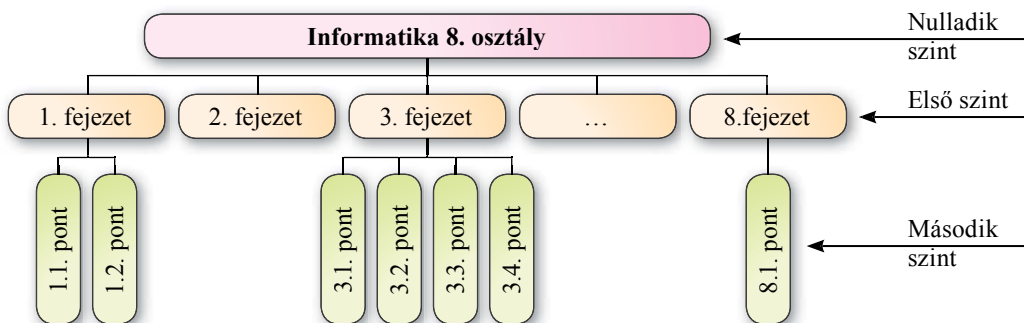
3.5. SZAKASZOK, FEJLÉCEK, STÍLUSOK. A DOKUMENTUM SZERKEZETE



1. Milyen tulajdonságait ismerjük a szöveg oldalainak? Hogy állíthatjuk ezeket be?
2. Milyen dokumentumnézetek léteznek **Word 2007**-ben? Hogyan váltunk ezek között?
3. Hogyan van strukturálva az informatika tankönyv? Milyen objektumokat találhatunk az oldalain?

A DOKUMENTUM STRUKTÚRÁJA

A hosszú, sok oldalt tartalmazó dokumentumokat (referátumok, évfolyammunkák, brosrák, könyvek stb.) kényelmes szerkezeti elemekre – részek, fejezetek, paragrafusok, pontok – kell felosztani, így módon a dokumentum egy hierarchikus szerkezetet kap. Ilyesféle struktúrákkal már találkozhatunk az irodalomórán, amikor a fogalmazásokhoz vázlatot készítettünk.



3.46. ábra. Dokumentum struktúrája

Ha megtekintjük ennek a tankönyvnek a felépítését (3.46. ábra), akkor láthatjuk, hogy annak első szintjén a fejezetek, a másodikikon a pontok, a harmadikon az alpontok találhatók.

A dokumentum struktúrája alatt tehát a dokumentum részeiből álló hierarchikus szerkezetet értjük.

A DOKUMENTUM SZAKASZAI


Ahhoz, hogy a dokumentumot olyan módon osszunk részekre, hogy ezek a részek ott kezdődjenek, ahol arra szükség van, **Szakasztöréseket** kell elhelyezni a dokumentumban. A dokumentumban minden szakasz előtt elhelyezhetünk ilyen töréspontot. Amennyiben ezt nem tesszük, a dokumentum egyetlen szakaszt alkot.

A szakasztörést többféleképpen helyezhetjük el:

- a következő oldalon – a szakasz a következő oldalon kezdődik;
- folytonosan – a szakasz a dokumentum aktuális pozíciójában kezdődik;




- páros oldalon – a szakasz a következő páros oldaltól kezdődik;
- páratlan oldalon – a szakasz a következő páratlan oldaltól kezdődik.

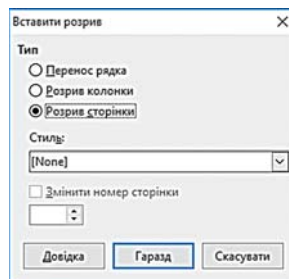
A szakaszokat **Vázlat** nézetben célszerű létrehozni, továbbá hasznos, ha be van kapcsolva a *nem látható jelek* mutatása is. Ezután a **Lap elrendezése** ⇒ **Oldalbeállítás** ⇒ **Töréspontok**  parancs végrehajtása után ki kell választani a megfelelő beillesztési módot. Ha be van kapcsolva a mindent mutató mód, a szövegben láthatjuk a **Szakasztörés** jeleket.

A szöveges dokumentum általában egy szakaszból áll, minden oldal beállítása (oldalmargó, margók, orientáció) ugyanolyan. Hosszú dokumentumok esetében azonban előfordulhat, hogy egyes oldalak paraméterei eltérőek legyenek. Például egy olyan dokumentumba, amelyben az oldalak tájolása álló, el kell helyezni egy olyan sok oszlopból álló táblázatot, amelyhez a fekvő tájolás a megfelelő. Ezt a problémát szakasztörések elhelyezésével orvosolhatjuk.

Ha el kell távolítani a dokumentumból a szakasztörést, törölnünk kell a szakasztörés rejtett szimbólumot a szövegből. Ekkor a szakasztörést követő szakasz egyesül az előzővel, emiatt formátuma is automatikusan megváltozik.

LibreOffice-használóknak

Új szakasz beszúrását a **Beszúrás** ⇒ **Töréspont** parancssal végezzük, majd a megnyíló párbeszédablakban (3.47. ábra) be kell állítani a töréspont elhelyezésének módját. Szakasztörést a **Standard** eszköztár **Töréspont**  eszközével is elhelyezhetünk a szövegben.



3.47. ábra. Töréspont beszúrása ablak

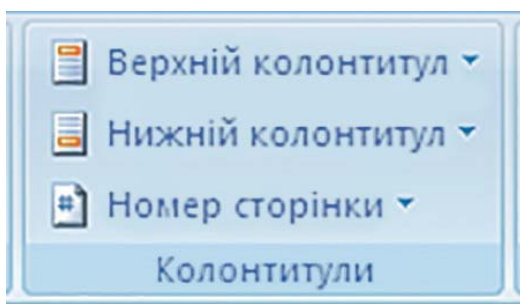
FEJLÉCEK ELHELYEZÉSE

Hosszú dokumentumok esetében gyakran alkalmaznak fejléceket (a francia *colonne*-oszlop és a latin *titulus* – cím összevonásából): ezek a dokumentum oldalainak margóin elhelyezkedő „szolgálati” információkat tartalmazó mezők.

A **Word 2007**-ben felső (*fejléc*), alsó (*lábléc*) és *oldalsó* fejléceket használhatunk. Ezek a fejlécek leginkább a dokumentum és szerző nevét, az oldalszámot, a fejezet címét, grafikai elemeket tartalmaznak. Nézd meg most figyelmesen ebből a szemszögből a tankönyv oldalait.

Az első oldal fejléce különbözhet a többitől, ugyanígy különbözhet a páros és páratlan oldalak fejlécei, vagy az egyes fejezetek fejlécei.

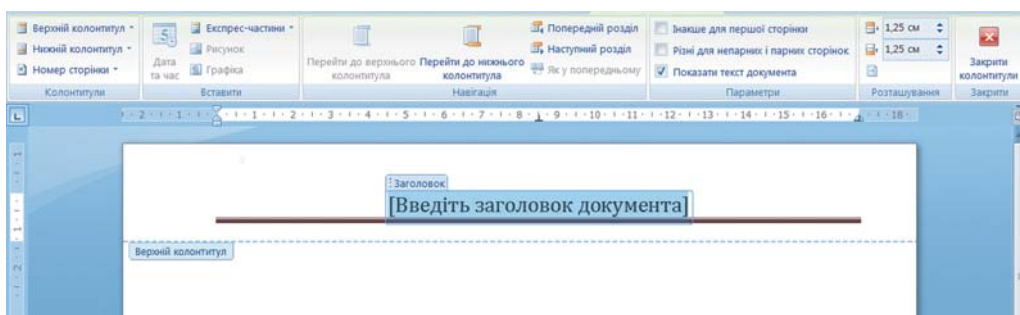
A **Word 2007**-ben beilleszthetünk sablon alapján készült fejléceket, de saját fejléceket is létrehozhatunk, amit aztán el is menthetünk a sablonok közé. A fejléceket csak **Nyomtatási elrendezés** és **Olvasás** nézetben láthatjuk. A fejléc szerkesztése közben a szöveg szerkesztése nem lehetséges.



3.48. ábra. Az **Élőfej és élőláb** csoport

Fejléceket a **Beszűrés** lap **Élőfej és élőláb** csoportjának eszközeivel helyezhetünk el a dokumentumban (3.48. ábra).

A fejléc beillesztése után a **Szalagon** elérhetővé válik a **Tervezés** ideiglenes eszköztár (3.49. ábra). Segítségével különböző elemeket – **Oldalszám**, **Dátum és idő**, **Kép**– helyezhetünk el a fejlécen. A fejléc helyét az oldalon a **Pozíció** lap elemeinek segítségével szabályozhatjuk.



3.49. ábra. Az élőfej sablonja és a **Tervezés** ideiglenes eszköztár

Ha kész sablonból szeretnénk elhelyezni fejléceket, a következőképpen járjunk el:

1. Nyissuk meg a **Beszűrés** lapot!
2. Válasszuk ki az **Élőfej és élőláb** csoportból az **Élőfej** vagy az **Élőláb** elemet!
3. Válasszunk a felajánlott sablonok közül!
4. A **Tervezés** ideiglenes eszköztár segítségével végezzük el a finomításokat!
5. Írjuk be a sablonba a megfelelő szöveget!
6. Zárjuk be a fejléceket az **Élőfej és élőláb eszközök** ⇒ **Tervezés** ⇒ **Bezárás** ⇒ **Bezárás** gombbal!

A létrehozott fejléceket alkalmazhatjuk a teljes dokumentumra, a páros vagy a páratlan oldalakra vagy csak az első oldalra. A fejléc alkalmazási területét a **Tervezés** lap **Beállítások** csoportjában található eszközökkel végezzük.

Az élőfej és élőláb között a **Navigáció** lap **Váltás az élőfej és élőláb között** eszközzel válthatunk. Ezt az eszközt is a **Tervezés** eszköztárban találjuk.

Amennyiben a dokumentumban különböző fejléceket szeretnénk alkalmazni, osszuk a dokumentumot szakaszokra, ekkor szakaszonként különböző fejléceket hozhatunk létre.



A fejléceket természetesen szerkeszthetjük és formázhatjuk is. Ezt a **Beszúrás** ⇒ **Élőfej és élőláb** ⇒ **Élőfej (élőláb)** ⇒ **Élőfej (élőláb) szerkesztése** eszközzel kezdeményezzük.

A **Beszúrás** ⇒ **Élőfej és élőláb** ⇒ **Élőfej (élőláb)** ⇒ **Élőfej (élőláb) eltávolítása** eszközzel törölhetjük a fejléceket.



3.50. ábra. Élőfej

LibreOffice-használóknak

Fejléceket a **Beszúrás** menü **Élőfej** és **Élőláb** eszközeivel helyezhetünk el a dokumentumba. A parancs végrehajtása és a stílus kiválasztása után a felhasználó elvégezheti a fejléc szerkesztését (3.50. ábra).

A szabványos elemeken (oldalszám, dátum, oldalak száma, szerző stb.) kívül a fejlécben a **Standard** eszköztár **Mező beszúrása** legördülő menüs gombok segítségével egyéb elemeket is elhelyezhetünk. Ha befejeztük a fejléc szerkesztését, kattintsunk a fejléc területén kívülre.

A **Formátum** ⇒ **Oldal** paranccsal kiválaszthatjuk és az **Élőfej és élőláb formázása** eszközével megadhatjuk, hogy a fejléceket páros vagy páratlan oldalon használjuk-e, vagy szeretnénk az első oldalhoz eltérő fejléceket alkalmazni. Ezeket az eszközöket a szerkesztés alatt álló fejléc mellett találjuk meg.

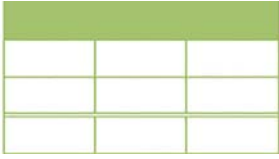

STÍLUSOK ALKALMAZÁSA A DOKUMENTUMBAN

Az előző órákon sokszor előfordult, hogy a dokumentum különböző objektumait – szöveget, táblázatot, képletet, **SmartArt** objektumokat – formáztatok. Ha a dokumentum hosszú és sok különböző objektumot tartalmaz, az objektumok formázása sok időt igényel. Ha ráadásul sok ugyanúgy formázott objektumot tartalmaz, külön erőfeszítést igényel a formázások megtartása.

A dokumentumok formázásának megkönnyítése és meggyorsítása céljából célszerű stílusokat alkalmazni. A **Stílus bizonyos objektum tulajdonságainak megadott értékeit tartalmazza**.

A **Word 2007**-ben alapértelmezetten elérhető a **stílustár**, amelyben az objektumok tulajdonságai harmonikusan vannak megválasztva, a mai dizájn kívánalmainak megfelelően. Ilyen stílusok a szöveges dokumentum különböző objektumai – szöveg, táblázat, grafikai objektumok – számára léteznek (3.10. táblázat). Minden stílus egyedi névvel rendelkezik. Ezek némelyikét már használtátok az eddigi szerkesztési munkáitok során.

Word 2007 stílusok

Objektum	Stílus neve	Minta	Tulajdonságok
Szöveg	Címsor4	<i>Ezek némelyikét már használtatok az eddigi szerkesztési munkáitok során</i>	Betűtípus: <i>Cambria</i> Méret: <i>12</i> Szín: <i>fekete</i> Stílus: <i>félkövér;dőlt</i> Sorköz: <i>1,5 sor</i> Első sor behúzása: <i>nincs</i> Igazitás: <i>sorkizárt</i> Térköz elötte: <i>3 pt</i>
Táblázat	Világos árnyékolás, 3 jelölőszín		Vonal színe: <i>zöld</i> Kitöltés: <i>első sor zöld</i> Vonalstílus: <i>folytonos</i> Vastagság: <i>0,5 pt</i>
Alakzat	Enyhe effektus, jelölőszín 6		Szín: <i>világosbarna</i> Átlátszóság: <i>0</i> Vonal: <i>folytonos</i> Vastagság: <i>1 pt</i> Vonalszín: <i>sötétbarna</i> Árnyék: <i>jobbra lent</i>

Stílusok alkalmazásával egyetlen művelettel több objektum tulajdonságát változtathatjuk meg. Ha például egy **Címsor4** stílusú bekezdést (3.10. tábl.) kézzel formázunk meg, az nyolc művelet elvégzését jelenti, míg a **Címsor4** stílus alkalmazásával egyetlen művelettel érhetjük el ugyanazt: a stílus kiválasztásával.

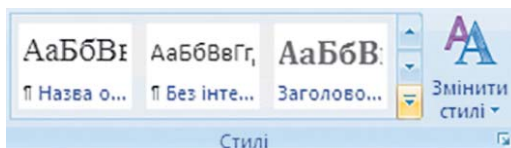
Nagyon fontos szerepe van a **Címsor1**, **Címsor2**, ..., **Címsor9** stílusoknak, amelyek alkalmazásával automatizálhatjuk a dokumentum szerkezetének kialakítását.

A stílusok gyűjteményét saját stílusokkal egészíthetjük ki, amelyeket a már létező stílusok alapján alakíthatunk ki. A dokumentum mentésekor automatikusan mentésre kerülnek az alkalmazott stílusok, hogy az újbóli megnyitáskor a dokumentum külalakja ne változzon.

A STÍLUSTÁR STÍLUSAINAK ALKALMAZÁSA

A stílustár stílusait a **KezdőlapStílusok** csoportjában (3.51. ábra) találjuk.

A stílustár minden eleme úgy került kialakításra, hogy annak külalakja megfelelően a választott stílusnak, a nevek melletti jelölés pedig azt mutatja, milyen



3.51. ábra. A stílustár stílusai



objektumra érvényesíthető az adott stílus: bekezdésekre – ¶, – a, vagy mindkettőre – ¶a. A címsorstílusokat a dokumentum strukturális egységeinek (fejezetek, pontok, alpontok) formázásakor használjuk.

A stílustár tetején mindig a legutóbb alkalmazott stílusokat találjuk. A ▲ és a ▼ gombokkal lapozhatjuk a további stílusok listáját, a ▾ gombra kattintva pedig megtekinthetjük a teljes stílustárat. Egy stílust mindig a szöveg aktuális elemére érvényesíthetünk, ha a stílus nevére kattintunk.

A stílus helyi menüjének segítségével a stílusokat átnevezhetjük és törölhetjük. Ezeket a műveleteket nem végezhetjük el a címsorstílusokon.

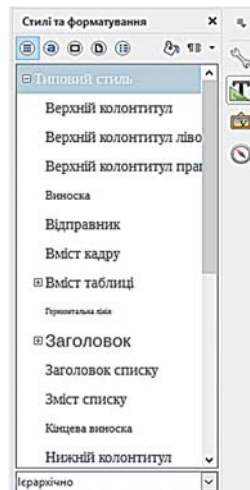
Ha el kívánunk tekinteni a stílusok alkalmazásától, és vissza szeretnénk térni az alapértelmezett beállításokhoz, kattintsunk a **Formázás törlése** gombra.

LibreOffice-használóknak

Stílusok alkalmazása céljából nyissuk meg a **Stílusok és formázások** (3.52. ábra) eszköztárat (*Formátum* ⇒ *Stílusok és formázások* paranccsal, vagy a **Stílusok és formázások** gombra kattintva).

Az eszköztárban a gombok a bekezdés , szimbólum , keret , oldal és listastílusok megnyitását teszik lehetővé.

A **Beszúrás** menü segítségével helyezünk el a dokumentumban hivatkozásokat, mutatókat, tartalomjegyzéket.

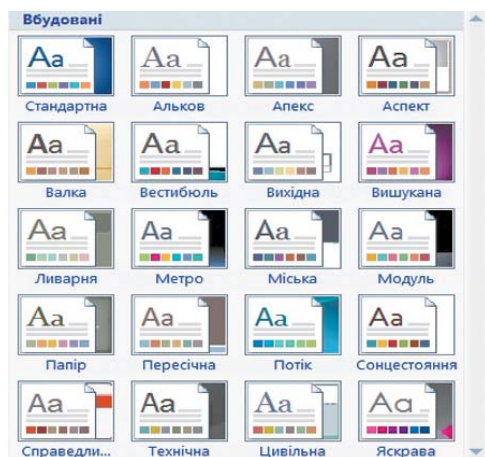


3.52. ábra. Stílusok és formázások ablak

Ha többet szeretnél tudni

Témák alkalmazása a dokumentumokban

Stílusokkal nemcsak egyes objektumokat formázhatunk meg, de a teljes dokumentumot is.



3.53. ábra. Beépített témák

A **Téma** a dokumentum egészére alkalmazott stílus. A témáknak egyedi neve van, és stíluselőírásokat tartalmaznak a dokumentumban alkalmazott objektumokra (szimbólumok színe, mérete, betűtípusa, bekezdések igazítása, sorköz, ábrák formázása stb.). Az 5. osztályban stílusok segítségével alakítottuk ki a prezentációkat.

A **Word 2007** beépített témáit a **Lap elrendezése** lapon a **Témák** csoportjában található **Témák** legördülő menüs gombbal érhetjük el (3.53. ábra).

Témát választhatunk a dokumentum létrehozásakor, de a szerkesztése során is. A témák áttekintése során a dokumentum dinamikusan formázódik az aktuális témának megfelelően. A téma kiválasztása értelem

szerűen kihat a szövegekre, táblázatokra, grafikai elemekre alkalmazott stílusokra is. Minden elem az egységes stílusnak megfelelően lesz formázva. *Fontos megjegyezni*, hogy a téma megváltoztatása csak a stílusokkal megformázott elemek külalakjára van kihatással.

A DOKUMENTUM SZERKEZETÉNEK KIALAKÍTÁSA ÉS ANNAK ALKALMAZÁSA




A dokumentum szerkezetét a címsorstílusok segítségével alakítjuk ki. A legmagasabb szintű strukturális elemhez értelemszerűen a **Címsor1** stílust rendeljük, a másodikhoz pedig a **Címsor2**-t és így tovább.

Ezt a következőképpen végezzük:

1. Jelöljük ki a dokumentum egy strukturális elemét (fejezetcím, pont, alpont neve stb.)!
2. Válasszuk ki a **Stílusok** csoportból a megfelelő szintű címsort!
3. Ismételjük az 1–2. pontokat valamennyi strukturális elemen!

Az így kialakított dokumentum szerkezetét **Vázlat** nézetben tekinthetjük meg, amit a **Nézet** lap **Dokumentumnézet** csoportjának eszközeivel, vagy a **StátuszsorVázlat** gombjára kattintva nyitunk meg.

Ez a dokumentumnézet eltér a megszokott nyomtatási nézettől (3.54. ábra). **Vázlat** nézetben a szerkezeti elemeket *összecsukva* mutatja, a képernyőn csak a szerkezeti elemeket láthatjuk. Hogy a szerkezet jól áttekinthető legyen, az azonos szintű elemekhez azonos mértékű behúzás tartozik. Az egyes szintek mellett a következő jeleket láthatjuk:

- , amennyiben a szinthez tartoznak alacsonyabb szintek és szöveg;
- , amennyiben a szinthez nem tartozik alárendelt szöveg;
- , amennyiben a szöveg a legalacsonyabb szinthez tartozik.

Програма ЮНЕСКО "Інформація для всіх"

1. Преамбула.

1. Інформація і знання є величезним загальнолюдським завоюванням. Вони необхідні для просування освіти, науки, культури і комунікацій, для збагачення культурної різноманітності і відвертості управління. Мандат ЮНЕСКО "сприяти вільному руху ідеї і справою" ясно указує на роль, яку організація покликана грати в тому, щоб зробити інформацію і знання доступними для всіх, з кінцевою метою подолання розриву між інформаційно-багатими й інформаційно-бідними.
2. Для досягнення цієї мети, відповідно до свого конституційного мандата ЮНЕСКО повинна "підтримувати, зберігати, збільшувати, і поширювати знання", гарантуючи і забезпечуючи "збереження і захист" зафіксованих знань людства. Більш того, Організація повинна сприяти "співпраці між народами... у обмін публікаціями" і іншими інформаційними матеріалами і ініціювати "способи міжнародної співпраці, розраховані на

Програма ЮНЕСКО "Інформація для всіх"

1. Преамбула.

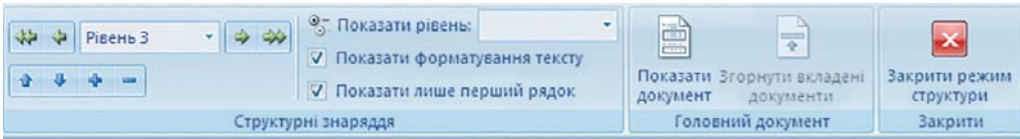
1. Інформація і знання є величезним загальнолюдським ...
2. Для досягнення цієї мети, відповідно до свого ...
3. Міжнародна інтелектуальна співпраця є надзвичайно ...
4. Технологічна конвергенція сама, що послідує, по ...
5. Нові способи доступу, обробки і захисту інформації ...
6. У цьому контексті ЮНЕСКО з своїм мандатом - ...

3. Цілі програми.

1. У дусі Міжнародної Декларації Прав Людини, яка формує ...
2. Як одна з основних програм ЮНЕСКО програма ...
3. Зокрема програма "Інформація для всіх" націлена на:
 - Заохочення і розширення доступу до інформації за ...
 - Розвиток міжнародної рефлексії і дискусій по етичних, ...
 - Сприяння тривалого, безперервної освіти і навчання у сфері ...
 - Просування високостандартних міжнародних стандартів і ...
 - Просування мережевої взаємодії у сфері інформації і знань ...

3.54. ábra. Dokumentum Nyomtatási nézetben és Vázlat nézetben

Vázlat nézetben a **Menüszalagon** megjelenik a **Vázlat** ideiglenes eszköztár (3.55. ábra). Ennek vezérlőelemeit használhatjuk a dokumentum szintjének kiválasztására, az elem szintjének megváltoztatására, új elemek beillesztésére és régiak törlésére.




3.55. ábra. Vázlat eszköztár

A **Vázlat** nézet legnagyobb előnye az, hogy a dokumentum tetszőleges szintjének beállítását úgy végezhetjük, hogy abban az alacsonyabb szintek nem zavarjanak, mivel el vannak rejtve. Ezt úgy érjük el, hogy a **Vázlat** eszköztárban a **Szerkezeti eszközök** között kiválasztjuk, melyik szintet szeretnénk megtekinteni. Ha a dokumentum egy szerkezeti elemének szeretnénk látni az alárendelt szintjeit, kattintsunk a megfelelő címsor előtti **+** jelre. A művelet ismétlése újra elrejtí a szöveget.

Ha egy alárendelt szövegnek túl nagy a betűmérete, esetleg túlságosan hosszú, emiatt zavarja a vázlat áttekinthetőségét, beállíthatjuk, hogy annak szövegét formázástól mentesen tekinthessünk meg (kapcsoljuk ki a **Szövegforgalmazás megtekintése** jelölőt), vagy kérhetjük, hogy csak a szöveg első sora jelenjen meg (**Csak az első sor megjelenítése** jelölő).

! Összefoglalás

A **dokumentum szerkezete** alatt a dokumentum összetevői által alkotott hierarchikus vázlat értjük. Ilyet leggyakrabban a hosszú, több fejezetből, paragrafusból álló dokumentumban hozunk létre.

A szöveget **Töréspontok** beszúrásával oszthatjuk szakaszokra. A szakaszokra bontást célszerű **Vázlat** nézetben végezni, eközben hasznos bekapcsolva tartani a *nem nyomtatható jelek mutatását* is. A töréspontokat a **Lap elrendezése** ⇒ **Oldalbeállítás** ⇒ **Töréspontok**  paranccsal illesztjük a szövegbe, ami után meg kell adjuk a beillesztés módját is.


A beszúrás helyén megjelenik a **Töréspont** nem nyomtatható jel. A szakaszokra bontott dokumentum részeinek formázása eltérhet egymástól.

A hosszú dokumentumok gyakran tartalmaznak **fejléceket**, amelyek a dokumentum margóin helyezkednek el. A **Word 2007**-ben *alsó*, *felső* és *oldalsó* fejléceket hozhatunk létre. A fejlécek leggyakrabban az oldalszámot, a dokumentum vagy a fejezet nevét, a szerző nevét és képeket tartalmaz. Az első oldal fejléce eltérhet a többitől, továbbá a páros és páratlan oldalak fejléce is eltérhet egymástól. A dokumentum szakaszaiban is eltérő lehet a fejléc.

A fejléceket csak **Nyomtatási nézetben** látjuk. A szöveg szerkesztése a fejlécek szerkesztésével egyidejűleg nem lehetséges. Fejléceket a **Beszúrás** lap **Élőfej és élőláb** csoportjának eszközeivel illesztünk be.

Az objektum formázásának megkönnyítése céljából stílusokat alkalmazhatunk. A **Stílus** az adott objektum paramétereit értékeinek névvel ellátott gyűjteménye. A **Word 2007**-ben a különböző objektumokhoz **stílusgyűjtemények** állnak rendelkezésünkre. Ezeket a stílus-

kat a dizájn követelményeinek alapján dolgozták ki, és harmonikusan egyesítik az objektumok tulajdonságainak értékeit. Stílusok a szöveges dokumentum különböző objektumaihoz – szöveg, táblázat, grafikai objektumok stb. – vannak kifejlesztve. A szöveges elemekhez rendelhető stílusokat a **KezdőlapStílustár**ában találjuk.

A dokumentum struktúráját a címsorstílusok – **Címsor1**, **Címsor2**, ..., **Címsor9**– segítségével hozzuk létre. Az ilyen módon strukturált dokumentum szerkezetét **Vázlat**nézetben tekinthetjük meg. Ebbe a nézetbe a **StátuszsorVázlat**  gombja vagy a **Nézet** lap **Dokumentumnézetek** csoportjának **Vázlat** eszköze segítségével léphetünk át.



Felelj a kérdésekre!

- 1°. Mit értünk a dokumentum struktúrája alatt? Hozz fel példákat!
- 2°. Mik a szakaszok? Mire szolgálnak?
- 3°. Milyen szakasztöréseket helyezhetünk el a dokumentumokban?
- 4°. Mi a fejléc? Milyen típusai vannak?
- 5°. Mire szolgálnak a fejlécek? Milyen objektumokat helyezhetünk el ezekben?
- 6*. Hogy állíthatunk be a dokumentum oldalaihoz különböző fejléceket?
- 7°. Mik a stílusok? Mire használják ezeket?
- 8°. A **Word 2007** milyen objektumaihoz alkalmazhatunk beépített stílusokat? Az objektumok mely tulajdonságaira vannak ezek hatással?
- 9°. Mik a címsorstílusok? Hányféle címsor áll rendelkezésünkre a **Word 2007**-ben?
- 10°. Mire szolgál a dokumentum **Vázlat** nézete?



Végezd el a feladatokat!

1. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\pukt 3.5\zrazok.docx**-et)! A szöveges dokumentum minden pontját formázd különálló szakaszként, ami új lapon kezdődik! A szakaszokat formázd az alábbiak szerint:
 - páratlan szakaszok: *álló* tájolás, *2 cm* margó, oldalszám *felül középen*;
 - páros szakaszok: *fekvő* tájolás, margók *1 cm*, oldalszám *balra lent*!Mentsd a dokumentumot **feladat 3.5.1.docx** néven a mappádba!
2. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\pukt 3.5\zrazok.docx**-et)! Hozz létre élőfejet, illeszd bele a mai dátumot *középre igazítva*, valamint élőlábat, ebbe illeszd bele a *nevedet balra igazítva*. Mentsd a dokumentumot **feladat 3.5.2.docx** néven a mappádba!
- 3*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\pukt 3.5\zrazok.docx**-et)! Végezd el a következő formázásokat: oldal *A4*, *álló* tájolás, felső margó *1,5 cm*, alsó *1 cm*, külső *2 cm*, belső *3 cm*, oldalszámozás a *lap alján*, kezdősorszám *1*; az élőfej a páratlan oldalon a dokumentum nevét tartalmazza, élőláb a páros oldalon a *dátumot és időt*! Mentsd a dokumentumot **feladat 3.5.3.docx** néven a mappádba!





4*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\punkt 3.5\zrazok1.docx**-et)! Formázd a dokumentumot az alábbi stílusokkal:

- Címek – *Cím* stílus.
- Szöveg – *Finom kiemelés*.
- Listák – *felsorolás* felsorolásjellel ♦.

Mentsd a dokumentumot **feladat 3.5.4.docx** néven a mappádba!

5*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\punkt 3.5\zrazok1.docx**-et)! Ez a dokumentum stílusok segítségével van formázva. Határozd meg, milyen értékeket rendeltek az objektumok tulajdonságaihoz az alkalmazott stílusokban! Töltsd ki a táblázatot!

<i>Objektum sorszáma</i>	<i>Objektum típusa</i>	<i>Stílus</i>	<i>Tulajdonság értékei</i>



6*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\punkt 3.5\zrazok1.docx**-et)! Ez a dokumentum stílusok segítségével van formázva. Töröld ki az előzőleg beállított formázásokat! Határozd meg, milyen értékeket rendeltek az objektumok tulajdonságaihoz az alapértelmezett stílusokban!

<i>Objektum sorszáma</i>	<i>Objektum típusa</i>	<i>Stílus</i>	<i>Tulajdonság értékei</i>



7*. A sűgó segítségével derítsd ki, hogy hozunk létre saját stílust a dokumentum már megformázott részlete alapján! Hozd létre az algoritmust, majd próbáld ki a gyakorlatban!

8*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 3\punkt 3.5\zrazok1.docx**-et)! Formázd a dokumentumot a következő stílusok segítségével:

- Első szintű címek – *Címsor1*;
- Második szintű címek – *Címsor2*;
- Harmadik szintű címek – *Címsor3*;
- Egyéb szöveg – *Normál*!

Tekintsd meg a dokumentum struktúráját! Mentsd a dokumentumot **feladat 3.5.8.docx** néven a mappádba!



3.6. A SZÖVEGES DOKUMENTUM LÉTREHOZÁSÁNAK AUTOMATIZÁLÁSA



1. Mik a stílusok? Mire használjuk ezeket?
2. Mit értünk a dokumentum struktúrája alatt? Hogy tekinthetjük meg?
3. Mi a hiperhivatkozás? Mire használjuk?

HIVATKOZÁSOK A SZÖVEGES DOKUMENTUMBAN

Az előző paragrafusban megismerkedtünk a hosszú dokumentumok kezelésének néhány sajátosságával: a dokumentum struktúrájának áttekintésével, a szakaszok létrehozásával és a stílusok alkalmazásával. A **Word 2007**-ben léteznek az összetett szerkezetű hosszú dokumentumok kezelésének egyéb eszközei is.

A hosszú dokumentumok különféle *hivatkozásokat* – *olyan speciális bejegyzéseket, amelyek a dokumentum egyéb helyeire mutatnak* – tartalmaznak. A hivatkozások a dokumentum egyes részeinek vagy objektumainak logikai összekapcsolására szolgálnak. A szövegben egy táblázatra, egy rajzra, egy másik honlapra vagy egy forrásdokumentumra mutató hivatkozást helyezhetünk el. Ha a hivatkozásra kattintunk, automatikusan átlépünk a hivatkozott objektumra.

A **Word 2007**-ben megtalálhatjuk a hivatkozások kezelésének eszközeit. Ezek a **Hivatkozás** lapra vannak összegyűjtve:

- a **Tartalomjegyzék** csoport eszközei a *tartalomjegyzék* létrehozására szolgálnak, ami a dokumentum szerkezeti egységeit és a megfelelő oldalszámokat tartalmazza.
- a **Tárgymutató** csoport eszközeivel a dokumentumban használt kifejezések **rendezett listáját** hozhatjuk létre azoknak az oldalaknak a megjelölésével, ahol ezek a kifejezések előfordulnak.

TARTALOMJEGYZÉK LÉTREHOZÁSA


Ha a hosszú dokumentum valamennyi strukturális egységét a megfelelő címsorstílussal formáztuk, a dokumentum tartalomjegyzéke automatikusan létrehozható. A műveletet **Nyomtatási kép** nézetben végezzük.


A tartalomjegyzéket a következő algoritmus szerint készítjük:




1. Helyezzük a kurzort a dokumentum azon pontjára, ahová a tartalomjegyzéket be szeretnénk illeszteni!
2. Hajtsuk végre a **Hivatkozás** ⇒ **Tartalomjegyzék** ⇒ **Tartalom** utasítást!
3. Válasszunk a beépített stílusok közül (3.56. ábra)!

A tartalomjegyzék egy olyan táblázat formájában lesz kialakítva, amely tartalmazza a dokumentum szerkezeti egységeit és azokat az oldalszámokat, ahol ezek az egységek kezdődnek (3.57. ábra). A tartalomjegyzéket általában a szöveg elején vagy végén helyezük el.



A tartalomjegyzék segítségével könnyedén navigálhatunk a dokumentumban, hamar megtalálhatjuk a fejezeteket, pontokat. E célból elegendő lenyomott **Ctrl** billentyű mellett a tartalomjegyzék megfelelő elemére kattintani. Figyeljétek meg, hogy a művelet elvégzése során megváltozik az egérmutató alakja .

Amennyiben a szerkesztés során megváltoztattuk a dokumentum szövegét vagy szerkezetét, a tartalomjegyzéket frissíteni kell. E célból hajtsuk végre a **Hivatkozások** ⇒ **Tartalomjegyzék** ⇒ **Frissítés** műveletet .

Вбудовані	
Автоматична таблиця 1	
Зміст	
Заголовок 1	1
Заголовок 2	1
Заголовок 3	1
Автоматична таблиця 2	
Зміст	
Заголовок 1	1
Заголовок 2	1
Заголовок 3	1
Ручний зміст	
Зміст	
Введіть заголовок розділу (рівень 1)	1
Введіть заголовок розділу (рівень 2)	2
Введіть заголовок розділу (рівень 3)	3
 Вставити таблицю змісту...	
 Видалити таблицю змісту	
 Зберегти виділений фрагмент до колекції таблиць змісту...	

3.56. ábra. Tartalomjegyzék-stílusok

Зміст	
Програма ЮНЕСКО "Інформація для всіх"	1
1. Преамбула	1
2. Мандат	1
3. Цілі програми	2
4. Основні розділи програми	2
Розділ 1. Розвиток інформаційної політики на міжнародному, регіональному і національному рівнях	2
Розділ 2. Розвиток людських ресурсів, навиків і вмій у століття інформації	3
Розділ 3. Посилення ролі інституцій у забезпеченні доступу до інформації	3
Розділ 4. Розвиток інструментів, способів і систем інформаційного менеджменту	3

3.57. ábra. Dokumentum tartalomjegyzéke (minta)

TÁRGYMUTATÓ LÉTREHOZÁSA

Tárgymutatókat leggyakrabban tankönyvekben, tudományos munkákban, szótárakban használunk. Ezeket általában a dokumentum végén helyezük el, külalakjukat a 3.58. ábra szemlélteti.

СЛОВНИЧОК

А

Абзац – об'єкт текстового документа, який має такі властивості: *вирінення, відступи, міжрядковий інтервал* та інші, с. 134.

Авторське право – право авторів розповсюджувати матеріали, які вони створили, підписувати їх своїм іменем, отримувати за них винагороду тощо, с. 182.

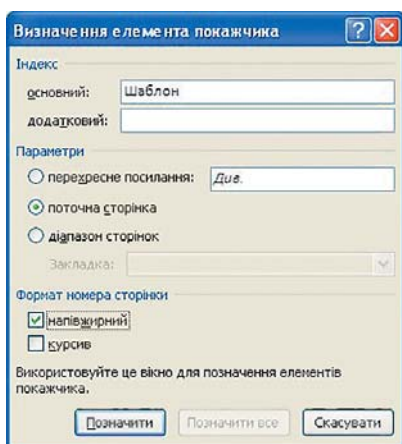
Алгоритм – скінченна послідовність команд, виконання яких приводить до розв'язання поставленої задачі, с. 19.

Б

Байт – одиниця вимірювання розмірів файлів і ємності носіїв даних, с. 54.

Блок-схема алгоритму – форма подання алгоритму у вигляді блоків. У блок-схемі алгоритму кожна команда записується в геометричній фігурі (блоці) певного вигляду. Блоки з'єднуються між собою стрілками, які вказують, яку команду алгоритму потрібно виконати наступною, с. 24.

3.58. ábra. Egy tankönyv tárgymutatója (minta)



3.59. ábra. A **Bejegyzés megjelölése** ablak

- szükség esetén végezzük el az elem nevének szerkesztését;
- adjuk meg az elem alternatív nevét, ha van;
- adjuk meg annak az elemnek a nevét, amellyel a kiválasztott elem össze van kapcsolva (kereszthivatkozás);
- ha csak az adott oldalhoz készítünk hivatkozást, jelöljük be a megfelelő rádiógombot;
- ha a kifejezés adott oldalak közötti összes előfordulásához készítünk hivatkozást, adjuk meg az oldalszámokat (ez esetben el kell készíteni az utolsó előforduláshoz is a mutatót);
- adjuk meg, hogy a tárgymutató oldalszámai félkövér vagy dőlt stílusúak legyenek-e!


4. Kattintsunk a **Jelölés** vagy az **Összes jelölés**

(ha intervallumot adtunk meg) gombra!

5. Ismételjük meg az 1–4. műveleteket valamennyi, a tárgymutatóban elhelyezendő elem esetében!

Aműveletek eredményeként valamennyi kijelölt kifejezés mellett megjelenik egy $\{\text{XE} \cdot \text{zміст документа} \cdot \text{A} \cdot \text{B} \cdot \text{A}\}$ alakú mező, amit azonban csak akkor láthatunk, ha bekapcsoltuk a nem nyomtatható jelek megjelenítését.

II. lépés. A tárgymutató elhelyezése a szövegben:

1. Vigyük a kurzort arra helyre, ahová szeretnénk beilleszteni a tárgymutatót (ez általában a dokumentum vége)!
2. Hajtsuk végre a **Hivatkozás** \Rightarrow **Tárgymutató – Beszúrás**  parancsot!
3. A megnyíló párbeszédablakban (3.60. ábra) adjuk meg a megjelenítés paramétereit:
 - legyenek-e az *oldalszámok jobbra igazítva*, amennyiben ilyen megjelenítés mellett döntöttünk;
 - a *tárgymutató formátuma* (válasszunk a listából);
 - *típus* (hogyan helyezzük el az oldalszámokat a tárgymutató eleme mellett);

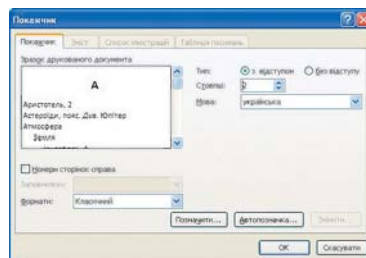
A tárgymutató létrehozása kétféle lépéses folyamat: először ki kell jelölni azokat a kifejezéseket, amelyeket fel szeretnénk venni a tárgymutatóba, majd be kell illeszteni a tárgymutatót a dokumentumba. Vizsgáljuk ezt meg részletesebben.

I. lépés. A tárgymutató elemeinek kijelölése:

1. Jelöljük ki azt a kifejezést, amit fel szeretnénk venni a tárgymutatóba ott, ahol az a dokumentumban először előfordul!

2. Hajtsuk végre a **Hivatkozás** \Rightarrow **Tárgymutató** \Rightarrow **Bejegyzés megjelölése**  parancsot!

3. A megnyíló párbeszédablakban (3.59. ábra) adjuk meg a következőket:




3.60. ábra. **Tárgymutató** ablak



- *hasábok* (hány hasábra legyen tördelve a tárgymutató);
- *nyelv* (milyen ábécé szerint legyen rendezve a tárgymutató).

4. Kattintsunk az **OK** gombra!

A tárgymutató be lesz illesztve a dokumentum kijelölt helyére. A tárgymutató előtt és után elhelyezésre kerül egy-egy **Szakaszvég** jel rejtett szimbólum is.

Ha a tárgymutató elemei megváltoztak (megváltozott a szöveg, vagy annak helye a dokumentumban), a tárgymutatót aktualizálni kell. Ezt a **Hivatkozás** ⇒ **Tárgymutató** ⇒ **Frissítés**  parancssal tehetjük meg.

Ha törölni kívánjuk a tárgymutatót, jelöljük ki azt, továbbá az előtte és utána elhelyezkedő sorokat (a rejtett szimbólumok törlése miatt), majd üssük le a **Delete** billentyűt.

LibreOffice-használóknak

A tárgymutató és a tartalomjegyzék kezelésére a **Beszűrés** menü **Tárgymutatók és tartalom** pontjának (3.61. ábra) parancsait használjuk.



3.61. ábra. A **Beszűrés** menü **Tárgymutatók** és tartalom pontja

SABLONOK A SZÖVEGES DOKUMENTUMOKBAN

Már tudjátok, hogyan hozunk létre és formázunk szöveges dokumentumokat a szövegszerkesztő eszközeinek segítségével. A **Word** bizonyos dokumentumok létrehozása során még egy eszközt bocsát a rendelkezésünkre: a sablonokat. Sablonokat alkalmaztunk az 5. osztályban a prezentációk létrehozása során.

A Sablon lényegében egy előre formázott dokumentum, amelyet kiindulási alapként használunk új dokumentum létrehozása során, és amely egy speciális fájl típusban tárolódik.

Léteznek naptár, levél, beszámoló, üdvözlőlap, névjegykártya stb. sablonok.

A sablon megnyitása után a felhasználó egy kész dokumentumot lát, amely szövegek, feliratok, képek, fejlécek, logók elhelyezésére szolgáló mezőket tartalmaznak. A dokumentumban különböző témákkal, stílusokkal lehet előreformázni (3.62. ábra).

A felhasználónak nincs más dolga, minthogy kitöltse a megfelelő mezőket, majd mentse a fájlt, mivel a struktúra és a formázás előre el volt készítve a sablonban.

A sablonok alkalmazása megkönnyíti a felhasználó munkáját, mivel lehetőséget biztosít azonos szerkezetű és külalakú, de eltérő tartalmú dokumentumok gyors elkészítésére, megtakarítva a formázásra fordítandó időt és erőfeszítést.

A dokumentumok és a sablonok között az alkalmazásukban rejlik az alapvető különbség. A sablon egy üres dokumentum előreformázott elemekkel, a dokumentum viszont valamely sablon alapján elkészült tartalom.

A sablonok **dotx** vagy **dotm** kiterjesztésű fájlokban tárolódnak. A sablonok makrovírusokat is tartalmazhatnak, ezért csak megbízható forrásból származó sablonokat szabad használni.

**СВЯТКУВАННЯ
Нового року**

МЕНЮ

[ПУНКТ МЕНЮ]

[Щоб замінити текст заміщення (наприклад, цей) власним, виділіть його та введіть власний текст. Не додавайте пробіли праворуч і ліворуч символів виділеного фрагмента.]

[ПУНКТ МЕНЮ]

[Щоб якнайшвидше привернути увагу гостей, описи страв мають бути короткими та привабливими]

[ПУНКТ МЕНЮ]

[Опис пункту меню]

[ПУНКТ МЕНЮ]

[Опис пункту меню]

[ПУНКТ МЕНЮ]



**СВЯТКУВАННЯ
Нового року**

МЕНЮ

Салат «Зимовий»

куряча печінка, корейська морквочка, мариновані огірки, гірчична заправка

Салат «Маскарад»

шинка, крабові палички, картопля, зелений горошок, листя салату, майонез

Десерт «Серпантин»

яблука в шоколаді

Коктейль «Феєрверк»

банан, полуниця, ананас з молоком



3.62. ábra. A menü sablonja és a sablon alapján elkészített dokumentum

A **Word 2007** sablonjai három csoportba vannak sorolva:

- *telepített sablonok* (levelek, faxok, beszámolók stb.), amelyek a **Microsoft Office 2007** csomag elemeként lettek a számítógépre telepítve;
- *Microsoft Office Online* sablonok, a **Microsoft Office Online** webhelyéről letölthető sablonok;
- *felhasználói sablonok*, értelemszerűen a felhasználó által lettek létrehozva.

A **Word 2007** alapértelmezett sablonja a **Normál** sablon (**Normal.dotm**), a program indításakor létrehozott üres dokumentum ebben a sablonban készül, és a következő beállításokat tartalmazza (3.11. táblázat):

3.11. táblázat

A Normal.dotm sablon alapértelmezett beállításai

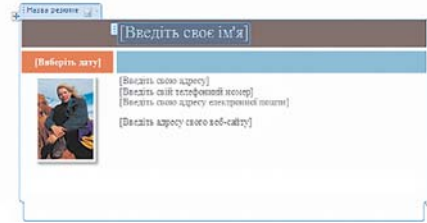
<i>Oldal</i>	<i>Bekezdés</i>	<i>Szimbólum</i>
<ul style="list-style-type: none"> • Tájolás: <i>álló</i> • Méret: <i>A4</i> • Felső margó: <i>1,5 cm</i> • Aló margó: <i>1 cm</i> • Bal margó: <i>2 cm</i> • Jobb margó: <i>1,5 cm</i> 	<ul style="list-style-type: none"> • Sorköz: <i>szimpla</i> • Bekezdések távolsága: <i>10 pt</i> • Igazítás: <i>balra</i> • Behúzások: <i>nincsenek</i> 	<ul style="list-style-type: none"> • Alapértelmezett betűtípus: <i>Calibri</i> • Méret: <i>11 pt</i> • Szín: <i>fekete</i> • Távolság: <i>normál</i>



A sablon tartalmazza továbbá a bekezdések, listák, táblázatok stílusait is. A szöveg beírása, az objektumok formázása, a szöveg struktúrájának kialakítása a felhasználó feladata. A **Normal.dotm** sablon alapján a felhasználó különféle dokumentumokat hozhat létre.

A **Word 2007** telepített sablonjai között találunk **Önéletrajz** sablont (**MedianResume.dotx**), amely szövegmezőket tartalmaz, amelyeket a felhasználó a saját adataival tölt fel: név, elérhetőség, végzettség, munkahely stb. (a sablon egy részlete a 3.63. ábrán látható). A beillesztett fényképet a felhasználó a sajátjára cseréli. A dátumok beillesztését a megfelelő mezők mellé rendelt naptár segíti. A sablon segítségével könnyen elkészíthetjük az önéletrajzunkat.


Önéletrajz – olyan dokumentum, amelyben röviden ismertetjük a tanulmányainkat és tevékenységünket, elért eredményeinket.

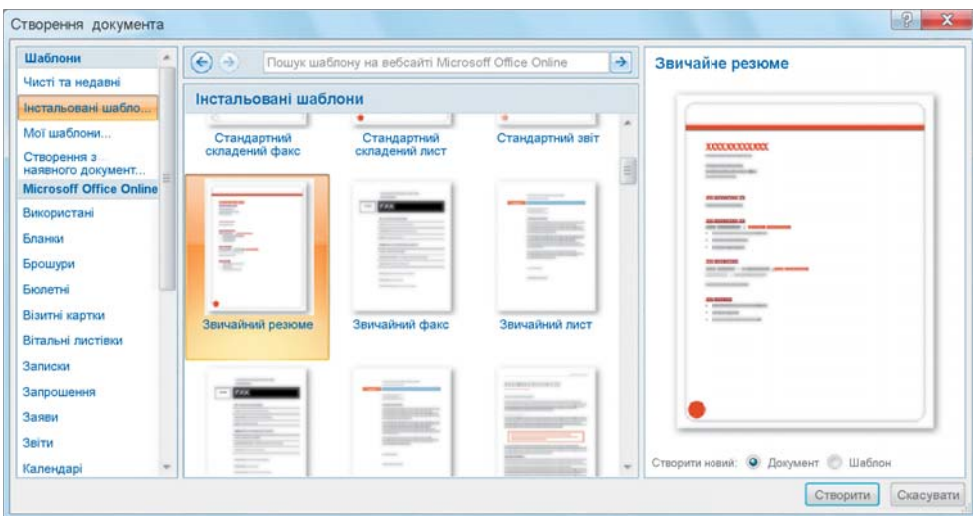


3.63. ábra. **Önéletrajz**-sablon

DOKUMENTUMOK LÉTREHOZÁSA SABLONBÓL

A **Word 2007**-ben a következőképpen hozunk létre dokumentumot sablon alapján:


1. Az **Office** gombra kattintva nyissuk meg a program főmenüjét!
2. Kattintsunk az **Új**  gombra!
3. A bal oldali listából válasszuk a telepített sablonokat (3.64. ábra)!
4. Válasszuk ki a sablont (például az **Önéletrajzot**)!
5. A minta mezőben (az ablak jobb oldali része) tekintsük meg a sablon szerkezetét!

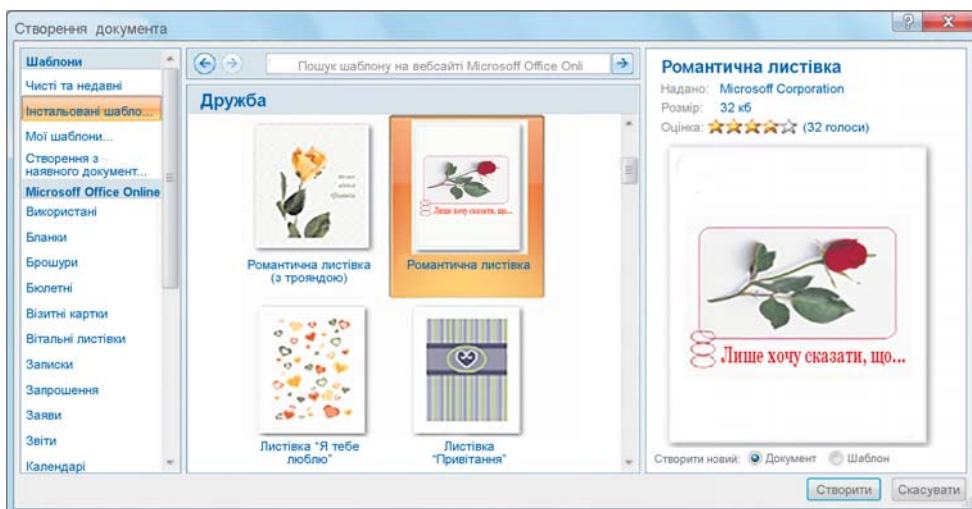


3.64. ábra. Dokumentum létrehozása sablon alapján

6. A minták alatt kapcsoljuk be a **Dokumentum** jelölőt!
7. Kattintsunk a **Létrehozás** gombra!
8. Töltsük ki a felajánlott beviteli mezőket!
9. Mentsük a dokumentumot!

Ha a **Microsoft Office Online** webhelyéről letöltött sablon alapján szeretnénk dolgozni:

1. Az **Office** gombra kattintva nyissuk meg a program főmenüjét!
2. Kattintsunk az **Új**  gombra!
3. A bal oldali listából válasszuk ki a **Microsoft Office Online** típust (3.65. ábra), például az **Üdvözlőlapot**!



3.65. ábra. Dokumentum létrehozása az internetről letöltött sablon alapján

4. Várjuk meg, amíg a szerverről letöltődik a sablonok listája!
5. Válasszuk ki a sablont!
6. A minta mezőben (az ablak jobb oldali része) tekintsük meg a sablon szerkezetét!
7. Kattintsunk a **Letöltés** gombra!
8. Várjuk meg a sablon letöltődését!
9. Töltsük ki a felajánlott beviteli mezőket!
10. Mentsük a dokumentumot!

Microsoft Word 2010-et használóknak

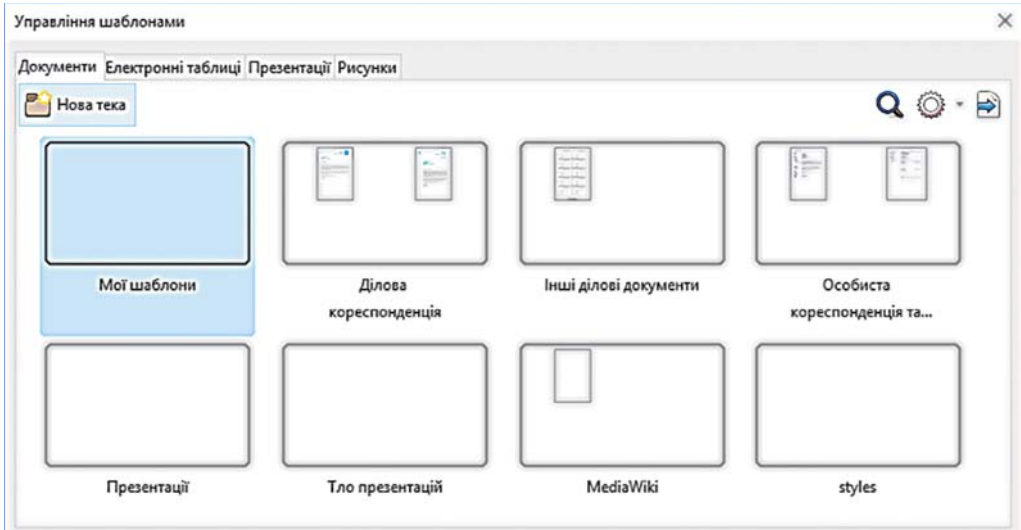
Word 2010-ben a **Fájl** ⇒ **Új** ⇒ **Használható sablonok** paranccsal hozunk létre új dokumentumot sablonból, ami után be kell kapcsolni a **Dokumentum** jelölőt, majd a **Létrehozás** gombra kell kattintani.

Amennyiben internetről szeretnénk a sablont letölteni, a használható sablonok helyett a **Sablonok az Office.com webhelyen** listából kell választani, majd le kell tölteni a választott sablont.



LibreOffice-használóknak

Sablon alapján létrehozott dokumentumot a **Sablonszervezővel** (3.66. ábra) készíthetünk, amit a **Fájl** ⇒ **Sablonok** ⇒ **Szervező** paranccsal nyitunk meg.






3.66. ábra. Sablonszervező ablaka

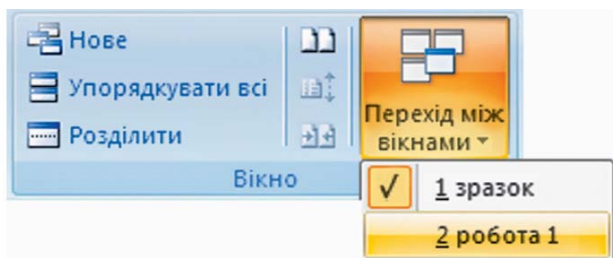
TÖBB DOKUMENTUM EGYIDEJŰ KEZELÉSE

A **Word 2007**-ben több dokumentummal is dolgozhatunk egyidejűleg. Ily módon a felhasználónak lehetősége van áttekinteni a dokumentumokat, összevetni azok tartalmát, részleteket átmásolni az egyikből a másikba, egyesíteni a dokumentumokat stb.

A megnyitott ablakok kezelésére a **Nézet** lap **Ablak** csoportjában (3.67. ábra) találunk eszközöket. Az ablakváltás például az alábbi eljárás szerint történik:

1. Kattintsunk a **Nézet** lap **Ablak** csoportjában az **Ablakváltás**  gombra!
2. Válasszuk ki a listából a megfelelő dokumentumot (az aktuális dokumentumot  jelöli)!

Ha ugyanazon dokumentum két különböző verziójával dolgozunk, célszerű azokat egymás mellett megjeleníteni. Ezt a **Nézet** ⇒ **Ablak** ⇒ **Párhuzamos nézet**  paranccsal állíthatjuk elő. A parancs végrehajtása után a képernyő függőlegesen két részre lesz osztva, minden

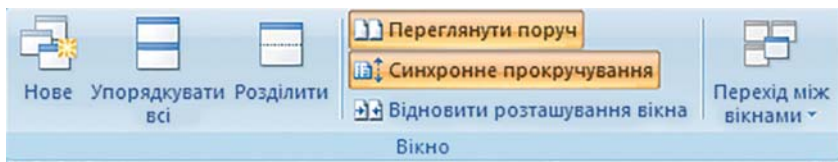


3.67. ábra. Az **Ablak** csoport összecukva


3. fejezet

dokumentum egy-egy félablaknyi helyet foglal majd el. A dokumentumok görgetése szinkronban történik.

A dokumentumok párhuzamos görgetését (3.68. ábra) az **Ablak** csoport megfelelő eszközével kapcsolhatjuk ki.



3.68. ábra. Az **Ablak** csoport

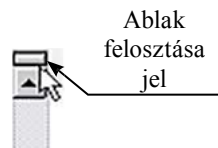
A **Párhuzamos nézet**  gomb újbóli lenyomásával kiléphetünk a párhuzamos nézet módból.



Ha többet szeretnél tudni

Előfordul, hogy egy nagyobb dokumentum szerkesztése során egyszerre szeretnénk látni ugyanannak a dokumentumnak két különböző részét. Ehhez a dokumentumot a **Nézet** ⇒ **Ablak** ⇒ **Felosztás** gombbal két részre kell bontani. A parancs hatására a képernyőn megjelenik egy szürke csík, amelyet a felosztás helyére kell húzni. Ugyanezt elvégezhetjük a függőleges gördítősáv tetején található jelecske mozgatásával is (3.69. ábra).

A **Nézet** ⇒ **Ablak** ⇒ **Felosztás** újbóli lenyomásával kiléphetünk az üzemmódból, ugyanezt elérhetjük akkor is, ha duplán kattintunk az ablakokat felosztó vonalra.





3.69. ábra. Ablak felosztása jel

SZÖVEGES FÁJLOK FORMÁTUMAI

A **Word 2007** alapértelmezett formátuma a **DOCX**. A szövegszerkesztő ezen felül az **RTF**, **TXT**, **DOC** és **PDF** formátumokat támogatja. A szöveges formátumok jellemzőit a 3.12. táblázatba foglaltuk össze.




3.12. táblázat

A Word 2007 által támogatott formátumok legfontosabb jellemzői

Formátum	Ikon	Jellemzők
TXT		A fájlban csak formázatlan, bekezdésekre tördelt szöveg tárolható. Sokféle szövegszerkesztő támogatja.
DOC		A fájlban formázott szöveg, képek, egyéb objektumok tárolhatók. A Word alapértelmezett formátuma a 2003 -as verzióval bezárólag.



A 3.12. táblázat folytatása

Formátum	Ikon	Jellemzők
DOCX		A fájlban formázott szöveg, képek, egyéb objektumok tárolhatók. A többi szöveges formátumhoz képest tömörebb tárolást (70% körül) tesz lehetővé. A Word alapértelmezett formátuma a 2007-es verziótól kezdve.
RTF		A fájlban formázott szöveg, képek, egyéb objektumok tárolhatók. Különböző operációs rendszerekben működő szövegszerkesztők által támogatott.
PDF		A fájl megtartja az alapjául szolgáló dokumentum formázásait. Nem szerkeszthető, megtekintéséhez szükség van az Adobe Reader programra.

Ha a fájlt nem az alapértelmezett formátumban szeretnénk menteni, a **Mentés** ablakban nyissuk meg a **Fájltípus** legördülő menüt, és válasszunk a támogatott formátumok közül.

LibreOffice-használóknak

A **LibreOffice Writer** alapértelmezett formátuma az **ODF** (*Open Document Format for Office Application* – nyitott fájlformátum Office alkalmazások számára), a fájl kiterjesztése **ODT**. A **LibreOffice Writer** támogatja még az **RTF**, **TXT**, **DOC** és **XML** formátumokat.

Összefoglalás

A hosszú dokumentumokban a **Word 2007** segítségével különböző **hivatkozásokat** hozhatunk létre. Ezek olyan speciális bejegyzések, amelyek összekötik a szöveges dokumentum különféle elemeit. Amikor a dokumentumban egy hivatkozásra kattintunk, átlépünk a dokumentum egyik részéből egy másikba.

A hivatkozások kezeléséhez szükséges eszközöket a **Hivatkozások** lapon találjuk. Ezek segítségével tartalomjegyzékeket és tárgymutatókat is létrehozhatunk.

A **Word 2007** különböző sablonokkal támogatja a szöveges dokumentumok automatikus létrehozását és formázását. A sablon egy szerkezetében és formázásában előre definiált dokumentum, amelyet speciális fájlokban (**dotx**, **dotm** a kiterjesztésük) tárolunk, és az új dokumentumok létrehozása során alkalmazunk.

Az **Office** ⇒ **Új** ⇒ **Sablon** utasítással hozhatunk létre új dokumentumot a megfelelő sablon alapján. Ekkor a megfelelő sablon kiválasztása után a **Létrehozás** gombra kell kattintani.

A **Word 2007** felhasználóinak lehetősége van több dokumentum egyidejű kezelésére. Ily módon a felhasználóknak lehetőségük van a dokumentumok összehasonlítására, vagy részletek másolására az egyik dokumentumból a másikba, egyesíteni a dokumentumokat. Az ablakok kezelését a **Nézet** lap **Ablak** csoportjának eszközeivel végezzük.

A **Word 2007**-ben készült dokumentumok alapértelmezett formátuma a **DOCX**. A szövegszerkesztő ezenkívül az **RTF**, **TXT**, **DOC** és **PDF** formátumokat támogatja.



Felelj a kérdésekre!

- 1°. Mik a hivatkozások egy dokumentumban? Mire használhatjuk?
- 2°. Mi a tartalomjegyzék? Mi a rendeltetése?
- 3°. Hogy hozunk létre **Word 2007**-ben tartalomjegyzéket?
- 4°. Mi a tárgymutató? Mire használhatjuk?
- 5°. Hogy hozunk létre tárgymutatót a szövegszerkesztőben?
- 6°. Mi a sablon? Hol használhatjuk?
- 7°. Milyen hatással van a dokumentum tulajdonságaira a sablon? Milyen sablonokat találhatunk a **Word 2007**-ben?
- 8°. Mi a különbség a sablon és a dokumentum között?
- 9°. Mi a sablont tartalmazó fájlok kiterjesztése? Hozz fel példákat!
- 10°. Hogy kezelhetünk egyszerre több dokumentumot a szövegszerkesztőben? Hogy tekinthetünk meg több dokumentumot egyidejűleg?
- 11°. Milyen fájlformátumokat támogat a **Word 2007**? Melyik ezek közül az alapértelmezett?
- 12°. Hogy adjuk meg a mentendő fájl formátumát?





Végezd el a feladatokat!

1. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 3\pukt 3.6\zrazok.docx** fájlt)! Formázd meg a dokumentum szerkezeti elemeit a megfelelő szintű címsorokkal! Illeszd be a dokumentum elejére a tartalomjegyzéket! Mentse a dokumentumot **feladat 3.6.1.docx** néven a mappádba!
2. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 3\pukt 3.6\zrazok.docx** fájlt)! Hozz létre a dokumentum végén tárgymutatót a zölddel kiemelt kifejezésekből és azok definícióiból! A tárgymutató formátumát alakítsd az ízlésed szerint! Mentse a dokumentumot **feladat 3.6.2.docx** néven a mappádba!
3. Nyisd meg a **Word 2007** szövegszerkesztőt! Tekintsd át a telepített sablonokat, állapítsd meg a számukat és azt, milyen típusú dokumentumok hozhatók létre ezen sablonok alapján! A választodat jegyezd be a füzetedbe!
- 4*. Keresd meg a számítógépeden a sablonfájlokat! Írd be a füzetedbe, melyik mappában találhatók! Hány ilyen sablon van, és mi ezek neve?
5. Írj levelet az *Ukraina's got talent* tehetségkutató verseny szervezőbizottságának, amelyben jelzed részvételi szándékodat! Használd az *Esszencia levél* sablont! Mentse a dokumentumot **feladat 3.6.5.docx** néven a mappádba!





- 6•. Hozd létre a névjegykártyádat a **Microsoft Office Online** webhelyéről letöltött sablon alapján! Mentsd a dokumentumot **feladat 3.6.6.docx** néven a mappádba!
-  7*. A **Súgó** segítségével állapítsd meg, hogy hozhatunk létre sablont a már létező dokumentumból! Hozz létre ily módon egy sablont a nyári gyakorlatról szóló beszámoló elkészítésére! Mentsd el a sablont a sablonok alapértelmezett helyére!
- 8•. Nyisd meg a tanárod által megadott három fájlt (vagy a **Rozdil 3\pukt 3.6\zrazok 1.docx, zrazok 2.docx, zrazok 3.docx** fájlokat)! Próbáld ki az ablakok közötti váltásokat a **Nézet** menü **Ablak** eszköztárának alkalmazásával! Másold az első dokumentum elejére a második első bekezdését, a végére pedig a harmadik dokumentum utolsó bekezdését! Mentsd az első dokumentumot **feladat 3.6.8.docx** néven a mappádba!
-  9•. Nyisd meg a tanárod által megadott két fájlt (vagy a **Rozdil 3\pukt 3.6\zrazok1.docx, zrazok 2.docx** fájlokat)! Helyezd a dokumentumablakokat egymás mellé! Próbáld ki a **Párhuzamos görgetést**, majd kapsold ki azt! Keresd meg az első dokumentumban a másodiktól eltérő részeket, és emeld ki ezeket háttérszínnel! Kapsold ki a párhuzamos nézetet! Mentsd a dokumentumot **feladat 3.6.9.docx** néven a mappádba!
- 10•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 3\pukt 3.6\zrazok.docx** fájlt)! Mentsd el a fájlt **DOC** és **TXT** kiterjesztéssel a mappádba! Hasonlítsd össze a fájlok méretét! Magyarázd meg az eredményt!

5. SZÁMÚ GYAKORLATI MUNKA

A dokumentum szerkezete. Tartalomjegyzék és tárgymutató létrehozása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 3\pukt 3.6\prakticsna 5\zrazok.docx** fájlt)!
2. Hozd helyre a dokumentum szerkezetét **Vázlat** nézetben! Szükség esetén változtasd meg az egyes elemek szintjét!
3. Formázd meg a szerkezeti elemeket a megfelelő szintű címsorokkal!
4. Helyezd el a dokumentum elejére a tartalomjegyzéket!
5. Hozz létre a dokumentum végén tárgymutatót a pirossal kiemelt szavakból!
6. Mentsd a dokumentumot **gyakorlati 5.docx** néven a mappádba!

Ebben a fejezetben megismerkedtek:

az audio- és videofájlok formátumaival

a multimédiás objektumokat feldolgozó programokkal

az audio- és videoformátumok átalakításának eszközeivel

a hang és a video rögzítésével

az audio- és videofájlok közzétételével az interneten

az audio- és videofájlok létrehozásával, amelyek:

- videoeffektusokat

- időben szinkronizált audio- és videofolyamot

- a részek közötti ugrásokat tartalmaz



4.1. MULTIMÉDIÁS OBJEKTUMOK FELDOLGOZÁSA



1. Milyen üzeneteket neveznek multimédiásnak?
2. Milyen multimédiás objektumokat feldolgozó programokat ismertek? Milyen célt szolgálnak a multimédia-lejátszó programok?
3. Milyen eszközöket használnak a multimédiás üzenetek létrehozására és feldolgozására?

AUDIO- ÉS VIDEOFORMÁTUMOK

Már megismerkedtetek a fájlformátum fogalmával. A fájl típus a fájlban tárolt adatok meghatározott struktúráját jelenti. Ismeritek a legelterjedtebb szöveges és grafikus formátumokat, a prezentációk és az elektronikus táblázatok formátumait. Az audio- és videoadatokat szintén meghatározott formátumú fájlokban tárolják, ezeket multimédiás fájl típusokként is emlegetik. Ezeket a fájl típusokat foglaltuk össze a 4.1. táblázatban.

4.1. táblázat

Multimédiás fájlformátumok

Formátum	Leírás	Kiterjesztés
<i>Tömörítés nélküli vagy veszteségmentes tömörítést használó formátumok</i>		
WAV vagy WAVE (az angol <i>waveform</i> – hullámalak szóból)	A Windows alapértelmezett hangformátuma tömörítetlen audioadatok tárolására	wav
MIDI (<i>Musical Instrument Digital Interface</i> – hangszeres digitális interfész)	Speciális eszköz vagy szoftver számára értelmezhető utasításokat tartalmazó fájl	mid, midi
FLAC (<i>Free Lossless Audio Codec</i> – szabad veszteségmentes audio kodek)	Szabad szoftver licenc alatt közzétett veszteségmentesen tömörített hangfájl	flac
<i>Veszteségesen tömörített formátumok</i>		
MP3 vagy MPEG -1/2/2.5 Layer 3 (<i>Motion Picture Experts Group</i> – mozgókép szakértői csoport, <i>Layer 3</i> – harmadik szint)	A legelterjedtebb hangfájltípus. Fájlcserélőkben, interneten alkalmazzák hangadatok továbbítására	mp3
MP4 vagy MPEG-4 Part 14	Videófilmek tárolására, valamint digitális televíziózásban használt fájl típus	mp4

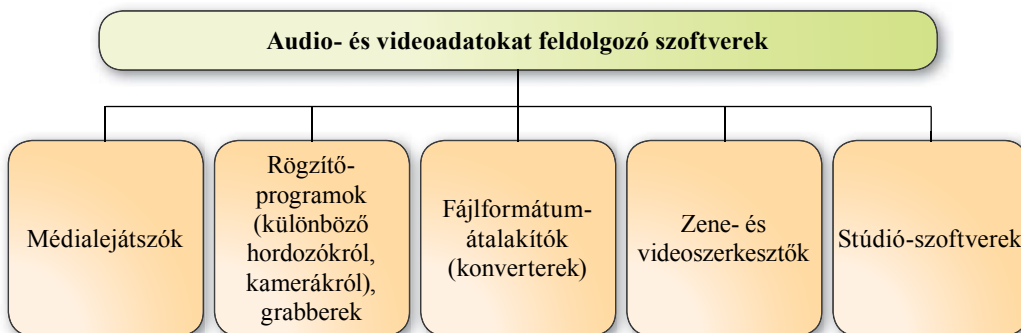
Formátum	Leírás	Kiterjesztés
WMA/WMV (<i>Windows Media Audio/Video</i>)	A Windows szabványos fájl típusa audio- és videó adatok tárolására és lejátszására	wma, wmv
QuickTime (angolul <i>gyors idő</i>)	Az Apple által kidolgozott fájlformátum audio- és videó adatok tárolására és lejátszására	mov
Flash video	Elsősorban interneten alkalmazott, erősen tömörített típus	flv

Mivel az audio- és videofájlok jelentős része tömörített, lejátszásuk során szükség van az adatok kitömörítésére is. Ehhez szükség van az adatok **kódolására** és **dekódolására** alkalmas szoftverre, az úgynevezett **kodekekre**. A kodekek általában részei az audio- és videofeldolgozó programoknak, de léteznek különálló programként is, ilyen például a **K-Lite Codec**.

MULTIMÉDIÁS OBJEKTUMOKAT FELDOLGOZÓ SZOFTVEREK

Már megismerkedtetek egyes multimédiás objektumokat feldolgozó programokkal, jelesül a médialejátszókkal. A 4.1. ábrán a multimédiás objektumokat feldolgozó programok általános klasszifikációját mutatjuk be.

Ismerkedjünk meg részletesebben ezekkel a programokkal és működésük algoritmusával.



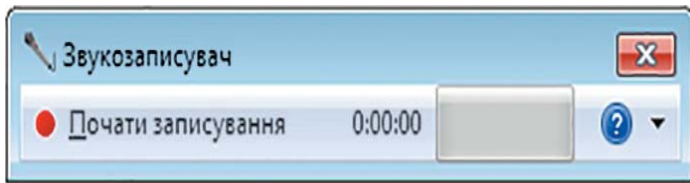
4.1. ábra. Audio- és videofeldolgozó programok klasszifikációja

AUDIO- ÉS VIDEÓ RÖGZÍTÉSE, CLIPEK LÉTREHOZÁSA

Az audio- és videó adatokat multimédiás **rögzítőprogramok** (**grabberek**, **rekorderek**) segítségével olvassuk be a mikrofonokról, kamerákról, egyéb eszközökről. Az adatokat ezek a programok mentik el a megfelelő formátumokba.



Ezek a programok teszik lehetővé, hogy hangos kommentárt csatoljunk a prezentációhoz, rögzítsük a **Skype** videokonferenciát vagy a webes sportközvetítést stb.



4.2. ábra. Hangrögzítő ablaka

Ilyen rögzítőprogram a **Windows Hangrögzítő** (4.2. ábra), az **Audacity**, az **Audio-grabber**, a **Streamripper**, az **AML Easy Audio Recorder**, a **Free Sound Recorder** stb. Manapság már internetes hang- és videórögzítő-szolgáltatásokat is igénybe vehetünk, például

<http://vocaroo.com>;

<http://online-voice-recorder.com>.

Ha tehát a mikrofonról szeretnénk hangot felvenni, a következőt kell tenni:

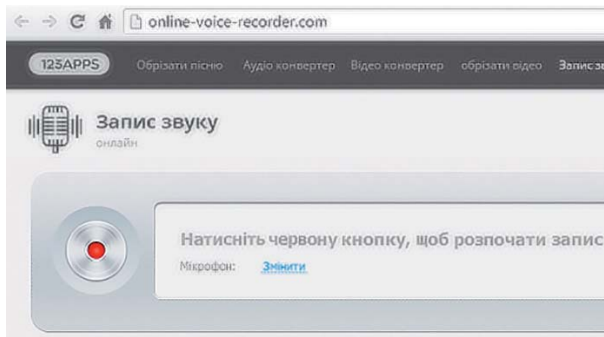
1. Kapcsoljuk be a mikrofont (vagy használjuk a beépítettet, mobil eszközökön)!
2. Indítsuk el a rögzítőprogramot (**Start** ⇒ **Minden program** ⇒ **Kellékek** ⇒ **Hangrögzítő**)!
3. A megfelelő gomb (például **Record**) megnyomásával (4.3. ábra) indítsuk el a felírást!
4. Mondjuk a mikrofonba a szöveget!
5. A megfelelő gombra kattintva állítsuk le a rögzítést (például a **Stop** gomb megnyomásával)!
6. Mentsük el a rögzített anyagot a megfelelő mappába!

A rögzített fájlt a későbbiekben meghallgathatjuk, szerkeszthetjük.

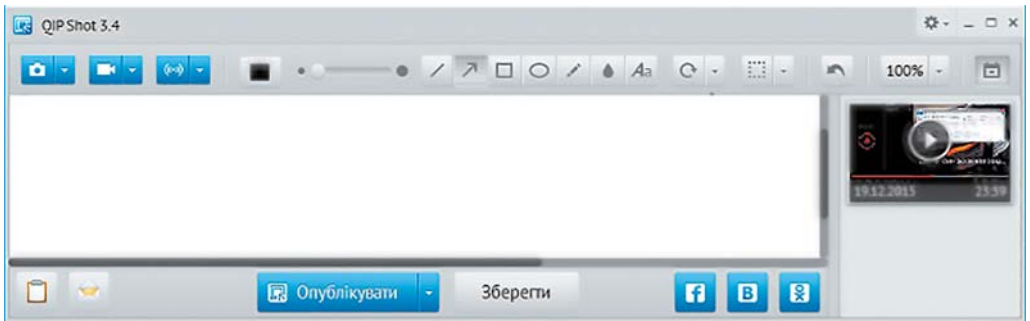
Videó rögzítésekor kiválaszthatjuk, hogy a képet és a hangot, vagy csak a képet szeretnénk rögzíteni. A számítógéphez természetesen a választásnak megfelelő külső eszközt (amely tehát videót vagy hangot képes lejátszani) kell csatlakoztatni.

Videót webkameráról, a lejátszó ablakából vagy egy webhelyről rögzíthetünk. Minden esetben a célnak megfelelő programot kell használni. Videó rögzítésére alkalmas program a **CamStudio**, a **QIP Shot**, a **Free Screen Video Recorder**, az **Ezvid**. A **QipShot**-tal (4.4. ábra) a következőképpen történik a rögzítés:


1. Nyissuk meg a programot, amelyből a videót rögzíteni szeretnénk, vagy nyissuk meg a weboldalt!
2. Indítsuk el a **Qip Shot**-ot!



4.3. ábra. Internetes hang- és videórögzítő



4.4. ábra. A QIP Shot ablaka

3. Válasszuk ki a program üzemmódját , vagyis az eszközt, ahonnan a rögzítés történik majd, például a képernyő egy téglalap alakú területe!
4. A speciális kurzor (4.5. ábra) segítségével jelöljük ki azt a képernyőrészt, ahonnan a videó rögzítése történik!
5. Indítsuk el a lejátszást a programból vagy a webhelyről!
6. Indítsuk el a rögzítést (4.6. ábra)!
7. A rögzítés befejeztével állítsuk azt le a **Stop** gombbal!
8. Ha el szeretnénk menteni a rögzített anyagot, kattintsunk a **Mentés** gombra!
9. Jelöljük ki a mentés helyét és a fájl nevét!



4.5. ábra.

A fenti algoritmus kisebb változtatásokkal más rögzítőprogram estében is alkalmazható.

Чемпіон України проведе перший спаринг на зборі в Іспанії



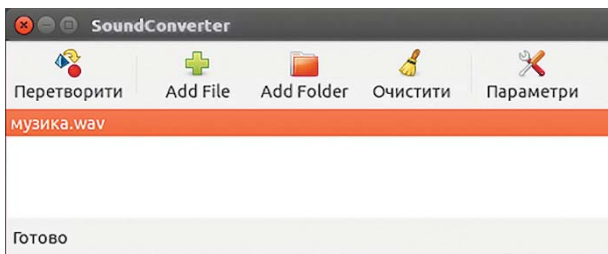
дебютний спаринг на першому тренувальному зборі в Іспанії. Суперником чемпіона України виступить представник третьої німецької Бундеслигі дрезденське "Динамо".

4.6. ábra. A QIP Shot ablaka, amint a képernyő téglalap alakú területéről rögzítést végez




AUDIO- ÉS VIDEOFORMÁTUM KONVERTEREK

A felhasználói igényektől függően sokszor szükség lehet a rögzített anyagok fájlformátumának átalakítására. Megesik, hogy a médialejátszónk nem alkalmas bizonyos formátumok lejátszására. Ilyen esetekben speciális szoftvereszközökre, úgynevezett konverterekre van szükség. Ilyenek a **HAMSTER Free Audio Converter**, a **SoundConverter XRECODE II**, **TAudioConverter**, **AudioConverter Studio** stb.



Miután elindítottuk a **SoundConverter** (4.7. ábra), az audiofájlok átalakítását a következőképpen végezzük:

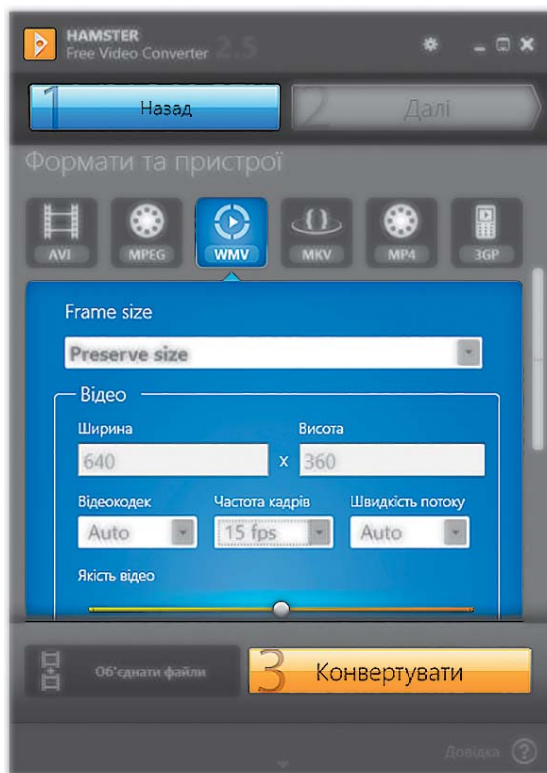
4.7. ábra. A **SoundConverter** ablaka átalakításra váró fájjal

1. Kattintsunk a  (**Add File** – fájl hozzáadása) gombra, és jelöljük ki a konvertálandó fájlt!
2. A **Paraméterek** gombra kattintva a megnyíló **Format** listán válasszuk ki, milyen formátumra szeretnénk a fájlt átalakítani!
3. Indítsuk el az átalakítást!

Az átalakított fájlt alapértelmezés szerint az eredetivel azonos mappába lesz mentve. A **Paraméterek** gombra kattintva más mappát is kijelölhetünk az átalakított fájl mentésére.

Videófájlok konvertálására rengeteg különböző licenc alatt terjesztett szoftver létezik. Ilyenek például a **HAMSTER Free Video Converter**, az **AVS Video Converter**, az **Any Video Converter Free**, a **SUPER** stb.

Vizsgáljuk meg, hogy történik az átalakítás a **HAMSTER Free Video Converter** esetében. Miután elindítottuk a programot:



1. A **Fájl hozzáadása** gombbal jelöljük ki az átalakítandó fájlokat!
2. Kattintsunk a **Tovább** gombra!
3. Adjuk meg a célformátumot, majd szükség esetén annak paramétereit (4.8. ábra)!
4. Kattintsunk az **Átalakítás** gombra!
5. Jelöljük ki, melyik mappába kívánjuk menteni az átalakított fájlt!

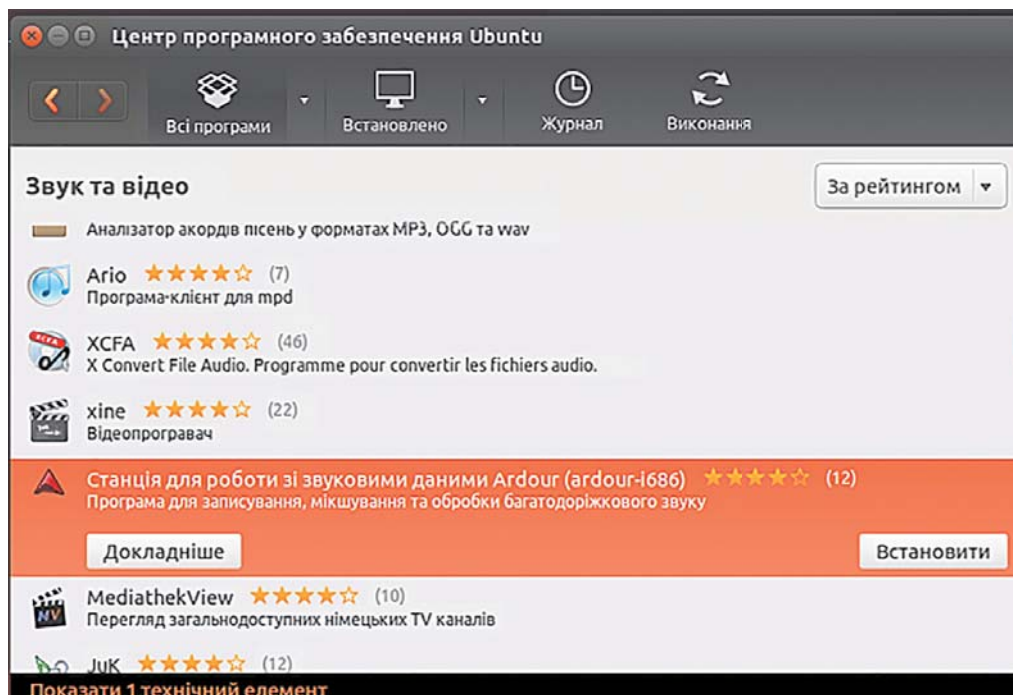
4.8. ábra. A **HAMSTER Free Video Converter** ablaka

A konvertereknek léteznek webes változatai is, úgy audio, mind videofájlok esetében, például a <http://audio.online-convert.com>, a <http://benderconverter.com>, a <https://online-audio-converter.com>, a <http://www.onlinevideoconverter.com>, a <http://www.online-convert.com> stb.



Linux-használóknak

Ha a telepítés során nem lettek volna telepítve az audio- és videofeldolgozó programok, az **Ubuntu szoftverközpont** lehet a segítségünkre. A **Programok** listájából keressük ki a hang- és videofeldolgozó programokat, választuk ki ezek közül azokat, amelyek nekünk szükségesek, majd kattintsunk a **Telepítés** gombra (4.9. ábra).



4.9. ábra. Az **Ubuntu szoftverközpont** ablaka

! Összefoglalás

Az audio- és videoadatokat különböző formátumokban tárolódnak. Az audiofájlok ismertebb kiterjesztései a **wav**, a **mid**, a **flac** és az **mp3**. A videofájloké **mp4**, a **wmv**, a **mov**, az **flv**.



A tömörített állományok **kódolását** és **dekódolását** végző programokat **kodekeknek** nevezzük.

Az audio- és videofájlokat feldolgozó programok között vannak lejátszók (**playerek**), rögzítőprogramok (**grabberek**, **rekorderek**), fájlformátumok átalakítására szolgáló programok (**konverterek**), szerkesztők, stúdióprogramok.

**Felelj a kérdésekre!**

- 1• Milyen audioformátumokat ismersz? Ezek közül melyikben alkalmaznak veszteséges tömörítést?
- 2° Milyen videoformátumokat ismersz?
- 3° Mi a kodek? Mi a rendeltetése?
- 4° Milyen típusai vannak a multimédiás adatokat feldolgozó szoftvereknek?
- 5° Mire szolgálnak a rekorderek? Nevez meg ilyen programokat!
- 6• Milyen lépésekből áll a hang rögzítése?
- 7* Miben különbözik a hangfelvétel és a videofelvétel algoritmus?
- 8• Milyen forrásokból rögzíthetünk videókat?
- 9• Milyen lépésekből áll az audio- és videofájlok konvertálása?

**Végezd el a feladatokat!**

- 1° Rögzítsd a saját előadásodban a kedvenc versedet a **Windows** hangrögzítőjével! Mentsd a hanganyagot **feladat 4.1.1.wav** néven a mappádba!
-  2• A <http://online-voice-recorder.com> internetes rögzítőszolgáltatás segítségével rögzítsd Petőfi Sándor *Itt van az ősz, itt van újra* című versét a saját előadásodban! Mentsd a hanganyagot **feladat 4.1.2.mp3** néven a mappádba!
-  3• Rögzíts a webkameráról egy 30 másodperces videót a **QIP Shot** program segítségével! Mentsd a fájlt **feladat 4.1.3.mp4** néven a mappádba!
- 4• Rögzítsd a **QIP Shot** segítségével a videolejátszóval éppen lejátszott **video2.avi** (**Rozdil 1\zavdannya 4.1\video2.avi**) fájlt! Mentsd a rögzített anyagot **feladat 4.1.4.wmv** néven a mappádba!
- 5*. Végezd el a tanárod által megadott két **MP3** formátumú fájlt (vagy a **Rozdil 1\zavdannya 4.1\Audio** mappa két fájljának) átalakítását **WMA** formátumba valamely audiokonverter segítségével! Mentsd a fájlokat a mappádban létrehozott **feladat 4.1.5** mappába! Hasonlítsd össze az eredeti és az átalakított fájlok méretét! Mivel magyarázható a tapasztalt különbség?
- 6*. Végezd el a tanárod által megadott két **MOV** formátumú fájlt (vagy a **Rozdil 1\zavdannya 4.1\Video** mappa két fájljának) átalakítását **WMV** formátumba valamely audiokonverter segítségével! Mentsd a fájlokat a mappádban létrehozott **feladat 4.1.6** mappába! Hasonlítsd össze az eredeti és az átalakított fájlok méretét! Mivel magyarázható a tapasztalt különbség?



4.2. VIDEÓK LÉTREHOZÁSA



1. Milyen audio- és videoformátumokat ismertek?
2. Milyen típusai vannak a multimédiás adatokat feldolgozó programoknak?
3. Milyen műveleteket végezhetünk a lejátszók, rekorderek, konverterek segítségével?

AUDIO- ÉS VIDEOSZERKESZTŐK

Az előző pontban nem vizsgáltuk a multimédiás programok két típusát: a szerkesztőket és a stúdiókat. Ezekkel a programokkal végezhetjük a multimédiás fájlok szerkesztését. Segítségükkel részleteket törölhetünk, másolhatunk, illeszthetünk be, megváltoztathatjuk a fájlok hosszát, szöveget és képeket illeszthetünk be a videókba stb.

A két típus között az a különbség, hogy a stúdióprogramok funkcionalitása magasabb, vagyis ezek több szerkesztési lehetőséget biztosítanak és professzionális hang-, film- és animációs stúdiókban használatosak. Ilyen programok a hangfeldolgozásban a **Linux MultiMedia Studio**, a **Psychedelics Psyce**, a **Steinberg Cubase**, a **Cakewalk Sonar**, az **Adobe Audition** stb. A videofeldolgozásban ide sorolható a **Pinnacle STUDIO**, a **Kino**, az **Adobe Premiere Pro**, az **Avid Media Composer**. A stúdióprogramok jellemzően kereskedelmi szoftverek.

Egyszerű hang- és videoszerkesztés céljára leginkább a különböző szerkesztőprogramok ajánlhatók. Hangfájlok esetén ez lehet az **Audacity**, a **Free Audio Editor**, a **Wave Editor**, videók estében pedig a **Windows Live Movie Maker**, az **Open Shot**, a **Virtual Dub**, a **ZS4 Video Editor**.

A WINDOWS LIVE MOVIE MAKER

A **Windows Live Movie Maker** a **Windows Movie Maker** új verziója. A **Windows 7** és **Windows 10** operációs rendszer alá a program ingyenesen letölthető a **Microsoft** webhelyéről. A **Windows** régebbi verziói esetében a **Movie Maker** része volt az installációs csomagnak.

A **Windows Live Movie Maker** videoszerkesztő, amellyel videorészleteket, fényképeket, hanganyagot, feliratokat fűzhetünk össze videóvá. A kész videót a program által támogatott formátumban elmenthetjük, vagy közzétehetjük az interneten.

A **Windows Live Movie Maker** videoszerkesztő ablakát a 4.10. ábra szemlélteti.

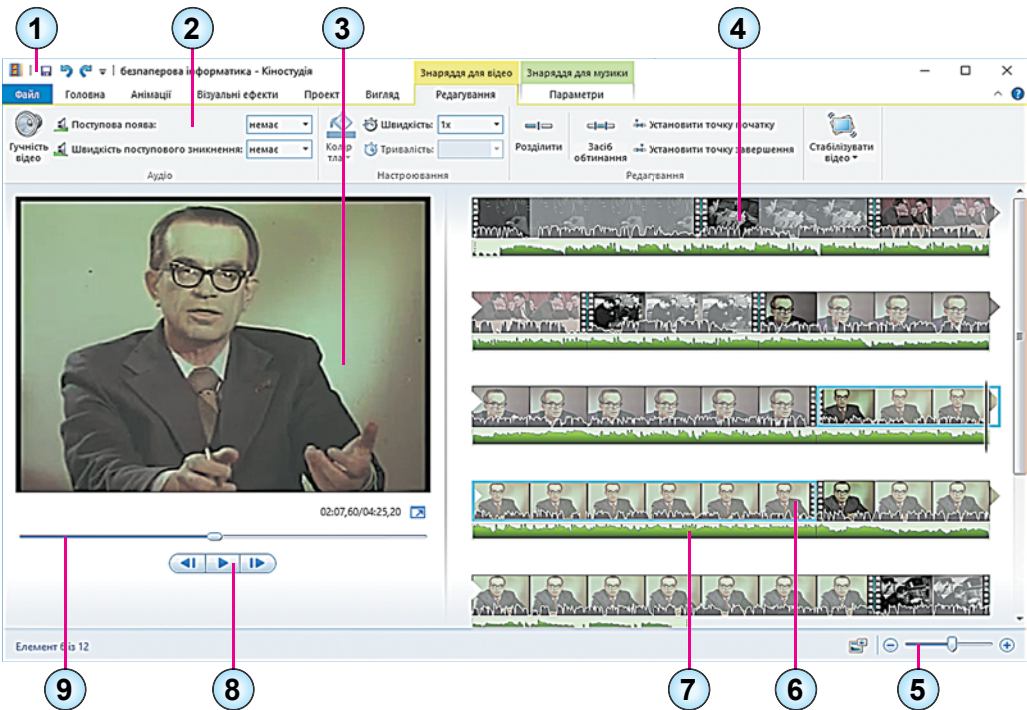
ÁLTALÁNOS ISMERETEK A VIDEÓRÓL

Videó- vagy **mozifilm** (ang. *cine-film* – filmszalag) alatt általában filmművészeti alkotásokat értünk. A felvevőeszközök nagy elterjedtsége ma már bármelyik felhasználó számára lehetővé teszi, hogy saját videót hozzon létre. Ezeket a filmeket természetesen otthon, a családban, vagy baráti társaságban fogják megtekinteni, nem mozikban. A rövid videoanyagokat manapság gyakran emlegetik **klip** vagy **videóklip** néven.



A videó **kockákból** – videorészletekből, állóképekből – áll. A kockák alkotják a videósávot, ezt a szerkesztők a videovonalon helyezik el. A videók a következő elemekből épülnek fel:

- **cím** – néhány szöveges objektum a film elején;
- **feliratok** – szöveges objektumok, amelyek a képi anyag fölött jelennek meg, magyarázzák a látványt, vagy a szerzőket mutatják be;
- **hangzó kíséret**, ami több elemet tartalmazhat:
 - a videó saját hangja;
 - narráció – hangos magyarázat a videóhoz;
 - zene – zenemű részlete, amit általában hangos háttérként használnak;
- **vizuális effektek** – a kép megjelenítésében bekövetkező változás, például áttérés fekete-fehér megjelenítésre régi film hatás céljából;
- **áttűnések** – a kockák váltását kísérő hatások.



- | | |
|---------------------------|------------------------------|
| 1. Gyors elérésű eszközök | 6. Videosáv |
| 2. Szalag | 7. Hangsáv |
| 3. Megtekintés ablak | 8. Lejátszást vezérlő gombok |
| 4. Szerkesztőterület | 9. Haladásjelző |
| 5. Nagyítás | |

4.10. ábra. A Windows Live Movie Maker ablaka

A vizuális effekteket és áttűnéseket általában a videosávon is láthatjuk. A hangos kíséret több sávot is elfoglalhat. A **Windows Live Movie Maker**-ben a videó saját hangsávját a videosávon láthatjuk.

A film készítése során célszerű bizonyos algoritmus szerint eljárni:

- **az első lépés** a forgatókönyv elkészítése. Ez egy részletes terv, amelyben meghatározzuk a kockák sorrendjét, az egyes objektumok elhelyezését. Minél részletesebb a forgatókönyv, annál könnyebb a film elkészítése;
- **a második lépés** a szükséges anyagok előkészítése: a helyszínek előkészítése, a forgatás, a képek és kísérőzene kiválasztása;
- **a harmadik lépés** a vágás – az egyes kockák és a film szerkezetének feldolgozása abból a célból, hogy egy egységes mű jöhessen létre. A vágás során történik a nyersanyag szerkesztése, a szükséges narráció elkészítése, a szövegek, feliratok létrehozása.

A FORGATÓKÖNYV ELKÉSZÍTÉSE

Vizsgáljuk meg a forgatókönyv elkészítésének menetét a *Papírmentes informatika* videó példáján. A videó V. M. Hluskovnak a számítógépes eszközök felhasználására vonatkozó elképzeléseinek megvalósulásáról szól. Hluskov előre látta, hogy a XXI. század elejére az információ döntő többsége nem papíron lesz tárolva, hanem számítógépek memóriájában.

Hluskov vezetésével ezen elképzelések megvalósítása céljából kifejlesztették a **MIR** (mérnöki célú számítógép) számítógép-sorozatot, a személyi számítógép elődjét. A fejlesztés legfontosabb célja a számítógépek alkalmazásának elterjesztése volt a lakosság széles köreiből kortól és végzettségtől függetlenül.

A videót az informatikaórán készülünk bemutatni V. M. Hluskov munkásságának ismertetése céljából. A forgatókönyv a kockák tartalmának, a kockák eseményeinek leírásából áll.

1. 1. kocka. **Címoldal:**

- cím – *Papírmentes informatika*;
- háttér – *világoszöld*, betűszín – *sötétzöld*;
- megjelenítés – 7 s;
- hang – *a háttér hangja*;
- effektus – *kioltás bal oldalról*;
- áttűnés – *mozaik*.

2. 2. kocka. **V. M. Hluskov fényképe:**

- fénykép fájlból;
- megjelenítés – 6 s;
- hang – *narratív szöveg, ebben a kockában kezdődik, a következőben ér véget, a hangzás hossza a felolvasás idejétől függ*;
- effektus – *nincs*;
- áttűnés – *mozaik*;



- a narratíva szövege: *V. M. Hluskov, az Ukrán TA Kibernetikai Intézetének igazgatója a papírmentes informatikára történő áttérést a számítástechnika egyik legfontosabb alkalmazási területének tekintette. Úgy gondolta, hogy a XXI. század elejére az információ mennyiségének jelentős része nem papíron, hanem a számítógépek memóriájában lesz tárolva. Az az ember pedig, aki nem lesz képes ezt az információt felhasználni, a XX. század elejének írástudatlanához lesz hasonló.*
3. 3. kocka. **Videofelvétel V. M. Hluskovról:**
- videó fájlból;
 - időtartam – *a narrátor szövegével egyszerre ér véget;*
 - hang – *általános háttérhang fájlból;*
 - effektus – *nincs.*
4. 4. kocka. **Cím:**
- szöveg: *Hallgassuk meg Viktor Mihajlovics Hluskovot;*
 - háttér – *világoszöld, betűszín – sötétzöld;*
 - időtartam – *7 s;*
 - hang – *általános háttérhang fájlból;*
 - effektus – *kioltás balról;*
 - áttűnés – *mozaik.*
5. 5. kocka. **Hluskov ismertetője a papírmentes informatikáról:**
- videó fájlból;
 - időtartam – *48 s;*
 - effektus – *nincs;*
 - áttűnés – *mozaik.*
6. 6. kocka. **V. M. Hluskov a MIR-1 számítógép fejlesztőinek körében:**
- fénykép fájlból;
 - időtartam – *6 s;*
 - hang – *általános háttérhang fájlból;*
 - felirat: *Ahhoz, hogy az emberek megtanulják a számítógép használatát, át kell térni a kis számítógépekre. V. M. Hluskov felveti a MIR család kifejlesztésének ötletét;*
 - betűszín – *narancs, háttér – sötétzöld;*
 - effektus – *nincs;*
 - áttűnés – *mozaik.*
7. 7. kocka. **A MIR-1 számítógép fényképe:**
- fénykép fájlból;
 - képfelirat: *A MIR számítógépek kis méretűek voltak és kényelmes beviteli eszközzel – elektromos írógéppel – voltak felszerelve;*
 - minden más a 6. kockának megfelelően.

8. 8. kocka. **A MIR-2 számítógép fényképe:**
 - fénykép fájlból;
 - képfelirat – *a MIR-2 új beviteli eszköze, a fényceruza;*
 - minden más a 6. kockának megfelelően;
9. 9. kocka. **A MIR-3 számítógép fényképe:**
 - fénykép fájlból;
 - képfelirat – *a MIR-3 számítógép;*
 - minden más a 6. kockának megfelelően.
10. 10. kocka. **Zárókép:**
 - felirat – *„Jövő” iskolai stúdió;*
 - operatőr – *Sztepan Mihajlenko;*
 - szöveg – *Okszana Kovalenko;*
 - vágó – *Mikola Ivanenko;*
 - háttér – *sárgászöld, betűk színe – fehér;*
 - időtartam – *10 s.*

ANYAGGYÚJTÁS

Nyomtatott kiadványok, internetről letöltött anyagok alapján összegyűjtjük a videó összeállításához szükséges információkat. Ezek listáját tartalmazza a 4.2. táblázat

4.2. táblázat

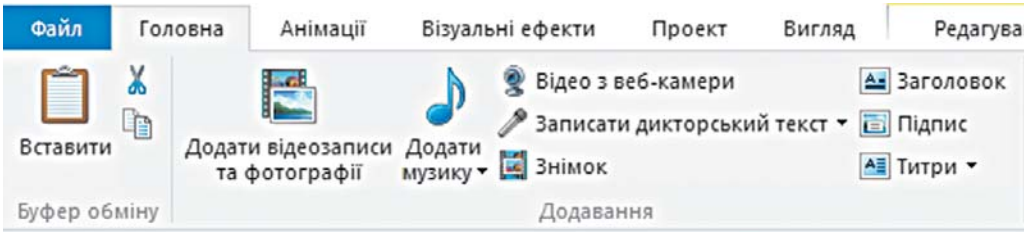
A videóhoz szükséges fájlok

<i>Fájl neve</i>	<i>Mappa</i>	<i>Hosszúság, s</i>	<i>Kocka</i>
Hluskov videó 0.wmv	Rozdil 4\pukt 4.2\video	53	3
Hluskov videó 1.wmv	Rozdil 4\pukt 4.2\video	48	5
Hlushkov.jpg	Rozdil 4\pukt 4.2\foto		2
Hlushkov_MIR-1.jpg	Rozdil 4\pukt 4.2\foto		6
MIR-1.jpg	Rozdil 4\pukt 4.2\foto		7
MIR-2.jpg	Rozdil 4\pukt 4.2\foto		8
MIR-3.jpg	Rozdil 4\pukt 4.2\foto		9
Physical.wma	Rozdil 4\pukt 4.2\audio	220	1–9., az 5. kivételével

A VIDEÓ LÉTREHOZÁSA

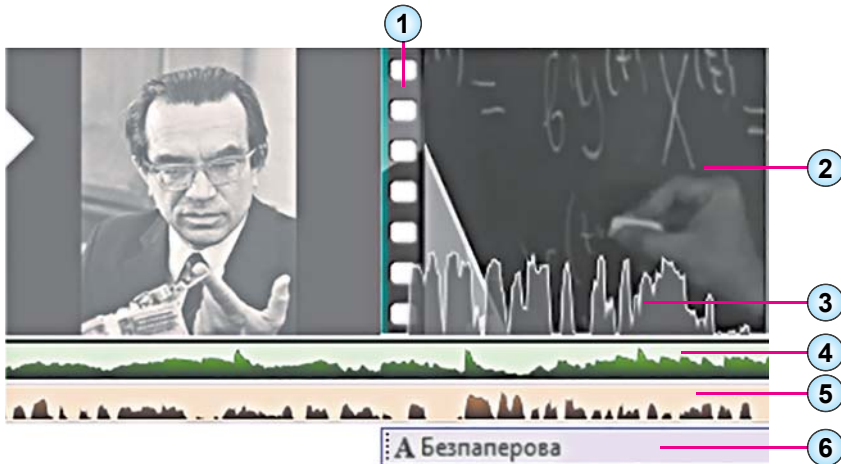
A **Windows Live Movie Maker**-ben a következőképpen készítjük el a filmet:

1. Indítsd el a programot!
2. Mentsd el az új projektet a **Fájl** ⇒ **Projekt mentése** paranccsal! Alapértelmezetten a program **wlmp** kiterjesztésű fájlba menti a projekteket!
3. A **Kezdőlap Hozzáadás** csoportjának vezérlőelemi segítségével helyezzük el a képeket és videoanyagokat a kockákban a forgatókönyvnek megfelelő rendben (4.11. ábra)!



4.11. ábra. A **Hozzáadás** csoport elemei

4. A **Hozzáadás** csoport elemeinek felhasználásával adjuk hozzá a címeket, feliratokat a forgatókönyv szerint!
5. A videoeszközök, hangeszközök, kísérőszövegek, cím és stáblista ideiglenes eszköztárak alkalmazásával végezzük el a videó objektumainak szerkesztését! Figyeljünk oda, hogy a videók és képek a videosávon (4.12.2.), a hangfájlok a hangsávon (4.12.4.), a narráció egy másik hangsávon (4.12.5.), a címek, feliratok, stáblista egy szövegsávon (4.12.6. ábra) jelenik meg. Amennyiben a videóhoz tartozik saját hang, az a videosáv alsó részében jelenik meg (4.12.3. ábra)!
6. Állítsuk be a videosávon a lejátszás idejét, a hangerőt, használjuk az objektumok mozgását a sávokon, a lejátszás kezdetének és végének beállítását!
7. Szükség esetén állítsunk be effektusokat az egyes kockákra (**Vizuális hatások** lap)!
8. Az **Animáció** lap alkalmazásával állítsuk be a kockák közötti áttünéseket!
9. Mentsük a fájlt a **Fájl** ⇒ **Mentés** parancs végrehajtásával!



- | | |
|--------------------------|-----------------------------|
| 1. Kockák határa | 4. Külső hangfájl hangsávja |
| 2. Videosáv | 5. Narráció hangsávja |
| 3. Videó saját hangsávja | 6. Szövegsáv |

4.12. ábra. Szerkesztési terület

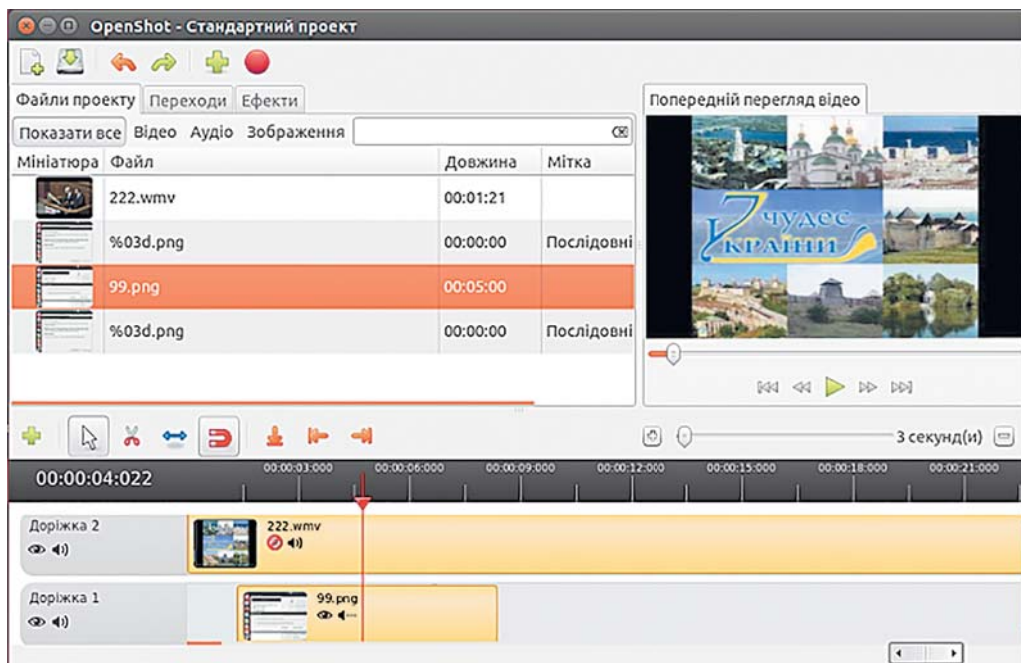
4. fejezet

A szerkesztés során célszerű a projektet időről időre elmenteni. Ha a videó szerkesztését nem fejeztük be, akkor a projektfájl megnyitása után folytathatjuk a szerkesztést. A mentett projekt lehetőséget biztosít arra, hogy egy-egy videót különböző változatokban készítsük el.



Linux-használóknak

Linux alatt az **OpenShot** programot használhatjuk videók szerkesztésére (lásd a képen). A program felhasználói felülete, a műveletek elvégzésének módja hasonlít a **Live Movie Maker**-ben megismertre.



Az Open Shot ablaka

! Összefoglalás

A zeneszerkesztők, videoszerkesztők, stúdióprogramok rendeltetése a hangfájlok és a videofájlok szerkesztése. Ezekkel a programokkal másolhatjuk, áthelyezhetjük, törölhetjük a hangfájlok és a videók részeit, képeket, szövegeket illeszthetünk a videókba.

A stúdióprogramok a szerkesztőkhöz képest többletfunkciókkal rendelkeznek, professzionális hang- és filmstúdióban vannak használatban.

A **Windows Live Movie Maker** segítségével videókból, képekből, narratív szövegből, feliratokból készíthetünk videókat. A létrehozott videót elmenthetjük a program által támogatott formátumokban, vagy közzétehetjük az interneten.



Videók készítése során célszerű betartani a műveletek következő sorrendjét:

- *első lépésként* el kell készíteni a forgatókönyvet;
- *második lépésként* készítsük elő a szükséges videót, hangfelvételeket, szöveges információkat;
- *harmadik lépésként* végezzük el a vágást!



Felelj a kérdésekre!

- 1°. Mi a hang- és videoszerkesztők rendeltetése?
- 2°. Mi a különbség a szerkesztőprogramok és a stúdióprogramok között?
- 3°. Milyen hangszerkesztőket ismersz?
- 4°. Milyen videoszerkesztőket ismersz?
- 5°. Mi a **Windows Live Movie Maker** rendeltetése?
- 6°. Ismertesd a 4.10. ábra alapján a **Live Movie Maker** ablakának objektumait!
- 7°. Mi a **Live Movie Maker Szerkesztőterületének** rendeltetése?
- 8°. Hogy készítünk videofilmet?
- 9°. Mi a forgatókönyv és miért van rá szükség?
- 10°. Hogy illesztünk a videóba fényképet?
- 11°. Hogy illesztjük be a narratív szöveget?
- 12*. Mit értünk a videosáv és a hangsáv szinkronizálásán? Hogy végezzük el ezt **Live Movie Maker**-ben?



Végezd el a feladatokat!

- 1°. Állíts össze forgatókönyvet, és készíts videót Leszja Ukraináról az ukrán irodalomórára! A videóban használj egy részletet a Leszja Ukrainát bemutató filmből, illeszd be a költőné képét és az emlékműveit bemutató fotókat! (**Rozdil 4 \punkt 4.2\zavdannya 4.2.1\Ukrainka**). Készíts a filmhez címkockát, feliratokat! Keress háttérzenét, ami a film elejétől a végéig szóljon! Mentse a videót **feladat 4.2.1.wmv** néven a mappádba!
- 2°. Készíts a 8. osztályos történelemlétkönyv, valamint a **Rozdil 4\punkt 4.2\zavdannya 4.2.2\Kolumb** anyagai alapján filmet Kolumbusz Kristófról! A narratív szöveget és a zenei aláfestést válaszd ki önállóan! Mentse a videót **feladat 4.2.2.wmv** néven a mappádba!
- 3°. Állíts össze forgatókönyvet, és készíts videót Olekszandr Dovzsenko neves ukrán íróról és filmrendezőről az ukrán irodalomórára! Használd fel az irodalomkönyv, valamint a **Rozdil 4\punkt 4.2\zavdannya 4.2.3\Dovzsenko** mappa anyagait! A narratív szöveget és a zenei aláfestést válaszd ki önállóan! Mentse a videót **feladat 4.2.3.wmv** néven a mappádba!



- 4*. Készíts az általad választott szerkesztőben egy videót, amiben a nyolcadik osztályban tanulmányozott fizikai folyamatokat mutatod be! A forgatókönyvet és a szükséges anyagokat állítsd össze önállóan!
- 5*. Készíts az általad választott szerkesztőben egy videót, amiben az osztályodat vagy egy aktuális iskolai eseményt mutatsz be! A forgatókönyvet és a szükséges anyagokat állítsd össze önállóan!

6. SZÁMÚ GYAKORLATI MUNKA

Videoklip készítése. Effektusok alkalmazása, a videosáv és hangsáv időzítése

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Készíts a videoszerkesztővel egy hőmérőket bemutató 2,5 perces videót!
2. Készítsd el a forgatókönyvet a következők figyelembe vételével:
 - tartalmazzon egy címkockát a videó nevével, ami lehet például **Hőmérők**;
 - tartalmazza a hőmérők különböző típusainak bemutatását (4–6 kép) a **Rozdil 4\prakticsna 6\Foto** mappából! Minden képhez készíts feliratot a hőmérő nevével, ami megegyezik a fájl nevével! A képek bemutatása ne tartson 5 másodpercnél tovább;
 - Galilei hőmérőjének bemutatását a **Rozdil 4\prakticsna 6\Foto** mappa 2–3 fotója és a **Rozdil 4\prakticsna 6\video** mappában található videó alapján végezd! A fotókon helyezz el feliratokat, a videót kísérd narratív szöveggel! A feliratok és a kísérszöveg összeállítása során alkalmazd a **Rozdil 4\prakticsna 6\Galilei-hőmérő.doc** fájl anyagait;
 - záró képkockákat a saját neveddel, a forgatókönyvíró és rendező nevével;
 - zenei aláfestést minden olyan kockához, amihez nem tartozik narratív szöveg, használd a **Rozdil 4\prakticsna 6\muzika** mappa fájljait.
3. Mentsd a videót **gyakorlati 6** néven a mappádba!



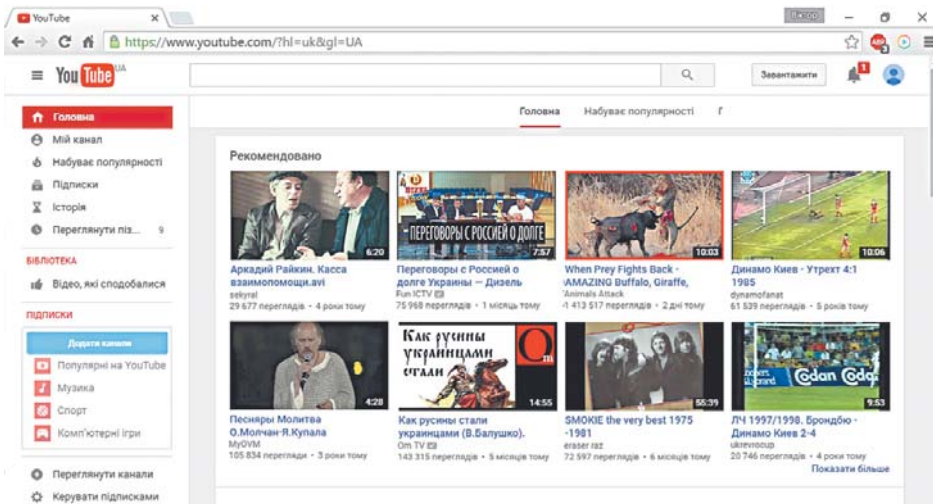
4.3. MULTIMÉDIÁS FÁJLOK KÖZZÉTÉTELE AZ INTERNETEN



1. Mi a felhasználói fiók? Hogy hozunk létre felhasználói fiókot a postaszerveren?
2. Hogy nyitunk meg egy régebben mentett projektet a **Windows Live Movie Maker**-ben?
3. Hogy adhatunk a filmhez kísérszöveget a videoszerkesztőben?


VIDEÓK ELHELYEZÉSE A YOUTUBE-ON

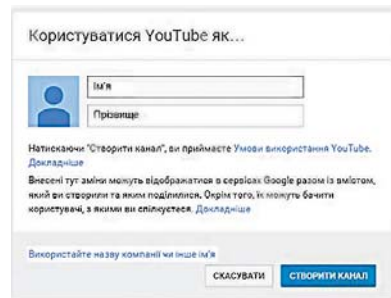
A YouTube (ang. *you tube* – a te telefonkagylód) az egyik legnépszerűbb videomegosztószolgáltatás az interneten. Bármelyik **Gmail** felhasználó jogosult elhelyezni az anyagait a megosztón, és hozzáférési jogokat rendelni a feltöltött fájlokhoz.



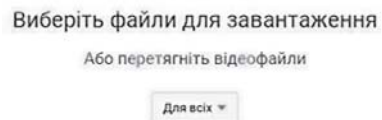
4.13. ábra. A YouTube nyitóoldala

Ahhoz, hogy multimédiás fájlokat helyezze el a **YouTube**-on, a következőképpen kell eljárni:

1. Nyisd meg a szolgáltatás főoldalát (4.13. ábra)!
2. Az ablak bal oldalán kattints a **Bejelentkezés** gombra!
3. A megnyíló párbeszédablakban add meg a **Gmail** felhasználói nevedet és jelszavadat!
4. Add meg, milyen néven szeretnél publikálni (alapértelmezetten a **Gmail** fiókad adatait használja a szolgáltatás) (4.14. ábra)!
5. Ismerkedj meg a felhasználási feltételekkel!
6. Kattints a **Csatorna létrehozása** gombra!
7. Kattints az ablak jobb oldalán található **Feltöltés** gombra!
8. Kattints az **Adatvédelem** gombra (4.15. ábra), és add meg a feltöltendő fájlhoz rendelt jogosultságokat (*nyilvános, privát, nemlistázott*)!
9. Kattints a  gombra, és válaszd ki a feltöltendő fájlt!
10. Várd meg a feltöltést, majd add meg a fájl tulajdonságait (4.16. ábra):
 - a videó nevét (4.16.3. ábra);
 - a rövid leírást – röviden ismerteted a témát (4.16.4. ábra);
 - a kulcsszavakat (tagok) a videó kereshetőségének meggyorsítására (4.16.8. ábra);



4.14. ábra. Csatorna létrehozása a YouTube-on

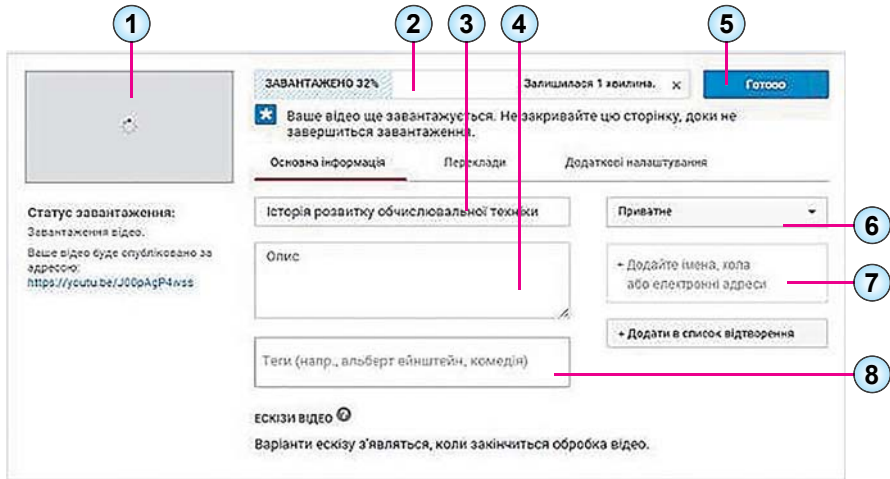


4.15. ábra. A fájl feltöltése ablak



4. fejezet

- a barátaid és köreid nevét, akiket értesíteni szeretnél a feltöltésről (4.16.7. ábra), amennyiben a hozzáférés privát.
11. Kattints az **OK** gombra!



- | | |
|----------------------------------|------------------------|
| 1. Vázlat nézet | 5. Kész gomb |
| 2. Feltöltés előrehaladás- jelző | 6. Jogosultságok lista |
| 3. Név mező | 7. Címzettek |
| 4. Leírás mező | 8. Kulcsszavak |

4.16. ábra. A videó tulajdonságainak beállítása ablak

Feltöltés után a videót megtekinthetjük, vagy a megtekintés során a képernyő alján látható eszközökkel (4.17. ábra) szerkeszthetjük is:

- megváltozathatjuk a leírást (**Információk és beállítások** gomb);
- megváltoztathatjuk a fényerőt, a kontrasztarányt, a színek telítettségét, a lejátszás sebességét (**Kiegészítő lehetőségek** gomb);
- hangot adhatunk hozzá (**Hang** gomb);
- feliratokat adhatunk hozzá (**Feliratok** gomb).

Konfidenciális (a latin *confidentia* – bizalom szóból) – korlátozott hozzáférésű, korlátozott terjesztésű.



4.17. ábra. Szerkesztési eszközök

A létrehozott csatornán keresztül a felhasználóknak lehetőségük van további fájlok feltöltésére.

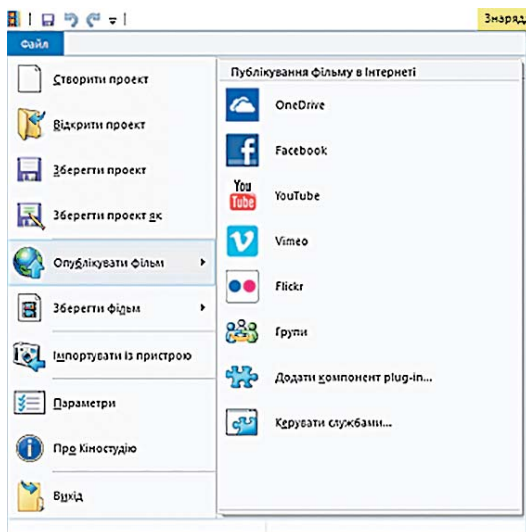
Videóink feltöltésére igénybe vehetjük egyéb megosztók szolgáltatásait is, ilyen lehet például a **Vimeo** (<https://vimeo.com>), a **dailymotion** (<http://daylimotion.com>), a **Video ukrhome** (<http://video.ukrhome.net>) vagy a közösségi oldalakat, például a **Facebookot**.



VIDEÓ FELTÖLTÉSE AZ INTERNETRE A WINDOWS LIVE MOVIE MAKER-REL

Amikor befejeztük a videó szerkesztését, a **Movie Maker** segítségével nemcsak elmenthetjük azt, de publikálhatjuk is az interneten. E célból nyissuk meg a projektet, majd:

1. Hajtsuk végre a **Fájl** ⇒ **Film közzététele** parancsot!
2. Válasszuk ki a szolgáltatók listájából azt (4.18. ábra), ahová el szeretnénk helyezni a fájlt, például a **You Tube**-ot!
3. A megnyíló párbeszédablakban adjuk meg a **Gmail** fiókunk felhasználói nevét és jelszavát!
4. Adjuk meg a film nevét, leírását, hozzáférési beállításait, kategóriáját, tag-jeit!
5. Kattintsunk a **Közzététel** gombra!
6. Várjuk meg a feltöltést!
7. Tekintsük meg a feltöltött videót online!
8. Zárjuk be a böngésző ablakát!



4.18. ábra. A szolgáltatók listája

HANGFÁJLOK KÖZZÉTÉTELE AZ INTERNETEN

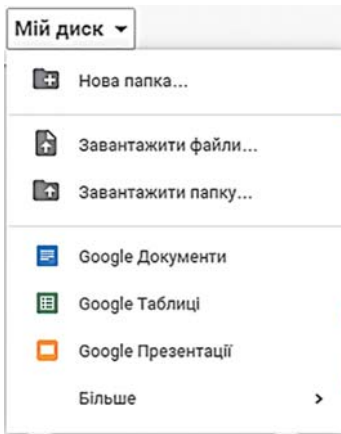
A hangfájlokat kezelő szolgáltatások döntő többsége azoknak lett szánva, akik saját zeneműveket hoznak létre. A másik irányzat azoknak szól, akik hangoskönyveket készítenek különböző – politikai, gazdasági, műszaki stb. – témákban.

A felsorolt hangfájlokat **podcast**oknak nevezik. A továbbiakban a **podcast** kifejezést használjuk a felhasználók által az interneten publikált hangfájlokra is.

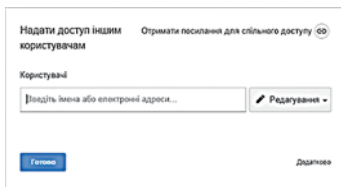
A hangfájlok közzétételét támogató szolgáltatások a **Civil podcast csatorna** (<http://cpod.co/>), az **Ukrajnai podcast terminál** (<http://podcaster.org.ua>), a **Best Podcast Directory** (<http://www.podcast.com>), a **BBC** média-vállalat angol nyelvet tanuló ukránok számára fenntartott oldala (<http://www.bbc.co.uk/ukrainian/learningenglish>) stb.

Amennyiben a barátainkkal, ismerőseinkkel szeretnénk hangfájlokat megosztani, kényelmes megoldás a **Google Drive** használata. Fájlokat **Google** fiókkal rendelkező felhasználók helyezhetnek el a **Google Drive**-on a következőképpen:



A **Podcast** szó az *ipod* (az Apple médialejátszója) és *broadcasting* (műsor-szórás) szó összeolvadásából keletkezett. A podcasting technológia lehetővé teszi hang-, videó- és más fájlok sorozatszerű közzétételét az interneten.





4.19. ábra. A **Saját meghajtó** legördülő menü



4.20. ábra. A megosztás ablak

1. Indítsuk el a böngészőt!
2. Nyissuk meg a **Gmail** fiókunkat! Adjuk meg a választott felhasználói nevünket és jelszavunkat, majd kattintsunk a **Bejelentkezés** gombra!
3. A  gombra kattintva nyissuk meg a **Google** alkalmazásokat!
4. Válasszuk ki a **Google Drive**-ot !
5. Gördítsük le a **Saját meghajtó** listát, majd válasszuk a **Fájl feltöltése** lehetőséget (4.19. ábra)!
6. Válasszuk ki a feltöltendő fájlokat, majd kattintsunk a **Megnyitás** gombra!

Ezután lehetőségünk van arra, hogy hozzáférési jogokat adjunk más felhasználóknak, ha a **Megosztás**  gombra kattintunk, a megnyíló ablakban (4.20. ábra). A **Felhasználók** mezőbe felvehetjük a barátaink e-mail címét, akiknek ezután automatikusan levelet küld a szolgáltatás a nevünkben, aminek tartalma a fájl elérési útja lesz. A címzeteknek ezután lehetőségük lesz a fájl meghallgatására.

Ha a **Megosztó Link létrehozása**  gombra kattintunk, azoknak a felhasználóknak lesz hozzáférése a fájlhoz, akik a megosztó linket megkapják. Ez a link automatikusan bemásolódik a **Vágólapra**, amit beilleszthetünk a leveleinkbe, amennyiben ezeket a leveleket mi magunk szeretnénk elküldeni anélkül, hogy kitöltenénk a **Felhasználók** mezőt.

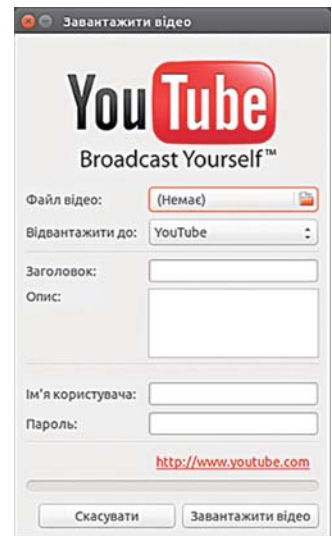


Linux-használóknak

Linux alatt az **OpenShot** szerkesztő segítségével tehetünk közzé videókat az interneten. A program indítása után:

1. Hajtsuk végre a **Fájl** ⇒ **Videó betöltése** parancsot!
2. Jelöljük ki a feltöltendő videót (4.21. ábra)!
3. Válasszuk ki a szolgáltatót, ahová fel szeretnénk tölteni a fájlt (például a **YouTube**)!
4. Töltsük ki a videó tulajdonságait tartalmazó mezőket (név, leírás)!
5. Adjuk meg a kiválasztott szolgáltatónál érvényes felhasználói nevünket és a jelszavunkat!
6. Kattintsunk a **Feltöltés** gombra!

Minden, a videók elhelyezésével kapcsolatos teendőnk hasonló a fentebb leírtakhoz.



4.21. ábra. A **Videó feltöltése** ablak



! Összefoglalás

Az interneten sok olyan webhelyet találhatunk, amelyen multimédiás fájlokot tehetünk közzé, és oszthatunk meg másokkal. A fájlok feltöltéséhez felhasználói hozzáférés szükséges.

A legelterjedtebb fájlmegosztó szolgáltatás a **YouTube** (<https://www.youtube.com>). Ide úgy tölthetünk fel videókat, ha létrehozunk egy csatornát, amely a felhasználó fájljait tartalmazza majd. A felhasználóknak lehetősége van arra, hogy szabályozzák mások hozzáférését a feltöltött fájljaikhoz.

A videoszerkesztők többsége támogatja a fájlok közzétételét a neten. Ilyenek például a **Windows Live Movie Maker** és az **OpenShot**.

Az internetre feltöltött multimédiás fájlokot **podcast**nak nevezik, a létrehozó felhasználókat pedig **podcaster**nek.

? Felelj a kérdésekre!

- 1°. Milyen szolgáltatóknál helyezhetünk el multimédiás tartalmakat az interneten?
- 2°. Hogy hozunk létre csatornát feltöltés céljából a **YouTube**-on?
- 3°. Milyen tulajdonságai vannak a **YouTube**-on elhelyezett videóknak?
- 4°. Milyen hozzáférési lehetőségeket biztosít a **YouTube**?
- 5°. Hogy rendelhetünk zenei háttérrel a **YouTube**-ra feltöltött fájlhoz?
- 6°. Hogy tehetünk közzé videókat a **Windows LiveMovie Maker** segítségével?
- 7°. Hogy töltsünk fel hangfájlokat a **Google Drive**-ra?
- 8°. Mi a **podcast**? Ki a **podcaster**?
- 9°. Hogy adhatunk hozzáférést a fájljainkhoz másoknak a **Google drive**-on?

📁 Végezd el a feladatokat!

- 1°. Tekintsd meg a <https://youtu.be/NcrOD9RysUo> címen található filmet, határozd meg a lejátszás idejét!
- 2°. Helyezd el a **YouTube**-on a **Rozdil 4\pункт 4.3\video\szejm sztare ricsiscse.mov** videót! Add meg a rövid leírást, zenei aláfestést és a tag-eket! Küldd el a tanárodnak a videó elérhetőségét!
- 3°. Töltsd fel a **YouTube**-ra a múlt foglalkozásokon **Dovzsenkóról** készült filmet a **Live Movie Maker** segítségével!
- 4°. Töltsd fel a **YouTube**-ra a tanárod által megadott fájlt (például a **Rozdil 4\pункт 4.3\video\ravlik.mov**-ot)! Add meg a rövid leírást és a tag-eket! Keress az interneten információkat a csigákról, és helyezz el ezek alapján feliratokat a filmkockákon!
- 5°. Helyezd el a **Google Drive**-on a tanárod által megadott fájlt (például a **Rozdil 4\pункт 4.3\video\scsedrivocska.wma**-t)! Adj hozzáférést azoknak a felhasználóknak, akik megkapják a fájl linkjét! Küldd el a linket e-mailban egy barátodnak és a tanárodnak!



7. SZÁMÚ GYAKORLATI MUNKA Hangfájlok és filmek elhelyezése az interneten

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Töltsd fel a **YouTube**-ra a tanárod által megadott fájlt (például a **Rozdil 4\prakticsna 7\video\komahi.wmv**-t)!
2. Állítsd be a következő tulajdonságokat:
 - név – *Rovarok*;
 - leírás – *Ukrajna központi régiójának rovarvilága*;
 - tag-ek – *természet, rovarok, szarvasbogár*;
 - hozzáférés szintje – *privát*;
 - vázlat – *szarvasbogár*;
 - hozzáférés e-mail címe – *a tanárod e-mail címe*;
 - feliratok – *00.01.0-tól 00.09.0-ig nagy szarvasbogár (Lucanus cervus) – a szarvasbogárfélék legtermetesebb képviselője*;
 - zene – stílus *country és folk*, név – *As We Go*.
3. Töltsd fel a **Google Drive**-ra a tanárod által megadott fájlt (például a **Rozdil 4\prakticsna 7\Audio\muzika.mp3**-at)! Adj hozzáférési jogot az osztálytársaidnak és a tanárodnak a zene meghallgatására!

5. fejezet

Az esemény- és objektumorientált programozás alapjai

Ebben a fejezetben megismerkedtek:

a programozási nyelvekkel és azok alkotóelemeivel

a **Lazarus** programozási környezettel

a form, a gomb, címke, a vezérlőelemekkel, azok tulajdonságaival, eseményeivel

az eseménykezelőkkel

az elemek tulajdonságainak megváltoztatásával a program futtatása előtt és közben

a párbeszédablakkal

a **Lazarus** projektek létrehozásával és az **Object Pascal** programozási nyelvvel



5.1. A SZÁMÍTÓGÉPES PROGRAMOK ÉS PROGRAMOZÁSI NYELVEK



1. Mi az algoritmus? Mi az algoritmus-végrehajtó utasításkészlete?
2. Mi a számítógépes program?
3. Milyen programozási nyelvekkel ismerkedtetek régebben? Idézd fel ennek néhány utasítását!

SZÁMÍTÓGÉPES PROGRAMOK

Már tudjátok, hogy a számítógép a szoftver vezérlése alatt működik, amit a különböző rendeltetésű programok alkotnak. Otthon vagy az iskolában használva a számítógépet, azon szövegszerkesztőt, képszerkesztőt, tömörítőprogramokat, táblázatkezelőt, prezentáció-készítőt, oktatási és ellenőrző szoftvereket, játékokat futtattatok.

Azt is tudjátok, hogy a **számítógépes program** egy adatfeldolgozó algoritmus, ami a számítógép számára értelmezhető nyelven van leírva.

Működésük során a programok **adatokat** dolgoznak fel.

Azok az adatok, amelyeket a program a bemeneti perifériától (billentyűzet, egér, érzékelők), vagy más forrásból (például egy szöveges állományból) megkap, a **bemenő adatok**. Vannak programok, amelyek bemenő adatok nélkül dolgoznak.

A program működése során az adatok feldolgozásra kerülnek, új adatok jönnek létre. Ezek a **köztes adatok**.

Ha a program célja bizonyos adatok előállításra, ezeket **kimeneti adatoknak** nevezzük.

A mai programok többsége működése során bizonyos lehetőséget biztosít a felhasználók részére, hogy beavatkozassanak a program és az eszközök működésébe. Ilyenek a gombok, menük, beviteli mezők, számlálók, címkék stb. Ezeknek az eszközöknek az összessége alkotja a **felhasználói felületet**.

PROGRAMOZÁSI NYELVEK

Amikor emberek számára állítunk össze algoritmust, emberi nyelveket használunk: ukránt, orosz, magyart, németet stb.

A számítógép részére összeállított algoritmus esetében nem használhatjuk a bonyolult és nem mindig egyértelmű (homonímiák) emberi nyelveket. Ezért a számítógépes algoritmusokat speciális nyelveken, úgynevezett **programozási nyelveken** írják.

Azt a nyelvet, amelynek rendeltetése, hogy a számítógépnek értelmezhető algoritmusokat fogalmazzunk meg, **programozási nyelvnek** nevezzük.

Az utóbbi 70 évben nagyjából 3000 programozási nyelvet hoztak létre. Egyeseket ma már nem használnak, másoknak új, kényelmesebben használható változatai jelentek meg. Időről időre új programozási nyelvek is megjelennek.



Vannak olyan nyelvek, amelyek a tudományos-műszaki, termelési és mindennapi problémák széles körében alkalmazhatóak, míg másokat speciális problémák megoldása céljából hoztak létre.

A programozási nyelvek **mesterséges nyelvek**. A kémiai reakciók leírásának nyelve, a matematikai egyenletek és egyenlőtlenségek nyelve, a postai irányítószámok nyelve, a sakkjátszmák leírásának nyelve szintén mesterséges nyelvek.

Egy nyelv a következő elemeket tartalmazza:

1. **Ábécé** – azok a szimbólumok, amelyeket a nyelv szavainak és mondatainak lejegyzésére használunk.

2. **Szótár** – a nyelvben használható szavak jegyzéke.

3. **Szintakszis** – a nyelvi szerkezetek megalkotásának szabályai.

4. **Szemantika** – a nyelvi szerkezetek értelmezése, alkalmazásának szabályai.

A 6–7. osztályban a **Scratch** programozási nyelvet tanultátok. Idézzük fel, ennek a nyelvnek az ábécéje az angol és a magyar nyelv betűiből, a $:$, $=$, $+$, $*$, $<$ stb. jelekből áll. A szótár tartalmazza az **ismétlés**, a **mozgás**, a **ha, amikor, pattanj vissza** utasításokat. A nyelv szintakszisa előírja, hogy az előtesztelt ciklus az **ismételd, amíg** szavakkal kezdődik. A nyelv szemantikája egyértelműen **lépj 10 lépést** utasítás az algoritmus végrehajtóját 10 lépéssel mozdítja el az állásának megfelelő irányba.

A meg nem engedett szimbólumok használata, a szótár szavainak nem pontos leírása, a szintaktikai szabályok megsértése értelmezhetetlenné teszi az algoritmust a számítógép számára. Az ilyen hibákat **szintaktikai hibáknak** nevezzük.

A 8. és 9. osztályban az **Object Pascal** (objektumorientált pascal) nyelvvel ismerkedünk majd. A nyelv alkalmas arra, hogy a tanulók és hallgatók megismerkedjenek a programozás alapjaival, de arra is, hogy professzionális programozók különféle programokat hozzanak létre a segítségével.



Érdekes történelmi tények

A **Pascal** programozási nyelvet a svájci műszaki egyetem szakemberei hozták létre 1970-ben Nicolaus Wirth (szül. 1934-ben, lásd 5.1. ábra) professzor vezetésével. A nyelvet a híres francia matematikus, fizikus, irodalmár és filozófus Blaise Pascal tiszteletére nevezték el.

A **Pascal** nyelvet kifejezetten oktatási céllal hozták létre, hogy a diákokat a programozási eljárások alkalmazására tanítsák. Idővel kiderült, hogy a programozási nyelv használata elég kényelmes, ezért elkezdtek gyakorlati problémák megoldására is alkalmazni. Megjelentek az első compilerok, amelyek a **Pascal** nyelven írott programokat a számítógép által végrehajtható programokká fordítják, később pedig a **Turbo Pascal** programozási környezet, ami nagy népszerűsége tette szert.

Az évek során a nyelvet átalakították, kibővítették, tökéletesítették, így jött létre a **Pascal objektum-orientált változata**, az **Object Pascal**. A 2000-es évek elején megjelent az **Object Pascal** nyelvre támaszkodó **Delphi** programozási környezet.



5.1. ábra. Nicolaus Wirth



A LAZARUS PROGRAMOZÁSI KÖRNYEZET

Az **Object Pascal** programozási nyelven írt programok létrehozására, szerkesztésére, tesztelésére és javítására a **Lazarus programozási környezetet** fogjuk használni.


A programozási környezet a következő részekből áll:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;
- **compiler**, ami az **Object Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**.

Lazarusban a fejlesztés során rendelkezésünkre áll a **vizuális komponensek** könyvtára (*Lazarus Component Library* – **Lazarus** komponenskönyvtár). Ezek tartalmazzák a számotokra már jól ismert gombokat, mezőket, címkéket, jelölőnégyzeteket, kapcsolókat, számlálókat, listákat, gördítősávokat stb. Amikor ezeket a komponenseket elhelyezzük a formon, automatikusan létrejön a megfelelő **Object Pascal** kód, amit nem kell begépelni, ezáltal sokkal könnyebb és gyorsabb a programok létrehozása. A programozásnak ez a módszere a **vizuális** programozás.

A **Lazarus** környezetben írt programokat **projekteknek** nevezzük. A projekt több fájlból áll.

A **Lazarus** a következőképpen indíthatjuk:

- a **Start** ⇒ **Programok** ⇒ **Lazarus** paranccsal;
- a **Munkaasztal**  parancsikonyjával.

A program indulásakor megnyílik a **Lazarus** rendszer ablaka (5.2. ábra), ami tartalmazza a **Főmenüt**, az eszköztárat, a vezérlőelemek lapjait és néhány alárendelt ablakot, amelyek között megtaláljuk:

- az **Objektum felügyelőt**, amelynek segítségével beállíthatjuk az objektumok tulajdonságait, elkészíthetjük az eseményvezérlőket stb.;
- a **Szövegszerkesztőt**, ahová beírhatjuk az **Object Pascal** nyelven írt program szövegét;
- a formot.

A **form** az az objektum, ahol el fogjuk helyezni a program vezérlőelemeit, például a gombokat, címkéket, menüket, gördítősávokat. A program futtatása során a formból lesz a programablak, ami tartalmazza az elhelyezett vezérlőket.

A formnak vannak **tulajdonságai**, ezeket az **Objektum felügyelő** ablakában tekinthetünk meg. Az ablak bal oldali része tartalmazza a tulajdonságokat, a jobb oldali pedig azok értékeit. A tulajdonságok egy részét beírhatjuk a billentyűzetről, másokat a legördülő listából választhatjuk ki.

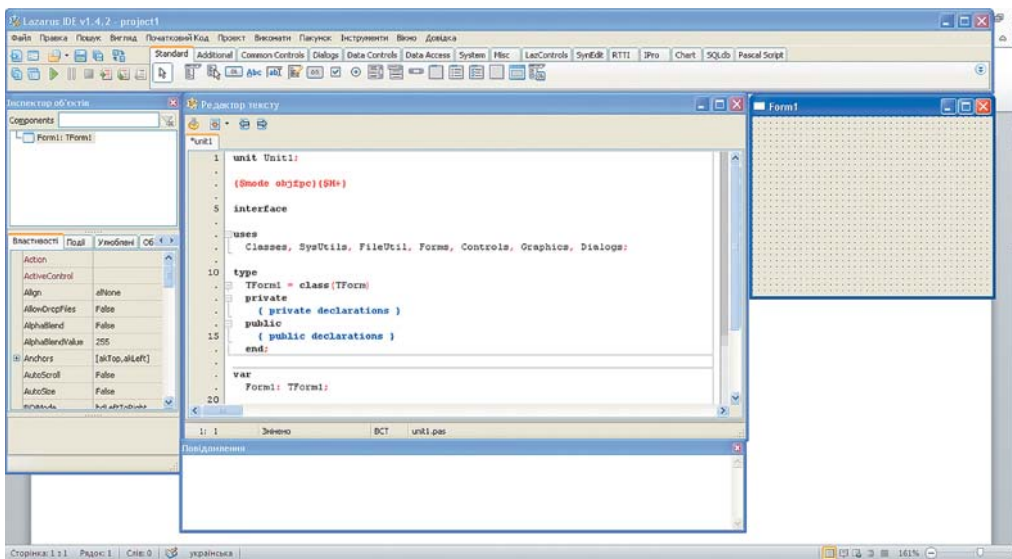
Lássunk néhány tulajdonságot:

Name (név) – a formnak, mint objektumnak neve;

Caption (cím) – ez jelenik meg az ablak címsorában;

Color (szín) – a form háttérszíne;

Height (magasság) – az ablak magassága pixelekből;



5.2. ábra. A Lazarus környezet az alárendelt ablakkal

Width (szélesség) – a form szélessége pixelekben;

Top (teteje) – az ablak bal felső sarkának pixelben mért távolsága a képernyő tetejétől;

Left (bal) – az ablak bal felső sarkának pixelben mért távolsága a képernyő jobb szélétől.

A **Top** és **Left** tulajdonságokat megváltoztatjuk a form képernyőn történő mozgásával, a **Height** és **Width** tulajdonságokat pedig a form szélein található markerek mozgásával.

Ahogy azt már említettük, a modern programozási rendszerekben – ezek közé tartozik a **Lazarus** – a program létrehozása részben automatizált, ami a kezdők és a szakemberek munkáját jelentősen megkönnyíti. Már a **Lazarus** indítása során, még mielőtt egyetlen sornyi programot beírtunk volna, a rendszer kigenerálja a projekt alapértelmezett szövegét. Ez a szöveg elég arra, hogy a projekt futtatható legyen.

Futtatás előtt a projektet ajánlott elmenteni. Ahogy azt már említettük, a projekt több fájlból áll. Ezért minden projektet célszerű külön mappába menteni. A mentést a **Fájl** ⇒ **Mindent menti** vagy a **Projekt** ⇒ **Projekt mentése** paranccsal történhet.

A rendszer egymás után a következő fájlokat ajánlja mentésre:

a **Project1** nevű **lpi** kiterjesztésű fájlt;

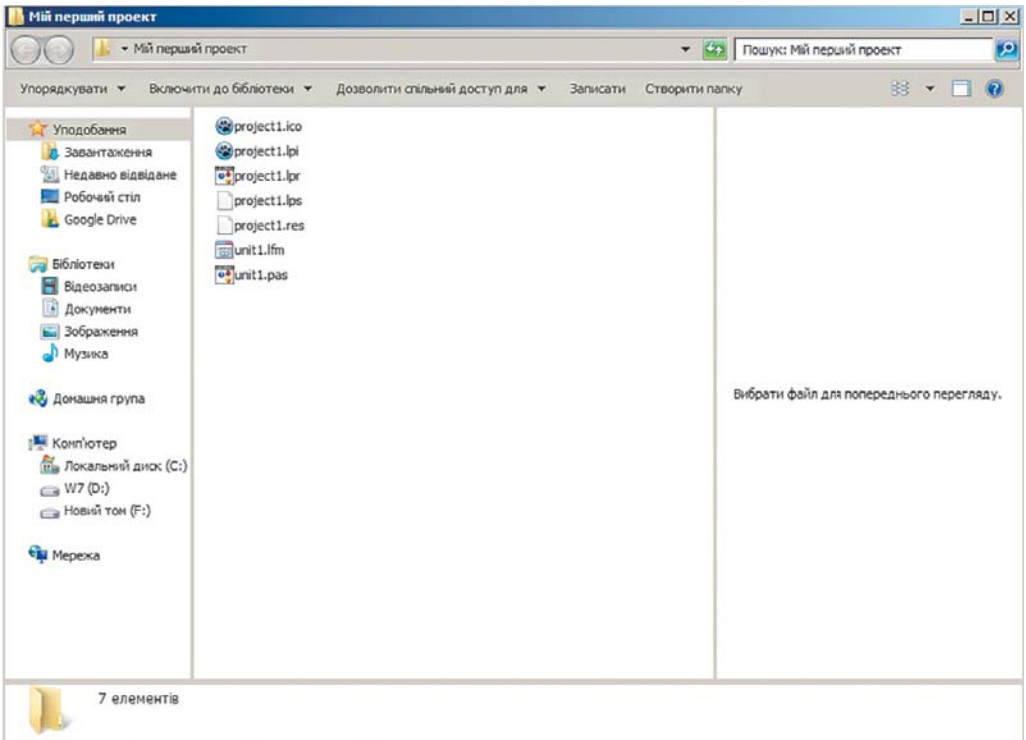
az **Unit1** nevű **pas** kiterjesztésű fájlt.

A fájlok neveit megváltoztathatjuk, a kiterjesztéseiket azonban nem.

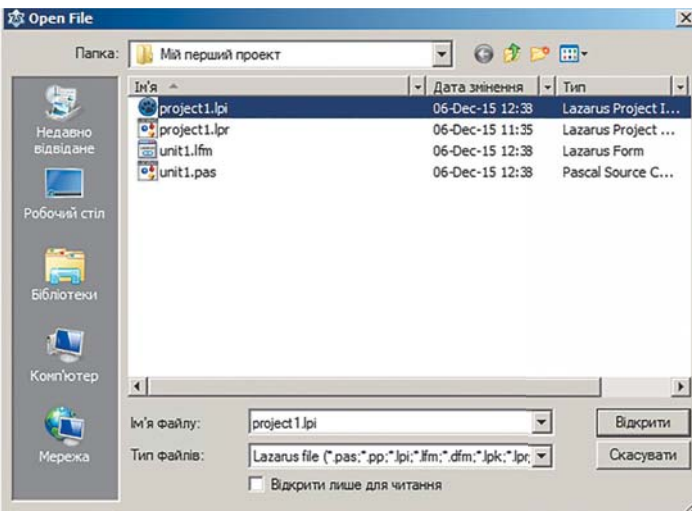
A projekt valamennyi fájlja a megadott mappába (például az **Első projektem** mappába) történik (5.3. ábra).

Ha a már mentett fájlt megváltoztatjuk, az újbóli mentés ugyanabba a mappába történik majd, a mentés során nem nyílnak meg újabb ablakok.

A mentett projektet elmenthetjük másik mappába is, ezt a **Projekt** ⇒ **Projekt mentése másként** paranccsal célszerű megtenni. Másolat mentése során meg lehet változtatni a fájlok neveit is.



5.3. ábra. A projekt fájljait tartalmazó mappa

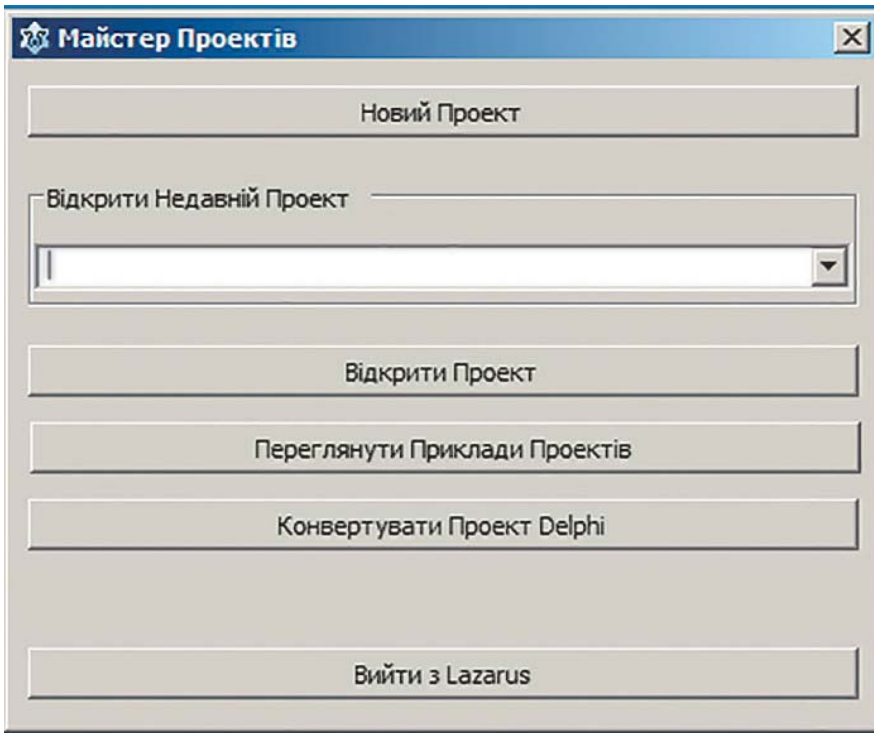


5.4. ábra. Projekt megnyitása ablak

A mentett projektet a **Fájl** ⇒ **Megnyitás** parancssal, az eszköztár **Megnyitás** gombjára kattintva, vagy a **Projekt** ⇒ **Projekt megnyitása** parancssal nyithatjuk meg. A megnyíló ablakban válasszuk ki az **lpi** kiterjesztésű fájlt, majd kattintsunk az **OK** gombra (5.4. ábra).

Előfordul, hogy a **Lazarus** indítása után a legutóbb szerkesztett projekt is meg lesz nyitva. Amennyiben nem ezzel a

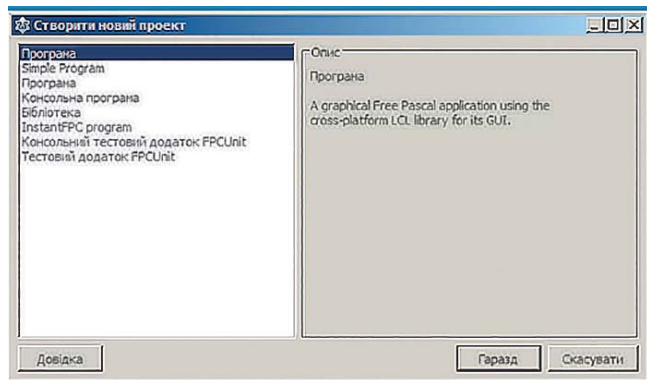
projekttel kívánunk dolgozni, a **Projekt** ⇒ **Projekt bezárása** parancssal bezárhatjuk azt. Ezután megnyílik a **Projekt varázsló** (5.5. ábra), amelynek segítségével új projektet hoz-



5.5. ábra. A Projekt varázsló ablak

hatunk létre, vagy egy **nemrég szerkesztett projektet** nyithatunk meg a munka folytatása céljából. Lehetőségünk van bármely mentett projekt megnyitására is.

Ha az **Új projekt** megnyitása mellett döntöttünk, a megnyíló **Projekt létrehozása** ablakban válasszuk a **Program** opciót, majd kattintsunk az **OK** gombra.



5.6. ábra. Az Új projekt létrehozása ablak

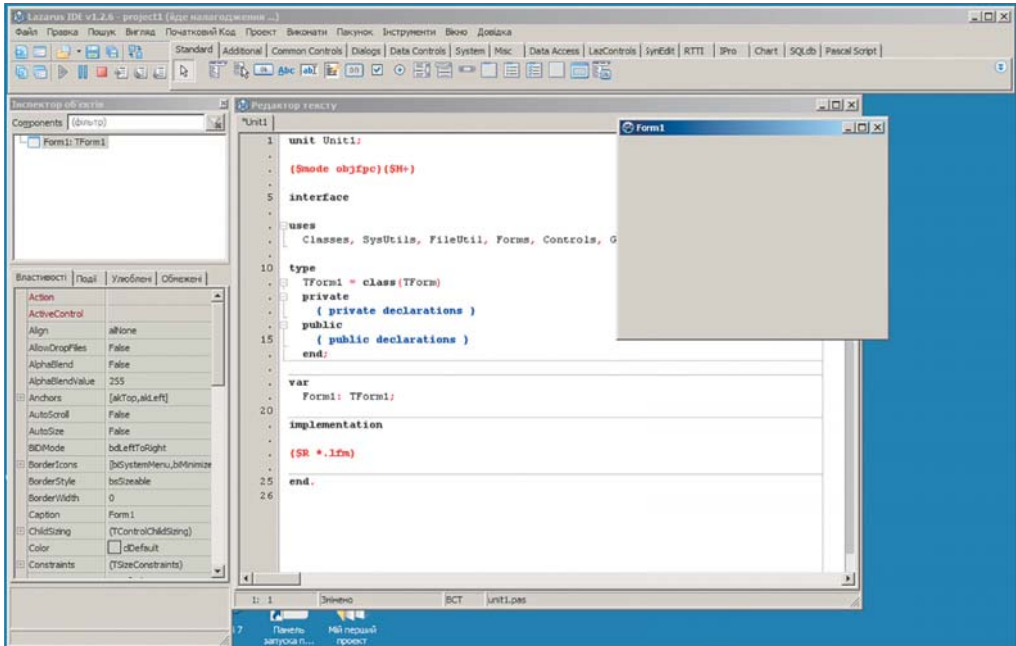
PROJEKT FUTTATÁSA

A projekt futtatását a **Futtatás** \Rightarrow **Futtatás** paranccsal, az eszköztár **Végrehajtás** gombjával vagy az **F9** funkcióbillentyű leütésével kezdeményezhetjük.

Amikor a projektet futtatjuk, először is a **compiler** végzi el a munkáját. Először elvégzi a program szövegének szintaktikai elemzését. Ha hibát talál, megszakítja munkáját és megmutatja az első megtalált hibát. A hiba kijavítása után újból el kell indítani

a futtatást. Ha nem talál hibát, akkor átalakítja az **Object Pascal** nyelvű programot a számítógép számára értelmezhető utasításokká, majd átadja azt a processzornak végrehajtás céljából.

Az alapértelmezett projekt futtatásakor a képernyőn megjelenik egy ablak, amelynek beállításai (szín, méretek, elhelyezkedés, cím stb.) megegyeznek az alapértelmezett értékekkel (5.7. ábra).



5.7. ábra. Az alapértelmezetten létrejövő projekt és futtatásának eredménye

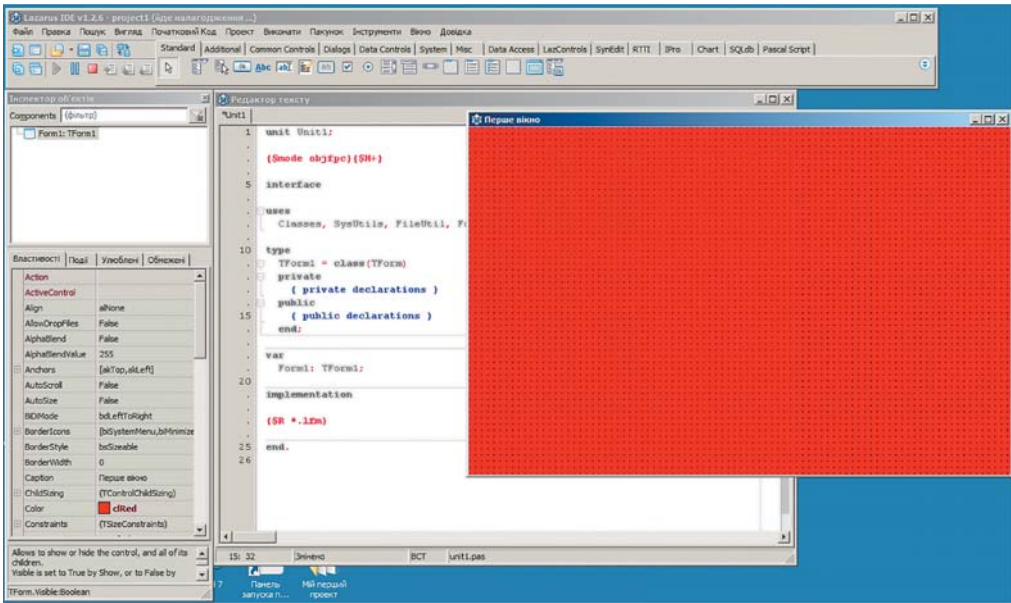
Az **Ablak bezárása** gombra kattintva vagy az **Alt + F4** billentyűkombinációt leütve leállíthatjuk a projekt futását.

Az alapértelmezett projektet megváltoztathatjuk. Változtassuk most meg az ablak tulajdonságait. Keressük meg a **Tulajdonságokat** az **objektumfigyelőben**, majd változtassuk meg azok értékeit a következőképpen:

- Caption** – *Első programom*
- Color** – *clRed* (*color* – szín, *red* – piros)
- Height** – *450*
- Width** – *700*
- Left** – *600*
- Top** – *350*.

Ha elmentjük a megváltozott projektet, majd lefuttatjuk azt, akkor az új tulajdonságokkal rendelkező ablak nyílik majd meg a képernyő megfelelő helyén (5.8. ábra).

A **Lazarus** környezetet a **Lazarus** ablak **Bezárás** gombjával vagy a **Fájl** menü **Kilépés** parancsával zárhatjuk be.



5.8. ábra. A megváltoztatott projekt és futtatásának eredménye

Összefoglalás

A **számítógépes program** a számítógép által értelmezhető nyelven megírt adatfeldolgozó algoritmus.

A számítógép **adatok** feldolgozását végzi.

A külső eszközök (billentyűzet, egér, érzékelők), vagy más források (például szöveges fájl, vagy másik program) által szolgáltatott adatok a **bemenő adatok**. Vannak programok, amelyek nem használnak bemenő adatokat.

A program futása során az adatok feldolgozásra kerülnek és **köztes adatok** jöhetnek létre.

Amennyiben a program célja valamilyen adatok előállítása, ezek lesznek a program **kimenő adatai**.

Azt a nyelvet, amelyen a számítógép számára értelmezhetően fogalmazunk meg algoritmusokat, **programozási nyelveknek** nevezzük.

A programnyelvek összetevői:

1. **Ábécé** – azok a szimbólumok, amelyeket a nyelv szavainak és mondatainak lejegyzésére használunk.
2. **Szótár** – a nyelvben használható szavak jegyzéke.
3. **Szintaxis** – a nyelvi szerkezetek megalkotásának szabályai.
4. **Szemantika** – a nyelvi szerkezetek értelmezése, alkalmazásának szabályai.

A **Lazarus programozási környezet** részei:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;

- **compiler**, ami az **Object Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**.

A **form** az az objektum, amelyen komponenseket – gombok, mezők, címkék, menük, gördítősávok stb. – helyezhetünk el. A program futtatása során a formból jön létre a program ablaka a vezérlőelemekkel.

A form tulajdonságai:

Name (név) – a formnak, mint objektumnak neve;

Caption (cím) – ez jelenik meg az ablak címsorában;

Color (szín) – a form háttérszíne;

Height (magasság) – az ablak magassága pixelekben;

Width (szélesség) – a form szélessége pixelekben;

Top (teteje) – az ablak bal felső sarkának pixelben mért távolsága a képernyő tetejétől;

Left (bal) – az ablak bal felső sarkának pixelben mért távolsága a képernyő jobb szélétől.



Felelj a kérdésekre!

- 1°. Mi a számítógépes program?
- 2°. Mik a program bemenő, köztes és kimenő adatai?
- 3°. Mi a programozási nyelv?
- 4°. Mikből áll egy programozási nyelv? Ismertesd ezeket!
- 5°. Mi a szintaktikai hiba?
- 6°. Mi a **Lazarus** környezet rendeltetése?
- 7°. Ismertesd röviden a **Lazarus** környezet elemeit!
- 8°. Mi a compiler rendeltetése? Mik a kompiláció lépései?
- 9°. Mi a form? Ismertesd néhány tulajdonságát!
- 10°. Hogy nyitunk meg egy **Lazarus**-projektet? Hogy menthetjük azt?



Végezd el a feladatokat!

1°. Hozz létre projektet, ami megnyit egy 400 x 350 pixeles piros háttérű ablakot, amelynek címsora családnevedet tartalmazza! Mentse a projektet a mappában létrehozott **feladat 5.1.1** nevű mappába!



2°. Hozz létre projektet, ami megnyit egy 200 x 450 pixeles kék háttérű ablakot, amelynek címsora nevedet tartalmazza! Mentse a projektet a mappában létrehozott **feladat 5.1.2** nevű mappába!



3• Hozz létre projektet, ami megnyit egy 300 x 300 pixeles kék hátterű ablakot, amelynek bal felső sarka 300 pixelnyire van a képernyő bal szélétől és 400 pixelnyire a tetejétől! Mentsd a projektet a mappádban létrehozott **feladat 5.1.3** nevű mappába!



4• Hozz létre projektet, ami megnyit egy 200 x 100 pixeles szürke hátterű ablakot, amelynek bal felső sarka 150 pixelnyire van a képernyő bal szélétől és 100 pixelnyire a tetejétől. Mentsd a projektet a mappádban létrehozott **feladat 5.1.4** nevű mappába!



5.2. ESEMÉNYEK. ESEMÉNYKEZELŐK



1. Mi a számítógépes program?
2. Mi a form? Milyen tulajdonságai vannak?
3. Mi a compiler rendeltetése és milyen lépéseket végez a munkája során?

ESEMÉNYEK ÉS AZOK KEZELŐI

A formhoz a tulajdonságokon kívül még **események** is tartoznak, amelyek a program futása során bekövetkezhetnek.

Ilyen események például:

- **Click** (kattintás) – bármely objektum kijelölése;
- **Close** (bezárni) – az ablak bezárása;
- **Create** (létrehozás) – a program ablakának megnyitása;
- **KeyPress** (billentyű leütés) – a billentyűzet valamely gombjának lenyomása;
- **MouseMove** (egér mozgása) – az egérmutató mozgatása az ablak területén

és mások.

Minden olyan eseménnyel, ami a program futása során bekövetkezhet, össze lehet kapcsolni a program egy részletét. Az ilyen programrészletet **eseménykezelőnek** nevezük. Valamely objektummal összekapcsolt eseménykezelőt az objektum **metódusának** nevezzük.

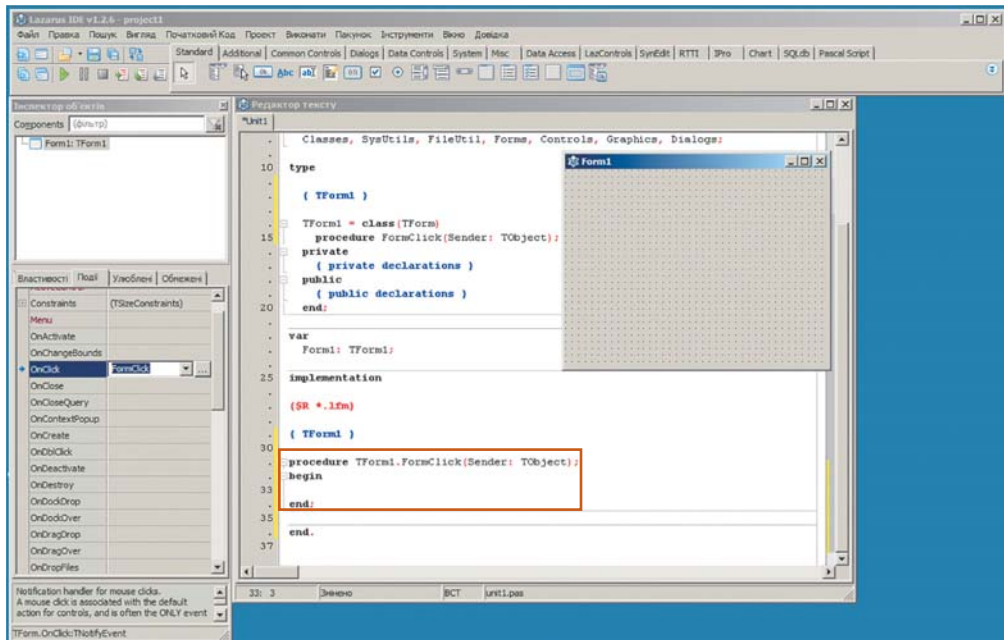
A **Lazarus** környezet az eseménykezelőt alapértelmezetten **eljárásként** hozza létre. Az **eljárás** a program névvel ellátott részlete, amelyet erre a névre hivatkozva indíthatunk el.

Az eljárás végrehajtása szintén esemény. Ha az eljárás annak eredményeképpen kerül végrehajtásra, hogy a formmal bekövetkezett egy esemény – például bekövetkezett a **Click** esemény –, akkor az eljárás futását **OnClick** eseménynek nevezik.

Ha egy eljárást a **Click** eseményhez kívánjuk kötni, a következőket kell tenni:

1. Kattintsunk a **Formra**!
2. Az **Objektum felügyelő** ablakában kattintsunk az **Események** fülre!
3. Jelöljük ki az **OnClick** eseményt! Az **OnClick** melletti mező aktívvá válik és megjelenik benne a kurzor.
4. Kattintsunk duplán erre a mezőre!

Ennek eredményeképpen az **OnClick** esemény mellett megjelenik a **FormClick** (kattintás a formon) felirat, az **Unit1** lapon pedig – ez tartalmazza a projekt szövegét – megjelenik a **TForm1.FormClick** eljárás sablonja, ahová be kell írni azokat az utasításokat, amelyeket a **Click** esemény bekövetkezésekor végre kell hajtani (5.9. ábra).



5.9. ábra. TForm1.FormClick eljárás sablonja

Az eljárás utasításait a **begin** (kezdet) és **end** (vége) szavak közé kell elhelyezni. *Itt jegezzük meg, hogy a **begin** és az **end** nem utasítások, inkább az eljárás elejét és végét jelző utasítás-zárójelek.*

Az eljárás végét jelző **end** kulcsszó után pontosvesszőnek (;) kell következni. Az eljárás valamennyi utasítását pontosvesszőnek (;) kell követni.

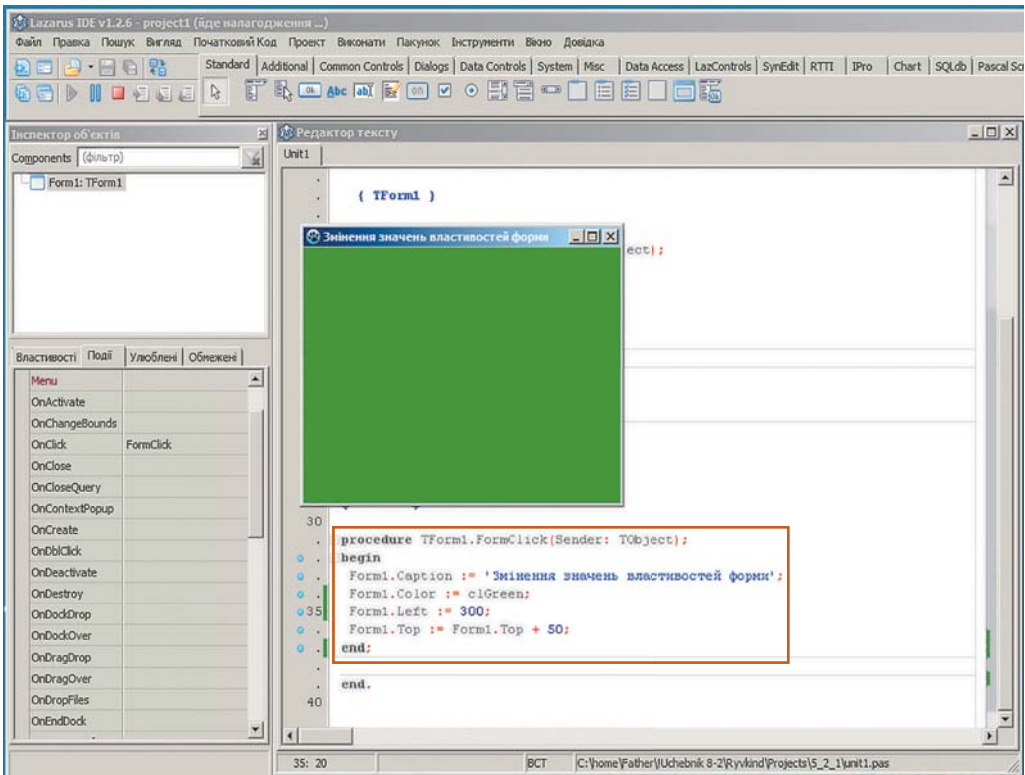
Ha azt szeretnénk, hogy az eljárás eredményeképpen megváltozzon a form valamely tulajdonsága, az eljárásnak egy **értékadó** utasítást kell tartalmazni. Ennek az utasításnak az általános alakja a következő:

<objektum neve>.<tulajdonság neve> := <érték vagy kifejezés>;

vagy:

<objektum neve >.<tulajdonság neve > := <objektum neve>.<tulajdonság neve > + <érték vagy kifejezés>;

A **Form1.Color := clGreen** utasítás például zöldre változtatja a form hátterét. A **Form1.Top := Form1.Top + 50** utasítás 50 pixellel növeli az ablak tetejének és a formnak a távolságát. Ez utóbbi utasítás értelmében a **Top** tulajdonság értéke előbb 50-nel növekszik, majd ez a szám lesz a **Top** tulajdonság új értéke.



5.10. ábra. A form **Caption**, **Color**, **Left** és **Top** tulajdonságainak megváltoztatása és a végrehajtás eredménye

A **Form1.Caption := 'a megváltozott cím'** értékadás eredményeképpen az idézőjelbe vett szöveg bekerül a form címsorába, a **Form1.Left := 300** pedig 300 pixelre állítja a form és a képernyő bal széle közötti távolságot.

Állítsuk össze most a **TForm1.FormClick** eljárást a fenti utasításokból, amelyek végrehajtása során megváltozik a form **Caption**, **Color**, **Left** és **Top** tulajdonságainak értéke (5.10. ábra).

A projekt indítása után meg kell várni az ablak létrejöttét, majd annak tetszőleges pontjára kell kattintani. Ekkor bekövetkezik a **Click** esemény, ennek eredményeképpen bekövetkezik az **OnClick** esemény, ami elindítja a **TForm1.FormClick** eseménykezelőt.

A PROJEKT LÉTREHOZÁSÁNAK ÉS SZERKESZTÉSÉNEK ESZKÖZEI

A **Lazarus** környezet kényelmes eszközökkel támogatja a projekt megírását:

- az objektum nevének első vagy néhány első betűjének beírása után meg lehet nyomni a **Ctrl + Szóköz** gombot, aminek eredményeképpen megnyílik egy lista, amiből kiválaszthatjuk az objektum nevét (5.11. ábra);



```

procedure TForm1.FormClick(Sender: TObject);
begin
  F
  var Form1 : TForm1
  procedure FormClick(Sender:TObject)
  var FActionLists : TList
  e var FFormBorderStyle : TFormBc
  var FFormState : TFormState
  procedure FocusControl (WinControl:TWinControl)

```

5.11. ábra. Az objektumok nevének kiválasztása

```

procedure TForm1.FormClick(Sender: TObject);
begin
  Form1.
end;
end.
property Top:Integer
property Left:Integer
property Color:TColor
procedure FormClick(Sender:TObject)
procedure Cascade
constructor Create(TheOwner:TComponent

```

5.12. ábra. Az objektum tulajdonságainak listája

A rendszer nem különbözteti meg a kis- és nagybetűket. Ha azonban a szó összetett, a könnyebb értelmezhetőség kedvéért a szókezdő betűket nagybetűvel szokás írni.

Már tudjátok, hogy a fordítás kezdetén a compiler szintaktikai ellenőrzést végez. Ha talál ilyeneket, akkor a szövegszerkesztő alatt található **Üzenetek** ablakban megjelenik a hibák listája azoknak a soroknak a számával, ahol a compiler a hibát találta, valamint a hiba valószínű okára utaló kommentárral. Ezenkívül háttérszínnel lesz kiemelve az első hibás sor, vagy az utána következő sor.

```

. procedure TForm1.FormClick(Sender: TObject);
. begin
.   Form1.Caption := 'Змінення значень властивостей форми';
34 Form1.Colour := clGreen;
35 Form1.Left := 300
.   Form1.Top := Form1.Top + 50;
.
.   end;
40
.   end.
42

```

34: 8 INS C:\home\Father\Uchebnik 8-2\Ryvkind\Pr

Messages

- unit1.pas(34,8) Error: identifier idents no member "Colour"
- unit1.pas(34,24) Error: Identifier not found "clGreen"
- unit1.oas(36,2) Fatal: Syntax error. ";" expected but "identifier FORM1" found

5.13. ábra. Szintaktikai hibák jelölése a Lazarusban

• miután az objektum neve után beírtuk a pontot, megnyílik az objektum tulajdonságait tartalmazó lista (5.12. ábra). Ezután már csak ki kell választani a tulajdonságot, és le kell ütni az **Enter**t.

Ha beírjuk a tulajdonság első néhány betűjét, a tulajdonságok listája egyre rövidebb lesz, mert csak azok maradnak meg, amelyek kezdőbetűi megegyeznek a beírt betűkkel. A módszer egyrészt meggyorsítja a program beírását, másrészt csökkenti a hibalehetőségek számát, különösen azok számára, akik nem ismerik az angol nyelvet.

A rendszer sok esetben kiteszi a megfelelő helyre a műveleti zárójeleket, értékadás (műveleti) jeleket, pontosvesszőket stb.

Az 5.13. ábrán a 34-es sor van kiemelve, mivel szintaktikai hibát tartalmaz: hibásan vannak leírva a **Color** és **Green** szavak. Ugyancsak a következő, 35. sor végéről hiányzik a pontosvessző. Ilyen esetben az üzenetek között azt az információt találjuk, hogy helytelen szimbólummal kezdődik a következő, 36. sor.

Ilyenkor ki kell javítani a hibákat, majd újra kell indítani a projekt futtatását.

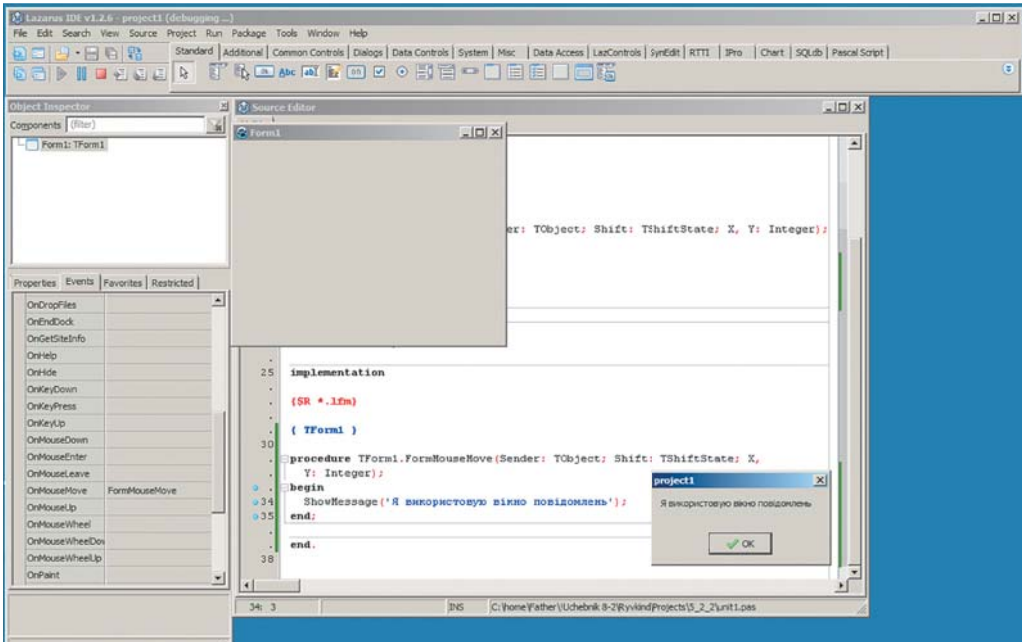


ÜZENETABLAKOK ÉS AZOK ALKALMAZÁSA

Szöveges üzeneteket **üzenetablakok** segítségével helyezhetünk el a képernyőn. Üzenetablakot a **ShowMessage** ('<szöveg>') (*show message* – üzenet megjelenítése) parancs segítségével nyithatunk meg, ahol zárójelek és idézőjelek között kell megadni a megjelenítendő üzenetet.

Az 5.14. ábrán látható esetben a **MouseMove** eseménykezelőjében helyeztünk el egy ilyen utasítást, ugyanezen az ábrán láthatjuk a végrehajtás eredményét is.

Ha elolvastuk az üzenetet, az üzenetablakot be kell zárni az **OK** gombbal. A projekt ablakát csak ezután tudjuk bezárni.



5.14. ábra. Üzenetablak alkalmazása

*Felhívjuk a figyelmeteket arra, hogy a példában a **MouseMove** eseményt használtuk. Ebben az az érdekes, hogy ez esetben a program ablakát nem lehet bezárni az egérrel. Ez azért van, mert amikor az egér megmozdul az ablak területén, azonnal bekövetkezik a **MouseMove** esemény, lefut az **OnMouseMove** eseménykezelő, ami elindítja a **TForm1.FormMouseMove** eljárást, ez pedig megjelenít egy üzenetablakot. A programot csak az **Alt + F4** billentyűkombinációval lehet ilyenkor bezárni. Ez a billentyűkombináció minden projekt esetében alkalmazható.*

Összefoglalás

Minden olyan eseménnyel, ami megtörténhet egy formmal, össze lehet kapcsolni egy programrészletet, ami az esemény bekövetkeztekor fog lefutni. Az ilyen programrészletet

eseménykezelőnek nevezzük. Az objektummal összekapcsolt eseménykezelőt **metódusnak** nevezzük.

A **Lazarus** környezet az eseménykezelőt alapértelmezetten **eljárásként** hozza létre. Az **Eljárás** a program névvel rendelkező része, amelyet erre a névre hivatkozva lehet futtatni.

Az eljárás utasításait a **begin** és **end** kulcsszavak között kell elhelyezni. Ezek a szavak nem utasítások, hanem az eljárás elejét és végét jelölő **utasítás-zárójelek**.

Az eljárás végét jelölő **end** kulcsszó után pontosvesszőt (;) kell tenni. Ugyancsak pontosvesszővel (;) kell lezárni az eljárást alkotó utasításokat.

Ha azt szeretnénk, hogy az eljárás eredményeképpen megváltozzon a form valamely tulajdonsága, az eljárásnak egy **értékadó** utasítást kell tartalmazni. Ennek az utasításnak az általános alakja a következő:

<objektum neve>.<tulajdonság neve> := <érték vagy kifejezés>;

vagy

<objektum neve>.<tulajdonság neve> := <objektum neve>.<tulajdonság neve> + <érték vagy kifejezés>;

Üzenetablakot a **ShowMessage** ('<szöveg>') utasítás segítségével nyithatunk, ahol zárójelek és idézőjelek között kell megadni a megjelenítendő üzenetet.



Felelj a kérdésekre!

- 1°. Milyen események történhetnek egy formmal?
- 2°. Mi a kapcsolat a **Click** és **OnClick** események között?
- 3°. Mi az eseménykezelő?
- 4°. Hogy kapcsoljuk az eseményhez az eseménykezelőt?
- 5°. Mi a metódus?
- 6°. Mi az eljárás?
- 7°. Mi az utasítás-zárójel? Mire szolgálnak?
- 8°. Milyen utasításokkal változtathatjuk meg a form tulajdonságait? Mondj példákat!
- 9°. Milyen utasítással nyithatunk meg egy üzenetablakot?



Végezd el a feladatokat!

- 1°. Hozz létre projektet, amelyben az **OnClick** esemény hatására a form mérete 400×300 lesz, a háttér zöldre vált, a címsorba beíródik a lakóhelyed neve! Mentsd a projektet a **feladat 5.2.1** néven létrehozott mappába!
- 2°. Hozz létre projektet, amelyben az **OnKeyPress** esemény hatására a form mérete 300×200 lesz, a háttér kékre vált, a címsorba beíródik a családneved! Mentsd a projektet a **feladat 5.2.2** néven létrehozott mappába!
- 3°. Hozz létre projektet, amelyben az **OnDbClick** (*double*–dupla) esemény hatására háttér is és a címsor is megváltozik, valamint 200 pixellel megnő a form és a képernyő teteje közötti távolság! Mentsd a projektet a **feladat 5.2.3** néven létrehozott mappába!



4. Hozz létre projektet, amelyben az **OnShow** (*show* – megmutatni) esemény hatására a form mérete valamint a háttér megváltozik és 100 pixellel csökken a form és a képernyő bal széle közötti távolság! Mentsd a projektet a **feladat 5.2.4** néven létrehozott mappába!
5. Hozz létre projektet, amelyben az **OnClick** esemény hatására a form mérete 500×500 lesz, a háttér barnára vált, a címsorba beíródik a „Projekt két eljárással” szöveg. Az **OnDbClick** eseménye váltsa ki egy üzenetablak megjelenítését *Ukrajna állampolgára vagyok* üzenettel! Mentsd a projektet a **feladat 5.2.5** néven létrehozott mappába!
6. Hozz létre projektet, amelyik az **OnClose** esemény hatására megjelenít egy üzenetablakot *A program futása befejeződött. Viszontlátásra!* üzenettel! Mentsd a projektet a **feladat 5.2.6** néven létrehozott mappába!

8. SZÁMÚ GYAKORLATI MUNKA

Üzenetablakot megjelenítő objektumorientált program létrehozása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Indítsd el a **Lazarust**!
2. A form tulajdonságait állítsd be a következőképpen:
 - háttér – *szürke*;
 - szélesség – *200 px*;
 - magasság – *100 px*;
 - távolság a bal oldaltól – *150 px*;
 - távolság a képernyő tetejétől – *100 px*;
 - címsor – *8. számú gyakorlati munka*.
3. Mentsd a projektet a mappádban létrehozott **Gyakorlati 8_1** mappába!
4. Futtasd a projektet!
5. Állítsd meg a projekt futását!
6. Hozz létre a formon a **Click** eseményhez egy eseménykezelőt, ami piros háttérszínt állít be, 200 pixelre állítja a form és képernyő teteje közötti távolságot, 300 pixellel növeli a form szélességét, 50 pixellel csökkenti a form és a képernyő bal széle közötti távolságot, a címsorba beírja az *Object Pascalt tanulunk* szöveget!
7. Mentsd a projektet a mappádban létrehozott **Gyakorlati 8_2** mappába!
8. Futtasd a projektet!
9. Állítsd meg a projekt futását!
10. Adj a projekthez eseménykezelőt a **KeyPress** eseményhez, ami a háttérre kékre állítja, a szélességét 200 pixellel növeli, a képernyő teteje és a form közötti távolságot 300 pixellel növeli!
11. Mentsd a projektet a mappádban létrehozott **Gyakorlati 8_3** mappába!
12. Futtasd a projektet!
13. Állítsd meg a projekt futását!
14. Zárd be a **Lazarus** programozási környezetet!




5.3. A GOMB

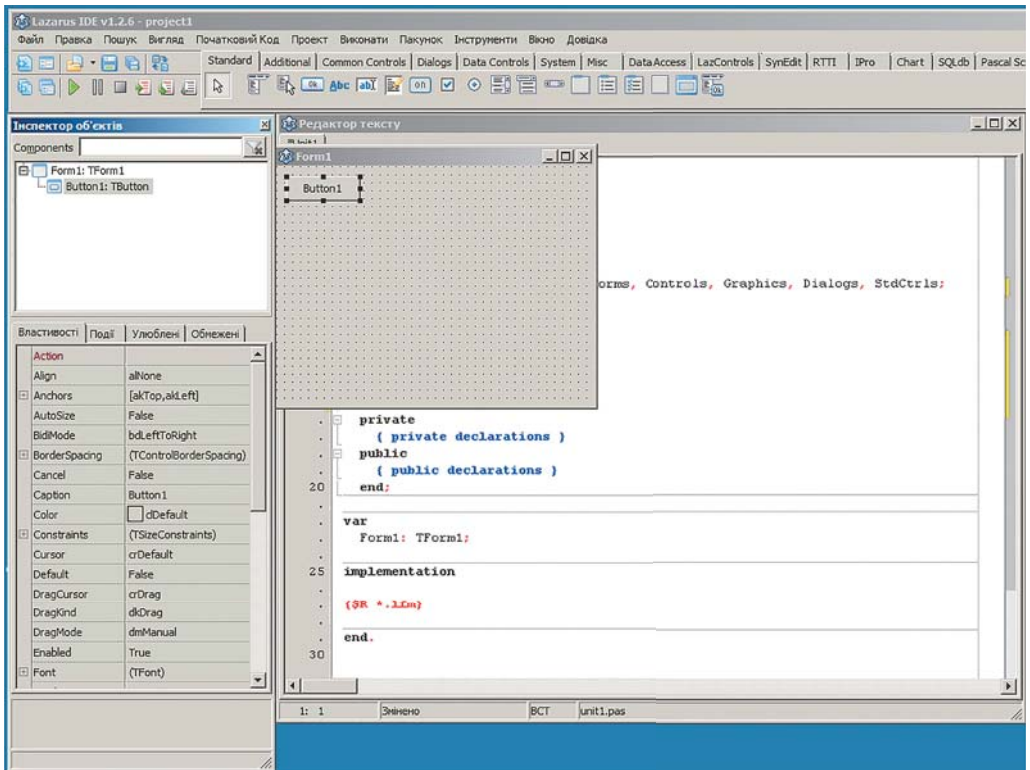


1. A form milyen tulajdonságait ismeritek? Hogy változtatjuk meg ezeket a tulajdonságokat a projekt futtatása előtt?
2. Milyen események következhetnek be a formon? Mit jelentenek ezek?
3. Mi az eseménykezelő? Hogy hozzuk létre? Hogy futtatjuk?

A GOMB, ANNAK TULAJDONSÁGAI ÉS ESEMÉNYKEZELŐI

A **gomb** az egyik legelterjedtebb vezérlőelem, amit a formon elhelyezhetünk.

Ha gombot szeretnénk elhelyezni, kattintsunk duplán az egér bal gombjával a **Standard** eszköztár  gombjára. A gomb az ablak bal felső részében jelenik meg (5.15. ábra), a gomb neve hozzáadódik az objektum felügyelő komponens-listájához, amit a felügyelő ablaknak a felső részében láthatunk. Szükség esetén a gombot áthúzzhatjuk az ablak tetszőleges helyére.





5.15. ábra. Gomb a formon

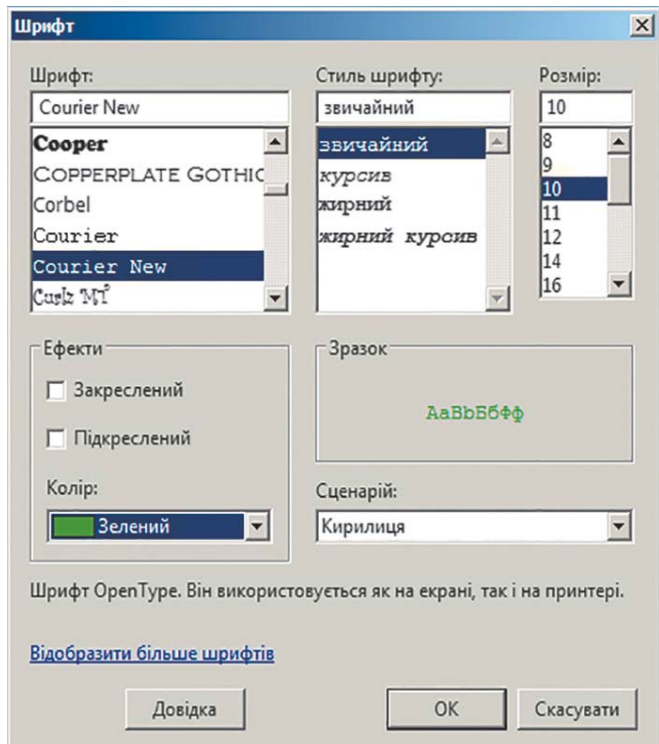


Ha kijelöljük a gombot a formon, vagy az **Objektum-felügyelő** ablakában, ekkor ebben az ablakban a gomb tulajdonságai lesznek aktívak. Ahogy a form esetén megtapasztalhattuk, a gombnak is vannak alapértelmezett tulajdonságai. Természetesen a gomb tulajdonságait is megváltoztathatjuk. Egyes tulajdonságok értékét be kell írni a megfelelő mezőbe, másokat a tulajdonsághoz tartozó listából választhatjuk ki; a gomb helyzetét áthúzással, méretét a méretezőpontok mozgatóásával változtathatjuk meg.

A gomb rendelkezik mindazokkal a tulajdonságokkal, amikkel a formmal kapcsolatban már megismerkedtünk (**Caption, Color, Height, Width, Top, Left, Name**). A gomb esetében a **Color** tulajdonság megváltoztatása nem változtatja meg a gomb alapértelmezett színét. A **Caption** tulajdonság természetesen a gombon látható feliratra vonatkozik, a **Top** és **Left** tulajdonságok viszonyítási pontja pedig nem a képernyő, hanem a form felső, illetve bal széle.

Tekintsünk át még néhány tulajdonságot, amellyel mind a form, mind pedig a gomb rendelkezik:

- **Enabled** (engedélyezett, hozzáférhető) – ennek értéke határozza meg, hogy az objektum tulajdonságain végezhető-e műveletek. Értéke lehet **True** (igaz, vagyis engedélyezett) vagy **False** (hamis, azaz tiltott).
- **Visible** (látható) – értéke azt határozza meg, hogy az objektum látható-e az ablakban. Ennek a tulajdonságnak is csak két értéke lehet, a **True** (látható), illetve a **False** (nem látható).
- **Font** (betűtípus). Ennek értéke határozza meg, milyen betűtípussal jelennek meg a feliratok az objektumon. Az eddigi tulajdonságoktól eltérően ez egy **komplex** tulajdonság. Ez azt jelenti, hogy értéke több részből (**Color, Height, Name, Size**) áll. Ezek mindegyikének értékét a **Fonttól** balra található  gomb lenyomása után megnyíló mezőkben érhetjük el. A tulajdonság értékét a **Fonttól** jobbra található betűtípus  gombra kattintva is



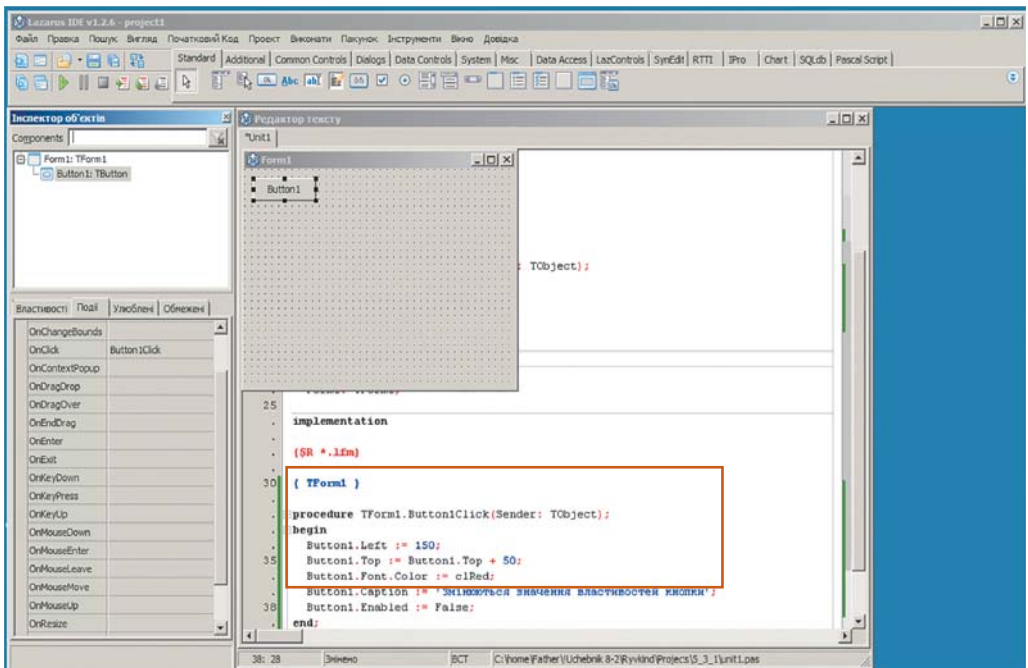
5.16. ábra. A **Betűtípus** ablak

elérhetjük (5.16. ábra). A gomb csak akkor használható, ha a font tulajdonság ki van jelölve.

A tulajdonságok értékét a projekt futása közben is megváltoztathatjuk, a **Button1.Text.Color := clBlue** utasítás hatására a gomb feliratának színe kékre változik.

Ha a gomb ki van jelölve, és az **Objektum felügyelő Események** lapján kiválasztunk egy eseményt, a rendszer a program szövegében automatikusan kigenerálja a megfelelő eseménykezelő sablonját. Ahogy a form esetében is, az eljárásokban elhelyezhetjük a gomb vagy a form tulajdonságait megváltoztató tulajdonságokat, majd futtathatjuk a projektet.

Az 5.17. ábrán látható példa esetében a **Click** esemény kezelője a következő utasításokat tartalmazza:




5.17. ábra. A gomb **Click** eseményének kezelője

- az ablak bal szélétől mért távolság legyen 150 pixel;
- az ablak felső szélétől mért távolság növekedjen 50 pixellel;
- a gomb feliratának színe legyen piros;
- a gombra íródjon ki a megfelelő szöveg;
- a gomb tulajdonságainak értékei váljanak elérhetetlenné.



Összefoglalás

Gombot úgy helyezhetünk el a formon **Lazarus** környezetben, hogy a **Standard-eszköztár**  gombjára duplán kattintunk.

A gombnak a formhoz hasonló tulajdonságai vannak, például a **Caption**, a **Color**, a **Height**, a **Width**, a **Top**, a **Left** és a **Name**. A gomb **Color** tulajdonságának megváltoztatása nem változtat a gomb háttérszínén, a **Caption** tulajdonság a gomb feliratát jelenti, a **Top** és **Left** tulajdonságok viszonyítási pontja pedig nem a képernyő, hanem az ablak.

A gomb és a form rendelkeznek még az alábbi tulajdonságokkal is:

- **Enabled** (engedélyezett, hozzáférhető) – ennek értéke határozza meg, hogy az objektum tulajdonságain végezhető-e műveletek. Értéke lehet **True** (igaz, vagyis engedélyezett), vagy **False** (hamis, azaz tiltott).
- **Visible** (látható) – értéke azt határozza meg, hogy az objektum látható-e az ablakban. Ennek a tulajdonságnak is csak két értéke lehet, a **True** (látható), illetve a **False** (nem látható).
- **Font** (betűtípus). Ennek értéke határozza meg, milyen betűtípussal jelennek meg a feliratok az objektumon. Ez eddigi tulajdonságoktól letérően ez egy **komplex** tulajdonság. Ez azt jelenti, hogy értéke több részből (**Color**, **Height**, **Name**, **Size**) áll.



Felelj a kérdésekre!

- 1°. Hogy helyezünk gombot a formra?
- 2°. Milyen tulajdonságai vannak a gombnak?
- 3°. Mit határoznak meg a gomb tulajdonságai?
- 4°. Mi a gomb esetében a **Color** tulajdonság különlegessége?
- 5°. Miben különbözik a **Caption** tulajdonság a form és a gomb esetében?
- 6°. Miben különböznek a **Top** és **Left** tulajdonságok a gomb és a form esetében?
- 7°. Milyen értékei lehetnek az **Enabled** és **Visible** tulajdonságoknak? Mit jelentenek ezek az értékek?
- 8°. Miért tekintjük a **Font** tulajdonságot komplexnek?



Végezd el a feladatokat!

- 1°. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője állítsa a gomb méretét 40×30 pixelre, a gomb feliratában pedig elhelyezi lakóhelyetek nevét! Mentd a projektet a mappában létrehozott **feladat 5.3.1** mappába!
- 2°. Hozz létre projektet, amelyben a gomb **KeyPress** eseményének kezelője állítsa a gomb méretét 20×40 pixelre, a gomb feliratában pedig elhelyezi a családnevedet! Mentd a projektet a mappában létrehozott **feladat 5.3.2** mappába!
- 3°. Hozz létre projektet, amelyben a gomb **MouseMove** eseményének kezelője növelje meg a gomb és a form bal széle közötti távolságot 20 pixellel és megváltoztatja a gomb feliratát! Mentd a projektet a mappában létrehozott **feladat 5.3.3** mappába!

- 4•. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője csökkentse a gomb és a form bal széle közötti távolságot 20 pixellel, és megváltoztatja a gomb méretét! Mentse a projektet a mappában létrehozott **feladat 5.3.4** mappába!
- 5•. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője állítsa a gomb méretét 50×50 pixelelre, a felirat színét barnára állítsa, a gomb feliratában pedig elhelyezi a *Projekt* szöveget! A **KeyPress** esemény hatására elmozdítja a gombot 30 pixellel jobbra és 40 pixellel le! Mentse a projektet a mappában létrehozott **feladat 5.3.5** mappába!
- 6*. Hozz létre egy projektet, amelynek formjában helyezze el három gombot! A futtatás kezdetén az első gomb legyen látható és elérhető, a második látható és tiltott, a harmadik láthatatlan és tiltott! Mindhárom gomb **Click** eseményéhez rendelj eseménykezelőt a következők szerint:
 - a) az első gomb esetében tegye láthatatlanná és elérhetetlenné a gombot, láthatóvá, de elérhetővé a második gombot, láthatóvá, de elérhetetlenné a harmadikat;
 - b) a második gomb esetében tegye láthatatlanná és elérhetetlenné a gombot, láthatóvá és elérhetővé a harmadik gombot, láthatóvá, de elérhetetlenné az első;
 - c) a harmadik gomb esetében tegye láthatatlanná és elérhetetlenné a gombot, láthatóvá és elérhetővé az első gombot, láthatóvá, de elérhetetlenné a másodikat.Mentse a projektet a mappában létrehozott **feladat 5.3.6** mappába!




5.4. A CÍMKE



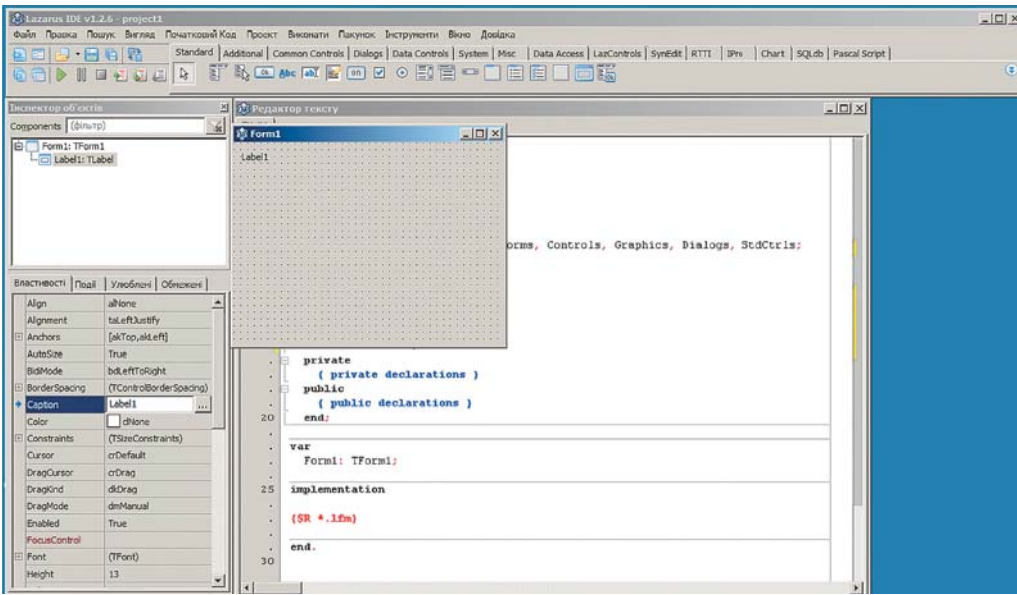
1. A gomb mely tulajdonságait ismeritek? Hogy változtathatjuk meg ezek értékeit?
2. Milyen események következhetnek be egy gombbal? Mit jelentenek ezek?
3. Milyen különbségek vannak a form és a gomb tulajdonságai között?

A CÍMKE ÉS ANNAK TULAJDONSÁGAI. A CÍMKE ESEMÉNYKEZELŐI

A formon elhelyezhető objektumok közül most a **címkével** foglalkozunk. A címkét szöveges üzenetek megjelenítésére használjuk.

A **Címke** (label) elhelyezése a gombhoz hasonlóan történik: kattintsunk duplán a **Standard** eszköztár **TLabel**  gombjára. A címke megjelenik a form bal felső sarkában (5.18. ábra), a neve bekerül az **Objektum felügyelő** ablakába. A címkét áthúzhatjuk a form tetszőleges helyére.

A címke kijelölése is a szokott módon, az ablakban vagy az **Objektum felügyelő**-ben történhet. Az objektum felügyelő ilyenkor a címke tulajdonságait mutatja. A **Tulajdonságok** értékeit itt változtathatjuk meg. Itt is vannak olyan tulajdonságok, amelyeket begépelünk, másokat a listából választjuk. A címke helyzetét áthúzással változtatjuk meg.



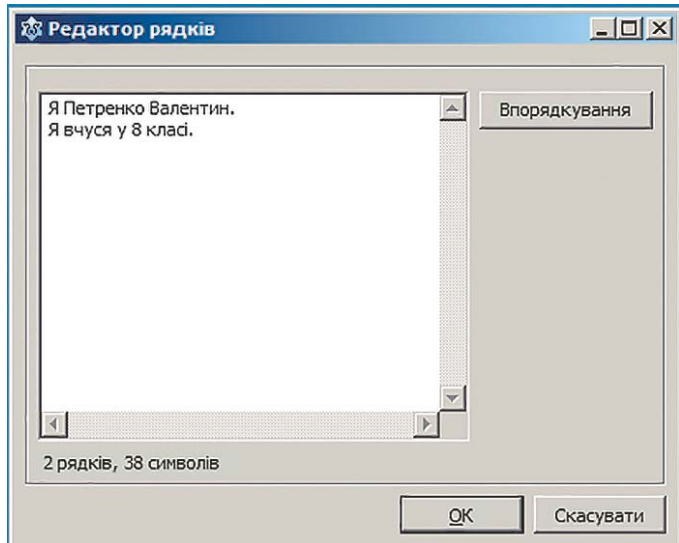
5.18. ábra. Címke a formon

A **Top**, **Left**, **Name**, **Enabled**, **Visible**, **Font** tulajdonságok ugyanazt jelentik, mint a gombok esetében. A **Color** tulajdonság kihatással van a címke megjelenésére. Ha értéke megváltozik, akkor a címke háttérszíne is módosul.

A gombtól eltérően a címke **Height** és **Width** tulajdonságait nem változtathatjuk meg az egér segítségével vagy a **Tulajdonságlapon**. A címke szélességét és hosszát annak a szövegnek a mérete határozza meg, amit a címke tartalmaz. A **Caption** tulajdonság értékének változása tehát megváltoztatja a címke méreteit.

A címke **Caption** tulajdonságát a **Karakterlánc-szerkesztőben** adhatjuk meg (5.19. ábra). Ebben az ablakban többsoros szöveget írhatunk be.

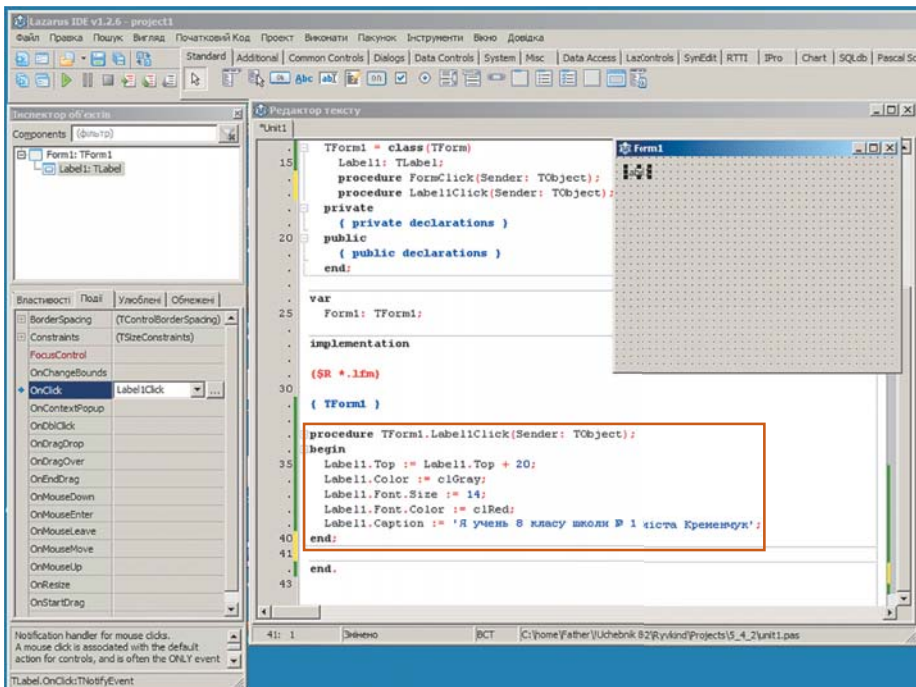
Ha kijelöltük a címkét, akkor az **Objektum felügyelő** ablakának **Események** lapján megjelölhetjük azokat az eseményeket, amelyekhez a rendszer



5.19. ábra. A Karakterlánc-szerkesztő

5. fejezet

elkészíti az eseménykezelők sablonjait, ahogy azt a form és a gomb esetében is tettük (5.20. ábra). Ahogy a form és a gomb esetében, ezekbe az eljárásokba elhelyezhetjük a tulajdonságok értékeit megváltoztató utasításokat és végrehajthatjuk a projektet.

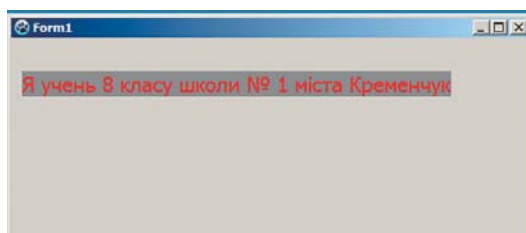


5.20. ábra. A címke **Click** eseménykezelője

Az 5.20. ábrán látható példában a **Click** esemény kezelője a következő utasításokat tartalmazza:

- 20 pixellel növeli a címke és az ablak teteje közötti távolságot;
- szürke háttérszínt állít be;
- 14-es betűméretet állít be a címke szövegéhez;
- piros szövegszínt állít be;
- szöveget ír ki a címkére.

Az 5.21. ábra a végrehajtás eredményét szemlélteti.



5.21. ábra. Az 5.20. ábrán látható eljárás futásának eredménye



Ha többet szeretnél tudni

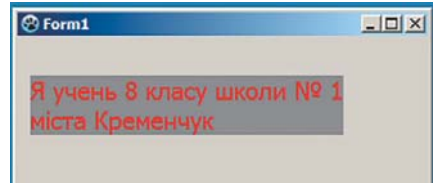
Ha azt szeretnénk, hogy a címke szövege a szükséges helyeken új sorban folytatódjon, #13 új sor szimbólumot kell elhelyezni a szövegben.

Az 5.20. ábrán látható szöveget például futásidőben a következő utasítással írhatjuk ki:

`Label1.Caption :=`

`'Я учень 8 класу школи № 1'`

`+ #13 + 'міста Кременчук'`



5.22. ábra. Az 5.20. ábrán látható eljárás futásának eredménye sortöréssel kiegészítve



Összefoglalás

A **címke** olyan komponens, amelyet szöveg megjelenítésére használunk.

A címkét a **Lazarus** rendszerben a **Standard** eszköztár **TLabel**  gombjára duplán kattintva helyezünk el a formon.

A **Top**, **Left**, **Enabled**, **Visible**, **Font** tulajdonságok ugyanazt jelentik, mint a gombok esetében. A **Color** tulajdonság kihatással van a címke megjelenésére. Ha értéke megváltozik, akkor a címke háttérszíne is megváltozik.

A gombtól eltérően a címke **Height** és **Width** tulajdonságait nem változtathatjuk meg az egér segítségével vagy a **Tulajdonságlapon**. A címke szélességét és hosszát annak a szövegnek a mérete határozza meg, amit a címke tartalmaz. A **Caption** tulajdonság értékének változása tehát megváltoztatja a címke méreteit.

A címke **Caption** tulajdonságát a **Caption** mezőben és **Karakterlánc-szerkesztőben** adhatjuk meg.

A címkét a formon a **Címkére** kattintva, vagy az **Objektum felügyelő** ablakában a címke nevére kattintva jelölhetjük ki.

Ha kijelöltük a címkét, az **Objektum felügyelő** események ablakában választhatjuk ki azt az eseményt, amely kezelőjének sablonját a rendszer a program szövegébe beilleszti.






Felelj a kérdésekre!

- 1°. Mire szolgál a címke?
- 2°. Hogy helyezünk el címkét a formon?
- 3°. Mit jelentenek a **Top**, **Left** és **Name** tulajdonságok a címke esetében?
- 4°. Mit jelentenek az **Enabled** és **Visible** tulajdonságok címke esetében?
- 5°. Mit jelentenek a **Font** komplex tulajdonság elemei a címke esetében?
- 6°. Miben különbözik a **Color** tulajdonság gomb és címke esetében?
- 7°. Miben különböznek a **Top** és **Left** tulajdonságok címke és form esetében?
- 8°. Miben különbözik a **Height** és **Width** tulajdonságok címke és form esetében?
- 9°. Hogy változtathatjuk meg a címke feliratát?



Végezd el a feladatokat!

- 1°. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke és a form teteje közötti távolság 120 pixel lesz, a címke szövegébe beírja annak az utcának a nevét, amelyen az iskolátok található! Mentsd a projektet a mappádban létrehozott **feladat 5.4.1.** mappába.
-  2°. Hozz létre projektet, amelyben a címke **KeyPress** eseményének eredményeképp a címke és a form bal oldala közötti távolság 80 pixel lesz, a címke szövegébe beírja a neved! Mentsd a projektet a mappádban létrehozott **feladat 5.4.2.** mappába!
- 3°. Hozz létre projektet, amelyben a címke **MouseMove** eseményének eredményeképp a címke szövegének színe piros lesz, és 20 pixellel nő a címke és a form teteje közötti távolság! Mentsd a projektet a mappádban létrehozott **feladat 5.4.3.** mappába!
-  4°. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke és a form bal széle közötti távolság 20 pixellel nő, a betűméret 12-es lesz, a betűszín kék lesz, a címkébe pedig beíródik az iskolád neve! Mentsd a projektet a mappádban létrehozott **feladat 5.4.4.** mappába!
- 5°. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke háttere barnára vált, a címkébe beíródik a *Projektem* szöveg, a **KeyPress** hatására elmozdul 30 pixellel jobbra és 40 pixellel le! Mentsd a projektet a mappádban létrehozott **feladat 5.4.5.** mappába!
- 6°. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke 60 pixellel jobbra és 10 pixellel felfelé mozdul el, a címkébe pedig beíródik a teljes neved! Mentsd a projektet a mappádban létrehozott **feladat 5.4.6.** mappába!
-  7°. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke 60 pixellel jobbra és 80 pixellel felfelé mozdul el, a címkébe szürke háttéren zöld betűkkel beíródik a teljes neved! Mentsd a projektet a mappádban létrehozott **feladat 5.4.7.** mappába!

9. SZÁMÚ GYAKORLATI MUNKA

Gombokat és címkéket tartalmazó projekt létrehozása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Indítsd el a **Lazarus** rendszert!
2. Helyezz el a formon két gombot és egy címkét!
3. Az első gomb tulajdonságai legyenek a következők:
 - szélesség – 60 px;
 - magasság – 20 px;
 - távolság a form bal szélétől – 120 px;
 - távolság a form tetejétől – 100 px;
 - felirat – *Form*.



4. A második gomb tulajdonságai legyenek a következők:
 - szélesség – 100 px;
 - magasság – 30 px;
 - távolság a form bal szélétől – 300 px;
 - távolság a form tetejétől – 100 px;
 - felirat – *Címke*.
5. Az címke tulajdonságai legyenek a következők:
 - szélesség – 120 px;
 - magasság – 40 px;
 - távolság a form bal szélétől – 150 px;
 - távolság a form tetejétől – 200 px;
 - felirat – az osztályod neve.
6. Mentsd a projektet a mappában létrehozott **gyakorlati 9_1** mappába!
7. Futtasd a projektet!
8. Állítsd le a projekt futását!
9. Hozd létre az első gomb **Click** eseményének kezelőjét, ami a form szélességét 800, magasságát 400 pixelre állítja be, a form háttérét zöldre, a gomb szélességét 200-ra állítja, magasságát 10 pixellel növeli, 50 pixellel balra, 30 pixellel felfelé mozditja, majd tulajdonságait zárolja!
10. Hozd létre a második gomb **MouseMove** eseményének kezelőjét, ami a címke háttérét pirosra állítja, távolságát a form tetejétől 200 pixelre állítja, csökkenti a címke és a form bal széle közötti távolságot 50 pixellel, a szöveg színét pirosra változtatja és megjeleníti az *Object Pascalt tanulunk* szöveget! (*Megjegyzés: piros háttérre piros színnel nem szerencsés írni, változtassuk meg inkább fehérre!*)
11. Mentsd el a projekt új változatát!
12. Futtasd a projektet!
13. Állítsd le a projekt futását!
14. Zárd be a **Lazarust**!



6. fejezet

Objektumokkal és mennyiségekkel dolgozó algoritmusok

Ebben a fejezetben megismerkedtek:

az állandókkal,
a változókkal és ezek
tulajdonságaival

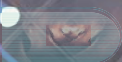
a projekt tesztelésével

a **mező, jelölőnégyzet**
listadoboz objektumokkal, és
ezek tulajdonságaival,
alkalmazásával

az elágazások, elől-
és hátultesztelő ciklusok
alkalmazásával

a változókkal és ezek
tulajdonságaival

a grafikus objektumok
futásidejű létrehozásával






6.1. A BEVITELI MEZŐ. MENNYISÉGEK. ÁLLANDÓ ÉS VÁLTOZÓ MENNYISÉGEK. A VÁLTOZÓK TULAJDONSÁGAI



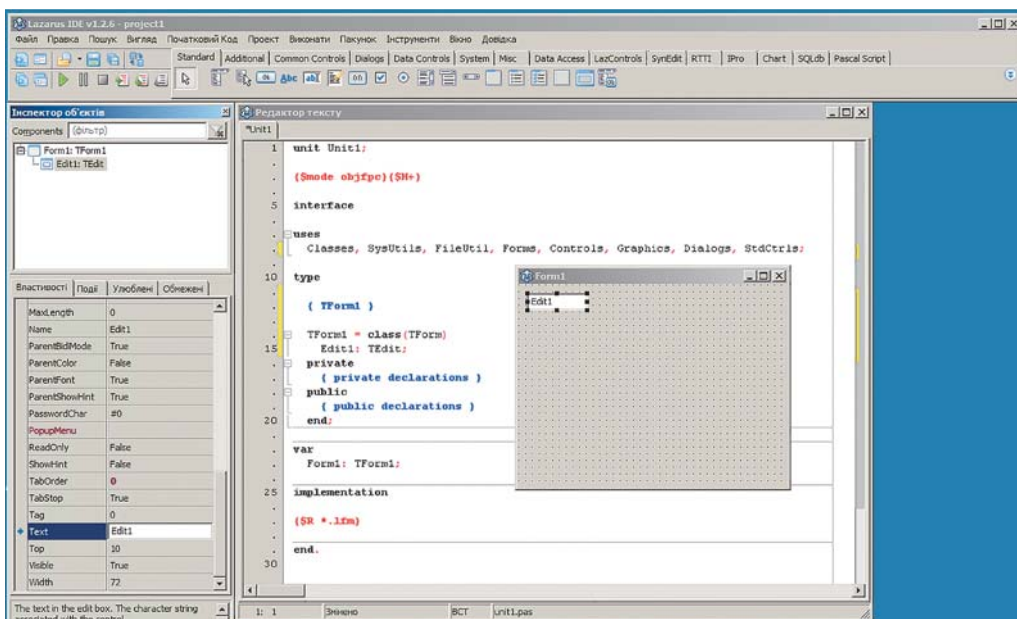
1. Mire szolgál az értékadó utasítás? Mi ennek az utasításnak az általános alakja?
2. Milyen tulajdonságai vannak a **címkének**? Mit jelentenek ezek a tulajdonságok?
3. Milyen tulajdonságai vannak a **gombnak**? Mit jelentenek ezek a tulajdonságok?

A BEVITELI MEZŐ, ANNAK TULAJDONSÁGAI ÉS ESEMÉNYKEZELŐI

A következő formra helyezhető komponens, amit megvizsgálunk, a **beviteli mező** lesz. Amint a neve is jelzi, adatok bevitelére szolgál.

A mezőt **Lazarus** környezetben úgy helyezzük el a formra, hogy duplán kattintunk a **Standard** eszköztár **TEdit**  elemére. Ekkor a form bal felső sarkában megjelenik a beviteli mező, az **Objektum felügyelő Komponensek** ablakához hozzáadódik a mező (6.1. ábra). A mezőt is szabadon áthelyezhetjük a formon belül.

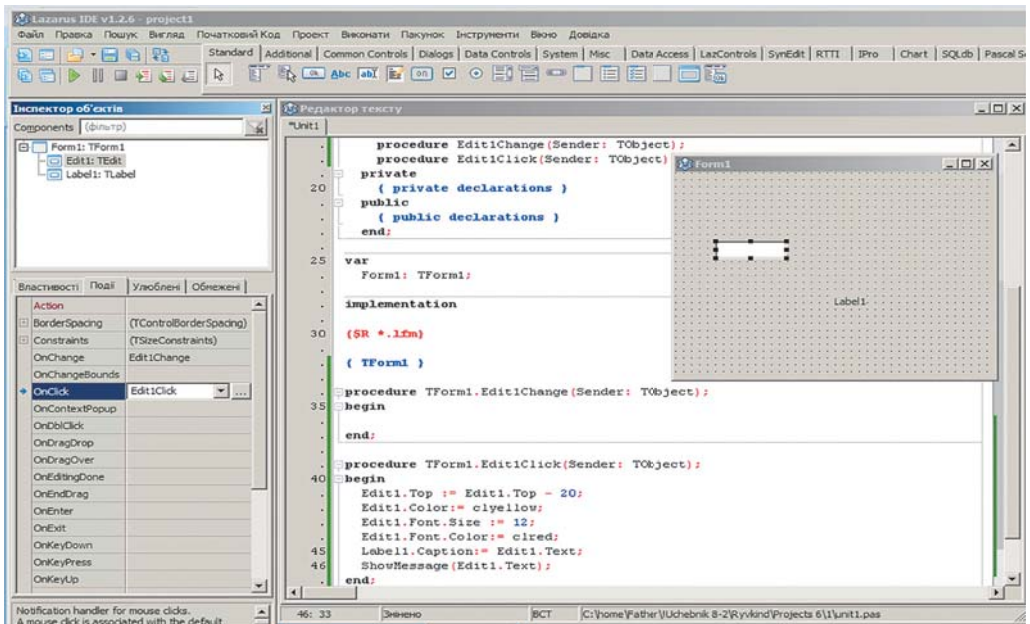
A mezőt is a szokott módon, a formon vagy az **Objektum felügyelő** ablakában az egér bal gombjával jelöljük ki. Ekkor a felügyelő ablakában a **Tulajdonság** ablak a mező tulajdonságait tartalmazza. Ezeket a tulajdonságokat is szerkeszthetjük. Egyes tulajdonságok értékeit a beviteli mezőkbe kell begépelni, mások esetében a legördülő listákból választhatunk; amikor a mezőt mozgatjuk a formon, szintén megváltoznak a megfelelő tulajdonságai.



6.1. ábra. Beviteli mező a formon

A **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** tulajdonságok értelmezése a mező és a címke esetében ugyanaz. A **Height** és **Width** tulajdonságokat a **Tulajdonság** lapon és a méretezőmarkerek mozgatásával állíthatjuk be, ahogy az a gomb esetében is volt. A beviteli mezőben olvasható szöveg megegyezik a **Text** tulajdonság értékével. Ennek a tulajdonságnak az értékét a **Tulajdonság** lap megfelelő mezőjébe írhatjuk be. A projekt futtatása előtt célszerű ezt a mezőt üresen hagyni.

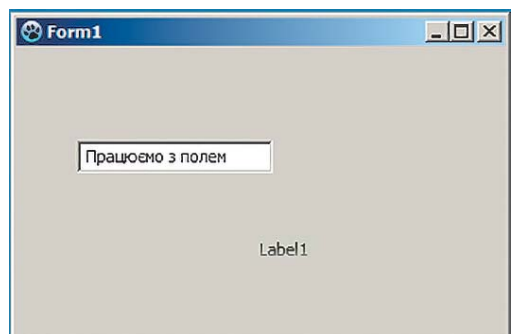
Ha a mezőt kijelöljük, az **Objektum felügyelő Események** lapján – az eddigi komponensekhez hasonlóan – kiválaszthatjuk azokat az eseményeket, amelyekhez a rendszer ki-generálja a megfelelő eseménykezelő sablonját. A 6.2. ábrán látható példa esetében a **Click** esemény kezelője a következő utasításokat tartalmazza:



6.2. ábra. A mező **Click** eseménykezelője

- a mező és az ablak felső széle közötti távolság csökkenjen 20 pixellel;
- a mező háttere legyen sárga;
- a mezőbe írt szöveg betűmérete legyen 12;
- a mező betűszíne legyen piros;
- szöveges üzenet elhelyezése a mezőbe.

A projekt indítása után a beviteli mező tetszőleges szöveg befogadására alkalmas (a 6.3. ábrán a mezőbe a *Mezőt szerkeszték* üzenet lett beírva), ehhez az egér bal gomb-

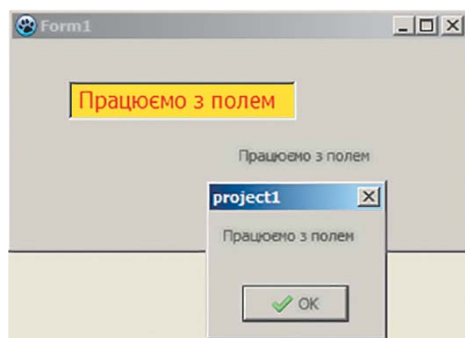


6.3. ábra. A projekt ablaka a **Click** eseménykezelő futása előtt



jával a mezőn belülre kell kattintani. A Click eseménykezelő futásának eredményét a 6.4. ábrán láthatjuk.

Bár a mező rendeltetése az adatok bevitele, de természetesen adatok kivételére is használhatjuk. Futási időben e célból meg kell változtatni a mező **Text** tulajdonságának értékét, például így: **Edit1.Text := 'Mi nyolcadikosok vagyunk'**. Amennyiben a mező szélesség túl kicsi a beírt szövegnek, a címkétől eltérően a mező szélessége nem változik meg, a szöveg egy része „kilóg” a mezőből, ezért nem látható.



6.4. ábra. A projekt ablaka a Click eseménykezelő lefutása után

MENNYISÉGEK. ÁLLANDÓK ÉS VÁLTOZÓK

Az informatika, fizika, matematika és sok egyéb tudomány mennyiségekkel dolgozik. A **mennyiség** az objektum valamely tulajdonságának számmal kifejezett értéke.

Az informatikában tehát mennyiség például az ablak szélesség és hosszúsága, a háttérszín, a gomb felirata, elhelyezkedése az ablakban. A matematikában mennyiség a téglalap szélessége, hossza, és területe, a szög mértéke stb. A fizikában mennyiség a megtett út, az idő, a sebesség, a sűrűség.

Azt a mennyiséget, amelynek értéke nem változik, **állandónak** nevezzük.

A matematikában állandó a π szám értéke. Tudjátok, hogy közelítő értéke 3,1416. A fizikában állandó valamely anyag sűrűsége, a levegőé például $1,293 \text{ kg/m}^3$.

Azt a mennyiséget, amelynek értéke megváltozhat, **változó mennyiségnek** vagy röviden **változónak** nevezzük.

MEZŐ HASZNÁLATA SZÁMOK BEVITELÉRE

Hozzunk létre egy projektet, amely összead két tetszőleges egész számot.

Helyezzünk el a formon két beviteli mezőt az összeadandóknak, egy gombot, amelynek lenyomása után megtörténik a mezőkbe beírt számok összeadása és kiírása az erre a célra létrehozott címkébe!

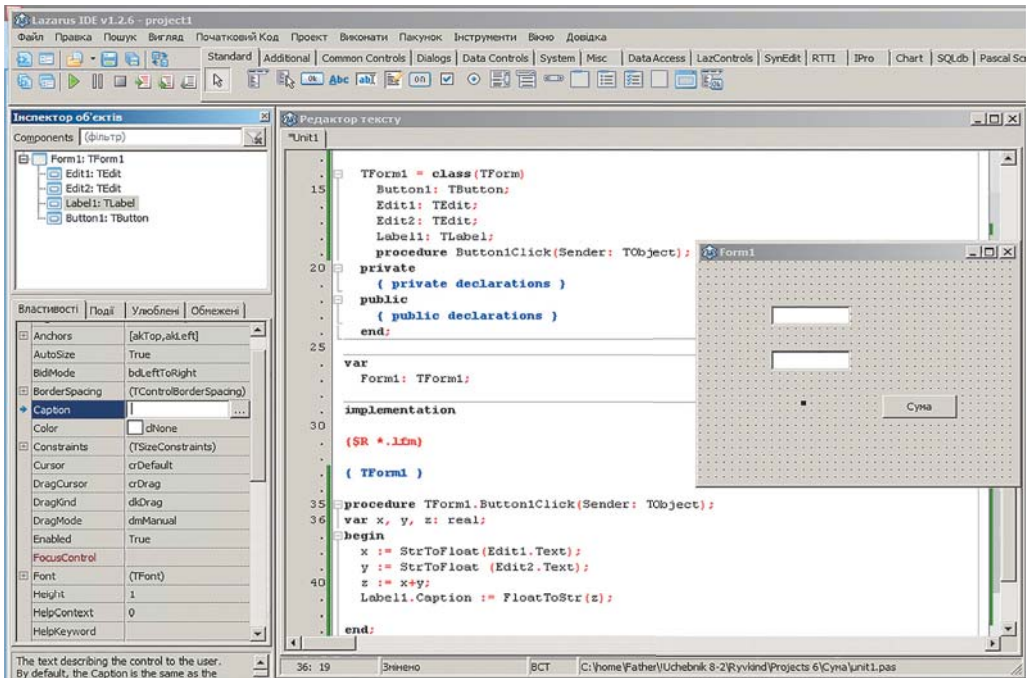
A 6.5. ábrán látható form ezeket az objektumokat tartalmazza, továbbá a két szám összeadására szolgáló eljárást.

Vizsgáljuk meg ezt az eljárást!

Bár a felhasználó a projekt futása során számokat ír a beviteli mezőkbe, a rendszer ezeket szöveggé értelmezi. A szövegek az **Edit1** és **Edit2** mezők **Text** tulajdonságába íródnak be. Ezért az eljárás első két utasítása, az **x := StrToFloat (Edit1.Text)** és az **y := StrToFloat (Edit2.Text)** (az angol *string to float* – szöveg tizedes törtté) átalakítja ezeket a szövegeket tizedes törttékké, és elhelyezi ezeket az **x** és **y** változóban.

A harmadik utasítás összeadja ezt a két változót, és az eredményt elhelyezi **z**-ben.

A negyedik utasítás – **Label1.Caption := FloatToStr (z)** – a kiszámított összeget számfórmátumból szöveggé alakítja, és elhelyezi ezt a szöveget a **Label1** címke **Caption** mezőjében (*float to string* – tizedes tört szöveggé).



6.5. ábra. Két mezőt, egy gombot és egy címkét tartalmazó form és a két szám összeadására szolgáló eljárás

Az **Object Pascal** programozási nyelv megköveteli, hogy minden változónak ismert **típusa** legyen.

A **változó típusa** meghatározza, hogy:

- milyen értékeket vehet fel a változó;
- milyen műveleteket lehet a változóval végezni;
- mekkora operatív tár szükséges a változó tárolására.

A változók nevét és típusát a **var** (*variable* – változó) kulcsszó után kell megadni.

Az eljárás során használt *x*, *y* és *z* változók a feltétel szerint tetszőleges valós értéket vehetnek fel. Az ilyen változókhoz a **real** (valós) típus a megfelelő választás. Az ilyen változók egész és tört értéket is felvehetnek, abszolút értékük maximálisan $1,7 \cdot 10^{38}$ lehet.

Minden ilyen szám 8 bájtot foglal el. Az eljárásban használt változókat és azok típusát a **var** sorba írjuk le, ami közvetlenül az eljárás fejléce után következik (6.5. ábra).

Ha egy változó csak egész értékeket vehet fel, akkor azt **integer** (egész) típusúnak kell deklarálni. Az ilyen változók megengedett értékei – **32 768** és **32 767** között vannak. A **longint** (*long integer* – hosszú egész) típusú változók 4 bájtot foglalnak el, ezért értékeik – **2 147 483 648** és **2 147 483 647** közöttiek lehetnek.

Abban a **var** sorban (6.5. ábra), amely az eljárásokon kívül helyezkedik el, olyan változókat kell deklarálni, amelyet bármelyik eljárás használhat.



Ha a **var** kulcsszó után különböző típusú változókat deklarálunk, akkor azt a következő alakban kell felírni: **var x, y: real; t: integer;**

Amikor a most létrehozott projektet lefuttatjuk, megnyílik a projekt ablaka. A felhasználó beírja a számokat a két beviteli mezőbe (a 6.6. ábrán ezek 2,7, illetve -3,43), majd megnyomja az *Összeg* gombot és a címkében megjelenik az összeg (-0,73). A felhasználó korlátlanul ismételheti a beviteli mezők értékeinek beírását és azok összeadását egészen az ablak bezárásáig.

Hogy a program futása során a felhasználóknak könnyen érthető legyen, mi a teendő, célszerű a mezők mellé címkéket elhelyezni, amelyeket magyarázó szövegekkel látjuk el, például *első összeadandó*, *második összeadandó*, *összeg* (6.7. ábra).

6.6. ábra. Két valós szám összeadásának projektje

6.7. ábra. A futó projekt ablaka

A SZÁMOKKAL VÉGZETT MŰVELETEK EREDMÉNYEINEK TÍPUSA

Már tudjátok, hogy a változó mennyiségekkel matematikai műveleteket (összeadás, kivonás, szorzás, osztás) lehet végezni.

Ha az $x := a + b$, $x := a - b$, $x := a * b$, $x := a / b$ műveletekben az a és a b **real** típusú, akkor az x **realként** kell legyen deklarálva.

Ha az $x := a + b$, $x := a - b$, $x := a * b$ műveletekben az a és a b **integer (longint)** típusú, akkor az x lehet **integer (longint)**, de lehet **real** is.

Ha az $x := a / b$ műveletben az a és b **integer (longint)**, az x **real** kell legyen.

ÁLLANDÓK ALKALMAZÁSA A PROJEKTBEN

A projektekben a változókon kívül állandókat is használhatunk.

Hozzunk létre projektet, amelyben meg kell határozni egy fenyőörök tömegét a térfogata alapján.

A fizikából tudjuk, hogy a fenyő sűrűsége 520 kg/m^3 , a tömeget pedig az $m = \rho V$ képlettel számítjuk ki.

Feltételezzük, hogy a térfogatot az **Edit1** mezőben köbméterekben adjuk meg, a tömeget pedig a **Label1** címkébe kilogrammokban fogjuk kiírni.

A projekt ennek megfelelően a következőképpen néz ki:

```
const p = 520;
var x, m: real;
begin
  x := StrToFloat(Edit1.Text);
  m := x*p;
  Label1.Caption := FloatToStr(m);
end;
```



Összefoglalás

A **mező** az a komponens, amelyet adatok bevitelére használunk.

A **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** tulajdonságok értelmezése a mező és a címke esetében ugyanaz. A **Height** és **Width** tulajdonságokat a **Tulajdonság** lapon és a méretezőmarkerek mozgatásával állíthatjuk be, ahogy az a gomb esetében is volt. A beviteli mezőben olvasható szöveg megegyezik a **Text** tulajdonság értékével. Ennek a tulajdonságnak az értékét a **Tulajdonság** lap megfelelő mezőjébe írhatjuk be.

A mezőt adatok bevitelére és kivitelére használjuk. E célból a mező **Text** tulajdonságába kell az adatokat beírni.

A **mennyiség** az objektum valamely tulajdonságának számmal kifejezett értéke.

Az **állandó** az a mennyiség, amelynek értéke nem változik.

A **változó** az a mennyiség, amelynek értéke megváltozhat.

Az **Object Pascal**ban minden változónak meghatározott **típusa** van, amit deklarálni kell a programban. A **változó típusa** meghatározza, hogy:

- milyen értékeket vehet fel a változó;
- milyen műveleteket lehet a változóval végezni;
- mekkora operatív tár szükséges a változó tárolására.





Az **Object Pascal** programozási nyelvben a változókon kívül állandókat is használhatunk.



Felelj a kérdésekre!

- 1°. Mire használjuk a beviteli mezőt?
- 2°. Hogy helyezünk el mezőt a formon?
- 3°. Mit jelentenek a mező esetében a **Top**, **Left**, **Name**, **Color**, **Text**, **Enabled**, **Visible** tulajdonságok?
- 4°. Mit határoznak meg a **Font** komplex tulajdonság értékei a mező esetében?
- 5°. Miben különböznek a form és a mező **Top** és **Left** tulajdonságai?
- 6°. Miben különböznek a **Height** és **Width** tulajdonságok a mező és a címke esetében?
- 7°. Hogy változtatjuk meg a szöveget a mezőben?
- 8°. Milyen utasításokkal alakítjuk át a szöveget számmá és vissza?
- 9°. Mit határoz meg a mező típusa?
- 10°. Hol, és hogy adjuk meg a változó típusát?

**Végezd el a feladatokat!**

- 1°. Hozz létre projektet, amelyben a mező **Click** eseményének hatására 100 pixellel növeli a mező és a form felső széle közötti távolságot, a mezőbe kiírja az *Informatika* szót! Mentd a projektet a mappádban létrehozott **feladat 6.1.1** mappába!
-  2°. Hozz létre projektet, amelyben a mező **KeyPress** eseményének hatására 120 pixellel növeli a mező és a form bal széle közötti távolságot, a mezőbe kiírja a nevedet! Mentd a projektet a mappádban létrehozott **feladat 6.1.2** mappába!
- 3°. Hozz létre projektet, amelyben a mező **MouseMove** eseményének hatására megváltoztatja a mező tartalmát, piros betűszínt és szürke hátteret állít be, és a mezőt 100 pixellel jobbra és 80 pixellel lefelé mozdítja! Mentd a projektet a mappádban létrehozott **feladat 6.1.3** mappába!
-  4°. Hozz létre projektet, amelyben a mező **Click** eseményének hatására elmozdítja a mezőt 80 pixellel jobbra és 20 pixellel felfelé, kiírja a nevedet sárga háttérre zöld színnel! Mentd a projektet a mappádban létrehozott **feladat 6.1.4** mappába!
- 5°. Hozz létre projektet a $(2a + b)(a + c)$ kifejezés értékének meghatározására! Mentd a projektet a mappádban létrehozott **feladat 6.1.5** mappába!
-  6°. Hozz létre projektet két valós szám különbségének meghatározására! Mentd a projektet a mappádban létrehozott **feladat 6.1.6** mappába!
-  7°. Hozz létre projektet a $(3a - 2b)c$ kifejezés értékének meghatározására! Mentd a projektet a mappádban létrehozott **feladat 6.1.7** mappába!
- 8°. Hozz létre projektet a körvonal hosszának meghatározására! A projektben használj állandót! Mentd a projektet a mappádban létrehozott **feladat 6.1.8** mappába!

10. SZÁMÚ GYAKORLATI MUNKA

Szekvenciális algoritmus összeállítása mennyiségek feldolgozására

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Nyisd meg a **Lazarust**!
2. Helyezz el a form-on egy gombot és egy beviteli mezőt!
3. A gomb tulajdonságai a következők legyenek:
 - szélesség – 70 px;
 - magasság – 30 px;
 - távolság a form bal szélétől – 150 px;
 - távolság a form felső szélétől – 200 px;
 - felirat a gombon – *Mezőtulajdonságok*.
4. A mező tulajdonságai a következők legyenek:
 - szélesség – 200 px;
 - magasság – 50 px;
 - távolság a form bal szélétől – 70 px;



6. fejezet

- távolság a form felső szélétől – 100 px;
 - felirat a mezőn – az osztályod neve.
5. Mentsd a projektet a mappádban **gyakorlati 10_1** néven létrehozott mappába!
 6. Futtasd a projektet!
 7. Állítsd le a projekt futását!
 8. Hozd létre a gomb **Click** eseménykezelőjét, ami a mezőnek piros háttérszínt állít be, a mező távolságát a form felső szélétől 150 pixelre állítja, csökkenti 50 pixellel a form bal széle és a mező távolságát, piros betűszínnel kiírja a mezőbe *A mezők tulajdonságaival ismerkedünk* szöveget!
 9. Mentsd a projekt új változatát!
 10. Futtasd a projektet!
 11. Állítsd le a projekt futását!
 12. Hozz létre projektet az $a(4b - 3c)$ kifejezés értékének kiszámítására!
 13. Mentsd a projektet a mappádban **gyakorlati 10_2** néven létrehozott mappába!
 14. Futtasd a projektet!
 15. Állítsd le a projekt futását!
 16. Zárd be a **Lazarust!**



6.2. PROJEKT TESZTELÉSE ÉS JAVÍTÁSA



1. Mi a compiler rendeltetése? Mik a működésének sajátosságai?
2. Mi a szintaktikai hiba? Hogy jelennek ezek meg a kompiláció során?
3. Hogy nyitunk meg korábban mentett projektet?

PROGRAMOK TESZTELÉSE

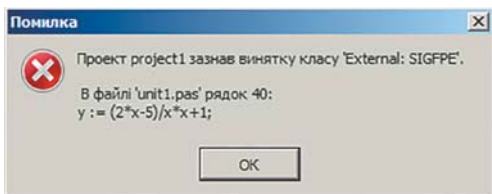
Már tudjátok, hogy a projekt létrehozása során szintaktikai hibák keletkezhetnek. A projekt szerzője által nem észlelt hibákat a compiler találja meg a program fordítása során. Ezeknek a hibáknak a javítása után kerülhet sor a projekt újbóli futtatására.

Ha a program futása folyton megszakad, vagy a futtatás nem a helyes eredményre vezet, akkor azt mondják, hogy a program **logikai hibákat** tartalmaz.

A logikai hibák kiszűrésének egyik módja, hogy a projektet **tesztadatokkal** vizsgáljuk. Minden ilyen tesztadatsorhoz előre meghatározzuk az elvárt eredményt. Amennyiben a tényleges eredmény eltér az elvárttól, meg kell határozni, milyen utasítások okozzák ezt az eltérést.

A logikai hibák tesztadatokkal történő felderítésének folyamatát nevezzük **tesztelésnek**.

Vizsgáljuk meg ezt a következő példán. Tegyük fel, hogy az $y = \frac{2x - 5}{x^2 + 1}$ függvény értékének meghatározására készítünk projektet. A függvény értékének kiszámítására szolgáló eljárás, amelyben az x értékét egy mező segítségével adjuk meg, a következő lesz:



6.8. ábra. A hibajelzés ablaka

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/x*x+1;
  Label1.Caption := FloatToStr(y);
end;
```

6.9. ábra. Az eljárás szövegében háttérszínnel kiemelt hibás sor

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/(x*x+1);
  Label1.Caption := FloatToStr(y);
end;
```

Viszonylag gyakori hiba, hogy a tanulók a kifejezés értékének meghatározására szolgáló $y := (2 \cdot x - 5) / (x \cdot x + 1)$ utasításban elfelejtik zárójelbe venni a nevezőt (időnként a számlálót is), ebből az utasításból tehát $y := (2 \cdot x - 5) / x \cdot x + 1$ vagy $y := 2 \cdot x - 5 / x \cdot x + 1$ lesz. Az ilyen hibát nem nehéz kiszűrni a megfelelő teszt segítségével. Ha a teszttérték $x = 0$, akkor a projekt az elvárt -5 végeredmény helyett működése során hibajelzéssel leáll (6.8. ábra).

Ebben az ablakban a hibajelzésen kívül a hibás sor számát és a tartalmát is megtaláljuk. Az **OK** lenyomása után az ablak bezáródik, a program szövegében a hibás sor háttérszínnel lesz kiemelve (6.9. ábra).

A projektet ilyen esetben a **Ctrl + F2** billentyűkombinációval, vagy a Leállítás gombot lenyomva állíthatjuk le. Ezután meg kell érteni, mi a hiba, majd annak kijavítása után újból futtathatjuk a projektet. A vizsgált esetben a hiba azért következett be, mert x 0 értéke mellett nullával való osztás következik be, aminek elvégzése nem lehetséges.

Ha másik teszttértéket választunk, például -1 -et, akkor a projekt futása nem áll le, de az eredmény a helyes $-3,5$ helyett -6 lesz. Ebben az esetben is át kell vizsgálni az eljárást, meg kell találni a hibát és ki kell javítani azt.



Ha többet szeretnél tudni

A hibakeresést jelentősen megkönnyíthetjük, ha a program szövegében **kommenteket** helyezünk el. A **komment** olyan szöveg, amelyet a compiler nem hajt végre és a program működésének magyarázata céljából helyezük el.

Hogy a compiler megkülönböztesse a kommentet az utasításoktól, figurális zárójelekbe kell venni azokat, például {**y értékének meghatározása**} (6.10. ábra). Néha a kommentet zárójelek és csillagok közé tesszük: (***y értékének kiírása***).

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text); //Команда введення значення x
  {Команда обчислення значення y}
  y := (2*x-5)/x*x+1;
  {Команда виведення значення y*}
  Label1.Caption := FloatToStr(y);
end;
```

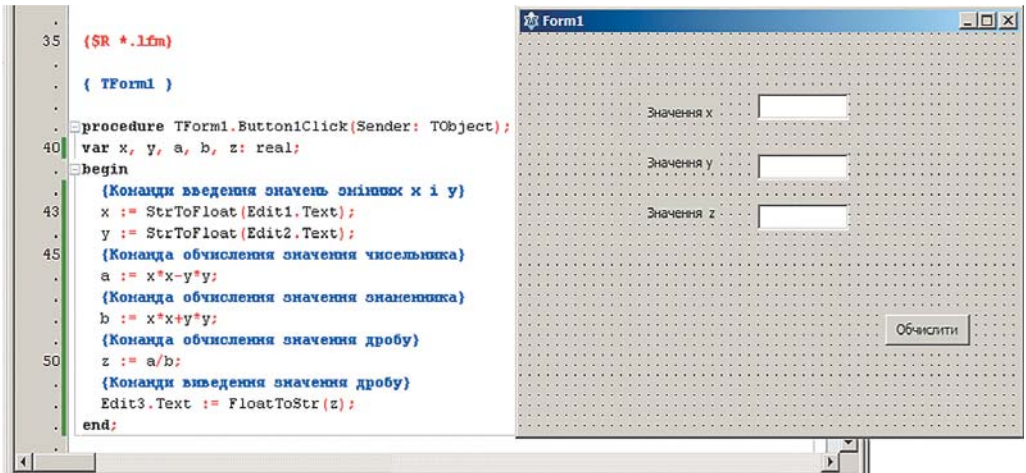
6.10. ábra. Az eljárás szövege logikai hibával és kommenttel

Ha a komment rövid, vagyis nem foglal el egy sornál többet, akkor a zárójelek helyett két dőlt vonalat tehetünk a komment elé: **//x értékének kiírása**.

Amennyiben a komment nem hosszabb egy sornál, akkor az utasítás után is elhelyezhetjük egy új sorban. A komment nem helyezkedhet el az utasításban.

A PROGRAM LÉPÉSENKÉNTI VÉGREHAJTÁSA

A logikai hibák felderítésének másik módja a program lépésenkénti végrehajtása, az úgynevezett **tracelés**. A **lépés** a projekt egy utasítását jelenti. A sor egy vagy több utasítást tartalmazhat. Vizsgáljuk meg a $z = \frac{x^2 - y^2}{x^2 + y^2}$ kifejezés kiszámítására szolgáló 6.11. ábrán látható eljárást. A lépésenkénti végrehajtás jobb megértése céljából a tört számlálóját és nevezőjét külön utasításokban számítottuk ki.

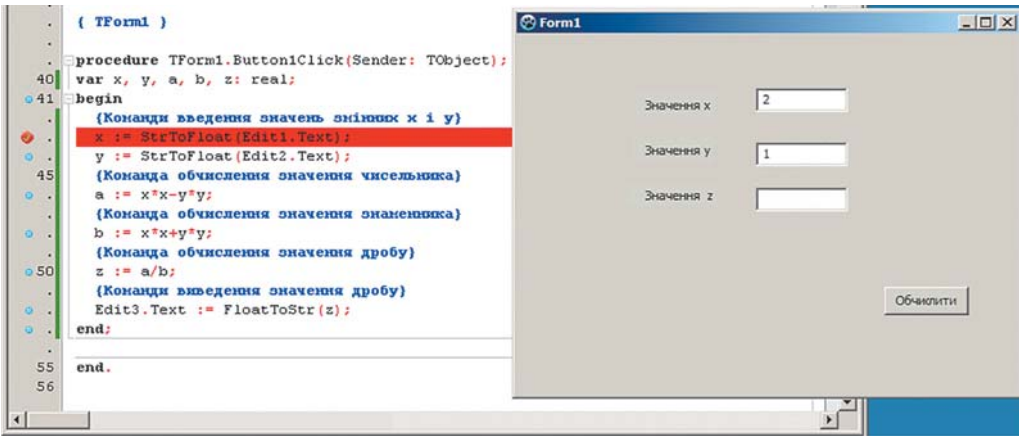


6.11. ábra. A $z = \frac{x^2 - y^2}{x^2 + y^2}$ értékének kiszámítására szolgáló projekt



Hogy gyorsabban átléphessünk a lépésenkénti végrehajtásra, célszerű az azt megelőző részt automatikus módban végrehajtani. Ezt úgy tehetjük meg, hogy a programban **töréspontot** helyezünk el ezen eljárás első sorában (6.12. ábra), majd elindítjuk a projekt futtatását. Ha a program bizonyos sorában töréspontot helyezünk el, ott az automatikus végrehajtás leáll. A kijelölt sor sem kerül végrehajtásra automatikus módban.

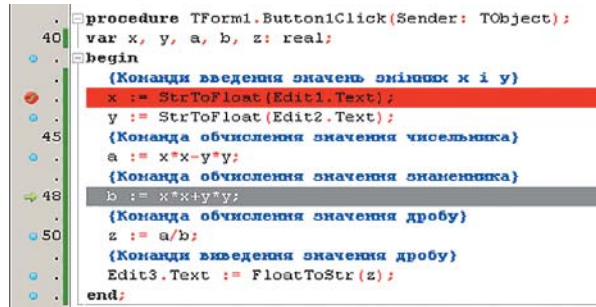
Töréspontot úgy helyezhetünk el a projektben, hogy a programsor sorszáma elé kattintunk. Ezután a sor piros háttérrel lesz kiemelve.

A projekt indítása után a megnyíló ablakban a megfelelő mezőkbe be kell írni az x és y értékét, majd a **Kiszámítás** gombra kell kattintani. Ezután kerül végrehajtásra a vizsgált eljárás. Mivel az eljárás első sorában töréspont van, a program futása itt megáll, és meg lehet kezdeni a lépésenkénti végrehajtást. Ezt a **Futtatás** \Rightarrow **Átlépni** paranccsal vagy az **F8** billentyű leütésével tehetjük meg. A soronkénti végrehajtás során az aktuálisan végrehajtandó parancs háttere szürkével lesz kijelölve (6.13. ábra).



6.12. ábra. Töréspont

A lépésenkénti végrehajtást a **Ctrl + F2** billentyűkombinációval vagy a **Leállítás**  vagy **Futtatás**  gomb lenyomásával bármikor felfüggeszthetjük. Az első két esetben a program futása megszakad, az utolsóban automatikus módban folytatódik.



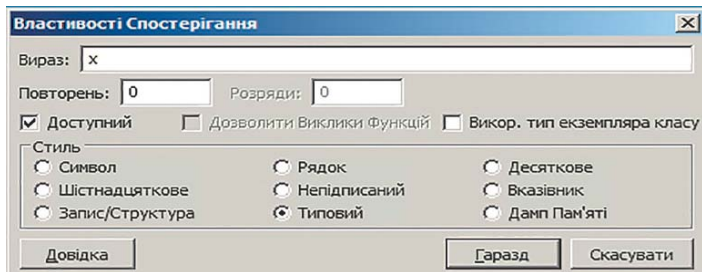
6.13. ábra. Lépésenkénti végrehajtás

A VÁLTOZÓK ÉRTÉKEINEK MEGTEKINTÉSE A LÉPÉSENKÉNTI VÉGREHAJTÁS SORÁN

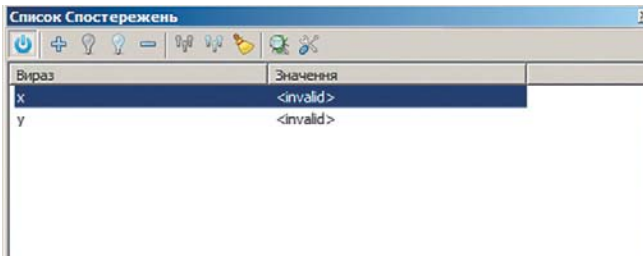
A lépésenkénti végrehajtásnak semmi értelme nem lenne, ha eközben nem vizsgálhatnánk meg, hogyan változnak a változók értékei. Éppen ez a vizsgálat teszi lehetővé, hogy megértsük a logikai hiba okát.

Ahhoz, hogy egy változó értékét megfigyelhessük a lépésenkénti végrehajtás során, el kell azt helyezni a **Figyelt elemek** listáján. Ezt a **Tulajdonság figyelő** ablakban tehetjük meg.

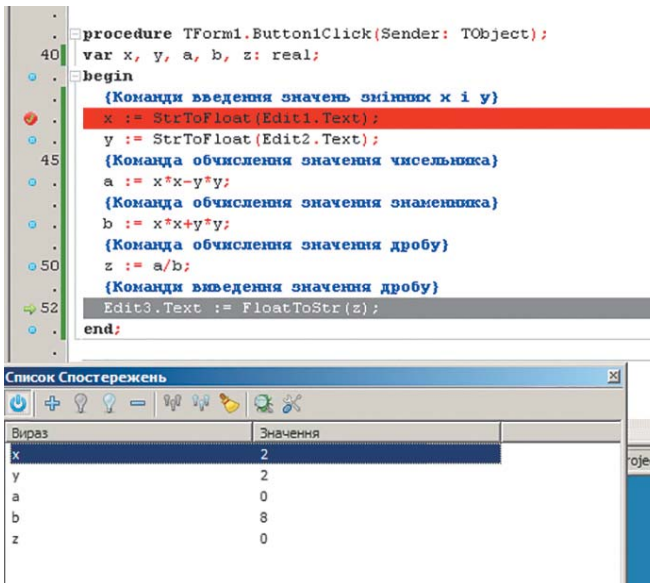
Hogy felvehessük a változót a **Figyelt elemek** listájára, üssük le a **Ctrl + F5**-öt, vagy hajtsuk végre a **Futtatás** ⇒ **Figyelő hozzáadása** parancsot (6.14. ábra). A **Kifejezés** mezőbe írjuk be a változó nevét, majd nyomjuk meg az **OK** gombot.



6.14. ábra. A Tulajdonság figyelő ablak



6.15. ábra. A Figyelt elemek listája



6.16. ábra. A változók értékei a program futásakor a Figyelőablakban

hogy melyik utasítás végrehajtása vezet a nem várt eredményhez. Ez az utasítás okozza a logikai hibát.






Összefoglalás

A **logikai hiba** olyan hiba a projektben, amely helytelen eredményhez vagy a program futásának megakadásához vezet.

Tesztadatok – a bemenő adatok olyan értékei, melyekhez ismerjük az elvárt kimenő adatok értékeit.

A **projekt tesztelése** – a logikai hibák keresésének folyamata tesztadatok igénybevételel.

Ezután megnyílik a **Figyelt elemek** listája (6.15. ábra), amely tartalmazza a felvett elemet. Ilyen módon a figyelt elemek listájára felvehetünk valamennyi olyan változót, amelynek értékét célszerű figyelni. A  gombra kattintva szintén felvehetünk új elemeket a figyelték közé, a  gombra kattintva eltávolíthatjuk a kijelölt elemet a listáról, a  gombra kattintva a lista valamennyi elemét törölhetjük.

A projekt futása előtt a figyelt változók értékei nem meghatározottak. Ezért a **Figyelt elemek listáján** az értékük **invalid** (nem létező, nem érvényes).

A soronkénti végrehajtás során ezen változók értéke változik, amit a **Figyelt elemek listáján** láthatunk is (6.16. ábra). Figyelemmel kísérve a változásokat és összevetve azokat a változók elvárt értékeivel, észre lehet venni,



Komment – olyan szöveg, amelyet a compiler a fordítás során figyelmen kívül hagy, és aminek célja a logikai hibák keresésének meggyorsítása. A kommenteket figurális zárójelek közé {}, zárójel-csillag párok közé (* *) tesszük, vagy dőlt vonalakkal // kezdünk.

Töréspont – a projekt azon sora, amelynél e projekt automatikus futtatása megszakad.

A **végrehajtás lépése** – a projekt egy sorába írt utasítások végrehajtása. Egy sorban egy vagy több utasítás lehet.

A program lépésenkénti végrehajtását *Futtatás* \Rightarrow *Átlépni* paranccsal vagy az **F8** leütésével hajtjuk végre.

A **Ctrl + F2** leütésével vagy a **Leállítás**  gomb lenyomásával a program futása megállítható, a **Futtatás**  gomb lenyomásával pedig folytatható.

Ahhoz, hogy egy változó értékét megfigyelhessük a lépésenkénti végrehajtás során, el kell azt helyezni a **Figyelt elemek** listáján. Ezt a **Tulajdonság figyelő** ablakban tehetjük meg.

Hogy felvehessük a változót a **Figyelt elemek listájára**, üssük le a **Ctrl + F5**-öt vagy a *Futtatás* \Rightarrow *Figyelő hozzáadása* paranccsal hozzá kell adni a figyelési listához. A **Kifejezés** mezőbe írjuk be a változó nevét, majd nyomjuk meg az **OK**-t.



Felelj a kérdésekre!

- 1°. Mit értünk tesztelés alatt?
- 2°. Mi célból végezzük a projekt tesztelését?
- 3°. Mi a komment? Hogy illeszthetünk kommentet a projektbe?
- 4°. Mi célból írunk a projektbe kommenteket?
- 5°. Mi célt szolgál a projekt lépésenkénti végrehajtása?
- 6°. Magyarázd el, hogy történik a projekt lépésenkénti végrehajtása?
- 7°. Mi a végrehajtás lépése? Hogy végezzük el?
- 8°. Mi a töréspont? Mire használjuk?
- 9°. Hogy lépünk ki a lépésenkénti végrehajtás módból?
- 10°. Hogy vizsgálhatjuk a változók értékeit a lépésenkénti végrehajtás során?



Végezd el a feladatokat!

1. Nyisd meg a **Téma 6/feladat 6.2/projekt 6.2.1** projektet, melyet a következő feladat megoldására hoztunk létre: *Két vonat egyidejűleg indult el egymás felé. Határozd meg, mikor találkoznak, ha ismert a kezdeti távolság közöttük és mind-egyik vonat sebessége!* A gomb **Click** eseménykezelője meg kell határozza a kérdéses időt, amennyiben a beviteli mezőkbe beírták a távolságot és a sebességeket. Futtasd a projektet a következő adatokkal:
 - a) távolság – 1000, sebességek – 120 és 130;



6. fejezet

- b) távolság – 600, sebességek – 80 és 120;
- c) távolság – 1200, sebességek – 115 és 58!

Ellenőrizd, hogy a kapott eredmények megegyeznek-e az elvártakkal! Amennyiben nem, magyarázd meg, miért következhetett ez be!



2•. Hozz létre három tesztadatsort a következő feladat megoldására szolgáló projekt tesztelésére: *Két vonat indult el ugyanarról a helyről ellentétes irányokban. Határozd meg, milyen messze lesznek ezek egymástól, ha ismert az indulás óta eltelt idő és a vonatok sebessége!* Határozd meg a kimenő adatok értékeit a megfelelő bemenő adatokhoz! Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.2** projektet! Teszteld a projektet az előkészített adatokkal! Döntsd el, tartalmaz-e a projekt hibákat! Ha igen, miért következtek ezek be?

3*. Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.3** projektet! Elemezd ezt a projektet a programban elhelyezett kommentek alapján, és határozd meg, milyen feladat megoldására szolgál! A következtetésed alapján készíts három tesztet a projekt teszteléséhez! Határozd meg az elvárt eredményeket! Teszteld a projektet a kidolgozott teszt sorokkal! Határozd meg, helyes volt-e a feltételezésed!

4°. Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.4** projektet! Futtasd le a projektet lépésenként! Figyeld meg a projekt végrehajtását!



5•. Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.5** projektet! Végezd el a projektet lépésenként, közben figyeld az s_1 és s_2 értékeit!

6•. A gyerekek ünnepre készülve csokit és süteményt vettek. Ismertek a csoki és a sütemények árai és a vásárolt mennyiségek. Hozz létre projektet a kifizetett összeg meghatározására! Mentsd el a projektet a mappában létrehozott **feladat 6.2.6** mappába! Futtasd a projektet lépésenkénti módban, figyeld meg a változók értékeit!

7•. Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.7** projektet! Lépésenkénti módban futtatva a projektet, figyeld meg az x , a és s változók értékeit! Állapítsd meg, hogyan kapcsolódnak az a értékei az x által felvett értékekhez! Változtasd meg a projektet, hogy az s értéke $x + x^2 + x^4$ legyen! Adj a programhoz kommentet, amiben leírod, milyen változásokat vittél a programba! Mentsd a projektet a mappában létrehozott **feladat 6.2.7** mappába!

8*. Nyisd meg a **Téma 6\feladat 6.2\projekt 6.2.8** projektet! A projekt célja, hogy határozza meg, mennyivel tart tovább ez egyik vonatnak a két város közötti út, mint a másiknak, ha ismertek a távolság és a sebességek. Állapítsd meg, tartalmaz-e a projekt logikai hibákat. Ha igen, állapítsd meg, melyik utasítások okozzák! Állíts össze tesztadatsorokat a logikai hibák kiszűrésére! Mentsd a javított projektet a mappában létrehozott **feladat 6.2.8**. mappába!



11. SZÁMÚ GYAKORLATI MUNKA

Program tesztelése és javítása

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Nyisd meg a **Lazarust!**
2. Nyisd meg a **Téma 6 gyakorlati 11** projektet, amely meghatározza, *hányszor nagyobb két valós szám szorzata ezen számok összegénél!*
3. Hozz létre három tesztadatsort a projekt vizsgálatára! Írd be ezeket a füzetedbe! Számítsd ki az elvárt eredményeket! Végezd el a projekt tesztelését! Vesd össze a projekt futásának eredményét az elvárttal! Vonj le következtetést!
4. Adj kommenteket azokhoz a sorokhoz, amelyek a számításokat végzik!
5. Hajtsd végre a projektet lépésenkénti módban! A futtatás során figyelj az összeg és a szorzat értékének változásait! Írd le, milyen értékek szerepelnek a **Figyelőablakban** a projekt futása előtt, közben és után valamennyi tesztsor esetében!
6. Mentsd a projektet a mappádban létrehozott **gyakorlati 11** mappába!



6.3. LOGIKAI KIFEJEZÉSEK. LOGIKAI VÁLTOZÓK. LOGIKAI MŰVELETEK



1. Mi a kijelentés? Mely kijelentéseket tekintünk igaznak, és melyeket hamisnak?
2. Milyen tulajdonságai vannak a változóknak?
3. Mit határoz meg az adattípus? Milyen típusai lehetek a számot tartalmazó változóknak?

A KIJELENTÉS MINT LOGIKAI KIFEJEZÉS

Már tudjátok, hogy a **kijelentés** egy olyan mondat, ami valamely objektumra, objektumok kapcsolatára nézve olyan állítást fogalmaz meg, amelyről egyértelműen eldönthető, hogy **igaz** vagy **hamis**. Igaz kijelentés például a *Kijev Ukrajna fővárosa* mondat, hamis a $3 > 5$ állítás.

A kijelentés legfontosabb tulajdonsága annak **logikai értéke**. Ha egy kijelentés igaz, akkor logikai értéke **true** (igaz). Ha a kijelentés hamis, akkor logikai értéke **false** (hamis).

A *Kijev Ukrajna fővárosa*, a $2 * 5 - 4 = 6$, a $4 < 12$ kijelentések logikai értéke tehát **true**, az *Ez a tankönyv hetedikeseknek szól*, a $2 * 7 + 3 = 12$, a $3 > 5$ logikai értéke pedig **false**.

A kijelentést **logikai kifejezésnek** tekinthetjük.

A **logikai kifejezés** olyan kifejezés, amely **true** és **false** értékeket vehet fel.



Logika (a görög *λόγος* – szó, gondolat, értelem, nyelv szóból) – a helyes gondolkodás törvényeit, módszereit és formáit, a következtetés módozatait vizsgáló tudomány.

Ha a kijelentés igaz, a megfelelő logikai kifejezés értéke is igaz. Vagyis *Kijev Ukrajna fővárosa* = **true**, $2 * 5 - 4 = 6$ = **true**, $4 < 12$ = **true**.

Ha a kijelentés hamis, a megfelelő logikai kifejezés értéke is hamis, tehát *Ez a tankönyv*

hetedikeseknek szól = **false**, $2 * 7 + 3 = 12$ = **false**, $3 > 5$ = **false**.

LOGIKAI VÁLTOZÓK

A logikai kifejezés értékét változó is felveheti. Vagyis az $x := 2 + 12/3 = 6$ vagy az $y = 32 < 13$ kifejezéseknek van értelme. Az x értéke **true**, az y pedig **false**.

Az olyan változókat, amelyek a **true** és **false** értékeket vehetik fel, **logikai változóknak** nevezzük.

A logikai változókat **boolean**-nak deklaráljuk, például **var x, y: boolean**.

A logikai változóknak gyakran az $x := \text{true}$ vagy $y := \text{false}$ utasításokkal adunk értéket.

A logikai változókat logikai kifejezéseknek tekintjük.

LOGIKAI MŰVELETEK

A logikai kifejezésekkel **logikai műveleteket** végezhetünk. Ilyenek a **tagadás**, a **konjunkció** (a latin *conjunctio* – egyesítés szóból) és a **diszjunkció** (latin *disjunctio* – szétválasztás) stb.

Az x **tagadása** olyan logikai művelet, amelynek eredménye **true**, ha x értéke **false**, illetve **false**, ha x értéke **true**.

Az x tagadását **not x**-szel jelöljük.

Vagyis ha $x = \text{true}$, akkor **not x = false**, ha $x = \text{false}$, akkor **not x = true**. Ezek alapján össze tudjuk állítani a tagadás műveletének igazságtáblázatát.

x	not x
true	false
false	true

A tagadás műveletének igazságtáblázata

A tagadás képzése a **nem** tagadószó használatával történik. Az $A = \text{Létezik legnagyobb prímszám}$ tagadása a **not A = Nem létezik legnagyobb prímszám**.

Az x tagadásának szokásos jelölése az \bar{x} vagy az \sqrt{x} .

Az x és y logikai kifejezések **konjunkciója** az a logikai kifejezés, amelynek értéke akkor **true**, ha az x és az y értéke is **true**, minden más esetben pedig **false**.

Az x és y kifejezések konjunkciójának szokásos jelölése x **and** y (az angol *and* – és szóból).

Két logikai kifejezés konjunkcióját az **és** kötőszóval fogalmazzuk meg. A *36 többszöröse 3-nak és 36 többszöröse 2-nek* kijelentés a *36 többszöröse 2-nek*, illetve a *36 többszöröse 3-nak* kifejezések konjunkciója.



Az x és y konjunkciójának szokásos jelölése még $x \wedge y$.

A **diszjunkció** az x és y logikai kifejezéseken végzett olyan művelet, amelynek eredménye akkor **true**, ha x és y közül legalább az egyik értéke **true** és akkor **false**, ha x és y mindegyike **false**.

Az x és y logikai kifejezések diszjunkciójának jelölése x or y (*or* – vagy).

Két kijelentés diszjunkciójának megfogalmazásakor a **vagy** kötőszót használjuk. A $11 \leq 21$ (11 kisebb vagy egyenlő 21-gyel) kifejezés a $11 < 21$ és a $11 = 21$ kijelentések diszjunkciója.

A diszjunkciót szokás még így jelölni: $x \vee y$. Gyakran használunk több **logikai** műveletet tartalmazó **kifejezéseket**, például $(x$ or $y)$ and $(x$ or $z)$.

A logikai műveleteknek is van **prioritási sorrendje**: először a tagadást hajtjuk végre, majd a konjunkciót, végül a diszjunkciót. Ha ettől a sorrendtől el szeretnénk térni, azt zárójelek alkalmazásával tehetjük meg.

A logikai kifejezések értékének kiszámításakor igazságtáblázatokat használunk. E célból a kifejezést alkotó logikai változók értékeinek valamennyi lehetséges kombinációját át kell tekinteni.

x	y	x and y
true	true	true
true	false	false
false	true	false
false	false	false

A konjunkció igazságtáblázata

x	y	x or y
true	true	true
true	false	true
false	true	true
false	false	false

A diszjunkció igazságtáblázata

x	y	z	1	2	3
			x or y	x or z	$(x$ or $y)$ and $(x$ or $z)$
true	true	true	true	true	true
true	true	false	true	true	true
true	false	true	true	true	true
false	true	true	true	true	true
true	false	false	true	true	true
false	true	false	true	false	false
false	false	true	false	true	false
false	false	false	false	false	false

Érdekes történelmi tények

A logikai típusú változókkal végzett műveleteket először **George Boole** (1815–1864) angol matematikus és filozófus (6.17. ábra) definiálta. Ő kutatta először ezek tulajdonságait. 1854-ben jelent meg *A gondolkodás azon törvényeinek vizsgálata, amelyeken a matematikai logika és valószínűség elmélete nyugszik* című munkája, amiben szisztematikusan megfogalmazza azt, amit ma *kijelentések algebrájának* vagy *Boole-logikának* neveznek. A Boole-logika a számítógépek létrehozásának matematikai eszközévé vált.



6.17. ábra. George Boole

Összefoglalás

Logikai kifejezésnek azt a kifejezést tekintjük, amelynek értéke **true** vagy **false** lehet. A kijelentés **logikai kifejezésnek** tekinthető. Ha egy kijelentés igaz, akkor logikai értéke **true** (igaz). Ha a kijelentés hamis, akkor logikai értéke **false** (hamis).

Azokat a változókat, amelyek **true** vagy **false** értékeket vehetnek fel, **logikai típusú változóknak** nevezzük. A logikai változókat **booleannak** kell deklarálni. A logikai változókat ugyancsak logikai kifejezéseknek tekintjük.

x	$not\ x$
true	false
false	true

A tagadás műveletének igazságtáblázata

x	y	$x\ and\ y$
true	true	true
true	false	false
false	true	false
false	false	false

A konjunkció igazságtáblázata

x	y	$x\ or\ y$
true	true	true
true	false	true
false	true	true
false	false	false

A diszjunkció igazságtáblázata

A fent áttekintett logikai műveletek **prioritási sorrendje** a következő: először a tagadás, majd a konjunkció, végül a diszjunkció. A műveletek sorrendjének megváltoztatása céljából zárójeleket használunk.

A logikai kifejezések értékének meghatározására igazságtáblázatokat használhatunk. Ezekben a bemenő logikai változók értékeinek összes lehetséges kombinációját át kell tekinteni.



Felelj a kérdésekre!

- 1°. Milyen kifejezést tekintünk logikainak? Mondj példákat!
- 2°. Milyen változókat nevezünk logikai változóknak?
- 3°. Hogy deklaráljuk a logikai változókat?



- 4•. Mi a logikai kifejezés tagadása?
- 5•. Mit értünk két logikai kifejezés konjunkcióján?
- 6•. Mit értünk két logikai kifejezés diszjunkcióján?
- 7°. Mi a logikai műveletek prioritásának sorrendje?
- 8*. Mit tanulmányoz a logika?



Végezd el a feladatokat!

- 1°. Mondd el a műveletek sorrendjét a következő kifejezésekben!
 - a) $a \text{ or } \text{not } a \text{ and } b$;
 - b) $\overline{\text{not } a \text{ or } \text{not } b} \text{ and } a$;
 - c) $(a \text{ or } \text{not } b) \text{ and } (\text{not } a \text{ or } \text{not } b)$;
 - d) $\text{not not } a \text{ and not } b \text{ or } c \text{ and } d$.
- 2•. Készítsd el az a és b logikai változókat tartalmazó kifejezések igazságtáblázatit!
 - a) $a \text{ or } a \text{ and } b$;
 - b) $a \text{ or } \overline{\text{not } a} \text{ and } b$;
 - c) $a \text{ or } b \text{ or } \text{not } b$;
 - d) $\text{not } a \text{ or } \text{not } b$;
 - e) $(a \text{ or } \text{not } b) \text{ and } (a \text{ or } \text{not } b)$;
 - f) $a \text{ and } \overline{\text{not } b} \text{ or } \overline{\text{not } a} \text{ and } b$.
- 3•. Fogalmazd meg az alábbi állítások tagadását!
 - a) A 3 osztója 545-nek.
 - b) A gépkocsi nem haladhat át a piros jelzésen.
 - c) Léteznek derékszögű paralelogrammák.
 - d) A $2x^2 - 3x + 1$ egyenletnek van egész gyöke.
 - e) Nem létezik 2-vel osztható természetes szám.
 - f) Létezik olyan egész szám, ami minden más egész számmal osztható.
- 4•. Az alábbi kijelentések közül válaszd ki a konjunkciókat és diszjunkciókat! Határozd meg ezek logikai értékét!
 - a) A 27 többszöröse 3-nak és 9-nek.
 - b) $17 < 42 < 18$.
 - c) A 2 prímszám vagy páros.
 - d) Az ABC háromszög hegyesszögű, derékszögű vagy tompaszögű.
 - e) Minden paralelogramma átlói derékszögben metszik és felezik egymást.
 - f) $7^2 = 49$ és $(-7)^2 = 49$.
 - g) $21 \leq 21$.
 - h) $21 \leq 18$.
- 5•. Legyen $A = \text{Szerhijenko angol nyelvet tanul}$ és $B = \text{Szerhijenkó 8-as osztályzatot kapott informatikából}$. Fogalmazd meg az alábbi kijelentéseket!
 - a) $A \text{ and } \text{not } \overline{B}$
 - b) $\overline{\text{not } A} \text{ or } B$
 - c) $A \text{ or } \overline{\text{not } A} \text{ and } B$
 - d) $\text{not } A \text{ and } \overline{\text{not } B} \text{ and } A$





6*. Állapíts meg az alábbi kifejezések logikai értékét!

a) $x > 0$ és $y > 0$ vagy $x < 0$ és $y < 0$

b) $x > 0$ és nem $y < 0$ vagy $x < 0$ és $y > 0$

A változók következő értékei mellett:

a) $x = 5, y = 8$;

b) $x = 5, y = -8$;

c) $x = -5, y = 8$;

d) $x = -5, y = -8$.



6.4. ELÁGAZÁST TARTALMAZÓ ALGORITMUSOK



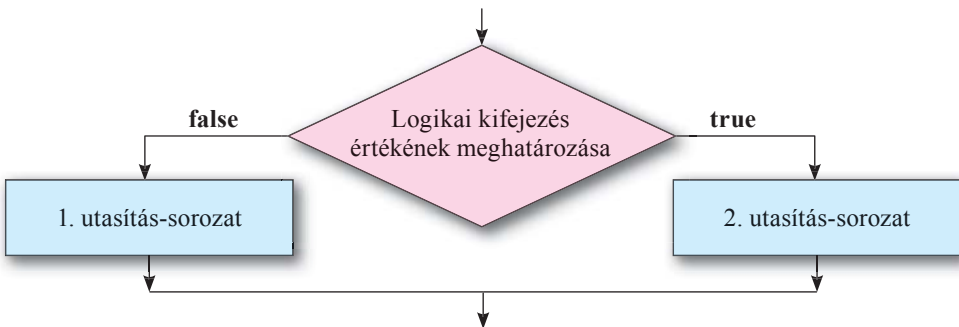
1. Milyen kifejezéseket nevezünk logikainak? Milyen értékeket vehetnek ezek fel?
2. Mik a logikai műveletek? Ismertesd ezek igazságtábláit!
3. Mi az elágazás? Milyen típusai vannak? Miben különböznek ezek?

ELÁGAZÁSOKAT TARTALMAZÓ ALGORITMUSOK

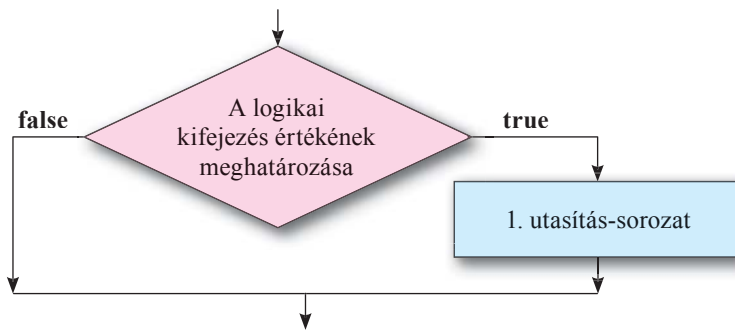
Már tudjátok, hogy az algoritmusok elágazásokat tartalmazhatnak. Azt is tudjátok, hogy az elágazás, mint az algoritmus része a feltétel ellenőrzését végző utasításból áll, aminek eredménye **igaz (true)** vagy **hamis (false)** lehet. Az ellenőrzés eredményétől függ, hogy ezt követően milyen utasítások kerülnek végrehajtásra.

Általános esetben ebben az utasításban egy logikai kifejezés értékét kell meghatározni, ami **true** vagy **false** lehet.

Az elágazás folyamatábrája általános esetben a következő (6.18. és 6. 19. ábrák):



6.18. ábra. Teljes elágazás



6.19. ábra. Nem teljes elágazás

ELÁGAZÁSOK OBJECT PASCALBAN

A teljes elágazás általános alakja **Object Pascal**ban a következő:

```

If<logikai kifejezés>
Then begin
    <1. utasítás-sorozat>
end
Else begin
    <2. utasítás-sorozat>
end;
    
```

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

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **1. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, a **2. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés.

Object Pascalban a nem teljes elágazás általános alakja a következő:

```

If<logikai kifejezés>
Then begin
    <utasítás-sorozat>
end;
    
```

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **utasítás-sorozat** végrehajtásra kerül, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, az utasítás-sorozat nem kerül végrehajtásra, a vezérlés azonnal az elágazás utáni első utasításra kerül.

Itt hívjuk fel a figyelmeteket a következőkre:

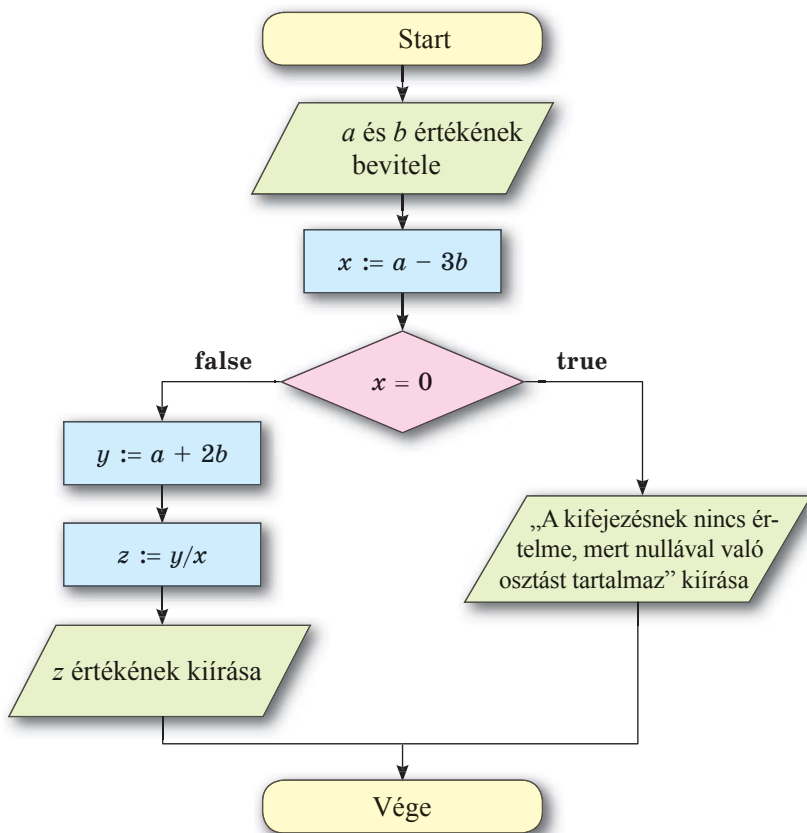
- ha a **Then** és **Else** után csak egy utasítás következik, akkor a **begin** és **end** utasítás-zárójeleket nem kötelező alkalmazni;
- az **Else** előtt pontosvesszőt tilos tenni.

Vizsgáljunk most meg egy elágazást tartalmazó algoritmust:

1. feladat. Számítsuk ki az $(a + 2b) / (a - 3b)$ kifejezés értékét, ahol a és b valós számok.

Mivel a kifejezés osztást tartalmaz, előfordulhat, hogy a nevező nullává válik (például $a = 3$, $b = 1$ esetében), a feladat megoldásának algoritmusá elágazást kell tartalmazzon, amely figyelembe veszi, az $a - 3b$ értéke nulla vagy sem.

Az algoritmus folyamatábráját a 6.20. ábrán láthatjuk:



6.20. ábra. Az 1. feladat megoldásának algoritmus

A 6.20. ábrán látható algoritmus **Object Pascal** nyelven a következő alakban írható fel:

```
a := StrToInt(Edit1.Text);
```

```
b := StrToInt(Edit2.Text);
```

```
x := a-3*b;
```

```
If x = 0
```

```
Then Label1.Caption := ('A kifejezésnek nincs értelme, nullával való osztást tartalmaz')
```

```
Else begin
```

```
y := a+2*b;
```

```
z := y/x;
```

```
Label1.Caption := FloatToStr(z);
```

```
end;
```

Felhívjuk a figyelmeteket, hogy az $a - 3b$ értékének kiszámítására nem kötelező külön utasítást alkalmazni és nem szükséges ezt az értéket külön az x változóban tárolni. Az $(a + 2b) / (a - 3b)$ értékét sem szükséges két lépésben kiszámítani. A vizsgált algoritmust a következőképpen is leírhatjuk:



```

a := StrToInt(Edit1.Text);
b := StrToInt(Edit2.Text);
If a-3*b = 0
Then Label1.Caption := ('A kifejezésnek nincs értelme, nullával való osztást
tartalmaz')
Else begin
z := (a+2*b) / (a-3*b);
Label1.Caption := FloatToStr(z);
end;
    
```

Tekintsünk át most egy olyan feladatot, amelynek elágazása konjunkciót alkalmaz.

2. feladat. Számítsd ki a következő függvény értékét:

$$y = \begin{cases} 2x - 12, & \text{ha } -3 < x < 5, \\ 7 - 8x, & x \text{ minden egyéb értéke mellett!} \end{cases}$$

A feladat megoldásának algoritmus a következő:

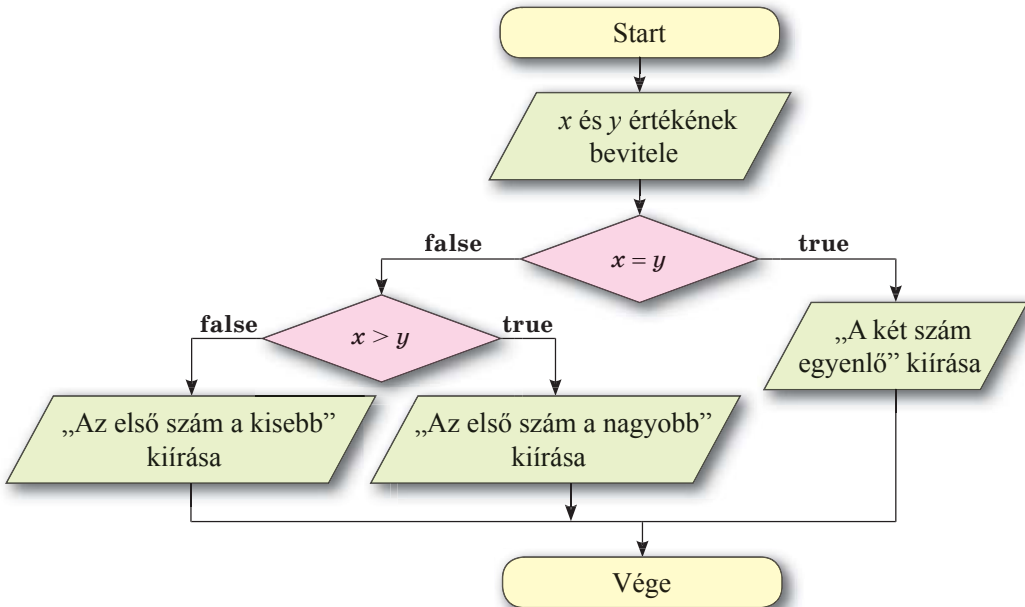
```

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

Végül lássunk egy olyan feladatot, amelynek megoldása során valamivel összetettebb elágazás alkalmazására van szükség.

3. feladat. Adva van két szám. Határozd meg, egyenlők-e ezek! Ha nem, akkor határozd meg, melyik a nagyobb!

A feladat megoldásának folyamatábrája a 6.21. ábrán látható:



6.21. ábra. A 3. feladat megoldásának folyamatábrája



6. fejezet

Ebben az eljárásban az egyik elágazás egy másik elágazást tartalmaz. A projekt elágazást tartalmazó részlete:

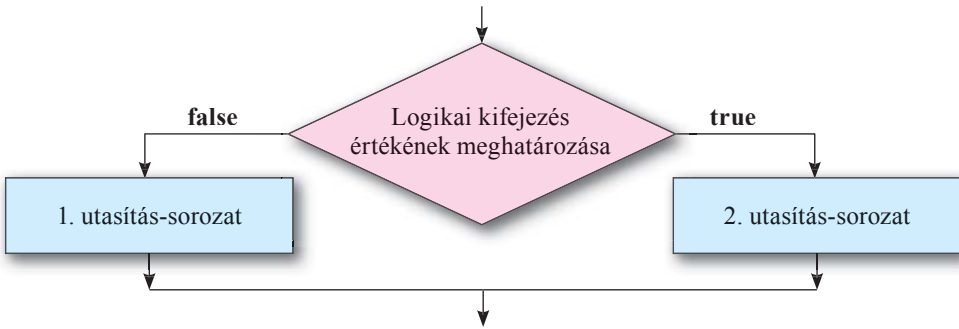
```

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

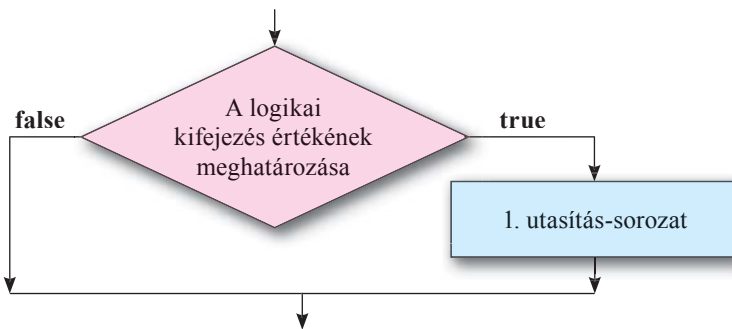


Összefoglalás

Az elágazás folyamatábrája általános esetben a következő (6.22. és 6.23. ábrák):



6.22. ábra. Teljes elágazás



6.23. ábra. Nem teljes elágazás

A teljes elágazás általános alakja **Object Pascal**ban a következő:

```

If <logikai kifejezés>
Then begin
    <1. utasítás-sorozat>
    end
Else begin
    <2. utasítás-sorozat>
    end;
  
```



Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor a **1. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, a **2. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés.

Object Pascalban a nem teljes elágazás általános alakja a következő:

```
If <logikai kifejezés>
Then begin
    <utasítás-sorozat>
end;
```

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **utasítás-sorozat** végrehajtásra kerül, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, az utasítás-sorozat nem kerül végrehajtásra, a vezérlés azonnal az elágazás utáni első utasításra kerül.



Felelj a kérdésekre!

- 1°. Milyen a teljes elágazás folyamatábrája? Ismertesd ennek végrehajtását!
- 2°. Milyen a nem teljes elágazás folyamatábrája? Ismertesd ennek végrehajtását!
- 3°. Miben különbözik a teljes és nem teljes elágazás végrehajtása?
- 4°. Mi a teljes elágazás általános alakja **Object Pascalban**?
- 5°. Mi a nem teljes elágazás általános alakja **Object Pascalban**?
- 6°. Miben különbözik a teljes és a nem teljes elágazás végrehajtása **Object Pascalban**?



Végezd el a feladatokat!

1. Ismertesd az utasítás végrehajtását:

```
If <1. logikai kifejezés>
Then begin
    <1. utasítás-sorozat>
end
Else If <2. logikai kifejezés>
Then begin
    <2. utasítás-sorozat>
end
Else begin
    <3. utasítás-sorozat>
end;
```

2. Magyarázd el a következő utasítás-sorozat működését:

```
If <1. logikai kifejezés>
Then begin
    <1. utasítás-sorozat>
end;
```



6. fejezet

If <2. logikai kifejezés>

Then begin

<2. utasítás-sorozat>

end;

3*. Magyarázd el, miben különbözik az 1. és 2. feladatok végrehajtása!

4. Hozz létre projektet az $(a + b) - c : (a - b)$ kifejezés értékének kiszámítására! MentSD a projektet a mappádban létrehozott **feladat 6.4.4.** mappába!



5. Hozz létre projektet az $a + b - c : (a - 2b)$ kifejezés értékének kiszámítására! MentSD a projektet a mappádban létrehozott **feladat 6.4.5.** mappába!

6. Hozz létre projektet, amely összehasonlít két valós számot, kiírja a kisebbet, vagy jelzi, ha a számok egyenlők! MentSD a projektet a mappádban létrehozott **feladat 6.4.6.** mappába!



7. Hozz létre projektet, amely meghatározza, hogy két nem egyenlő valós szám közül melyik a nagyobb! MentSD a projektet a mappádban létrehozott **feladat 6.4.7.** mappába!



8. Hozz létre projektet egy valós szám abszolút értékének meghatározására! MentSD a projektet a mappádban létrehozott **feladat 6.4.8.** mappába!

9*. Hozz létre projektet, amely meghatározza, hogy három nem egyenlő szám közül melyik a legkisebb! MentSD a projektet a mappádban létrehozott **feladat 6.4.9.** mappába!

10*. Hozz fel példákat elágazásokra a nyelvtan, matematika, más tantárgyak köréből!

11*. Hozz fel példákat a mindennapi életből, amelyek elágazásokkal írhatók le!

12*. Hozz létre projektet, amelyben egy gomb 20 pixellel jobbra mozdul, ha az egérmutatót fölé visszük! Ha a gomb eléri a form jobb szélét, jelenjen meg a bal oldalon! MentSD a projektet a mappádban létrehozott **feladat 6.4.12.** mappába!

13*. Hozz létre projektet az $ax = b$ egyenlet megoldására! MentSD a projektet a mappádban létrehozott **feladat 6.4.13.** mappába!

14* Hozz létre projektet az $ax + b = c$ egyenlet megoldására! Hajtsd végre a projektet, ha 1) $a = 2, b = -8, c = 18$; 2) $a = 20, b = 5, c = 5$; 3) $a = 0, b = 12, c = 3$; 4) $a = 0, b = 10, c = 10$! MentSD a projektet a mappádban létrehozott **feladat 6.4.14.** mappába!

15*. Hozz létre projektet, amely meghatározza, hogy három adott szakaszból képezhető-e háromszög! MentSD a projektet a mappádban létrehozott **feladat 6.4.15.** mappába!

16. Hozz létre projektet a következő kifejezés értékének kiszámítására:

$$y = \begin{cases} 15 - 3x, & \text{ha } -1 < x < 4, \\ 6 + 4x, & x \text{ minden egyéb értéke mellett!} \end{cases}$$

MentSD a projektet a mappádban létrehozott **feladat 6.4.16.** mappába!



6.5. A JELÖLŐNÉGYZET. A KOMBINÁLT LISTA. A VÁLASZTÓGOMB



1. Milyen a teljes és a nem teljes elágazás általános alakja **Object Pascal**ban?
2. Milyen tulajdonságai vannak a címkének és mezőnek, hogy változtatjuk meg ezeket?
3. Mire használták az általuk használt programokban a jelölőnégyzetet, kombinált listákat és választógombokat?

A JELÖLŐNÉGYZET ÉS ALKALMAZÁSA

Vizsgáljuk meg a **jelölőnégyzet** komponenst! Ennek egyes tulajdonságait, például a **Checked** (ellenőrzött) tulajdonságot jól lehet használni az elágazások programozása során.

Jelölőnégyzetet a **Standard** eszköztár **TCheckbox** (ellenőrző doboz) gombjára duplán kattintva helyezhetünk el a formon. A jelölőnégyzet a form bal felső sarkában jelenik meg (6.24. ábra), tulajdonságai pedig megjelennek az **Objektum Felügyelő komponenslistájában**. A jelölőnégyzet a form területén áthelyezhető.

A jelölőnégyzet **Caption**, **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font**, **Width**, **Height** tulajdonságok értelmezése megegyezik azzal, amit a címke esetében tapasztaltunk. A **Width** és **Height** tulajdonságok értékei a jelölőnégyzet melletti szöveg hossza és betűmérete által meghatározott. Ez a szöveg a jelölőnégyzet **Caption** tulajdonságának értéke. Ezt a szöveget, ahogy a címkénél is, beírhatjuk a **Caption** mezőbe, vagy a gombra kattintva megnyíló **Karakterlánc-szerkesztőbe**.

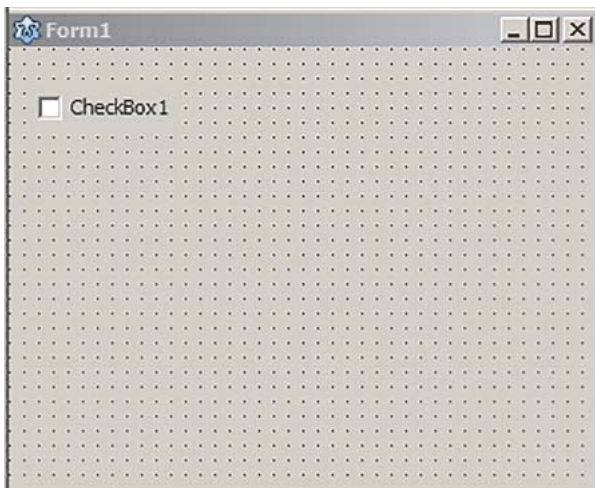
A jelölőnégyzet **Checked** tulajdonságának két értéke lehet, a **true** (a jelölőnégyzetben a pipa be van kapcsolva) vagy **false** (a pipa nincs bekapcsolva).

Ha a formon több jelölőnégyzet van, akkor be lehet kapcsolva valamennyi, néhány, esetleg egyik sem. Ha a jelölőnégyzet be van kapcsolva, akkor egy pipa látható benne.

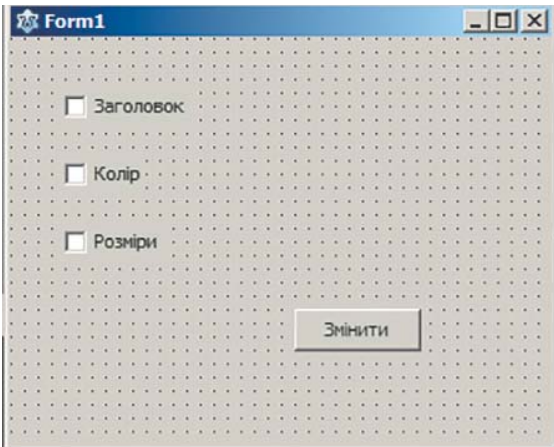
Jelölőnégyzet segítségével valamely objektum tulajdonságainak értékeit is be lehet állítani.

Vizsgáljuk meg, hogy használható a jelölőnégyzet elágazás programozása során.

Tegyük fel, hogy a projekt futása során az ablak bizonyos tulajdonságait szeretnénk megváltoztatni (cím, szín, méretek), vagy ezek közül néhányat, esetleg egyiket sem.



6.24. ábra. Jelölőnégyzet a formon



6.25. ábra. Form három jelölőnégyzettel és gombbal

```

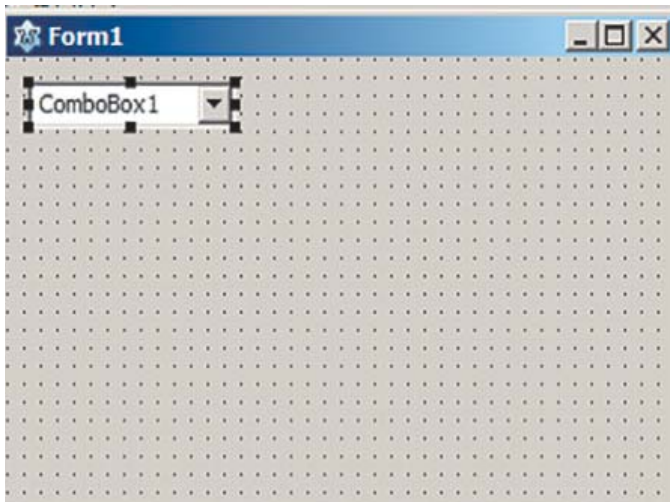
If CheckBox3.Checked
Then begin
  Form1.Width := 500;
  Form1.Height := 200;
end;

```

Ezekben az utasításokban egymástól függetlenül végezzük a jelölőnégyzetek állapotát. A jelölők állapotától függően történik meg az ablak tulajdonságainak megváltoztatása.

KOMBINÁLT LISTÁK ÉS AZOK ALKALMAZÁSA

Egy másik olyan komponens, amelynek tulajdonságait az elágazás logikai ellenőrzése során használni lehet, a **kombinált lista**.



6.26. ábra. Kombinált lista a formon

E célból helyezünk el a formon egy gombot és három jelölőnégyzetet (6.25. ábra). A jelölőnégyzetek elhelyezésekor azok **Checked** állapota alapértelmezésben **false**. Mivel a projekt futása során a jelölőnégyzetek állapota lehet **false**, célszerű ezek **Checked** állapotát alapértelmezésben hagyni.

Hozzuk létre a gomb **Click** eseményének kezelőjét, és helyezünk el ebben elágazásokat:

```

If CheckBox1.Checked
Then Form1.Caption := 'A cím
megváltozott';
If CheckBox2.Checked
Then Form1.Color := clRed;

```

Kombinált listát a **Standard** eszköztár **TComboBox** (kombinált lista) gombjára duplán kattintva helyezhetünk el a formon. A kombinált lista a form bal felső sarkában jelenik meg (6.26. ábra), tulajdonságai pedig megjelennek az **Objektum felügyelő komponenslistájában**. A kombinált lista a form területén áthelyezhető.

A kombinált lista **Top**, **Left**, **Height**, **Width**, **Name**, **Color**, **Enabled**,



Visible, **Font**, **Text** tulajdonságok értelmezése a mező és a kombinált lista esetében ugyanaz.

A kombinált lista elemeit a **Karakterlánc-szerkesztő**ben adhatjuk meg, ami az **Items** tulajdonság mellett található **...** gombra nyílik meg.

A projekt futása közben lehetőségünk van arra, hogy a lista bármelyik elemét kiválaszthatjuk. Ekkor a kiválasztott elem lesz látható a mezőben, továbbá a kiválasztott elem lesz a **Text** tulajdonság értéke.

Vizsgáljuk meg, hogy használhatjuk a kombinált listát az elágazások programozása során.

Tegyük fel, hogy a projekt futása során az ablak egy tulajdonságát – címét, színét vagy a méreteit – szeretnénk megváltoztatni.

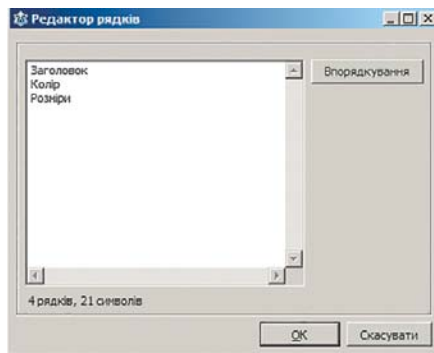
Helyezzünk el a megfelelő kombinált listát a formon (6.27. ábra). Helyezzünk el továbbá egy gombot is, ennek **Click** eseménykezelője ellenőrzi majd, hogy a lista (6.28. ábra) melyik eleme van kiválasztva, illetve elvégzi a megfelelő tulajdonság beállítását.

Válasszuk ki a *Címet* a listából, és kattintsunk a gombra. Akkor a **ComboBox1.Text = 'Cím'** értéke **true** lesz, vagyis lefut a **Form1.Caption := 'A cím megváltozott'** utasítás, és megváltozik az ablak címe, majd az elágazás és az eljárás is befejeződik.

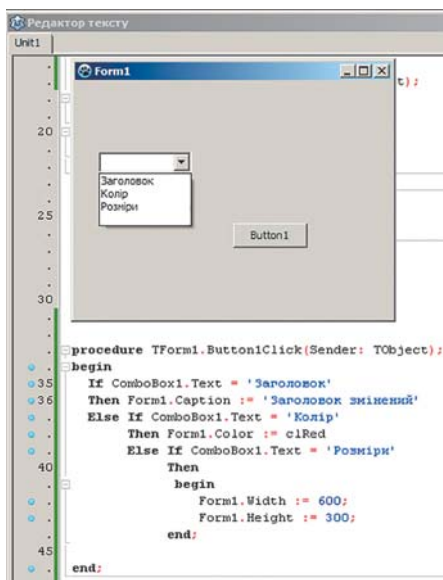
Ha a *Szint* választjuk ki a listából, akkor a **ComboBox1.Text = 'Cím'** értéke **false** lesz, a **ComboBox1.Text = 'Szín'** értéke **true**, vagyis lefut a **Form1.Color := clRed** utasítás, és megváltozik az ablak címe, majd az elágazás és az eljárás is befejeződik.

Ha a listából a *Méreteket* választjuk, majd a gombra kattintunk, akkor a **ComboBox1.Text = 'Cím'** értéke **false** lesz, a **ComboBox1.Text = 'Szín'** értéke szintén **false**, a **ComboBox1.Text = 'Méretek'** pedig **true**, ezért lefutnak a **Form1.Width := 600** valamint a **Form1.Height := 300** utasítások, megváltozik az ablak mérete, majd az elágazás és az eljárás is befejeződik.

Ha a lista egyik elemét sem választjuk ki és a gombra kattintunk, akkor valamennyi logikai kifejezés értéke **false** lesz, ezért az ablak egyetlen tulajdonsága sem változik meg és az eljárás véget ér.



6.27. ábra. Karakterlánc-szerkesztő ablaka a listaelemekkel




6.28. ábra. Az ablak címét, színét, méretét megváltoztató eljárás és futásának ablaka




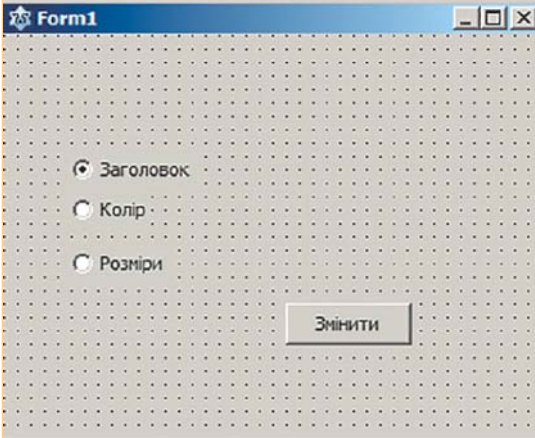
Ha többet szeretnél tudni

Választógombok és azok alkalmazása

Ha a lehetséges variációk közül egyet kell kiválasztani, például az objektum tulajdonságainak lehetséges értékei közül kell egyet kiválasztani, a **választógombok** használata a jó megoldás.

Választógombot a **Standard** eszköztár **TRadioButton**  (rádiógomb) gombjára duplán kattintva helyezhetünk el a formon.

Ha a formon több választógomb van, ezek közül csak egy lehet kiválasztva. Ha a választógomb ki van választva, akkor annak alakja  lesz.



6.29. ábra. Form gombbal és három választógombbal

A választógomb mindazokkal a tulajdonságokkal rendelkezik, mint a jelölőnégyzet, többek között a **Checked** tulajdonsággal is, amit az elágazás logikai kifejezésében használhatunk.

Figyeljük meg, hogy változtathatjuk meg az ablak egy tulajdonságát választógomb alkalmazásával!

Helyezzünk a formra egy gombot és három választógombot (6.29. ábra)!

A választógombok beillesztésekor azok **Checked** tulajdonsága alapértelmezetten **false**. A projekt futása előtt változtassuk meg az első választógomb **Checked** tulajdonságát **true**-ra. Ekkor a form külalakja pontosan olyan lesz, ahogy azt megterveztük, ilyen lesz a futtatás kezdetén is.

Hozzuk létre a gomb **Click** eseménykezelőjét, és helyezzük el benne a következő utasításokat:

```

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

```

Az utasítás végrehajtása a **RadioButton1.Checked** kiértékelésével kezdődik.

Ha a **Click** esemény bekövetkezésekor az első választógomb volt bekapcsolva, akkor **RadioButton1.Checked = true**. Ezért az ablak címe lesz megváltoztatva, az elágazás egyéb utasításai pedig kimaradnak.

Ha a **Click** esemény bekövetkezésekor a második választógomb volt bekapcsolva, akkor **RadioButton1.Checked = false**. Ezért az **Else** utáni utasítás kerül végrehajtásra. Ez az utasítás szintén egy elágazás. A végrehajtása során a **RadioButton2.Checked** kerül kiértékelésre. Mivel most a második gomb volt bekapcsolva, ezért **RadioButton2.Checked = true**. Ezért az ablak háttérszíne lesz megváltoztatva, az elágazás többi utasítása kimarad.



Ha a **Click** bekövetkezésekor a harmadik gomb volt bekapcsolva, akkor **RadioButton1.Checked = false**, ezért az **Else** utáni utasításra kerül a vezérlés, vagyis a **RadioButton2.Checked** kerül kiértékelésre. Mivel a második gomb sem volt bekapcsolva, értéke ennek is **false**. Ezért a második **Else** utáni utasítás kerül végrehajtásra, vagyis az ablak méretének megváltoztatása.

Ha a projekt indítása előtt egyik gomb **Checked** tulajdonságát se változtattuk meg, akkor a projekt indításakor egyik se lesz kijelölve. Ha a projekt futása során sem kapcsoljuk be egyiket se, és lenyomjuk a gombot, akkor az ablak mérete változik meg (gondold át önállóan, miért történik ez!). Hogy ez ne történjen így, mindenképpen be kell kapcsolni futtatás előtt az egyik választógombot, vagy a második **Else** után el kell helyezni még egy elágazást:


```

If RadioButton1.Checked
Then Form1.Caption := 'A cím megváltozott'
Else If RadioButton2.Checked
Then Form1.Color := clRed
Else If RadioButton3.Checked
Then begin
    Form1.Width := 600;
    Form1.Height := 300;
end;


```

! Összefoglalás

A jelölőnégyzet **Checked** tulajdonságának értékeit jól lehet használni az elágazás feltételeként. Ennek értéke **true**, ha a jelölő be volt kapcsolva és **false**, amennyiben nem.

Jelölőnégyzetet a **Standard** eszköztár **TCheckbox** (ellenőrző doboz)  gombjára duplán kattintva helyezhetünk el a formon. Ha a formon több jelölőnégyzet van, akkor be lehet kapcsolva valamennyi, néhány, esetleg egyik sem. Ezért a jelölőnégyzeteket akkor használjuk, ha a projekt futása során a lehetőségek egyikét szeretnénk kiválasztani, esetleg többet vagy egyet sem.

A jelölőnégyzetek állapotának (ki van jelölve vagy nincs) feldolgozására a projektbe független elágazásokat illesztünk be.

Ha a projekt futása során a lehetőségek egyikét szeretnénk csak kiválasztani, **kombinált listát** alkalmazhatunk. Kombinált listát a **Standard** eszköztár **TCombobox**  (kombinált lista) gombjára duplán kattintva helyezhetünk el a formon.

? Felelj a kérdésekre!

- 1• Milyen célra használhatjuk a jelölőnégyzetet?
- 2• Milyen értékeket vehet fel a jelölőnégyzet **Checked** tulajdonsága?
- 3• Mire szolgál a kombinált lista?
- 4• Hogyan kell leírni a jelölőnégyzet **Checked** tulajdonságát ellenőrző utasítást?
- 5• Hogyan kell elhelyezni a jelölőnégyzetek **Checked** tulajdonságát ellenőrző tulajdonságokat?



6. fejezet

- 6*. Miben különbözik a jelölőnégyzet alkalmazása az elágazásokban a kombinált lista alkalmazásától?
- 7*. Mire szolgálnak a választógombok?
- 8*. Milyen értékeket vehet fel a jelölőnégyzet **Checked** tulajdonsága?



Végezd el a feladatokat!

- 1• Rajzold meg az alábbi utasítások folyamatábráját, és magyarázd el a működését:

```
If CheckBox1.Checked
```

```
Then begin
```

```
<1. utasítássor>
```

```
end;
```

```
If CheckBox2.Checked
```

```
Then begin
```

```
<2. utasítássor>
```

```
end;
```



- 2• Rajzold meg az alábbi utasítások folyamatábráját, és magyarázd el a működését:

```
If ComboBox1.Text = '<Tekst 1>'
```

```
Then begin
```

```
<1. utasítássor>
```

```
end
```

```
Else If ComboBox1.Text = '<Tekst 2>'
```

```
Then begin
```

```
<2. utasítássor>
```

```
end
```

```
Else If ComboBox1.Text = '<Tekst 3>'
```

```
Then begin
```

```
<3. utasítássor>
```

```
end;
```

- 3*• Rajzold meg az alábbi utasítások folyamatábráját, és magyarázd el a működését:

```
If RadioButton1.Checked
```

```
Then begin
```

```
<1. utasítássor>
```

```
end
```

```
Else If RadioButton2.Checked
```

```
Then begin
```

```
<2. utasítássor>
```

```
end
```

```
Else begin
```

```
<3. utasítássor>
```

```
end;
```



- 4*. Magyarázd meg, miben különbözik az 1. feladatban szereplő elágazás működése a 2. feladatban leírt elágazás működésétől!
- 5•. Hozz létre projektet, amiben két valós szám összegét vagy különbségét, vagy szorzatát, vagy hányadosát számítod ki egy lista kiválasztott elemének megfelelően! Mentd a projektet a mappádban létrehozott **feladat 6.5.5** mappába!
- 6•. Hozz létre projektet, amiben két valós szám összegét vagy különbségét, vagy szorzatát, vagy hányadosát számítod ki jelölőnégszögek állapotának megfelelően! Mentd a projektet a mappádban létrehozott **feladat 6.5.6** mappába!
- 7•. A munkás alkatrészeket készít. Ismerjük az egy hónap alatt elkészített alkatrészek számát és az alkatrész árát. Amennyiben a munkás teljesíti a tervet, 10% jutalmat kap, ha túlteljesíti, akkor 30%-ot, amennyiben nem teljesíti, 20%-ot levonnak a béréből. Hozz létre projektet, amely meghatározza a munkás havi fizetését! Mentd a projektet a mappádban létrehozott **feladat 6.5.7** mappába!
- 8•. A munkásnak megállapított fix havibére van. Ha a munkásnak van gyereke, akkor 20% kiegészítést kap a béréhez. Ha a hónap során a munkás megsértette a munkarendet, akkor a fizetéséből 15%-ot levonnak. Hozz létre projektet, amely meghatározza a munkás havi fizetését! Mentd a projektet a mappádban létrehozott **feladat 6.5.8** mappába!



6.6. SZÁMLÁLÓS CIKLUS



1. Milyen folyamatot nevezünk számlálós ciklusnak? Hozz fel példákat!
2. Mi a ciklus egy algoritmusban? Mi a ciklusmag?
3. Mi az **Ismételd N-szer** ciklus általános alakja? Hogy történik a végrehajtása?

SZÁMLÁLÓS CIKLUS OBJECT PASCALBAN

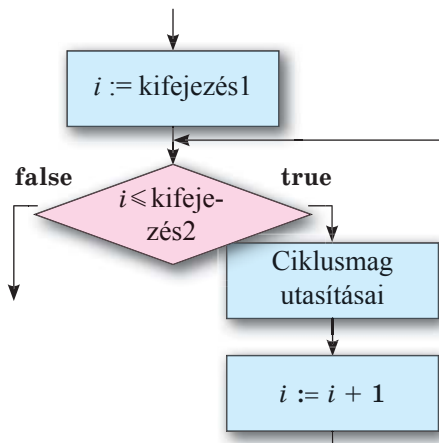
Az **Object Pascal**ban többféle ciklusokat megvalósító utasítás létezik. Ezek egyike a **számlálós ciklus**. Ezek alkalmazása akkor célszerű, ha az ismétlések száma már a ciklus végrehajtása előtt ismert. A számlálós ciklus általános alakja a következő:

```

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

A **for** <változónév> := <kifejezés1> **to** <kifejezés2> **do** (*for* – valami részére, *to* – valameddig, *do* – tenni) sor a **számlálós ciklus feje**. A ciklusfejben szereplő változó (i) a **ciklusváltozó**.

A számlálós ciklus folyamatábráját a 6.30. ábrán láthatjuk.



6.30. ábra. A számlálós ciklus folyamatábrája



6. fejezet

A **ciklusszámláló** és a **két kifejezés** csak egész értékeket vehetnek fel. Ha a ciklusmag csak egyetlen utasítást tartalmaz, a **begin** és **end** utasítás-zárójelek elhagyhatók.

A számlálós ciklus a következőképpen működik:

1. Az **i** ciklusváltozó felveszi a **kifejezés1** értéket.
2. Kiértékelődik az **i <= kifejezés2** logikai kifejezés.
3. Ha a kifejezés értéke **true**, végre kell hajtani a ciklusmagot, majd át kell lépni a 4. utasításra, ha **false**, a vezérlést át kell adni a ciklus utáni első utasításnak.
4. A ciklusváltozót meg kell növelni eggyel.
5. Át kell lépni a 2. utasításra.

A ciklus működésének bemutatására lássuk az $1^2 + 2^2 + 3^2 + 4^2$ összeg kiszámításának algoritmusát.

```

var s, i: integer;
begin
  s := 0;
  for i := 1 to 4 do
    begin
      s := s + i*i;
    end;
  Label1.Caption := IntToStr (s);
end;

```

Hajtsuk végre ezt a programrészletet:

<i>Utasítás</i>	<i>A végrehajtás eredménye</i>
<code>s := 0</code>	<code>s = 0</code>
<code>i := 1</code>	<code>i = 1</code>
<code>i <= 4</code>	<code>(1 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 0 + 1*1 = 1</code>
<code>i := i + 1</code>	<code>i = 2</code>
<code>i <= 4</code>	<code>(2 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 1 + 2*2 = 5</code>
<code>i := i + 1</code>	<code>i = 3</code>
<code>i <= 4</code>	<code>(3 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 5 + 3*3 = 14</code>
<code>i := i + 1</code>	<code>i = 4</code>
<code>i <= 4</code>	<code>(4 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 14 + 4*4 = 30</code>
<code>i := i + 1</code>	<code>i = 5</code>
<code>i <= 4</code>	<code>(5 <= 4) = false</code>
<code>Label1.Caption := IntToStr (s)</code>	<code>Label1.Caption = 30</code>



Felhívjuk a figyelmeteket arra, hogy a ciklus lefutása után a ciklusváltozó értéke 1-gyel nagyobb, mint a ciklusfejen a **kifejezés2** értéke. Ezt az értéket szükség esetén fel lehet használni a program következő részében.



Ha többet szeretnél tudni

Van a számlálós ciklusnak olyan változata, amelyben a ciklusváltozó nem növekszik, hanem csökken 1-gyel:

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

(a *downto* – lefelé valameddig)

Ez a ciklus addig fut majd, amíg a ciklusváltozó értéke nagyobb vagy egyenlő a **kifejezés2** értékénél (vagyis amíg kisebb nem lesz annál).



Összefoglalás

A számlálós ciklus általános alakja a következő:

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

A **for <változónév> := <kifejezés1> to <kifejezés2> do** sor a számlálós ciklus feje. A ciklusfejen szereplő változó a **ciklusváltozó**. A **ciklusszámláló** és a **két kifejezés** csak egész értékeket vehetnek fel.



Felelj a kérdésekre!

- 1°. Mi a ciklus az algoritmusban?
- 2°. Milyen a számlálós ciklus általános alakja az **Object Pascalban**?
- 3°. Hogy működik a számlálós ciklus az **Object Pascalban**?
- 4°. Mutasd be a számlálós ciklus folyamatábráját! Ismertesd ennek végrehajtását!
- 5°. Előfordulhat-e, hogy a számlálós ciklus csak egyszer kerüljön végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 6°. Előfordulhat-e, hogy a számlálós ciklus egyszer sem kerül végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 7*. Előfordulhat-e, hogy a számlálós ciklus végtelenszer kerül végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 8*. Mi a különbség a szekvencia, az elágazás és a ciklus között?



Végezd el a feladatokat!

1• Állítsd össze az alábbi programrészletek folyamatábráját, és hajtsd végre azt!

a) $n := 1;$ for $i := 1$ to 4 do $n := n * i;$	b) $p := 0; a := 2;$ for $i := 1$ to 5 do begin $a := 3 * a + 1;$ $p := p + a;$ end;
--	--



2• Állítsd össze az alábbi programrészletek folyamatábráját, és hajtsd végre azt!

a) $n := 0;$ for $i := 1$ to 4 do $n := n + i * i;$	b) $p := 0; a := 4;$ for $i := 1$ to 5 do begin $a := 2 * a - 1;$ $p := p + a;$ end;
--	--

3° Állíts össze algoritmust 5 szám összegének kiszámítására, amelyek közül az első 7 és minden szám 3-mal haladja meg az előzőt! Hajtsd végre az algoritmust!



4° Állíts össze algoritmust 6 szám összegének kiszámítására, amelyek közül az első 100 és minden szám 5-ször kisebb az előzőnél! Hajtsd végre az algoritmust!

5* Állítsd össze a következő feladat megoldásának folyamatábráját! Határozd meg, hány négyzetre lehet feldarabolni a téglalap alakú papírdarabot, ha minden lépésben akkora négyzetet vágunk le belőle, amelynek oldala a megmaradt csík kisebbik oldalával egyenlő! Hajtsd végre az algoritmust két különböző bemenő adattal! Válaszd meg a bemenő adatokat úgy, hogy a ciklusmag többször kerüljön végrehajtásra; egyszer kerüljön végrehajtásra!

6• Állíts össze projektet n szám összegének kiszámítására, amelyek közül az első x és minden szám 5-ször nagyobb az előzőnél! Mentd a projektet a mappában létrehozott **feladat 6.6.6** mappába!



7• Állíts össze projektet n szám összegének kiszámítására, amelyek közül az első x és minden szám az előző négyzete! Mentd a projektet a mappában létrehozott **feladat 6.6.7** mappába!

8• Hozz létre projektet a következő feladat megoldására: Az ügyfél bizonyos összeget helyezett el a bankban. A bank 20% kamatot fizet az előző év végén a számlán lévő összeg után. Határozd meg, mekkora összeg lesz a számlán n év múlva! Mentd a projektet a mappában létrehozott **feladat 6.6.8** mappába!



9• Hozz létre projektet a következő feladat megoldására: Az ügyfél bizonyos összeget helyezett el a bankban. A bank meghatározott kamatot fizet az előző év végén a számlán lévő összeg után. Határozd meg, mekkora összeg lesz a számlán n év múlva! Mentd a projektet a mappában létrehozott **feladat 6.6.9** mappába!



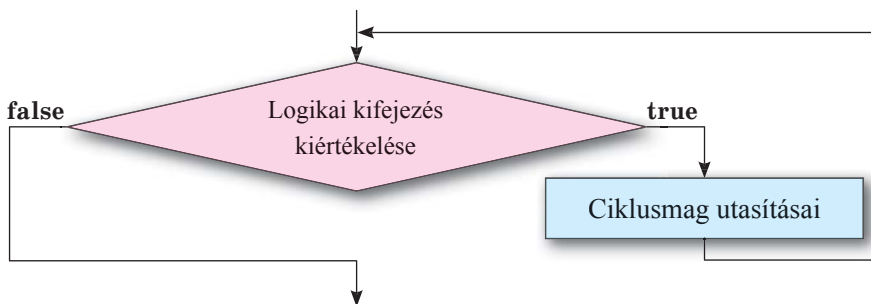
6.7. ELŐLTESZTELŐ CIKLUS



1. Mi a számlálós ciklus általános alakja, és hogy működik a számlálós ciklus **Object Pascalban**?
2. Milyen az előltesztelő ciklus folyamatábrája, és hogy működik?
3. Az algoritmusokban mikor alkalmazunk számlálós ciklust, és mikor előltesztelőt?

A CIKLUS ELŐFELTÉTELE

Már tudjátok, hogy az előltesztelő ciklus, mint az algoritmus része a feltétel ellenőrzésével kezdődik, aminek eredménye **igaz (true)** vagy **hamis (false)** lehet. Az ellenőrzés eredményétől függően vagy a ciklusmag kerül végrehajtásra, vagy a ciklus utáni első utasítás.



6.31. ábra. Az előltesztelő ciklus folyamatábrája

Általános esetben tehát a ciklus egy logikai kifejezés kiértékelésével kezdődik, ami **true** vagy **false** értéket vehet fel.

Az előltesztelő ciklus általános folyamatábráját a 6.31. ábrán láthatjuk.

ELŐLTESZTELŐ CIKLUS OBJECT PASCALBAN

Object Pascalban a ciklusutasítás általános alakja a következő:

```
While<logikai kifejezés>
Then begin
  <ciklusmag>
end;
```

(*while* –amíg).

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, végrehajtásra kerülnek a ciklusmag utasításai, ezután újra kiértékelődik a logikai kifejezés, és ha az eredmény újra **true**, újból végrehajtásra kerülnek a ciklusmag utasításai; ha a kiértékelés eredménye **false**, a **ciklusmag** nem kerül végrehajtásra, a vezérlés a ciklus utáni első utasításra kerül.



6. fejezet

Itt hívjuk fel a figyelmeteket a következőkre:

- ha a ciklusmagot egyetlen utasítás alkotja, a **begin** és **end** utasítás-zárójelek alkalmazása nem kötelező;
- a ciklusmagok szekvenciákat, elágazásokat és további ciklusokat tartalmazhatnak.

Vizsgáljunk meg egy feladatot, amely megoldásának algoritmusá elöltesztelő ciklust tartalmaz.

Feladat. Hány összeadandót kell összeadni ahhoz, hogy azok összege meghaladja a 100-at, ha az első összeadandó az adott valós szám és minden következő összeadandó 3-mal nagyobb az előzőnél?

A feladatban egyetlen bemenő adat az első összeadandó, amit egy beviteli mezőn keresztül adunk meg.

A ciklus kezdete előtt az összeg meghatározására szolgáló *s* változó értékét egyenlővé tesszük az első összeadandóval. Az összeadandók megszámlálására szolgáló *n* kezdőértéke legyen 1, mivel az összeg egyelőre az első összeadandóval egyenlő.

A számok összeadását addig kell folytatni, amíg az összeg kisebb, vagy egyenlő 100-zal. Ezért a ciklusfejben az $s \leq 100$ logikai kifejezést helyezzük el. A ciklusmagban meg kell határozni a következő összeadandót, azt hozzáadjuk az összeghez, az összeadandók számlálóját meg kell növelni 1-gyel.

A megfelelő projektrészlet a következő:

```

var a, s: real; n: integer;
begin
  a := StrToFloat (Edit1.Text);
  s := a; n := 1;
  while s <= 100 do
    begin
      a := a + 3;
      s := s + a;
      n := n + 1;
    end;
  Label1.Caption := IntToStr(n);
end;

```



Összefoglalás

Az elöltesztelő ciklus általános alakjának folyamatábráját a 6.31. ábra mutatja.

Object Pascalban a ciklusutasítás általános alakja a következő:

```

While <logikai kifejezés>
Then begin
  <ciklusmag>
end;

```

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, végrehajtásra kerülnek a ciklusmag utasításai, ezután újra kiértékelődik a logikai kifejezés, és ha az eredmény újra **true**, újból végrehajtásra kerülnek a ciklusmag utasításai; ha a kiértékelés eredménye **false**, a **ciklusmag** nem kerül végrehajtásra, a vezérlés a ciklus utáni első utasításra kerül.



Felelj a kérdésekre!

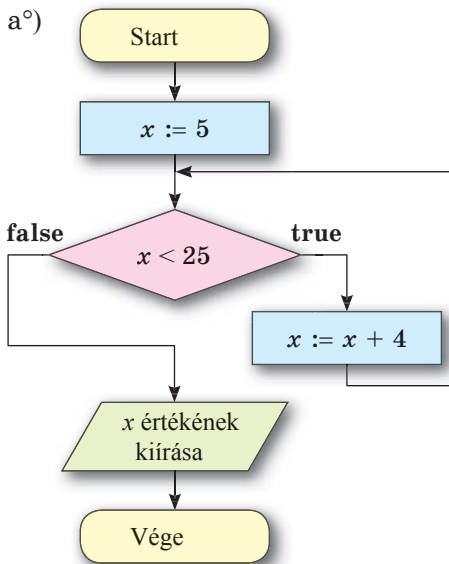
1. Milyen az előtesztelt ciklus általános alakjának folyamatábrája? Ismertesd a ciklus végrehajtását!
2. Előfordulhat-e, hogy az előtesztelő ciklus magja egyszer sem kerül végrehajtásra? Magyarázd meg a válaszodat! Hozz fel példát!
- 3*. Előfordulhat-e, hogy az előtesztelő ciklus a végtelenségig fut? Magyarázd meg a válaszodat! Hozz fel példát!
- 4*. Miben különböznek egymástól az előtesztelő és a számlálós ciklus?
- 5*. Miben különbözik egymástól a következő algoritmikus szerkezetek végrehajtása: szekvencia, számlálós ciklus, előtesztelő ciklus?
6. Milyen az előtesztelő ciklus általános alakja **Object Pascalban**?
7. Hogy történik az előtesztelt ciklus végrehajtása **Object Pascalban**?



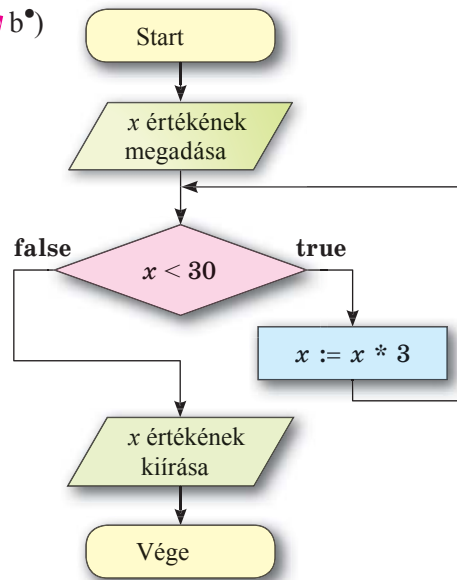
Végezd el a feladatokat!

1. Hajtsd végre az alábbi algoritmusokat!

a°)



b°)



2. Hajtsd végre a programrészletet, és állapítsd meg, milyenek lesznek a változók értékei a végrehajtás után:

```

a) k := 1; a := 12;
   while a < 100 do
   begin
     a := 2*a - 4;
     k := k + 1;
   end;
    
```

```

b) s := 0; a := 1; k := 1;
   while a < 50 do
   begin
     s := s + a;
     k := k + 1;
     a := k*k;
   end;
    
```



3. Hajtsd végre a programrészletet, és állapítsd meg, milyenek lesznek a változók értékei a végrehajtás után:

<pre>a) k := 1; a := 100; while a > 10 do begin a := a/2; k := k + 1; end;</pre>	<pre>b) s := 0; a := 5; while a <= 100 do begin s := s + a; a := a*2; end;</pre>
--	--

4. Állíts össze folyamatábrát azoknak a 100-at meg nem haladó számoknak az összeadására, amelyek közül az első 7, és minden következő 5-tel haladj meg a megelőzőt! Hajtsd végre az algoritmust!



5. Állíts össze algoritmust azoknak a pozitív számoknak az összeadására, amelyek közül az első összeadandó 50, és minden következő 8-cal kisebb az előzőnél!

6. Hozz létre projektet, amely megszámlálja azokat a kétjegyű számokat, amelyek közül az első 10, és minden következő egy adott számmal haladj meg az előzőt! Mentse a projektet a mappában létrehozott **feladat 6.7.6** mappába!



7. Hozz létre projektet, amely meghatározza azoknak a pozitív számoknak az átlagát, amelyek közül az első 100, és minden következő egy adott számmal kevesebb az előzőnél! Mentse a projektet a mappában létrehozott **feladat 6.7.7** mappába!

8. Hozz létre projektet a következő feladat megoldására: Az ügyfél bizonyos összeget helyezett el a bankban. A bank minden évben meghatározott kamatot fizet az előző évben a számlán lévő összegre. Hány év múlva éri el az ügyfél számláján az összeg az S -t? Mentse a projektet a mappában létrehozott **feladat 6.7.8** mappába!

12. SZÁMÚ GYAKORLATI MUNKA

Elágazásokat és ciklusokat tartalmazó projektek létrehozása

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Indítsd el a **Lazarust**!
2. Hozz létre projektet a következő feladat megoldására: Ismerjük két téglalap oldalait. Határozd meg a téglalapok területeit, és hasonlítsd össze azokat! Határozd meg, négyzetek-e a téglalapok! Határozd meg, elfér-e az egyik téglalap a másikban?
 - 1) Helyezz el a formon feliratokat az adatok kiírására, valamint három gombot!
 - 2) A projektben:
 - állítsd össze az első gomb **Click** eseménykezelőjét, amely meghatározza a téglalapok területeit és összehasonlítja azokat;
 - állítsd össze a második gomb **Click** eseményének kezelőjét, amely a megfelelő jelölőnégyzetek állapotától függően eldönti, hogy az egyik vagy a másik téglalap négyzet-e;



- állítsd össze a harmadik gomb **Click** eseményének kezelőjét, amely a választógombok állapotától függően eldönti, elfér-e az egyik téglalap a másikban, illetve a másik az egyikben!
3. Mentsd a projektet a mappádban létrehozott **gyakorlati 12_1** néven létrehozott mappába!
 4. Hozz létre projektet a következő feladat megoldására: *Áradás előtt a víz szintje a folyóban H méter. Az áradás során a vízszint óránként P százalékkal növekszik. Mennyi lesz a vízszint N óra elteltével? Hány óra múltán éri el a folyó szintje a K métert?*
 - 1) Helyezz el a formon beviteli mezőket, magyarázó címkéket és három gombot!
 - 2) A gombok feliratai legyenek *Feladat1*, *Feladat2*, *Előlről*, a mezők *legyenek üresek!*
 - 3) Készítsd el az első gomb **Click** eseményének kezelőjét, amely elvégzi az első feladat megoldását, és kiírja az eredményt!
 - 4) Futtasd a projektet, és győződj meg annak helyességéről!
 - 5) Készítsd el a második gomb **Click** eseményének kezelőjét, amely elvégzi a második feladat megoldását, és kiírja az eredményt!
 - 6) Futtasd a projektet, és győződj meg annak helyességéről!
 - 7) Állítsd össze a harmadik gomb **Click** eseménykezelőjét, ami törli a második feladatban érintett mezők és címkék tartalmát!
 5. Mentsd a projektet a mappádban létrehozott **gyakorlati 12_2** néven létrehozott mappába!
 6. Zárd be a **Lazarust!**



6.8. GRAFIKAI OBJEKTUMOK MEGJELENÍTÉSE A PROGRAMOZÁSI NYELV ESZKÖZEIVEL



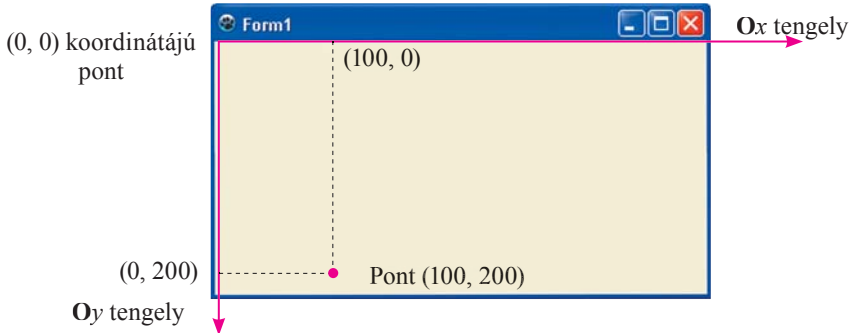
1. Milyen elemeket alkalmazhatunk képszerkesztőben és szövegszerkesztőben rajzolás során?
2. Milyen tulajdonságait ismered a grafikai elemeknek?
3. Milyen tulajdonságai vannak a címkének, gombnak, formának?

RAJZOLÁSI ESZKÖZÖK OBJECT PASCALBAN

Lazarus környezetben különféle eszközök állnak rendelkezésünkre képi elemek megjelenítésére. Az egyik lehetőség a grafikai alapelemek megjelenítése a koordináta-rendszerben.

Ilyen koordináta-rendszer helyezkedik el a formon. A koordináta-rendszer kezdőpontja a form bal felső sarka. Az Ox tengely vízszintesen jobbra, az Oy pedig függőlegesen

lefelé mutat. A form koordináta-rendszerének méretei megegyeznek a form méreteivel. A 6.32. ábrán a (100, 200) koordinátájú pont van kiemelve piros színnel.



6.32. ábra. Koordináta-rendszer a formon

Rajzolás során a form **Canvas** (rajzvászon) tulajdonságát használjuk.

A form minden pontjának vannak koordinátái és van színe. A koordináták csak egész értékeket vehetnek fel. A pont színét a **TColor** típusú állandóval adhatjuk meg. A 6.1. táblázat az állandók néhány lehetséges értékét és azok színeit tartalmazza.

6.1. táblázat

Színkonstansok Lazarusban

Állandó	Szín	Állandó	Szín
<i>clBlack</i>	Fekete	<i>clSilve</i>	Ezüst
<i>clWhite</i>	Fehér	<i>clGray</i>	Szürke
<i>clRed</i>	Piros	<i>clGreen</i>	Zöld
<i>clBlue</i>	Kék	<i>clNavy</i>	Sötétkék
<i>clLime</i>	Zöldcitrom	<i>clOlive</i>	Olíva

Az (x, y) koordinátájú pontot a következő utasítással állíthatjuk pirosra:

```
Canvas.Pixels[x, y] := clRed;
```

A rajzolás során alkalmazható utasításokat a 6.2. táblázatban foglaltuk össze.

6.2. táblázat

Rajzolás során alkalmazható utasítások

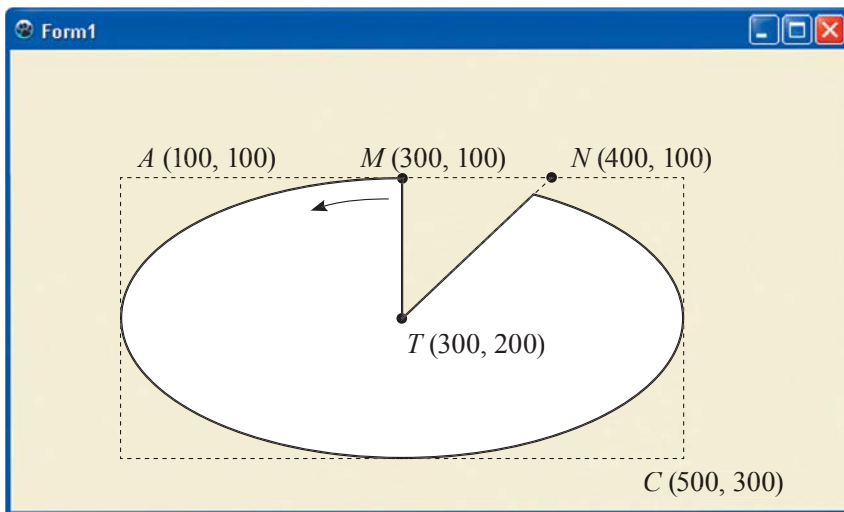
Utasítás	Magyarázat
Canvas.MoveTo (x, y)	A koordináta-rendszer (x, y) kezdőpontjának kijelölése a formom



Utasítás	Magyarázat
Canvas.LineTo (x, y)	Szakasz rajzolása az előző művelet végpontjából az (x, y) koordinátájú pontba. Ilyen utasítások sorozatával szakaszt vagy töröttvonalat lehet rajzolni
Canvas.Rectangle ($x1, y1, x2, y2$)	Olyan téglalap rajzolása, amelynek szemben fekvő csúcsai ($x1, y1$) és ($x2, y2$), oldalai pedig párhuzamosak a tengelyekkel
Canvas.Ellipse ($x1, y1, x2, y2$)	Olyan ellipszis rajzolása, amely pontosan illeszkedik arra a téglalpra, amelynek szemben fekvő csúcsai ($x1, y1$) és ($x2, y2$). A téglalap nem látszik
Canvas.Pie ($x1, y1, x2, y2, x3, y3, x4, y4$)	Ellipsziscikk rajzolása. Az ellipszis pontosan illeszkedik arra a téglalpra, amelynek szemben fekvő csúcsai ($x1, y1$) és ($x2, y2$). Az ív kezdőpontja egybeesik az ellipszisnek és annak az egyenesnek a metszéspontjával, amely áthalad a középponton, valamint az ($x3, y3$) ponton. Végpontja egybeesik az ellipszisnek és annak az egyenesnek a metszéspontjával, amely áthalad a középponton, valamint az ($x4, y4$) ponton. Az ív rajzolását a kezdőpontból az óramutató járásával ellenkező irányban kezdjük és a végpontban fejezzük be

A 6.33. ábrán látható ellipszis az $A(100, 100)$ és $C(500, 300)$ pontokra kifeszített téglalapba van írva, a kezdőpont az ellipszis középpontján ($T(300, 200)$) és az $M(300, 100)$ ponton áthaladó egyenes metszéspontja, a végpont az ellipszis középpontján és az $N(400, 100)$ ponton áthaladó egyenes metszéspontja.

Minden megrajzolható alakzathoz megadhatjuk a kontúrvonal színét, és a bezárt terület kitöltését. Ezek a tulajdonságok megegyeznek a form 6.3. táblázatban összefoglalt tulajdonságaival.



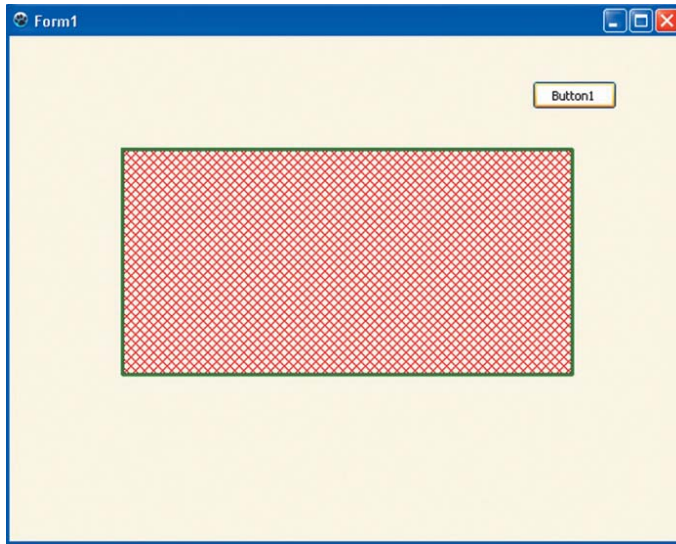
6.3. ábra. A Canvas.Pie(100, 100, 500, 300, 300, 100, 400, 100) utasítással kirajzolt elliptikus cikk



A form vonaltulajdonságokat és kitöltéseket szabályozó tulajdonságai

Tulajdonság	Rendeltetés	Alapértelmezett érték
Canvas.Pen.Color	Vonalszín	Fekete
Canvas.Pen.Width	Vonalvastagság	1 pixel
Canvas.Pen.Style	Vonalstílus: <i>psSolid</i> – folytonos, <i>psDash</i> – szaggatott, <i>psDot</i> – pontozott vonal	Folytonos
Canvas.Brush.Color	Kitöltőszín	Fehér
Canvas.Brush.Style	Kitöltés stílusa: <i>bsSolid</i> – egyszínű, <i>bsCross</i> – függőleges és vízszintes vonalokból alkotott rács, <i>bsDiagCross</i> – átlós vonalak által alkotott rács	Egyszínű

Rajzolás során célszerű előbb beállítani a vonal és a kitöltés tulajdonságait, aztán elkészíteni magát az alakzatot. A 6.34. ábrán a következő eljárással készült színes téglalapot láthatjuk.



6.34. ábra. A rajzolás eredménye

```

procedure TForm1.Button1Click (Sender: TObject);
begin
    Canvas.Pen.Color := clGreen; //zöld vonalszín
    Canvas.Pen.Width := 3; // 3 pixel vonalvastagság
    Canvas.Brush.Color := clRed; //piros kitöltőszín
    Canvas.Brush.Style := bsDiagCross; //átlós háló stílus
    Canvas.Rectangle (100, 100, 500, 300); a (100, 100) és (500, 300) pontokra
    kifestített téglalap megrajzolása
end;


```

Ha azt szeretnénk, hogy a rajz már a projekt indítása alatt elkészüljön, az utasításokat a **Paint** esemény kezelőjében kell elhelyezni.



AZ ALAK (SHAPE) ESZKÖZ HASZNÁLATA A RAJZOLÁS SORÁN

Lazarusban az **Alak (Shape)** eszközt is használhatjuk a rajzolás során.

Az **alak (shape)** eszközt úgy helyezhetjük el a formon, hogy duplán kattintunk az **Additional** (kiegészítő) lap **TShape** (alak, forma)  elemére.

Az **Alak** komponens **Name**, **Enabled**, **Visible** tulajdonságainak értelmezése ugyanaz, ami a többi komponens esetében. A **Top**, **Left**, **Height**, **Width** tulajdonságok értékét ugyanúgy a **Tulajdonságok** lapon változtathatjuk meg, mint a form esetében.

Az **Alak** komponens rendelkezik a vonal és a kitöltés megjelenését meghatározó tulajdonságokkal (6.4. táblázat).

6.4. táblázat

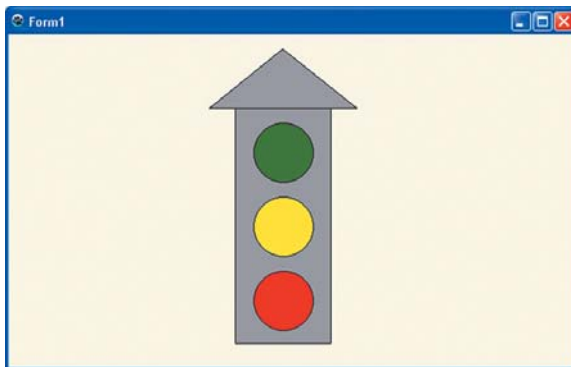
Az Alak komponens tulajdonságai

Tulajdonság	Értelmezés
Pen.Color	Vonalszín
Pen.Width	Vonalvastagság
Pen.Style	Vonalstílus
Brush.Color	Kitöltőszín
Brush.Style	Kitöltőminta

A komponens ezen tulajdonságait beállíthatjuk az **Objektum felügyelő Tulajdonságok** lapján, vagy a megfelelő értékadó utasításokkal. A tulajdonságok lehetséges értékei megegyeznek a rajzeszközök tulajdonságainál megismertekkel.


Az **Alak** rendelkezik egy **Shape** tulajdonsággal, amivel a formáját adhatjuk meg. Lehetséges értékei például **stRectangle** (téglalap), **stCircle** (kör), **stSquare** (négyzet), **stDiamond** (rombusz), **stTriangle** (háromszög). Az alapértelmezett beállítás az 1 pixel vastagságú fekete kontúrvonalú fehér kitöltésű négyzet.

A 6.35. ábrán látható rajzot különbözőképpen beállított **Shape** komponens segítségével készítettük.



6.35. ábra. A **Shape** komponenssel készített rajz

KÜLSŐ KÉPFÁJLOK MEGJELENÍTÉSE

A projektekből lehetőségünk van külső fájlokban tárolt képek megjelenítésére is. E célból egy **Képterületet** kell elhelyezni a formon, amire az **Additional** lap **Kép (TImage)** komponense  biztosít lehetőséget. Elhelyezése a már megszokott dupla kattintással történik.

A **Kép Name, Enabled, Visible** tulajdonságai megegyeznek a form hasonló tulajdonságaival. A **Top, Left, Height, Width** tulajdonságok értékét ugyanúgy a **Tulajdonságok** lapon és mozgatással változtathatjuk meg.

A fájl megjelenítését szabályozó tulajdonságokat a 6.5. táblázatban foglaltuk össze.

6.5. táblázat

A Kép komponens tulajdonságai

Tulajdonság	Értelmezés
Picture	A megjelenítendő fájl kiválasztása
AutoSize	Ha az értéke true , a képterület mérete felveszi a betöltött kép méretét, ha false , akkor a képterület a betöltött képtől függetlenül a megfelelő tulajdonságokkal (Height, Width) megadott
Stretch	Amennyiben az értéke true , a betöltött kép visszaméreteződik oly módon, hogy elférjen a képterületen, egyébként nem tölti azt ki, vagy csak egy részletét láthatjuk
Proportional	Ha az értéke true , akkor a betöltött kép oldalaránya az átméretezés során megmarad, egyébként nem

A 6.36. ábrán egy külső fájlból betöltött képet láthatunk.



6.36. ábra. Külső fájl megjelenítő projekt

Összefoglalás

A form koordináta-rendszerében lehetőségünk van grafikai alapelemek elhelyezésére. Erre a célra a form **Canvas** tulajdonságát használjuk. A pontok koordinátákkal és színnel rendelkeznek. Valamennyi grafikai alapelemhez tartozik bennfoglaló vonal és kitöltés.

Grafikai alapelemeket elhelyezhetünk ezenkívül az **Alak (Shape)** komponensen is. A komponens tulajdonságai határozzák meg az elem vonalstílusát és kitöltését.

Ha a formon elhelyezünk egy **Képterületet (TImage)**, lehetőségünk lesz külső képfájl megjelenítésére is.

Felelj a kérdésekre!

1. Hogy helyezkedik el a formon a koordináta-rendszer?
2. A form mely tulajdonságait használjuk rajzolás során?
3. Milyen utasítások szolgálnak a szakasz, töröttvonal, ellipszis, ellipsziscikk rajzolására?
4. Hogy változtatjuk meg az alakzatok kontúrvonalát és kitöltését?

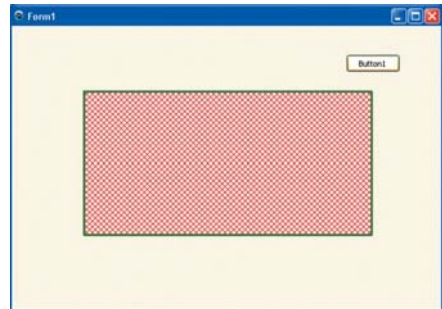


- 5°. Milyen komponens segítségével helyezhetünk el alakzatokat a formon?
- 6°. Az **Alak** komponens mely tulajdonságai határozzák meg a kontúr és a kitöltés kinézetét?
- 7°. Az **Alak** mely tulajdonságai határozzák meg a rajzolt alakzat formáját?
- 8°. Melyik komponens teszi lehetővé azt, hogy a formon külső grafikai fájlt megjelenítsünk?
- 9°. Hogy helyezünk el fájlt a **Képterületen**?



Végezd el a feladatokat!

- 1°. Hozz létre projektet, amely a gomb lenyomása után megjeleníti a képen látható alakzatot! Mentse a projektet a mappában létrehozott **feladat 6.8.1** mappába!
- 2°. Hozz létre projektet, amely az indítás után megjeleníti a következő grafikai elemeket:
 - a) A (10, 10) és a (200, 100) pontokat összekötő 5 pixel széles, piros szaggatott vonalat;

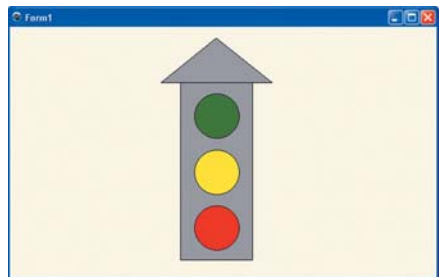


Az 1. feladathoz

- b) négy szakaszból álló 3 pixel vastag, kék, pontozott töröttvonalat;
- c) az (50, 50) és (250, 150) pontokra kifeszített téglalapot, vonala 1 pixel vastag, kék, folytonos, a kitöltés ezüst, rács;
- d) a (100, 20) és (200, 300) pontokra kifeszített téglalapba írt ellipszist, vonala 2 pixeles, salátazöld, szaggatott; kitöltés: rács;
- e) elliptikus cikket, bennfoglaló téglalap csúcsai (50, 50) és (150, 150), ívek végpontjai (150, 100) és (100, 150), vonala 4 pixeles, fehér, folytonos, kitöltés olívozöld, dőlt rács.

Mentse a projektet a mappában létrehozott **feladat 6.8.2** mappába!

- 3°. Hozz létre projektet, amely indítása után elkészíti az ábrán látható rajzot! Mentse a projektet a mappában létrehozott **feladat 6.8.3** mappába!
- 4°. Hozz létre projektet, amely futtatása során megrajzol:
 - a) egy teherautót;
 - b) egy robotot;
 - c) egy számítógépet;
 - d) egy rakétát!



A 3. feladathoz

- A vonalak és kitöltések stílusát alakítsd be-látásod szerint! Mentse a projektet a mappában létrehozott **feladat 6.8.4** mappába!
- 5°. Hozz létre projektet, amely megjeleníti a **Rozdil 6** **punkt 6.8** **zavdannya 6.8.5.jpg** fájlban található képet! Mentse a projektet a mappában létrehozott **feladat 6.8.5** mappába!



- 6*. Hozz létre projektet, amelyben a form háttére a **Rozdil 6\pukt 6.8\zavdannya 6.8.6.jpg** fájl lesz, és erre a háttérre egy autót rajzol ki! Mentsd a projektet a mappában létrehozott **feladat 6.8.6** mappába!
- 7*. Helyezz el a formon gombot, egy alakot (shape), három jelölőnégyzetet és egy le-gördülő menüt! Hozz létre projektet, amely a gomb lenyomásakor a listából ki-választott elemnek megfelelően négyzetet, kört vagy háromszöget rajzol, a jelölő-négyzetek állapotától függően 1 vagy 4 vonalszélességgel, zöld vagy sárga színnel, egyszínű vagy dőlt rácsos kitöltéssel! Mentsd a projektet a mappában létrehozott **feladat 6.8.7** mappába!

13. SZÁMÚ GYAKORLATI MUNKA

Adatok grafikai megjelenítését tartalmazó algoritmusok összeállítása és végrehajtása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Nyisd meg a **Lazarust!**
2. Hozz létre projektet, amelynek végrehajtásakor:
 - a) a form felső részében grafikai alapelemekből kirajzolsz egy napot;
 - b) a form alsó részében megjeleníti a **Rozdil 5\pukt 6.8\prakticsna 13.jpg** képet;
 - c) az alak eszköz segítségével megjelenít egy házat!
3. Mentsd a projektet a mappában létrehozott **gyakorlati 13** mappába!
4. Futtasd a projektet!
5. Elemezd a futás eredményét!
6. Állítsd le a projektet!
7. Zárd be a **Lazarust!**

Ebben a fejezetben megismerkedtek:

az abszolút,
relatív és vegyes
cellahivatkozásokkal

különböző grafikonok és
diagramok létrehozásával
és beállításával

az elektronikus táblázat
oldalbeállításával és
nyomtatásával

a feltételes
formázással

az adatok
rendezésével

a táblázatkezelő
matematikai, statisztikai
és logikai függvényeinek
alkalmazásával

az automatikus és
irányított szűréssel
és a részösszegekkel





7.1. ABSZOLÚT, RELATÍV ÉS VEGYES CELLAHIVATKOZÁSOK. OLDALBEÁLLÍTÁS ÉS NYOMTATÁS ELEKTRONIKUS TÁBLÁZATOKBAN



1. Hogy írjuk le a cellahivatkozást a táblázatkezelőben? Mire szolgál ez?
2. Mit értünk a képlet módosulása alatt? Mikor történik ez? Mik a szabályai?
3. Milyen tulajdonságai vannak a szöveges dokumentumban az oldalnak? Milyen értékeket vehetnek ezek fel? Hogy állítjuk be ezen tulajdonságok értékeit?

RELATÍV, ABSZOLÚT ÉS VEGYES CELLAHIVATKOZÁSOK

A 7. osztályban már tanultátok, hogy az elektronikus táblázat képleteiben cellákra vagy cellatartományokra hivatkozhatunk. A cellahivatkozásokat tartalmazó képletek értéke a cellák változásakor automatikusan újraszámolódnak. Azt is tudjátok, hogy a képletek másolásakor a képletek módosulnak. Előfordulhat azonban olyan eset is, amikor a képletek módosulása nem kívánatos.

Azokat a cellahivatkozásokat, amelyek a másolás során módosulnak, *relatív* cellahivatkozásoknak nevezzük.

Azokat a cellahivatkozásokat, amelyek a másolás során nem módosulnak, *abszolút* cellahivatkozásoknak nevezzük.

Ahhoz, hogy egy relatív hivatkozásból abszolút legyen, a sor száma és az oszlop neve elé a \$ jelet kell tenni.

A **B10** cellahivatkozás tehát relatív, a **\$B\$10** pedig abszolút.

Lássunk egy példát, amely megmutatja az abszolút cellahivatkozások alkalmazásának célszerűségét.

A **C3:C7** cellatartomány (7.1. ábra) azt tartalmazza, hány kitűnő diák tanul az iskola 8–10. osztályaiban. A **C8** cellában ezen számok összegét találjuk. A **D3:D7** cellákban meg kell határozni, hogy az adott osztályban az iskola kitűnő diákjainak hány százaléka tanul.

	A	B	C	D	E
1					
2			Відмінники	Відсоток	
3	8 А		3	18,75	
4	8 Б		1	6,25	
5	9 А		5	31,25	
6	9 Б		4	25,00	
7	10		3	18,75	
8	Усього		16		

7.1. ábra. Relatív és abszolút cellahivatkozások alkalmazása

A százalékok kiszámítása céljából a **D3:D7** tartomány celláiba a következő képleteket kell beírni:

$$D3 = C3/C8*100$$

$$D4 = C4/C8*100$$

$$D5 = C5/C8*100$$

$$D6 = C6/C8*100$$

$$D7 = C7/C8*100$$



Ha megfigyeljük ezeket a képleteket, arra a következtetésre jutunk, hogy a nevezőkben a hivatkozás nem változik, csak a számlálóban. Ha beíránk a **D3**-ba a $=C3/C8*100$ képletet, majd azokat lemásolnánk a **D4:D7** tartományba, akkor a nevező is módosulni fog. Arra volna szükség, hogy a **C3** hivatkozás módosuljon, a **C8** pedig ne. Vagyis a **D3** cellába az $=C3/SC\$8*100$ képletet kell beírni.

Ha az \$ jelet csak az oszlop neve vagy a sor száma elé írjuk be – például **\$B10** vagy **B\$10** –, akkor a képlet másolásakor részlegesen módosul: vagy csak az oszlop neve, vagy csak a sor száma fog megváltozni. Az ilyen cellahivatkozást **vegyes** cellahivatkozásnak nevezzük.

Azokat a cellahivatkozásokat, amelyekben másolás során csak az oszlop neve, vagy a sor száma változik meg, vegyes cellahivatkozásoknak nevezzük.

A cellahivatkozást megváltoztathatjuk a beírás során, de később is szerkeszthetjük, sőt, ehhez elég az **F4** billentyűt leütni. Ebben az esetben a hivatkozás típusa ciklikusan változik relatív – abszolút – vegyes sorrendben.

Ha a képletben nem hivatkozást használunk, hanem annak a nevét, az nem változik meg a másolás során. Vagyis a **cella neve lényegében abszolút cellahivatkozás**. Vagyis ha a vizsgált esetben a **C8** cellát elnevezzük **Jeleseknek**, a **D3**-ba a következő képletet is beírhatjuk: $=C3/jelesek*100$.

A 7.1. táblázatban összefoglaltuk a cellahivatkozásokra vonatkozó tudnivalókat.

7.1. táblázat

Hivatkozástípusok

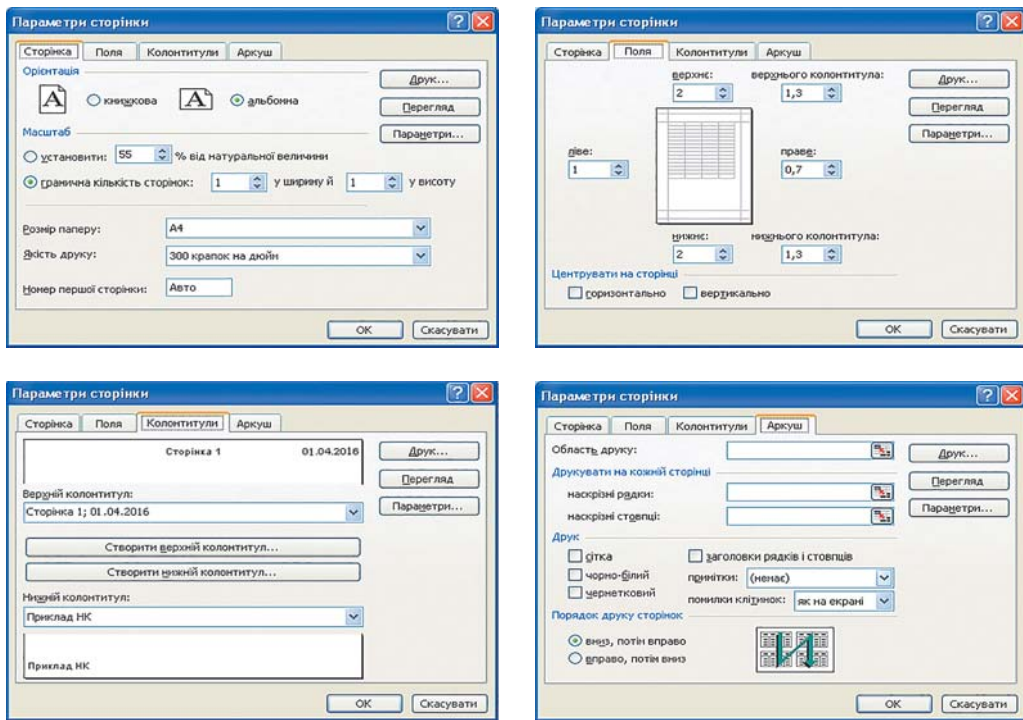
Hivatkozás	Típus	Magyarázat
C3	Relatív	Másolás során az oszlop neve és a sor száma is változik
\$A\$1	Abszolút	Másolásakor sem az oszlop neve, sem a sor száma nem változik
kyc	Abszolút	Másolásakor sem az oszlop neve, sem a sor száma nem változik
C\$1	Vegyes	Másolásakor az oszlop neve megváltozik, a sor száma nem
\$A3	Vegyes	Másolásakor az oszlop neve nem változik, a sor száma igen

OLDALBEÁLLÍTÁS A TÁBLÁZATKEZELŐBEN

Már nyomtatottak szöveges dokumentumot, tudjátok, hogy nyomtatás előtt el kell végezni az oldalbeállítást (papírméret, tájolás, margók stb.), aztán **Nyomtatási kép** nézetben meg kell győződni a beállítások helyességéről, majd csak ezután érdemes elvégezni a nyomtatást.

Az eljárás a táblázatkezelőben is hasonló, bár az **Excel 2007** és a **Word 2007** között vannak különbségek.

Az oldalbeállítást **Lap elrendezése** lap **Oldalbeállítás** csoportjának eszközeivel, vagy az **Oldalbeállítás** párbeszédablakban végezzük (7.2. ábra).



7.2. ábra. Az Oldalbeállítás ablak lapjai

Az **Oldal** lapon a következő tulajdonságok beállítását végezhetjük:

- a lap tájolása;
 - a táblázat nyomtatandó részének méretaránya;
 - papírméret;
 - nyomtatás minősége;
 - az első lap oldalszáma
- stb.

A **Margók** fülön beállíthatjuk a margók méreteit, az élőfej és élőláb méretét, a táblázat középre igazítását a lapon.

Az **Élőfej és élőláb** lapon a megfelelő gombra kattintva kiválaszthatjuk, szeretnénk-e **Élőfejet** vagy **Élőlábat** elhelyezni a lapon és megadhatjuk annak tartalmát.

A **Lap** fülön a következő tulajdonságokat állíthatjuk be:

- a nyomtatási területet, vagyis a nyomtatandó cellatartományt;
 - a nyomtatáskor fent ismétlődő sorokat és bal oldalon ismétlődő oszlopokat, ha vannak ilyenek;
 - a nyomtatás sorrendjét – le, majd jobbra, vagy előbb jobbra, aztán lefelé
- stb.



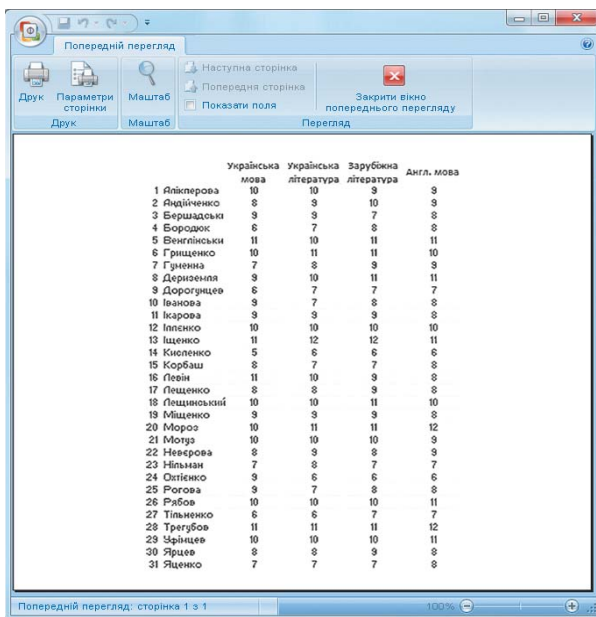
AZ ELEKTRONIKUS TÁBLÁZAT NYOMTATÁSA

Ahhoz, hogy megtekintsük, hogy mutat majd a táblázat a papíron, az **Office** ⇒ **Nyomtatás** ⇒ **Nyomtatási kép** paranccsal át kell lépni **Nyomtatási kép** nézetbe (7.3. ábra), vagy **Lap elrendezése** nézetben (7.4. ábra) kell megtekinteni a táblázatot, amit a **Státuszsor Lap elrendezése** gombjára kattintva tehetünk meg.

Nyomtatási kép nézetben a táblázatot oldalakra osztva tekinthetjük meg. Ha a nyomtatási tartomány nem fér el egy oldalon, az **Excel 2007** automatikusan oldalakra osztja (ilyen helyzetet láthatunk a 7.4. ábrán).

Nyomtatási kép nézetben az oldalakat egyenként tekinthetjük meg, míg **Lap elrendezése** módban egyszerre több, szaggatott vonallal elválasztott oldalt. A szaggatott vonalak mozgatásával meg lehet változtatnia a táblázat oldalakra bontását.

A **Nyomtatási kép** nézet bezárása után a táblázatot függőleges és vízszintes szaggatott vonalak jelennek meg, amelyek a táblázat oldalakra bontását mutatják.

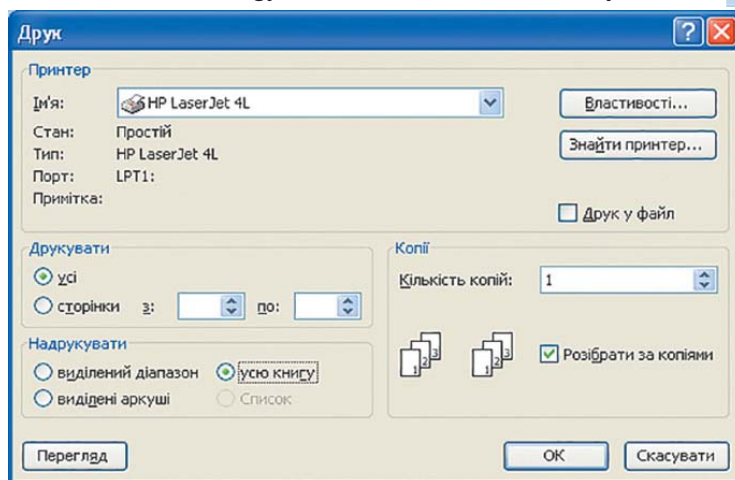


7.3. ábra. Nyomtatási kép nézet

	Українська мова	Українська література	Зарубіжна література	Англ. мова	Історія України	Всєєвітня історія	Алгебр	Геометрія	Фізика	Хїмія	Інформатик	Бїологїя	Економїка	Фїзїка культура
1														
2														
3														
4														
5	1 Анкієрова	10	10	9	9	9	8	7	8	6	7	9	7	11
6	2 Андїєченко	8	9	10	9	10	10	11	11	9	8	10	11	7
7	3 Бершадські	9	9	7	8	11	11	5	6	5	6	6	6	7
8	4 Борошок	6	7	8	8	8	9	8	10	10	11	11	10	10
9	5 Венгліньські	11	10	11	11	10	10	9	9	9	8	11	11	11
10	6 Грищенко	10	11	11	10	8	8	9	8	7	7	9	12	12
11	7 Гуєнна	7	8	9	9	9	9	7	7	7	6	8	9	10
12	8 Деризенля	9	10	11	11	11	11	9	9	8	8	9	11	10
13	9 Дорогунцев	6	7	7	7	8	8	7	7	8	6	7	8	9
14	10 Іванова	9	7	8	8	9	9	7	7	7	7	9	9	8
15	11 Ісарова	9	9	9	8	7	7	7	8	9	7	7	9	9
16	12 Ілпєско	10	10	10	10	10	11	8	9	8	9	8	8	9
17	13 Іщенко	10	11	12	11	10	10	11	11	11	11	12	12	12
18	14 Кисєлєно	5	6	6	6	7	7	5	5	5	6	6	7	7
19	15 Корбаш	8	7	7	8	8	8	8	8	7	6	7	8	9
20	16 Лєвін	11	10	9	8	8	8	9	9	9	10	10	11	11
21	17 Лещєнєко	8	8	9	8	9	8	8	8	8	9	9	9	10
22	18 Лещєньєкий	10	10	11	10	9	9	9	8	9	8	7	8	10
23	19 Мищенко	9	9	9	8	10	10	7	7	8	9	9	11	11
24	20 Мороз	10	11	11	12	12	12	11	11	11	10	9	11	11
25	21 Мотуза	10	10	10	9	9	8	8	9	9	10	10	10	11
26	22 Неєєрова	8	9	9	9	8	9	9	9	9	8	10	10	11
27	23 Нїєман	7	8	7	7	8	8	9	8	8	8	9	9	9
28	24 Оптєєно	9	6	6	6	7	8	9	8	10	10	10	11	11
29	25 Рогова	9	7	8	8	9	10	10	11	11	10	9	9	10
30	26 Рябов	10	10	10	11	11	11	9	9	9	11	10	10	10
31	27 Тїєньєко	6	6	7	7	8	8	8	9	10	10	11	11	9
32	28 Трегубов	11	11	11	12	12	12	10	10	10	11	11	10	10
33	29 Уряєєє	10	10	10	11	11	9	9	9	8	10	10	9	9
34	30 Уряєєє	8	8	9	8	9	8	9	8	9	7	7	9	9
35	31 Уряєєє	7	7	7	8	8	7	7	6	6	5	5	8	8

7.4. ábra. Lap elrendezése mód

Ha az **Excel 2007** gyorselérésű eszköztárán van **Nyomatás**  gomb, akkor arra katintva kezdeményezhetjük a dokumentum nyomtatását az alapértelmezett nyomtatóra az alapértelmezett beállításokkal. Ha nem lenne ilyen gomb, akkor az **Office** ⇒ **Nyomatás** ⇒ **Nyomatás** parancssal kezdeményezhetjük ezt.



7.5. ábra. Nyomatás ablak

parancsra megnyíló **Nyomatás** ablakban (7.5. ábra) megváltoztathatjuk ezeket: megváltoztathatjuk a nyomtatási területet, kiválaszthatjuk a nyomtatót, megadhatjuk a másolatok számát stb. Beállíthatjuk, hogy az egész dokumentumot, vagy annak meghatározott oldalait szeretnénk kinyomtatni.


Összefoglalás


A képletekben többféle hivatkozást használhatunk: **relatívát** (módosul másolás során), **abszolútát** (nem módosul a másolás során) és **vegyeset** (részben módosul a másolás során).

A cellahivatkozásból úgy lesz abszolút, hogy az oszlop neve vagy a sor száma elé \$ jelet írunk. A vegyes hivatkozásban csak az oszlop neve vagy a sor száma elé teszünk \$ jelet.

Ha a képletben egy cellára annak nevével hivatkozunk, az a másolás során nem módosul, vagyis abszolút hivatkozásnak minősül.

Az elektronikus táblázat oldalának paramétereit a **Menüszalag Lap elrendezése** lapján **Oldalbeállítás** csoportjának, vagy az **Oldalbeállítás** párbeszédablaknak az eszközeivel végezzük.

A **Nyomatási kép** nézetben, vagy a **Lap elrendezése** nézetben megtekinthetjük, milyen lesz a dokumentum külalakja papíron. Ezeket az üzemmódokat az **Office** ⇒ **Nyomatás** ⇒ **Nyomatási kép**, vagy a **Státuszsor Lap elrendezése**  gombjával kapcsoljuk be.

A dokumentum nyomtatását a gyorselérésű eszköztár **Nyomatás**  gombjával vagy az **Office** ⇒ **Nyomatás** ⇒ **Nyomatás** parancssal kezdeményezzük.

**Felelj a kérdésekre!**

- 1°. Milyen cellahivatkozás-típusokat használhatunk az elektronikus táblázat képleteiben?
- 2°. Hogy írjuk le a **B2**-re mutató relatív, abszolút vagy vegyes cellahivatkozást?
- 3°. Miben különbözik a különböző hivatkozások használata?
- 4°. Milyen típusú hivatkozás a cella neve?
- 5°. Milyen tulajdonságai vannak az oldaldnak az elektronikus táblázatban?
- 6°. Hogy végezzük az oldalbeállítást az elektronikus táblázatban?
- 7°. Mire szolgál a nyomtatási kép nézet? Milyen lehetőségeket biztosít?
- 8°. Hogy tekinthetjük meg, hogyan lesz oldalakra bontva egy elektronikus táblázat?
- 9°. Hogy végezzük a dokumentum gyorsnyomtatását?
- 10°. Hogy végezzük az elektronikus táblázat nyomtatását?

**Végezd el a feladatokat!**

1. Az **E3** cellában az $=\text{BS\$3+C4}$ képlet szerepel. Hogy módosul a képlet, ha azt a következő cellákba másoljuk:

a) E7 ;	c) C3 ;	e) K15 ;	g) C3 ;
b) E7 ;	d) E11 ;	f) T34 ;	h) A2 ?
2. Az **E3** cellában az $=\text{BS\$3+C4}$ képlet szerepel. Hogy módosul a képlet, ha azt a következő cellákba másoljuk:

a) C9 ;	c) B4 ;	e) D8 ;	g) I2 ;
b) K7 ;	d) E9 ;	f) T23 ;	h) F5 ?
3. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pункт 7.1\zrazok 7.1.3.xlsx-et**)!
 - a*) A **Munka1** lap **F3** cellájába írd képletet, amely meghatározza az éves bevételt, majd másold ezt az **F4:F9** tartományba! Milyen hivatkozásokat kell tartalmazzon ez a képlet: abszolút, relatív vagy vegyeset? A válaszod indokold meg!
 - b*) A **Munka2** munkalap **E4** cellájába írd képletet, amely meghatározza az áruk árát hrvnyában (az árfolyamot a **B3** cellában találod), majd másold ezt az **E5:E12** tartományba! Milyen hivatkozásokat kell tartalmazzon ez a képlet: abszolút, relatív, vagy vegyeset? A válaszod indokold meg!
 - c*) A **Munka3** lap **D10** cellájába írd képletet, amely meghatározza az iskola 8. osztályos tanulóinak számát! Nevezd el a cellát **Létszám8**-nak! Használd ezt a nevet azokban a képletekben, amelyekben a 8. A osztály tanulóinak százalékos arányát kell meghatározni! Másold ezeket a képleteket azokba a cellákba, ahol ilyen százalékos arányt kell meghatározni! Módosul-e a képlet? A válaszod indokold meg! Mentsd a táblázatot **feladat 7.1.3.xlsx** néven a mappádba!

- 4*. Hozz létre táblázatot, amely az Ukrajnában forgalomban lévő bankjegyek értékét határozza meg dollárban, euróban és fontban! Az árfolyamot három különböző cellában tárold! Mentsd a táblázatot **feladat 7.1.4.xlsx** néven a mappádba!
5. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.1\zrazok 7.1.5 .xlsx**-et)! Tekintsd meg a dokumentum **Munka1** lapját **Nyomatási kép** és **Lap elrendezése** nézetben! Mindkét nézetben tekintsd meg a dokumentum valamennyi oldalát! Végezd el a munkalap nyomtatását!

14. SZÁMÚ GYAKORLATI MUNKA

Számítási feladatok megoldása

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\prakticsna 41\zrazok.xlsx**-et)!
2. A **Munka1** lap **F2** cellájába írd képletet, amely meghatározza a turista által az első 3 nap alatt megtett utat! Másold ezt a képletet az **F3:F7** tartományba!
3. A **Munka2** lap **F3** cellájába írd képletet, amely meghatározza az árakat dollárban (az árfolyamot az **E3**-ban találod), majd másold ezt a képletet az **E4:E10** tartományba!
4. A **Munka3** lap **B10** cellájába írd képletet, amely meghatározza az iskola tanulójának összlétszámát! Nevezd el a cellát **Tanulóknak**! Használd ezt a nevet abban a képletben, amellyel kiszámítod, hogy az iskola tanulójának hány százaléka jár első osztályba! A kitöltőnégyzet segítségével másold a képletet azokba a cellákba, amely a különböző osztályfokok részarányát határozza meg!
5. Mentsd a táblázatot **gyakorlati 14.xlsx** néven a mappádba!



7.2. A TÁBLÁZATKEZELŐ MATEMATIKAI, STATISZTIKAI ÉS LOGIKAI FÜGGVÉNYEINEK RENDELTETÉSE ÉS ALKALMAZÁSA



1. Az **Excel** mely függvényeit ismered? Hogy használjuk a függvényeket **Excelben**?
2. Milyen paraméterei lehetnek a függvényeknek **Excelben**? Mi lehet a függvény argumentuma?
3. Hogy írunk elágazásokat **Object Pascalban**? Hogy történik ennek végrehajtása?

Amint azt már tanultátok, az **Excel 2007** több mint 300 függvényt számláló beépített függvénykönyvtárral rendelkezik, amely kényelmi okokból különböző kategóriákba – *matematika, statisztikai, szöveg, logikai, pénzügyi* stb. – vannak sorolva.



A 7. osztályban már megismerkedtetek a táblázatkezelő néhány függvényével, mint például a számok összegzésére szolgáló **SZUM**, az átlag meghatározására szolgáló **ÁTLAG**, a legkisebb, illetve legnagyobb érték meghatározására alkalmas **MAX**, illetve **MIN**. Ezek közül az első a matematikai, a többi a statisztikai kategóriába tartozik.

MATEMATIKAI FÜGGVÉNYEK

Vizsgáljunk meg néhány matematikai függvényt (7.2. táblázat).

7.2. táblázat

Néhány matematikai függvény

Függvény	Argumentumok száma	Eredmény	Példa
ABS(szám)	1	A szám abszolút értéke	ABS(C10)
KEREKÍTÉS (szám, tizedesjegyek)	2	Szám kerekített értéke; ha a tizedesjegyek száma pozitív, akkor a tizedesvessző után, ha negatív, akkor előtte értendő	ROUND(C1;3)
GYÖK(szám)	1	A szám számtani gyöke	SQRT(B10)
HATVÁNY(szám; kitevő)	2	A szám kitevőedik hatványa	POWER(C5;5)
PI()	0	A π közelítő értéke, 3,14159265358979	PI()

LOGIKAI FÜGGVÉNYEK

Az Excel képleteiben a függvényeken, matematikai műveleteken kívül logikai kifejezések is előfordulhatnak, amelyek összehasonlítás-jeleket $>$ (nagyobb), $<$ (kisebb), $=$ (egyenlő), $<=$ (kisebb vagy egyenlő), $>=$ (nagyobb vagy egyenlő) – tartalmaznak. Ilyen például az **A2 + 15 = B4-1**, **SZUM (A2:C10) > 100** stb.

Ahogy azt már tudjátok, az ilyen kifejezés **True** vagy **False** értékeket vehet fel.

Az Excelben logikai függvényeket is használhatunk.

Az olyan függvényeket, amelyeknek eredménye True vagy False értékeket vehet fel, logikai függvényeknek nevezzük.

Az Excel logikai függvényei a **HA**, az **ÉS**, a **VAGY** és a **NEM**.

Logikai függvényeket akkor használunk, amikor a táblázatkezelőnek egy logikai kifejezés értékének alapján kell eldönteni, milyen műveletet kell végrehajtania. Ilyen feladatokkal már találkoztatok, amikor például a terv teljesítésétől függően kell a munkásnak felszámolni a 20% prémiumot. Vagy amikor az áru értékét a szavatossági idő lejártához közeledve 50%-kal csökkentik.

A HA függvény általános alakja a következő:

HA(logikai_kifejezés;érték_ha_igaz;érték_ha_hamis).

A függvény kiértékelése a következőképpen történik:

- kiértékelődik a **logikai_kifejezés** (**true** vagy **false**);
- ha az **érték true**, kiszámításra kerül az **érték_ha_igaz** kifejezés;
- ha az **érték false**, kiszámításra kerül az **érték_ha_hamis** kifejezés.

A **HA(A1+B1>100;C1*0,2;C1*0,1)** függvény kiértékelése például a következőképpen zajlik. Az **Excel** meghatározza az **A1+B1** értékét, majd meghatározza az **A1+B1>100** értékét. Ha ennek értéke **true**, a függvény értéke a **C1*0,2** lesz. Ha a kifejezés értéke **false**, a függvény értéke **C1*0,1** lesz.

Lássunk egy másik példát a **HA** függvény alkalmazására. Ismerjük a munkások által egy nap alatt legyártott alkatrészek mennyiségét és a tervezett mennyiséget. Minden munkás esetében el kell dönteni, teljesítette-e a tervet. A 7.6. ábrán a feladat megoldása látható.

Ahogy látjátok, a logikai függvény alkalmazása a táblázatkezelőben lényegében megegyezik annak végrehajtásával a programozásban.

Az **Excelben** használhatjuk a **HA** rövidített változatát, ami megfelel a nem teljes elágazásnak:

HA (logikai_kifejezés;érték_ha_igaz).

Vizsgáljuk meg a **VAGY**, **ÉS**, **NEM** függvényeket (7.3. táblzat).

	A	B	C	D	E	F	G
		Робітник	Норма	Виготовлено	Виконання норми		
3							
4		Величко	100	95	=IF(D4>=C4;"Виконав";"Не виконав")		
5		Іванов	120	112			
6		Петренко	120	122			
7		Шмуклер	130	143			

Аргументи функції

IF

Лог_вираз: D4>=C4 = FALSE

Значення_якщо_істина: "Виконав" = "Виконав"

Значення_якщо_хибність: "Не виконав" = "Не виконав"

Значення_якщо_істина: "Не виконав" = "Не виконав"

Перевіряє, чи виконується умова, і повертає одне значення, якщо вона виконується, та інше значення, якщо ні.

Значення_якщо_істина: значення, яке повертається, якщо лог_вираз має значення ІСТИНА. Якщо не вказано, повертається ІСТИНА. Дозволяється глибина вкладення до 7 рівнів.

Значення: Не виконав

OK Скасувати

7.6. ábra. A HA függvény alkalmazása a feladat megoldása során



A VAGY, ÉS, NEM logikai függvények

Függvény	Argumentumok száma	Eredmény	Példa
ÉS (logikai1, logikai2,...)	1-től 255-ig, csak az első kötelező	True , ha minden argumentum értéke True , egyébként False	AND(A1>2;B1>10;B1<20;C1=5)
VAGY (logikai1, logikai2,...)	1-től 255-ig, csak az első kötelező	True , ha legalább az egyik argumentum True , egyébként False	OR(A1>2;B1>10;B1<20;C1=5)
NEM (logikai)	1	True , ha az argumentum False , False , ha az argumentum True	NOT(F1>25)

Ezeket a függvényeket leggyakrabban a **HA** függvény argumentumaként használjuk, önállóan nagyon ritkán.

A 7.4. táblázatban a logikai függvények értékei láthatók az **A1** és **B1** cellák értékeitől függően.

A VAGY, ÉS, NEM logikai függvények táblázati értékei

A1	B1	AND(A1;B1)	OR(A1;B1)	NOT(A1)
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Az **ÉS** függvény a konjunkció, a **VAGY** függvény a diszjunkció, a **NEM** a negáció műveletét valósítja meg.

A **VAGY, ÉS, NEM** függvények segítségével nemcsak egyszerű összehasonlítások, hanem azoknál összetettebb logikai kifejezések is kiértékelhetők. A **HA** függvény argumentumaként használunk most egy kettős egyenlőtlenséget. Ekkor a **10<A3<20** helyett az **ÉS(A3>10;A3<20)** használjuk. Vagy az

$$y = \begin{cases} 2x - 5, & \text{ha } x < -2 \text{ vagy } x > 10 \\ \text{nem létezik } x \text{ egyéb értékei mellett} \end{cases}$$

képlettel megadott függvényt a következőképpen számíthatjuk ki, ha x értékét az **A5** cellában találjuk: **HA(VAGY(A5<-2;A5>10);2*A5-5; 'A függvény értéke nem meghatározott')**.



Ha többet szeretnél tudni

Az **Excel 2007**-ben összetett függvényeket is alkalmazhatunk (64 beépítettségig szintig). Így a **HA** függvény második és harmadik argumentuma is tartalmazhat függvényeket, akár egy újabb **HA** függvényt is.

Például: **HA (A1<0; -B1/A1; HA(A1>0;B1/A1; 'Az osztás nem végezhető el'))**.



Összefoglalás

Az **Excel 2007** több mint 300 függvényt számláló beépített függvénykönyvtárral rendelkezik, amely kényelmi okokból különböző kategóriákba – *matematikai, statisztikai, szöveg, logikai, pénzügyi* stb. – vannak sorolva.

Matematikai függvények például az abszolút érték meghatározására szolgáló **ABS**, a számtani gyököt kiszámító **GYÖK**, a hatvány értékét kiszámító **HATVÁNY**, az összeget kiszámító **SZUM**.

Az **Excel** logikai függvényei a **HA**, a **VAGY**, az **ÉS** és a **NEM**. Ezeket akkor használjuk, ha a táblázatkezelőnek egy logikai kifejezés értékétől függően kell egyik vagy másik műveletet végrehajtani.

A **HA** függvény általános alakja a következő:

HA(logikai_kifejezés;érték_ha_igaz;érték_ha_hamis).

A függvény kiértékelése a következőképpen történik: kiértékelődik a **logikai_kifejezés** (**true** vagy **false**); ha az érték **true**, akkor kiszámításra kerül az **érték_ha_igaz** kifejezés; ha az érték **false**, kiszámításra kerül az **érték_ha_hamis** kifejezés.







Felelj a kérdésekre!

- 1• Milyen az **ABS**, **KEREKÍTÉS**, **PI**, **HATVÁNY** függvények általános alakja? Hány argumentuma van a függvénynek?
- 2• Milyen a **GYÖK** és **SZUM** függvények általános alakja? Hány argumentuma van a függvénynek?
- 3• Milyen az **ÁTLAG**, **MAX**, **MIN** függvények általános alakja? Hány argumentuma van a függvénynek?
- 4• Mi a logikai kifejezés? Milyen jeleket tartalmazhatnak?
- 5° Milyen értékeket vehetnek fel a logikai kifejezések?
- 6• Milyen a **HA** függvény általános alakja? Hány argumentuma van a függvénynek?
- 7• Hogy kerül kiszámításra a **HA** függvény értéke?
- 8• Milyen az **ÉS**, **VAGY**, **NEM** függvények általános alakja? Hány argumentuma van a függvénynek?
- 9• Hogy történik az **ÉS**, **VAGY**, **NEM** függvények kiértékelése?
- 10* Mi az összetett függvény? Mennyi a beépítettség maximális szintje?

**Végezd el a feladatokat!**

- 1°. Indítsd el az **Excel 2007**-et! Hozz létre táblázatot 5 ismert oldalú téglalap átlójának kiszámítására! Az eredményt 2 tizedesjegy pontossággal jelenítsd meg! Mentse a táblázatot **feladat 7.2.1.xlsx** néven a mappádba!
- 2°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.2\zrazok 7.2.2.xlsx**-et)! A **Munka1** munkalapon egy öttusaverseny résztvevői által elért pontszámokat találod. Illesz a táblázatba képleteket, amelyek meghatározzák a versenyzők összpontszámát, az egyes versenyzők által elért legmagasabb pontszámot, valamint az egyes versenyszámokban elért átlagos pontszámot! Az átlagokat kerekítsd egészekre! Mentse a táblázatot **feladat 7.2.2.xlsx** néven a mappádba!
- 3°. Az **A5** cellában a 10 , a **B5**-ben a -7 , a **C5**-ben a 0 szám van. Határozd meg a következő függvények értékét:

- a) **AND(A5>5;A5<20);**  e) **NOT(B5<20);**
 b) **OR(C5<10;C5>=20);**  f) **AND(OR(B5>5;B5<=-5);NOT(B5<>10));**
 c) **OR(C5<10;A5>20);**  g) **OR(AND(A5>2;A5<=10);B5<0);**
 d) **AND(A5>=-2;B5>0);**  h) **NOT(AND(A5<100;C5=0))!**



- 4°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.2\zrazok7.2.4.xlsx**-et)! A **Munka1** munkalapon a munkások által egy nap alatt elkészített alkatrészek száma található. Az **A1** cella a napi tervet tartalmazza. Egészítsd ki a táblázatot egy képlettel, amely minden munkás esetében meghatározza, teljesítette-e a tervet! Az eredmény formátuma **Igen** vagy **Nem** legyen! Mentse a táblázatot **feladat 7.2.4.xlsx** néven a mappádba!



- 5*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.2\zrazok 7.2.5.xlsx**-et)! A **Munka1** munkalapon a diákolimpia első két fordulójának eredményeit találod. I. helyezést nyernek azok a diákok, akik az elérhető pontszám legalább 85%-át érték, a II. helyezéshez a pontszám 75%-át, harmadik helyezéshez pedig az 50%-át. Minden más diák a részvételét igazoló oklevelet kap. Egészítsd ki a táblázatot képlettel, amely minden résztvevő esetében meghatározza, milyen eredményt ért el! Mentse a táblázatot **feladat 7.2.5.xlsx** néven a mappádba!



- 6*. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.2\zrazok 7.2.5.xlsx**-et)! A **Munka1** munkalap az x változó értékeit tartalmazza. Egészítsd ki a táblázatot az y változó értékét meghatározó képlettel, ha

$$y = \begin{cases} 2x - 5, & \text{ha } x < -2 \text{ vagy } x > 10 \\ 3x + 1, & \text{ha } 2 \leq x \leq 3, \\ x^5 - 4x^3 & \text{minden egyéb értéke esetében} \end{cases}$$

Mentse a táblázatot **feladat 7.2.6.xlsx** néven a mappádba!

- 7*. Indítsd el az **Excel 2007**-et! Készíts táblázatot annak meghatározására, mi kifizetődőbb az ügyfélnek, az évi 24% kamat, amelyet havonta számolnak fel, vagy évi 26% kamat, ha azt félévente számolják fel! Mentsd a táblázatot **feladat 7.2.7.xlsx** néven a mappádba!



7.3. DIAGRAMOK LÉTREHOZÁSA ÉS FORMÁZÁSA



1. Mi a diagram? Mire használják?
2. Milyen objektumokból áll a diagram **Excel**ben? Milyen tulajdonságai vannak ezeknek?
3. Hogy hozunk létre diagramot? Milyen szerkesztési és formázási műveleteket végezhetünk a diagramokkal?

DIAGRAMOK EXCEL 2007-BEN

A 7. osztályban már tanultátok, hogy a szám adatok jobb szemléltetése és könnyebb áttekintése céljából diagramokat használunk, amelyek az adatok arányait mértani alakzatok segítségével mutatják be. Az alakzatok mérete arányos azokkal a mennyiségekkel, amelyeket ábrázolnak.

Leggyakrabban oszlop- és kördiagramokat használunk, ezekkel ismerkedtetek meg a múlt esztendőben. A **kördiagramok** azt mutatják, mekkora rész esik egy-egy mennyiségre azok összegéhez képest. Az **oszlopdigramot** akkor célszerű használni, ha a mennyiségek arányait szeretnénk bemutatni.

Az **Excel**ben ezeken kívül használhatunk még *vonalt, pont, sugár, buborékdiagramot, grafikon* stb. Minden típusnak több altípusa van. Ezeket a **Beszűrés** lap **Diagram** csoportjában tekinthetjük meg.

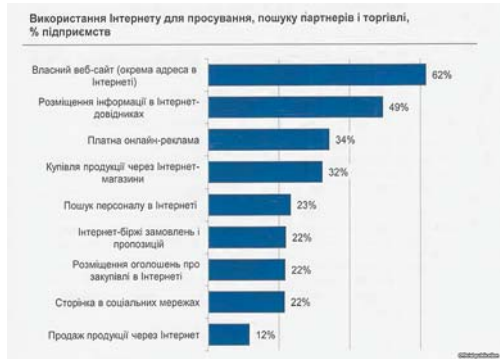
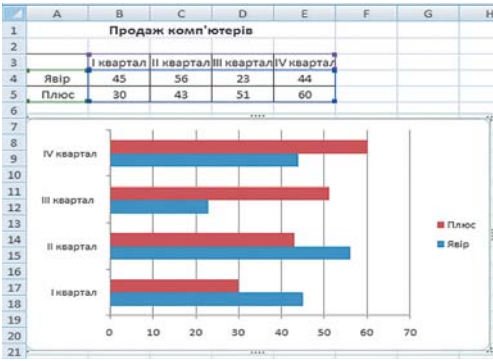
Vizsgáljunk meg ezek közül néhányat.

A **sávdigram** hasonlít az *oszlopdigramra*, de itt az oszlopok elhelyezése vízszintes (7.7. ábra). Ezek alkalmazása akkor célszerű, amikor a vízszintes elemek szemléletesebbek a függőlegeseknél, például, ha a vízszintes tengelyen idő, hőmérséklet növekvő intervallumai vannak ábrázolva. Akkor is érdemesebb ezeket alkalmazni, ha a függőleges tengelyek feliratozása hosszúak, és áttekinthetőbbek vízszintes szövegírással.

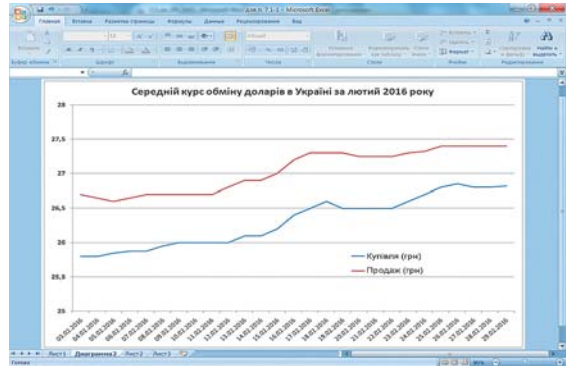
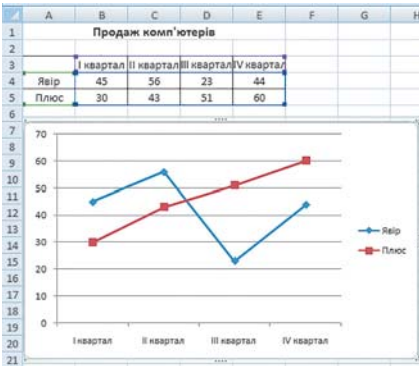
A **grafikon** (vonaldiagram, pontdiagram). Akkor alkalmazzuk, ha valamilyen folyamatot, az adatok dinamikáját, egymástól való függését szeretnénk bemutatni (7.8. ábra). A grafikonon ábrázolhatjuk az adatpontokat, vagy eltekinthetünk ettől.

A pontdiagramokat leginkább akkor alkalmazzuk, ha két változó összefüggését szeretnénk bemutatni, amikor az egyik változó értéke a másik által meghatározott (7.9. ábra).

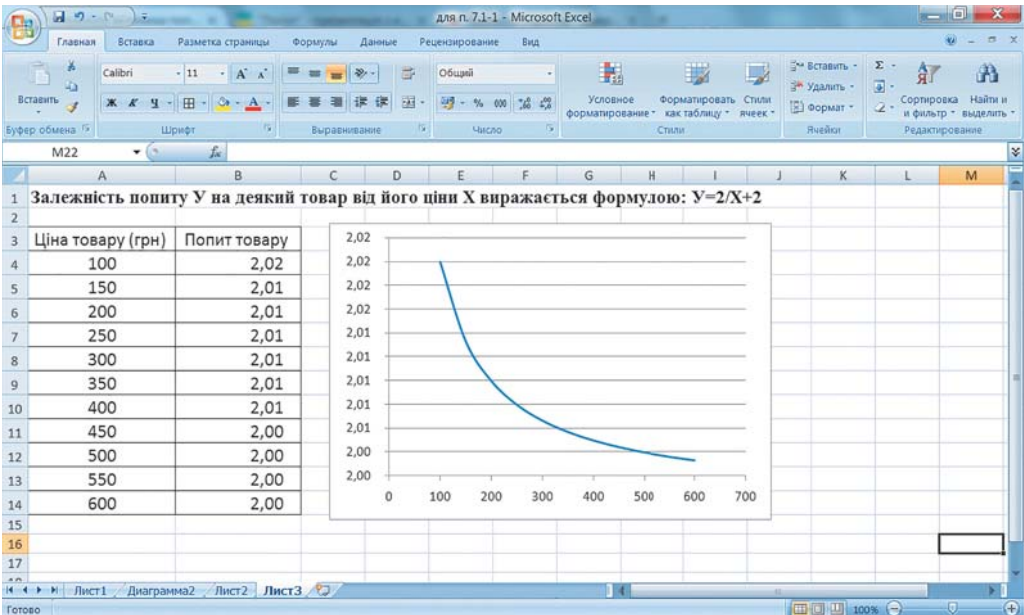
A pontdiagramokat leggyakrabban függvénygrafikonok ábrázolására használjuk (7.10. ábra). Egy diagramon két függvény grafikonját is elhelyezhetjük, ez hasznos lehet az egyenletek közelítő gyökeinek kiszámításakor (7.11. ábra).



7.7. ábra. Sávdigramok

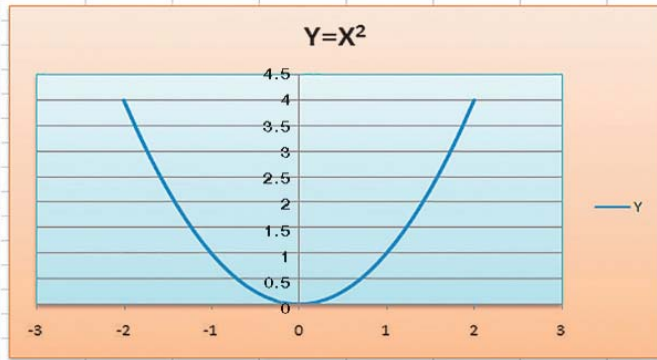


7.8. ábra. Grafikonok adatpontokkal és azok nélkül



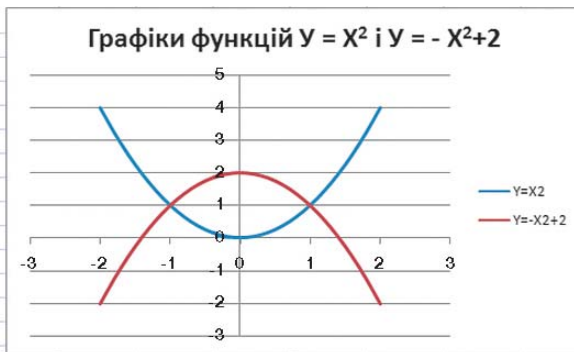
7.9. ábra. Pontdiagram

X	Y
-2	4
-1,8	3,24
-1,6	2,56
-1,4	1,96
-1,2	1,44
-1	1
-0,8	0,64
-0,6	0,36
-0,4	0,16
-0,2	0,04
0	0
0,2	0,04
0,4	0,16
0,6	0,36
0,8	0,64
1	1
1,2	1,44
1,4	1,96
1,6	2,56
1,8	3,24
2	4



7.10. ábra. Függvénygrafikon

X	$Y=X^2$	$Y=-X^2+2$
-2	4	-2
-1.8	3.24	-1.24
-1.6	2.56	-0.56
-1.4	1.96	0.04
-1.2	1.44	0.56
-1	1	1
-0.8	0.64	1.36
-0.6	0.36	1.64
-0.4	0.16	1.84
-0.2	0.04	1.96
0	0	2
0.2	0.04	1.96
0.4	0.16	1.84
0.6	0.36	1.64
0.8	0.64	1.36
1	1	1
1.2	1.44	0.56
1.4	1.96	0.04
1.6	2.56	-0.56
1.8	3.24	-1.24
2	4	-2



7.11. ábra. Két függvény grafikonja

DIAGRAMOK LÉTREHOZÁSA

Ahogy azt a múlt évben megtanultátok, a diagramokat a táblázat adataiból készítjük. A diagramok dinamikusak, azaz az adatok változásakor megváltoznak. A diagramokat a **Beszűrés** lap **Diagram** csoportjának eszközeivel készíthetjük. E célból:

1. Jelöljük ki a táblázatban azokat az adatokat, amelyeket ábrázolni szeretnénk!
2. A **Beszűrés** lap **Diagram** csoportjából válasszuk ki a megfelelő diagramtípust!
3. Kattintsunk a kiválasztott diagramon!



Itt jegyezzük meg, hogy az ábrázolandó adatok alkothatnak összefüggő tartományt, de ez nem kötelező. Kívánatos, hogy a kijelölés tartalmazza az oszlopok és sorok neveit.

A fenti algoritmus végrehajtásának eredményeképpen a munkalapon létrejön egy diagram, amelynek tulajdonságai megegyeznek az adott diagramtípus alapértelmezett tulajdonságaival.

A diagramon a következő objektumokat különíthetjük el: diagramterület, rajzterület, diagramcím, adatpont, adatsor, elsődleges vízszintes tengely, elsődleges függőleges tengely, tengelycímek, adatfeliratok.

DIAGRAMOK SZERKESZTÉSE ÉS FORMÁZÁSA

A diagram szerkesztése alatt a következőket értjük:

- a diagram adatterületének megváltoztatása, ezen belül a sorok és oszlopok felcserélése, új adatok hozzáadása, sorok vagy oszlopok eltávolítása;
- a diagramcím és tengelycímek hozzáadása, eltávolítása, szerkesztése, jelmagyarázat szerkesztése;
- a tengely és a tengelyek osztásközének beállítása;
- a diagram egyes objektumainak kijelölése és szerkesztése;
- a diagramtípus megváltoztatása;
- a diagram helyének megváltoztatása (mozgatása a munkalapon, külön diagramlapra helyezése)

stb.

A diagram formázása alatt a diagram egészének vagy egyes objektumainak formázását értjük, például:

- a diagram átméretezése;
- a diagram stílusának megváltoztatása;
- a betűtípus megváltoztatása, az egyes objektumok méretének, színének, kitöltésének megváltoztatása;
- a rács és a tengelyek megjelenésének beállítása

stb.

A szerkesztési és formázási műveleteket a **Diagramszközök** ideiglenes eszköztár **Tervezés**, **Elrendezés** és **Formátum** lapjain található eszközökkel végezzük.

Emlékezzünk vissza ezek némelyikére:

- a diagramtípust a **Tervezés** ⇒ **Más diagramtípus** ⇒ **Diagramtípus módosítása** paranccsal változtathatjuk meg;
- a diagram sorait és oszlopaikat a **Tervezés** ⇒ **Adatok** ⇒ **Sor/oszlop váltása** paranccsal cserélhetjük fel;
- a diagram méreteit a méretezőpontok mozgatásával változtathatjuk meg;
- a diagram helyzetét a munkalapon a diagram mozgatásával változtathatjuk meg.

Azzal is megismerkedtetek a 7. osztályban, hogy a diagram objektumának kijelölése után van lehetőségünk annak formázására:

- a **Formátum** lap **Alakzatstílusok** csoportjában új stílust választhatunk az objektumhoz;
- a **Formátum** lap **WordArt Stílusok** csoportjában a szöveges objektumokhoz rendelhetünk stílusokat;
- a **Formátum** lap **Alakzatstílusok** csoportjában megadni a körvonalat, a kitöltést, effektusokat rendelni a kiválasztott objektumhoz;
- végrehajtani a **Formátum** ⇒ **Aktuális kijelölés** ⇒ **Kijelölés formázása** parancsot, és a megnyíló párbeszédablakokban elvégezni a szükséges beállításokat;
- alkalmazni az objektumok helyi menüjét.



Összefoglalás

A sávdiaagramokat – amelyek hasonlóak az oszlopdiaagramokhoz – akkor használjuk, ha az elemek vízszintes megjelenítése szemléletesebb a függőlegesnél.

A grafikonokat akkor célszerű alkalmazni, amikor folytonosan változó mennyiségeket kell ábrázolni, vagy két mennyiség összefüggését kell megmutatni.

A pontdiaagramokat leginkább akkor alkalmazzuk, ha két változó összefüggését szeretnénk bemutatni, amikor az egyik változó értéke a másik által meghatározott.

A diagramok beillesztését a **Beszúrás** lap **Diagram** csoportjának eszközeivel végezzük.

A diagramok szerkesztését és formázását a **Diagrameszközök** ideiglenes eszköztár **Tervezés, Elrendezés** és **Formátum** lapjainak eszközeivel végezhetjük el.



Felelj a kérdésekre!

- 1°. Milyen diagramtípusok léteznek **Excel 2007**-ben?
- 2°. Milyen objektumokat találhatunk az **Excel 2007** diagramjain? Röviden írd le ezeket!
- 3°. Milyen tulajdonságai vannak a diagramok objektumainak **Excel 2007**-ben?
- 4°. Hogy hozunk létre sávdiaagramot? Mi a rendeltetése?
- 5°. Hogy hozunk létre grafikont? Mi a rendeltetése?
- 6°. Hogy hozunk létre pontdiaagramot? Mi a rendeltetése?
- 7°. Miben különbözik a grafikon és a pontdiaagram alkalmazása?
- 8*. Használhatunk-e pontdiaagramot egyenletek közelítő grafikai megoldása során?
- 9°. Milyen szerkesztési műveleteket végezhetünk a diagramon? Hogy végezhetjük el ezeket?
- 10°. Milyen formázási műveleteket végezhetünk a diagramon? Hogy végezhetjük el ezeket?
- 11°. Hogy válasszuk ki a megfelelő diagramtípust?



Végezd el a feladatokat!

- 1°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.3\zrazok 7.3.1.xlsx**-et)! A **Munka1** munkalapon az utóbbi hónapban lehullott csapadék adatait találod. Készíts az adatok alapján kördiaagramot, sávdiaagramot és grafi-



kont! Melyik diagram szemlélteti leginkább átláthatóan az adatokat? Formázd a kiválasztott diagramot (cím, adatfeliratok, szövegformátum, kitöltés stb.), majd helyezd át külön diagramlapra! Mentsd a fájlt **feladat 7.3.1.xlsx** néven a mappádba!

- 2• Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.3\razok 7.3.2.xlsx**-et)! A **Munka1** munkalapon adatokat találsz az Ukrajna területén különböző vállalatok által kitermelt gáz mennyiségéről az utóbbi három év során! Az adatok alapján készíts sávdigramot az egyes vállalatok által kitermelt mennyiségről, valamint a teljes kitermelt mennyiségről! Helyezd a diagramokat külön lapokra! Formázd a diagramokat (cím, adatfeliratok, szövegformátum, kitöltés stb.)! Mentsd a fájlt **feladat 7.3.2.xlsx** néven a mappádba!



- 3• Keresz az interneten adatokat lakóhelyed lakosságának (vagy Ukrajna lakosságának) alakulásáról az utóbbi 5 év során! Hozz létre táblázatot az adatokból! Hozz létre az adatok alapján kördigramot, sávdigramot és grafikont! Helyezd el ezeket külön diagramlapokon, és formázd meg az elképzelésed szerint! Mentsd a fájlt **feladat 7.3.3.xlsx** néven a mappádba!

- 4• Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\pukt 7.3\razok 7.3.4.xlsx**-et)! A **Munka1** munkalapon található táblázat bizonyos anyagok oldhatóságát tartalmazza a hőmérséklet függvényében. Készíts az adatok alapján adatpontokat tartalmazó és nem tartalmazó vonaldigramot! Helyezz el a diagramon tengelycímeteket! Alkalmaz diagramstílust! Mentsd a fájlt **feladat 7.3.4.xlsx** néven a mappádba!

- 5• Nyisd meg az **Excel 2007**-et! A **Munka1** munkalapon készítsd el az $y = x^3 - 3x$ függvény grafikonját a $[-3; 3]$ intervallumon $0,2$ lépéssel! Mentsd a fájlt **feladat 7.3.5.xlsx** néven a mappádba!



- 6• Nyisd meg az **Excel 2007**-et! A **Munka1** munkalapon készítsd el az $y = \frac{1}{x^2 + 1}$ függvény grafikonját a $[-3; 3]$ intervallumon $0,2$ lépéssel! Mentsd a fájlt **feladat 7.3.6.xlsx** néven a mappádba!

- 7*• Nyisd meg az **Excel 2007**-et! A **Munka1** munkalapon készítsd el az $y = x^4 - 2x^2$ és az $y = 1/4 x - 0,5$ függvények grafikonjait a $[-3; 3]$ intervallumon $0,2$ lépéssel, és határozd meg az $x^4 - 2x^2 = 1/4 x - 0,5$ egyenlet közelítő gyökeit ezen az intervallumon! Mentsd a fájlt **feladat 7.3.7.xlsx** néven a mappádba!



- 8*• Indítsd el az **Excel 2007**-et! A **Súgó** segítségével állapítsd meg a **Buborékdiagram** és a **Sugárdiagram** rendeltetését! Keresz rá az interneten vidéked időjárásai adataira! Hozz létre ezek alapján táblázatot az utóbbi hónap felhős, derült és borult napjairól! Hozz létre az adatok alapján buborék és sugárdiagramot! Helyezd el a diagramokat a táblázat mellett! Formázd meg a diagramokat! Mentsd a fájlt **feladat 7.3.8.xlsx** néven a mappádba!

15. SZÁMÚ GYAKORLATI MUNKA

A táblázatkezelő matematikai, logikai és statisztikai függvényeinek alkalmazása. Diagramok beszúrása

Figyelem! A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\prakticsna 15.3\zrazok pr15.xlsx**-et)!
2. A **Munka1** munkalapon lévő táblázat az elektromos mérőóra állását tartalmazza az elmúlt év minden hónapjában. Az **A1** cella tartalmazza $1 kWh$ energia árát. Egészítsd ki a táblázatot képletekkel, amelyekkel meghatározható a fogyasztásért havonta kifizetett összeg, az év során kifizetett teljes összeg, valamint a havi számlák átlaga!
3. Készítsd el külön diagramlapra a havonta kifizetett összegeket ábrázoló grafikonot! Helyezz el a grafikonon tengelyfeliratokat és adatfeliratokat! Formázd meg a diagramot!
4. A **Munka2** munkalapon osztályod tanulóinak magasságát találod. Határozd meg minden tanulóra nézve, hogy magasabb-e az osztályátlagnál!
5. Készítsd el külön diagramlapra a tanulók magasságát ábrázoló sávdigramot! Helyezz el a grafikonon tengelyfeliratokat és adatfeliratokat! Formázd meg a diagramot!
6. A **Munka3** munkalapon készítsd el az $y = x^4 - 4x$ függvény értéktáblázatát a $[-3; 3]$ intervallumon!
7. Készítsd el külön diagramon a függvény grafikonját! Adj címet a diagramnak! Helyezz el a grafikonon tengelyfeliratokat és adatfeliratokat! Formázd meg a diagramot!
8. Mentsd a táblázatot **gyakorlati 15.xlsx** néven a mappádba!



7.4. ADATOK RENDEZÉSE A TÁBLÁZATOKBAN. AUTOMATIKUS ÉS IRÁNYÍTOTT SZŰRŐK



1. Milyen a szimbólumok sorrendje az **Unicode** táblában?
2. Milyen értékeket vehet fel az **ÉS** és a **VAGY** függvény az argumentumok értékétől függően?
3. Mik a * és ? szimbólumok jelentése a fájlok nevében?

ADATOK RENDEZÉSE A TÁBLÁZATOKBAN

A táblázat adatait rendezhetjük, vagyis megváltoztathatjuk azok sorrendjét valamilyen kritériumoknak megfelelően.

A táblázatkezelőben **növekvő** és **csökkenő** sorrendet alakíthatunk ki. A rendezés módja a rendezendő adatoktól függ. A szöveges adatok rendezése karakterenként történik *a szimbólumok Unicode-táblában elfoglalt helye alapján*.

Az **Excel 2007**-ben az adatok rendezése a következő szabályok szerint történik:

- ha a táblázat különböző típusú adatokat tartalmaz, azok rendezése a következő sorrendben történik: számok, dátum/idő, szöveg, logikai;
- a számok rendezése a legkisebbtől a legnagyobbig történik;



- a dátumok rendezése a legrégebbitől a legújabb irányába történik;
- a latin betűs szöveg a cirill betűs előtt helyezkedik el;
- a kisbetűs szöveg a nagybetűs előtt helyezkedik el;
- a szöveg összehasonlítása betűnként történik;
- a logikai értékek közül a **False** megelőzi a **True**-t;
- az üres cellák a lista végére kerülnek (bár a **Szóköz**nek van a legkisebb kódja az **UNICODE**-ban).

Csökkenő rendezés esetében a sorrend értelemszerűen ellenkezőjére fordul.

A 7.12. ábrán vegyes típusú adatok láthatók növekvő, illetve csökkenő sorrendben.

Ahhoz, hogy egy tartomány adatait gyorsan sorba rendezzük az első oszlop adatai alapján, a következőt kell tenni:

1. Jelöld ki a rendezendő cellatartományt!
2. Hajtsd végre a **Kezdőlap** ⇒ **Szerkesztés** ⇒ **Rendezés és szűrés** ⇒ **Rendezés növekvő** (vagy **Rendezés csökkenő**) **sorrendben** parancsát (7.13. ábra)!

A művelet eredményeként a kijelölt sorok úgy cserélődnek ki, hogy az első sor adatai a kiválasztott módon legyenek rendezve. A többi oszlop adatai így természetesen együtt mozognak az elsővel.

Itt hívjuk fel a figyelmeteket a következőkre:

1. A rendezési utasítások neve automatikusan alkalmazkodik a rendezendő adat-típushoz:
 - szöveges utasítások esetében **Rendezés A-Z**, **Rendezés Z-A**;
 - dátum és idő esetében **Rendezés a legrégebbtől a legújabbig** vagy a **Rendezés legújabbtól a legrégebbig**.

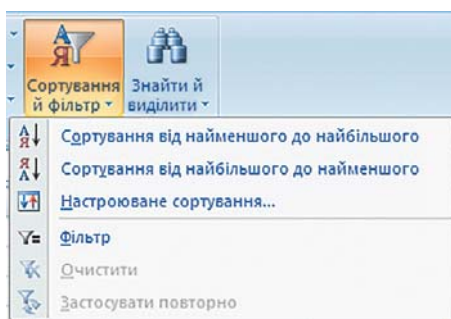
2. Ha a rendezésnél csak egy oszlop lesz kijelölve, megjelenik egy figyelmeztető ablak (7.14. ábra), amely azt ajánlja, hogy terjesszük ki a kijelölést a szomszédos cellákra, de megengedi, hogy a rendezés az aktuális kijelölésen belül történjék.

A kijelölt tartományt több oszlop adatai figyelembevételével is rendezhetjük. Ez a következőképpen történik:

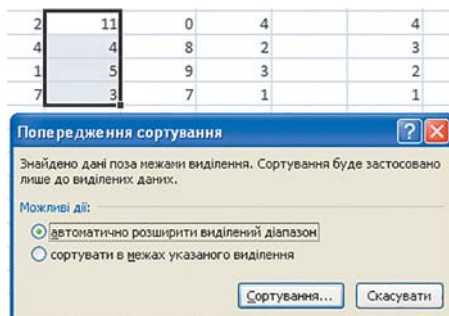
- először megtörténik az adatok rendezése az első oszlop adatai alapján;

Уведені дані	Відсортовані за зростанням	Відсортовані за спаданням
35	-10	TRUE
школа	0,5	FALSE
-10	35	школа
	Lyceum	Сіренко
TRUE	Сіренко	Lyceum
	школа	35
0,5	FALSE	0,5
FALSE	TRUE	-10
Lyceum		
Сіренко		

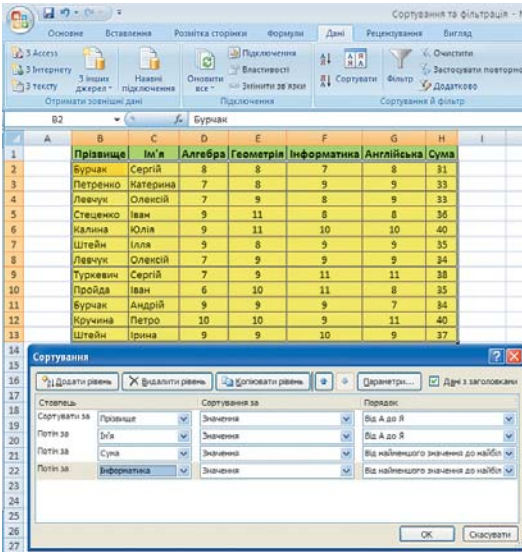
7.12. ábra. Példák rendezésre



7.13. ábra. A **Rendezés és szűrés** gomb legördülő menüje



7.14. ábra. **Figyelmeztető** üzenet ablak



7.15. ábra. A Rendezés párbeszédablak

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Сергій	8	8	7	8	31
Петренко	Катерина	7	8	9	9	33
Левчук	Олександр	7	9	8	9	33
Стеценко	Іван	9	11	8	8	36
Калина	Юлія	9	11	10	10	40
Штейн	Ілля	9	8	9	9	35
Левчук	Олександр	7	9	9	9	34
Туркевич	Сергій	7	9	11	11	38
Пройда	Іван	6	10	11	8	35
Бурчак	Андрій	9	9	9	7	34
Кручина	Петро	10	10	9	11	40
Штейн	Ірина	9	9	10	9	37

7.16. ábra. A 7.15. ábrán megadott adatok alapján történt rendezés

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олександр	7	9	8	9	33
Левчук	Олександр	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

A harmadik kritériumot (**Összeredmény**) csak az ötödik és hatodik sorok esetében – mindkettőben Olekszij Levcsuk szerepel – kell figyelembe venni. A negyedik kritériumot egyszer sem kell figyelembe venni, mivel nem maradtak olyan sorok, amelyeket ne lehetett volna az első három kritérium alapján egyértelműen elhelyezni.

Hogy a kijelölt tartományt a kiválasztott oszlopok szerint rendezzük:

1. Ki kell jelölni a rendezendő tartományt (kívánatos, hogy a kijelölés tartalmazza az oszlopok neveit és nem tartalmazhat egyesített cellákat)!
2. Végre kell hajtani az **Adatok** ⇒ **Rendezés és szűrés** ⇒ **Rendezés** parancsot!
3. A megnyíló párbeszédablakban állítsuk be a rendezés paramétereit: a **Rendezés** mezőben kiválasztani a *családnevet*, a rendezés alapjául az *értékek* szolgálnak majd, a sorrendben pedig a *növekvő* vagy *csökkenő* sorrendet (7.15. ábra).

- a többi kritérium csak akkor kerül kiértékelésre, ha az első oszlop adatai alapján a sorrend nem dönthető el (azaz az első oszlopban vannak ismétlődő adatok).

Minden, a rendezéshez kijelölt oszlop egy *rendezési szintet* alkot.

Figyeljük meg ennek működését egy konkrét példán. A 7.15. ábrán az iskola tanulói által elért tanulmányi eredmények egy még rendezetlen táblázatát láthatjátok. A 7.16. ábrán ezt a táblázatot 4 szint – **Családi név (A–Z)**, **Név (A–Z)**, **Összeredmény** (a legkisebttől a legnagyobbig), **Informatikából** elért eredmény (a legkisebttől a legnagyobbig) – alapján történt rendezés után mutatja.

A rendezés a következőképpen történik: először a táblázat **Családi név** szerint rendeződik **A–Z sorrendben**. Amennyiben nem lenne két azonos családnevű, a rendezés ezzel be is fejeződne. De a táblázat három ismétlődést – Burcsak, Levcsuk, Stein – is tartalmaz. A további rendezés már a második szint – **Név** – alapján történik, de már csak a három sorpár esetében. Az első és második, az ötödik és hatodik, valamint a tizenegyedik és tizenkettedik sorok tehát úgy rendeződnek majd, hogy a nevek növekvő sorrendbe kerüljenek (**A–Z**).



4. Az **Újabb szint** gombra kattintva adjunk hozzá egy rendezési szintet, majd a megnyíló **Azután** sorban állítsuk be a második oszlopot a rendezés alapjául!
5. Ismételjük meg a negyedik lépést, ahányszor az szükséges!
6. Ha a kijelölésbe bekerültek az oszlopok nevei, be kell kapcsolni az **Adatok fejléccet tartalmaznak** jelölőt (ekkor a kijelölés első sora helyén marad a rendezés során).
7. El kell végezni a rendezést az **OK** gombra kattintva!



Ha többet szeretnél tudni

Rendezni nemcsak az oszlopokat, hanem a sorokat is lehet. Alapértelmezés szerint az **Excel 2007** a sorokat rendezi az oszlopok értékei szerint. Ha ettől el kívánunk térni, kattintsunk a **Rendezés** ablakban a **Beállítások** gombra, majd válasszuk ki a megfelelő irányt. Ugyanebben az ablakban állíthatjuk be, hogy a rendezés során a kis- és nagybetűket meg kell-e különböztetni.


AUTOMATIKUS SZŰRÉS

A **szűrés** a táblázat adatainak bizonyos kritérium alapján történő kiválogatását jelenti.

A szűrés után a táblázatnak csak azokat a sorait láthatjuk, amelyek megfelelnek a **szűrőfeltételnek**. A táblázat többi sora ideiglenesen láthatatlanná válik.

A szűrést a következőképpen végezzük:

1. Helyezzük el a kurzort a szürendő táblázatban, vagy jelöljük ki a szürendő cellatartományt!
2. Hajtsuk végre a **Kezdőlap** \Rightarrow **Szerkesztés** \Rightarrow **Rendezés és szűrés** \Rightarrow **Szűrés** (vagy az **Adatok** \Rightarrow **Szűrés és rendezés** \Rightarrow **Szűrés**) parancsot!

A parancs végrehajtása után a táblázat első sorában minden cella mellett megjelenik egy legördülő lista  a szűrési lehetőségekkel (7.17. ábra). Ebben a listában szín, szöveg és számszűrőket találunk.

Vizsgáljunk meg néhány példát!

A szűrést leggyakrabban a táblázat egyik oszlopának adatai alapján végezzük. Tegyük fel, hogy a táblázatban csak azoknak a diákoknak az adatait szeretnénk látni, akik **9**-est értek el **informatikából**. E célból:

1. Nyissuk meg az **Informatika** oszlop szűrőmenüjét!
2. Kapcsoljuk ki az **Összes kijelölése** jelölőnégyzetet!
3. Kapcsoljuk be a **9**-es szám melletti jelölőnégyzetet!
4. Kattintsunk az **OK** gombra!

Ezután a táblázatnak csak azok a sorai lesznek láthatók, amelyekben az **Informatika** oszlopban a **9**-es szerepel. A fenti algoritmus végrehajtását és az eredményét a 7.18. és 7.19. ábrán láthatjuk.

Prívisce	Im'a	Algebra	Geometria	Informatika	Anglil'ska	Suma
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	8	9	33
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

7.17. ábra. Cellatartomány **Szűrés** után

7.18. ábra. Az **Informatika** oszlop szerinti szűrés beállítása

7.19. ábra. A szűrés eredménye

7.20. ábra. A **Név** szűrése két érték alapján

7.21. ábra. A szűrés eredménye

7.22. ábra. Számszűrők alkalmazása

Figyeljétek meg, hogy a szűrés végrehajtása után az **Informatika** oszlop neve mellett megváltozik a szűrés jele (7.19. ábra). Ha ilyen gomb van az oszlop neve mellett, az azt jelenti, hogy az adott oszlop alapján szűrés van érvényben.

A szűrést két vagy több érték alapján is végezhetjük. Ha például a táblázatban csak az **Ivan** és **Olekszij** nevű tanulókat szeretnénk látni, a következőképpen járunk el:

1. Nyissuk meg a **Név** mező szűrőfeltételeinek listáját!
2. Kapcsoljuk ki az **Összes kijelölése** jelölőnégyzetet!
3. Kapcsoljuk be az **Ivan** és az **Olekszij** melletti jelölőnégyzetet (7.20. ábra)!
4. Kattintsunk az **OK** gombra!

A szűrés eredményét a 7.21. ábra mutatja.

Amennyiben az oszlopban csak számok szerepelnek, alkalmazhatunk **Számszűrőket** is. Ezekben az **Egyenlő**, **Nem egyenlő**, **Kisebb**, **Nagyobb**, **Között** stb. feltételeket alkalmazhatjuk (7.22. ábra).

Ha például azokat a diákokat szeretnénk kiszűrni, akik összpontszáma **legalább 35**, de **nem haladja meg a 40**-et:

1. Nyissuk meg az **Összesen** oszlop legördülő szűrőmenüjét!
2. Válasszuk a **Számszűrőket**!
3. Írjuk be a megfelelő mezőkbe a **35**-öt és a **40**-et (7.23. ábra)!
4. Kattintsunk az **OK** gombra!

A 7.23. ábrán látható szűrőfeltétel szerinti szűrés eredményét a 7.24. ábrán láthatjuk.

A **Toplista Szűrőfeltétel** kiválasztása után megnyíló ablakban (7.25. ábra) kiválaszthatjuk, hogy csak a néhány leg-



nagyobb vagy legkisebb elem kerüljön megjelenítésre.

Az **Átlag alatt** vagy **Átlag felett** feltételekkel kiválogathatjuk azokat a sorokat, amelyek az adott oszlop átlagánál kisebb vagy nagyobb értékeket tartalmaznak.

A szövegszűrők esetében (7.26. ábra) más lehetőségeket tartalmaz, például **Egyenlő**, **Nem egyenlő**, **Kezdődik**, **Végződik**, **Tartalmaz**.

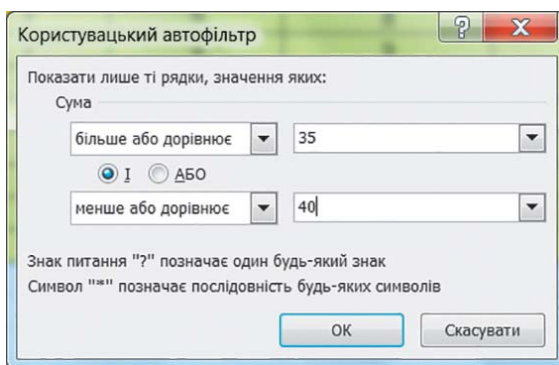
Ha tehát azokat a diákokat szeretnénk kiszűrni, akiknek a neve **P** betűvel kezdődik, a következőképpen járunk el:

1. Nyissuk meg a **Családnév** szűrőfeltételeit!
2. Válasszuk a **Szövegszűrők** ⇒ **Kezdődik...** szűrőt!
3. A mezőbe írjuk be a **P** betűt (7.27. ábra)!
4. Kattintsunk az **OK** gombra!

Szűrést több szűrőfeltétel alapján egyszerre is végezhetünk. Ekkor minden oszlop értékeire az adott oszlopban beállított szűrőfeltételek kerülnek alkalmazásra. Vagyis az egyesített szűrőfeltétel az egyes feltételeken elvégzett **ÉS** művelet eredménye. Ha tehát az **Összesen** mezőben **nagyobb, mint 35**, az **Informatika** oszlopban pedig **több, mint 8** van beállítva, akkor a szűrés az **Összesen nagyobb, mint 35 ÉS Informatika nagyobb, mint 8** feltétel alapján történik. A szűrés eredményét a 7.28. ábrán láthatjuk.

A szűrést többféleképpen is megszüntethetjük:

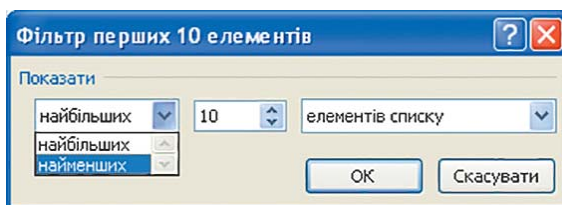
- a **Kezdőlap** ⇒ **Szerkesztés** ⇒ **Rendezés és szűrés** ⇒ **Szűrő törlése** paranccsal;



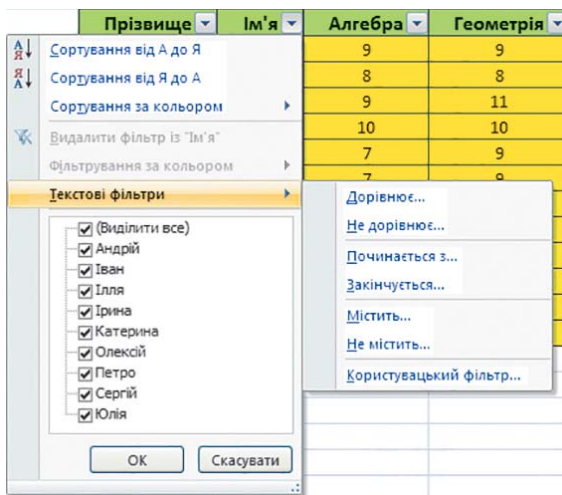
7.23. ábra. Számszűrő beállítása

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійський	Сума
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

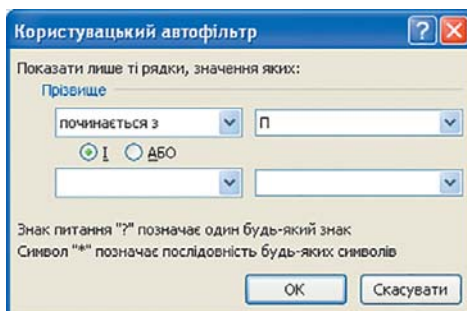
7.24. ábra. A 7.23. ábrán látható szűrőfeltétel alkalmazásának eredménye



7.25. ábra. A Toplista



7.26. ábra. Szövegszűrő kiválasztása



7.27. ábra. Szövegszűrő beállítása

- a **Szűrő törlése** paranccsal abban az oszlopban, ahol szűrés volt beállítva;
- az **Összes kijelölése** jelölőnégyzet bekapcsolásával abban az oszlopban, ahol szűrés volt beállítva, ami után még az **OK** gombra kell kattintani.

A szűrés üzemmódból a szűrés bekapcsolása algoritmusának újbóli alkalmazásával léphetünk ki.

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Туркевич	Сергій	7	9	11	11	38
Штейн	Ірина	9	9	10	9	37

7.28. ábra. Az **Összesen** és **Informatika** mezők egyidejű szűrésének alkalmazása

IRÁNYÍTOTT SZŰRÉS

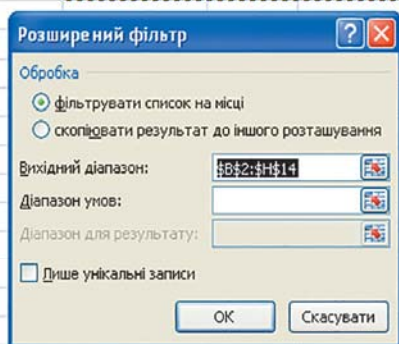
Az automatikus szűrés nem minden esetben oldja meg a szűréssel kapcsolatos problémákat. Nem tudjuk például megoldani, hogy az egyes szűrőfeltételekre a **VAGY** műveletet alkalmazzuk (**az összpontszám több, mint 35 VAGY az informatika több, mint 8**).

Excelben ezeket a problémákat az **irányított szűréssel** oldhatjuk meg. A következőképpen járjunk el.

1. Másoljuk le a táblázat oszlopainak neveit a táblázat szabad celláiba (ahogy azt a 7.29. ábrán az *Összesen* és *Informatika* oszlopokkal tettük)!
2. Az oszlopnevek alá írjuk be a megfelelő szűrőfeltételeket! Ha ezekre a feltételekre az **ÉS** műveletet szeretnénk alkalmazni, akkor egy sorban kell elhelyezkedniük, ha a **VAGY** műveletet, akkor különböző sorokban (a 7.29. ábrán a >35 , >8).
3. Hajtsuk végre az *Adatok* \Rightarrow *Rendezés és szűrés* \Rightarrow *Speciális szűrés* parancsot!
4. A megnyíló **Irányított szűrő** párbeszédablakban:
 - 1) Válasszunk a helyben szűrés és más helyre másolás választógombok közül (a 7.29. ábrán *helyben szűrni*)!
 - 2) A **Listatartomány** mezőbe adjuk meg a szűrendő adatokat tartalmazó cellatartományt (a 7.29. ábrán *B2:H14*)!
 - 3) A szűrőtartomány ablakban adjuk meg, hol helyeztük el a **Szűrőfeltételeket**!



Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	8	9	33
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37



Сума	Інформатика
>35	>8

7.29. ábra. Szűrőfeltételek az Irányított szűrés ablakban

- 4) Ha a **más helyre másolás** lehetőséget választottuk, adjuk meg, hol szeretnénk elhelyezni a szűrt adatokat!
- 5) Kattintsunk az **OK** gombra!

A 7.30. ábrán a 7.29. ábrán látható szűrőfeltétel alkalmazásának eredményét láthatjuk.

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

7.30. ábra. A 7.29. ábrán látható szűrőfeltétel alkalmazásának eredménye

Összefoglalás

A táblázat adatait rendezhetjük, vagyis megváltoztathatjuk azok sorrendjét valamilyen kritériumoknak megfelelően. A táblázatkezelőben **növekvő** és **csökkenő** sorrendet alakíthatunk ki.

Ahhoz, hogy egy tartomány adatait gyorsan sorba rendezzük az első oszlop adatai alapján, a **Kezdőlap** ⇒ **Szerkesztés** ⇒ **Rendezés és szűrés** ⇒ **Rendezés növekvő** (vagy **Rendezés csökkenő**) **sorrendben** parancsát.

A kijelölt tartományt több oszlop adatai figyelembevételével is rendezhetjük. Ehhez a **Kezdőlap** ⇒ **Szerkesztés** ⇒ **Rendezés és szűrés** ⇒ **Rendezés** parancsa után a párbeszédablakban adjuk meg, milyen szempontok alapján szeretnénk rendezni az egyes oszlopokat. Először megtörténik az adatok rendezése az első oszlop adatai alapján, a többi kritérium csak akkor kerül kiértékelésre, ha az első oszlop adatai alapján a sorrend nem döntethető el (azaz az első oszlopban vannak ismétlődő adatok).

A **szűrés a táblázat adatainak bizonyos kritérium alapján történő kiválogatását jelenti**. A szűrés után a táblázatnak csak azokat a sorait láthatjuk, amelyek megfelelnek a **szűrőfeltételnek**. A táblázat többi sora ideiglenesen láthatatlanná válik.

A szűrés előtt helyezzük a kurzort a szürendő tartományba, vagy jelöljük ki a szürendő cellatartományt, majd hajtjuk végre a **Kezdőlap** ⇒ **Szerkesztés** ⇒ **Rendezés és szűrés** ⇒ **Szűrés** (vagy az **Adatok** ⇒ **Szűrés és rendezés** ⇒ **Szűrés**) parancsot.

A szűrést értékek alapján vagy számszűrőkben (szövegszűrőben) leírt feltételek alapján végezhetjük.

A szűrésnél több feltételt is figyelembe lehet venni. Ebben az esetben minden újabb szűrést az előző szűrés eredményére fogja a táblázatkezelő alkalmazni.

A szűrést az **Adatok** ⇒ **Rendezés és szűrés** ⇒ **Szűrő eltávolítása** parancssal szüntethetjük meg.

Ha az egyes szűrőfeltételekre a **VAGY** műveletet szeretnénk alkalmazni, akkor irányított szűrést kell alkalmazni. Ehhez a táblázat szabad celláiban el kell helyezni a szűrőfeltételeket, majd végre kell hajtani az **Adatok** ⇒ **Rendezés és szűrés** ⇒ **Speciális** parancsot, aztán elvégezni az **Irányított szűrő** megfelelő beállítását.

Felelj a kérdésekre!

- 1°. Mit értünk a rendezés alatt **Excel 2007**-ben?
- 2°. Milyen szabályok szerint történik a rendezés **Excel 2007**-ben?
- 3°. Hogy végezzük a gyorsrendezést?
- 4°. Milyen esetekben jelenti a növekvő sorrend a **Rendezés A–Z**-t?
- 5°. Hogy végezzük a rendezést több oszlop adatai alapján?
- 6°. Mi a szűrés?



- 7•. Milyen szűrést végezhetünk számokat tartalmazó cellákon és milyeneket szöveges tartalom esetében?
- 8•. Mi célt szolgál az irányított szűrő?
- 9°•. Hogy adjuk meg a szűrőfeltételeket irányított szűrő esetében?
- 10*•. Mi a különbség az automatikus és az irányított szűrés között?

**Végezd el a feladatokat!**

- 1°•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 7\pukt 7.4\zrazok 7.4.1.xlsx**-et)! A **Munka1** munkalapon a kijevei repülőterekről induló járatok adatait találod. Rendezd az adatokat indulási idő szerint növekvő sorrendbe! A **Munka2** munkalapon kapcsold be a szűrést! Szűrd ki a Boriszpol repülőtéréről induló járatokat! Mentsd a fájlt **feladat 7.4.1.xlsx** néven a mappádba!
- 2•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 7\pukt 7.4\zrazok 7.4.2.xlsx**-et)! A **Munka1** munkalapon egy háromtusa-versenyen elért eredményeket találod. A versenyszámok vívás (pontok), futás (idő) és magasugrás (cm). Rendezd a táblázatot a következő kritériumok szerint: **Pontszám** szerint csökkenő, **Idő** szerint növekvő, **Magasság** szerint csökkenő sorrendbe! A **Munka2** munkalapon kapcsold be a szűrést! Jelenítsd meg azokat a versenyzőket, akik legalább 40 pontot szereztek vívásban, valamint 2,20 és 2,25 m között teljesítettek magasugrásban! Mentsd a fájlt **feladat 7.4.2.xlsx** néven a mappádba!
- 3•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 7\pukt 7.4\zrazok 7.4.3.xlsx**-et)! A **Munka1** munkalapon a kijevei repülőterekről induló járatok adatait találod. Rendezd a táblázatot úgy, hogy az azonos repülőtéréről induló járatok egymás után helyezkedjenek el A–Z sorrendben, ezeken belül pedig a járatok legyenek indulási idő szerint rendezve! A **Munka2** munkalapon kapcsolj be szűrést! Jelenítsd meg a 9 és 15 óra között Varsóba induló járatokat! Mentsd a fájlt **feladat 7.4.3.xlsx** néven a mappádba!
- 4•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 7\pukt 7.4\zrazok 7.4.4.xlsx**-et)! A **Munka1** munkalap táblázat merevlemezekről tartalmaz adatokat. Add meg a következő szűrőfeltételt: a kapacitás legyen több mint 500 GB, az ár 1000 hrvnyva alatt! Mentsd a fájlt **feladat 7.4.4.xlsx** néven a mappádba!
- 5*•. Nyisd meg a tanárod által megadott fájlt (vagy a **Rozdil 7\pukt 7.4\zrazok 7.4.5.xlsx**-et)! A **Munka1** munkalapon a TFT-monitorokról találsz adatokat. Add meg a következő szűrőfeltételt: a képátló nagyobb mint 19", az ár kisebb mint 3000 hrvnyva! Mentsd a fájlt **feladat 7.4.5.xlsx** néven a mappádba!





7.5. RÉSZÖSSZEGEK. FELTÉTELES FORMÁZÁS



1. Mit értünk a táblázat adatainak rendezése alatt? Hogyan hajthatjuk végre?
2. Mit értünk a táblázat adatainak szűrése alatt? Hogyan hajthatjuk végre?
3. Mi az irányított szűrő? Mondj példát az alkalmazására!

RÉSZÖSSZEGEK

A táblázatok gyakran tartalmaznak ismétlődő adatokat. Ilyenek lehetnek például a diákok által látogatott szakkörök, a beszállító vállalatok, árucikkek nevei, vonatok célállomásai stb.

A 7.31. ábrán egy könyvesboltba beszállított és ott leadott könyvekről találunk adatokat.

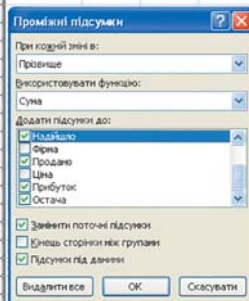
Láthatjuk, hogy a **Név**, **Műfaj**, **Kiadó** oszlopokban gyakran ismétlődnek az adatok. A táblázat adataiból fontos meghatározni, hogy az egyes szerzők műveiből mennyi érkezett be és fogyott el, mennyi fogyott a prózából, mennyi a verseskönyvekből, mennyi könyv érkezett az egyes kiadóktól és azokból mennyi fogyott el stb. Az ilyen elemzés segíti a könyvesbolt tulajdonosait annak megállapításában, hogy a prózai vagy a verses kötetek fogynak-e jobban, kik a népszerű szerzők, melyik kiadó könyvei fogynak a leginkább.

Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача
Бедрик	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1
Андрухович	Юрій	Середмістя	Вірші	8	Селена	5	25	125	3
Андрухович	Юрій	Дванадцять обручів	Проза	7	Орфей	3	30,5	91,5	4
Бердник	Олесь	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0
Бердник	Олесь	Вогнесміх	Проза	12	Тріумф	8	19,8	158,4	4
Бураго	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2
Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2
Вінграновський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5
Вінграновський	Микола	Київ	Вірші	15	Орфей	12	27	324	3
Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2
Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16
Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5
Забіла	Наталія	Ясочина книжка	Проза	15	Орфей	12	12,5	150	3
Забіла	Наталія	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2
Кущнір	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7
Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11
Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13
Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10
Симоненко	Василь	Вибране	Вірші	30	Орфей	25	20	500	5
Гончар	Олесь	Собор	Проза	40	Орфей	28	27,6	772,8	12
Щербак	Юрій	Чорнобиль	Проза	12	Тріумф	3	19,4	58,2	9
Хорунжий	Юрій	Вірую	Проза	14	Селена	2	32	64	12
Хорунжий	Юрій	Таємна грамота	Проза	9	Селена	1	30	30	8
Стельмах	Михайло	Чотири броди	Проза	23	Селена	14	24	336	9
Старицький	Михайло	Історичні романи	Проза	14	Тріумф	5	27,8	139	9

7.31. ábra. A könyvesboltba beszállított és ott eladott könyvek adatai



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача				
2	Андрухович	Юрій	Десять обрубків	Проза	7	Орфей	3	30,5	91,5	4				
3	Андрухович	Юрій	Середмістя	Вірші	8	Селена	5	25	125	3				
4	Бедрик	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1				
5	Бердник	Олесь	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0				
6	Бердник	Олесь	Вогнесімі	Проза	12	Тріумф	8	19,8	158,4	4				
7	Бурого	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2				
8	Вінграновський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5				
9	Вінграновський	Микола	Київ	Вірші	15	Орфей	12	27	324	3				
10	Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2				
11	Гончар	Олесь	Собор	Проза	40	Орфей	28	27,6	772,8	12				
12	Забіла	Наталі	Ясочина книжка	Проза	15	Орфей	12	12,5	150	3				
13	Забіла	Наталі	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2				
14	Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5				
15	Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2				
16	Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16				
17	Кушнір	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7				
18	Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11				
19	Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13				
20	Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10				
21	Симоненко	Василь	Вибране	Вірші	30	Орфей	25	20	500	5				
22	Старицький	Михайло	Історичні романи	Проза	14	Тріумф	5	27,8	139	9				
23	Стельмах	Михайло	Чотири броди	Проза	23	Селена	14	24	336	9				
24	Хорунжий	Юрій	Віруко	Проза	14	Селена	2	32	64	12				
25	Хорунжий	Юрій	Таємна грамота	Проза	9	Селена	1	30	30	8				
26	Щербак	Юрій	Чорнобиль	Проза	12	Тріумф	3	19,4	58,2	9				



7.32. *ábra.* A **Családnév** alapján rendezett táblázat és a **Részösszegek** ablak

Hogy a feltett kérdésekre válaszoljunk, az adatokat az ismétlődő értékek szerint csoportosítani kell, majd el kell végezni az összegek, átlagok, maximális és minimális elemek stb. meghatározását a csoportokon belül.

Ennek egyik módja a **részösszegek** alkalmazása.

A részösszegek kiszámítása céljából a következőt kell tenni:

1. Jelöljük ki azokat a cellákat, amelyeket ki szeretnénk értékelni!
2. Rendezzük az adatokat a szerint az oszlop szerint, amely a csoportosítandó adatokat tartalmazza!
3. Hajtsuk végre az **Adatok** ⇒ **Tagolás** ⇒ **Részösszeg** parancsot!
4. A **Részösszegek** ablakban (7.32. *ábra.*) válasszuk ki az alkalmazandó függvényt (**Összeg**, **Darab**, **Átlag**, **Maximum** stb.)!
5. Az **Összegzendő oszlopok** listában jelöljük ki azokat az oszlopokat, amelyekre nézve az összegzéseket el szeretnénk végezni!
6. Kattintsunk az **OK** gombra!

A 7.33. *ábra.* a fent vázolt algoritmus végrehajtásának eredményét mutatja. A táblázat adatai a **Családnév** mező alapján vannak csoportosítva, minden csoport alatt létrejött egy **Összező** sor, meg vannak határozva az **Érkezett**, **Eladott**, **Bevétel** és **Maradék** mezőkben az összegek. Mindegyik csoport teljesen kifejtett, amiről a sor előtt látható **-** jel is tanúsodik. Ha erre a jelre kattintunk, a megfelelő csoport összecsukódik, csak az **Összező** sor marad látható, a gomb pedig **+** alakra vált. Ha erre a gombra kattintunk, a csoport újra ki lesz fejtve.

A kibontás/összezáras gombok felett találjuk az **1**, **2** és **3** gombokat, amelyek a különböző csoportszintek gyors kibontására/összecsukására szolgálnak. Az első szint az egész táblázat, a második a csoportok, a harmadik a csoporttagok szintje.

	A	B	C	D	E	F	G	H	I	J
1	Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача
2	Андрухович	Юрій	Дванадцять обрuchів	Проза	7	Орфей	3	30,5	91,5	4
3	Андрухович	Юрій	Середмістя	Вірші	8	Селена	5	25	125	3
4	Андрухович	Підсумок			15		8		216,5	7
5	Бедрик	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1
6	Бедрик	Підсумок			5		4		62	1
7	Бердник	Ольсь	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0
8	Бердник	Ольсь	Вогнесміх	Проза	12	Тріумф	8	19,8	158,4	4
9	Бердник	Підсумок			22		18		375,4	4
10	Бураго	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2
11	Бураго	Підсумок			3		1		15	2
12	Вінграновський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5
13	Вінграновський	Микола	Київ	Вірші	15	Орфей	12	27	324	3
14	Вінграновський	Підсумок			36		28		804	8
15	Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2
16	Вороний	Підсумок			7		5		130	2
17	Гончар	Ольсь	Собор	Проза	40	Орфей	28	27,6	772,8	12
18	Гончар	Підсумок			40		28		772,8	12
19	Забіла	Наталя	Ясочина книжка	Проза	15	Орфей	12	12,5	150	3
20	Забіла	Наталя	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2
21	Забіла	Підсумок			27		22		237	5
22	Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5
23	Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2
24	Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16
25	Забужко	Підсумок			63		40		1296	23
26	Кушнір	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7
27	Кушнір	Підсумок			11		4		68	7
28	Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11
29	Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13
30	Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10
31	Павличко	Підсумок			66		32		784	34

7.33. ábra. Családnév alapján rendezett táblázat részösszegekkel

FELTÉTELES FORMÁZÁS

A *feltételes formázás* az egyik módja annak, hogy a táblázatban bizonyos feltételeket kielégítő értékeket kiemeljünk.

Feltételes formázás segítségével a bevitt adatokat tartalmazó celláknak automatikusan meg tudjuk változtatni a formátumát különböző feltételek alapján.

Az elektronikus naplóban különböző színekkel lehet kiemelni az alacsony, közepes és magas eredményességű tanulókat. Azokat a cellákat, ahol az 1 és 3 közötti számok találhatóak, pirossal, a 4 és 6 közöttiek zölddel, a 7 és 9 közöttiek sárgával, a 10 és 12 közöttiek kékekkel lesznek kiemelve.

Felhívjuk a figyelmeteket, hogy a feltételes formázás nem rejti el az adatokat, mint a szűrés, csak meghatározott módon kiemeli az adott feltételt kielégítő cellákat.

Az **Excel 2007**-ben 5 módja van a feltételes formázásnak (7.34. ábra):

- cellakijelölési szabályok megadása;
- legfelső és legalsó értékeinek kiszámítására vonatkozó szabályok megadása;
- adatsávok;
- színskálák;

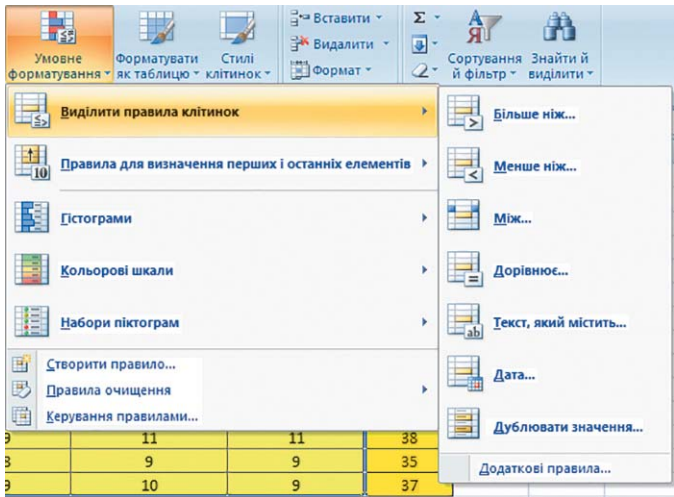


- ikonkészletek.

A **cellakijelölési szabályokat** a következőképpen készítjük:

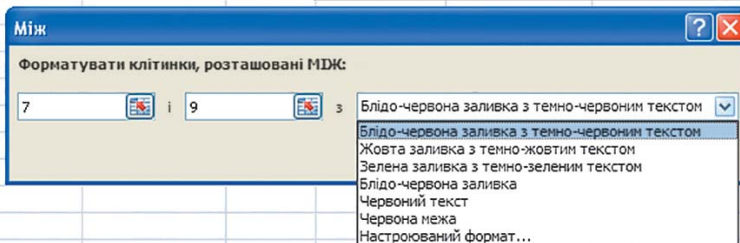
1. Jelöljük ki a formázandó cellatartományt!
2. Hajtsuk végre a **Kezdőlap** ⇒ **Stílusok** ⇒ **Feltételes formázás** parancsot!
3. A **Feltételes formázás** listán (7.34. ábra) válasszuk ki a megfelelő formázási módszert!
4. A listából jelöljük ki a formázás szabályát!
5. A megnyíló párbeszédablakban válasszunk formátumot arra az esetre, amikor a feltétel teljesül (7.35. ábra)! A formátumot kiválaszthatjuk a listából, de lehetőség van egyéni formátum kialakítására.
6. Kattintsunk az **OK** gombra!

A 7.35. ábrán látható mintán a **Két érték között 7 és 9** feltétel, a kész formátumok listája és a formázás előképe látható.



7.34. ábra. A **Feltételes formázás** gomb és a **Cellakijelölési szabályok**

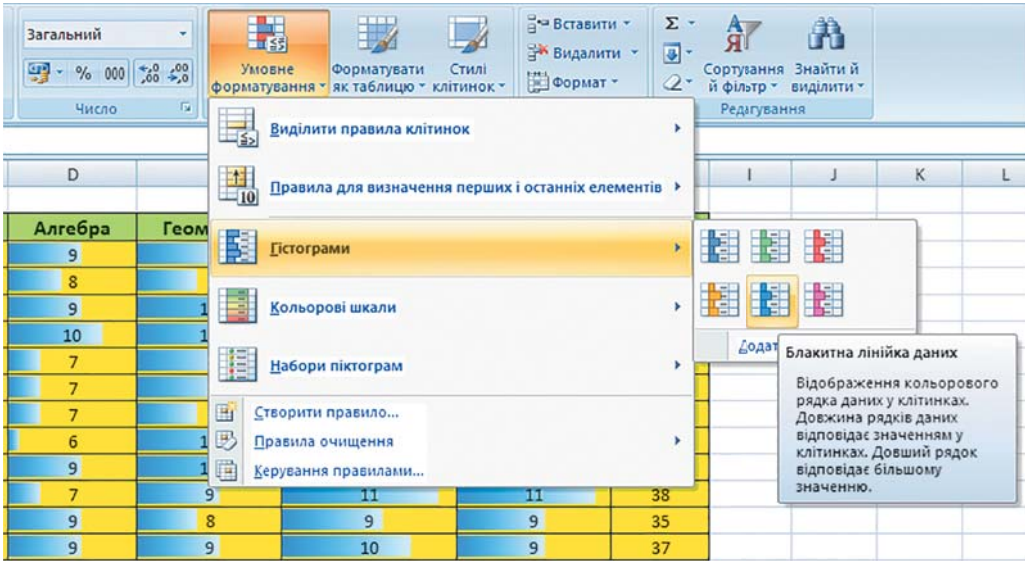
Príзвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	8	9	33
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37



7.35. ábra. A kijelölt cellatartomány, a **Két érték között** ablak és a formátumok listája

7. fejezet

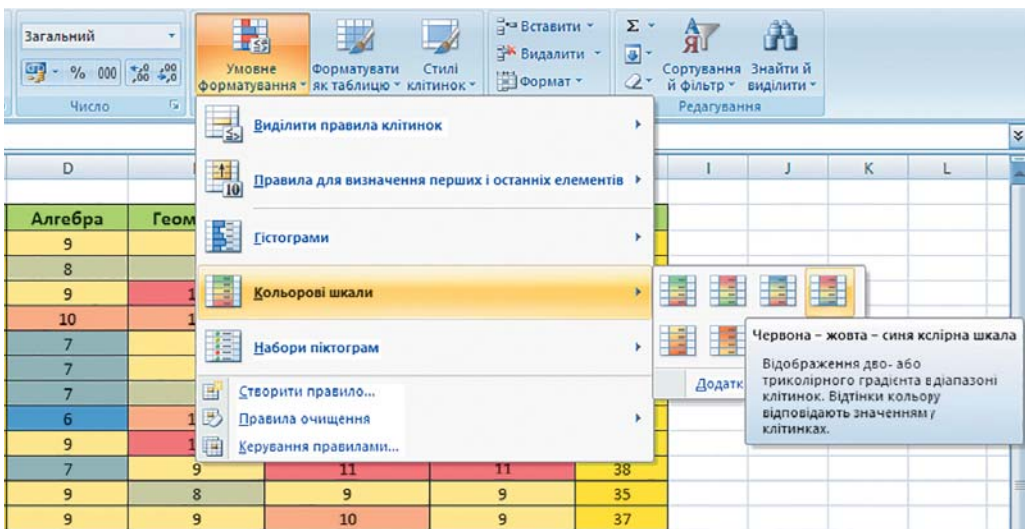
A feltételes formázás másik módja az **Adatsávok** alkalmazása, amikor is a cella a tartalmozott értékkel arányos méretű színes oszloppal van formázva (7.36. ábra)



7.36. ábra. Feltételes formázás adatsávokkal

A **Ші́нскі́лалі́к** alkalmazásakor (7.37. ábra) a kijelölt cellatartomány az azonos értékeket tartalmazó cellái azonos színnel lesznek formázva.

Alkalmazhatjuk ezeken kívül még az **Ikonkészletet** is feltételes formázásként. Ha ezt tesszük, akkor a kijelölt cellatartomány minden cellájában az ikonkészlet egy ikonja jelenik



7.37. ábra. Feltételes formázás ші́нскі́лалі́ккал



meg. Egy konkrét ikon megjelenése a cellában azt jelenti, hogy a cellába tárolt érték kielégíti a csoportban az adott ikonhoz tartozó formázási feltételt.

Itt hívjuk fel a figyelmeteket arra, hogy a kijelölt cellatartományra egymás után több feltételes formázást is rendelhetünk.

A feltételes formázást a **Feltételes formázás** ⇒ **Szabályok törlése** ⇒ **Szabályok törlése a kijelölt cellákból (a teljes munkalapról)** paranccsal szüntethetjük meg.



Felelj a kérdésekre!

- 1°. Mik a részösszegek? Mikor célszerű az alkalmazásuk?
- 2°. Hogy számítjuk ki a részösszegeket **Excel 2007**-ben?
- 3°. Milyen függvényeket használhatunk a részösszegekben?
- 4°. Mi a feltételes formázás? Mire alkalmazzuk?
- 5°. Milyen feltételes formázásokat használhatunk?
- 6°. Hogy végezzük a feltételes formázást?
- 7°. Hogy szüntetjük meg a feltételes formázást?



Végezd el a feladatokat!



- 1°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.1.xlsx**-et)! A **Munka1** munkalapon a merevlemezekről találsz adatokat. Határozd meg a részösszeget a **Gyártó cég** oszlop értékei alapján! Mentsd a fájlt **feladat 7.5.1.xlsx** néven a mappádba!
- 2°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.2.xlsx**-et)! A **Munka1** munkalapon a monitorokról találsz adatokat. Határozd meg a részösszeget a **Képtáló** oszlop értékei alapján! Mentsd a fájlt **feladat 7.5.2.xlsx** néven a mappádba!
- 3°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.3.xlsx**-et)! A **Munka1** munkalapon a TFT-monitorokról találsz adatokat. Emeld ki rózsaszínnel a 21". Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.4.xlsx**-et)! A **Munka1** munkalapon a merevlemezekről találsz adatokat. Emeld ki kék színnel azokat, amelyeknél a cache-memória meghaladja a 8 Mb-ot! Mentsd a fájlt **feladat 7.5.4.xlsx** néven a mappádba!
- 4°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.5.xlsx**-et)! A **Munka1** munkalapon az utóbbi hónap során Kijevben mért hőmérsékleti adatokat találsz. Formázd **Adatsávok (ibolyaszínű skála)** segítségével a hőmérsékleteket! Mentsd a fájlt **feladat 7.5.5.xlsx** néven a mappádba!
- 6°. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\punto 7.5\zrazok 7.5.6.xlsx**-et)! Keresd meg a lakóhelyedre vonatkozó népességi adatokat az interneten! A **Munka1** munkalapon helyezd el az utóbbi 5 év adatait! Formázd az adatokat **Ikonkészlet** segítségével (**közlekedési lámpa**)! Mentsd a fájlt **feladat 7.5.6.xlsx** néven a mappádba!



16. SZÁMÚ GYAKORLATI MUNKA

**Adatok rendezése a táblázatokban. Automatikus és irányított szűrés.
Feltételes formázás**

Figyelem! *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Nyisd meg a tanárod által megadott fájlt (például a **Rozdil 7\prakticsna 16\zrazok-pr.16.xlsx**-et)!
2. A **Munka1** munkalapon rendezd az adatokat a **Családnév** szerint növekvő sorrendbe!
3. A **Munka2** munkalapon végezd el a rendezést a következő kritériumok alapján: **Családnév** (növekvő), **Név** (növekvő) **Születési év** (csökkenő)!
4. A **Munka3** munkalapon szűrd ki azokat a tanulókat, akik magassága meghaladja az 1m 80 cm-t!
5. A **Munka4** munkalapon szűrd ki azokat a tanulókat, akik a rajz vagy az ének szakköröket látogatják!
6. A **Munka5** munkalapon határozd meg a részösszegeket a **Testsúly** alapján, amelyek megállapítják az átlagos és maximális testsúlyt!
7. A **Munka6** munkalapon állíts be feltételes formázást! A 80 kg-ot meghaladó testsúlyokat piros, a 75 kg-nál kisebbeket kék színnel emeld ki.
8. Mentsd a munkafüzetet **gyakorlati 16.xlsx** néven a mappádba!



8. fejezet

Kompetenciafeladatok megoldása, projektmunkák végrehajtása

Ebben a fejezetben megismerkedtek azzal:

Milyen lépésekből áll az infokommunikációs eszközökkel támogatott problémamegoldás különböző tevékenységi területeken?

Milyen stratégiákat alkalmazhatunk az információ keresése során?

Milyen tevékenységeket rejt az infokommunikációs eszközökkel támogatott kompetenciafeladatok megoldása?

Hogyan hajtunk végre tanulmányi projekteket informatikából?



8.1. KOMPETENCIAFELADATOK MEGOLDÁSA



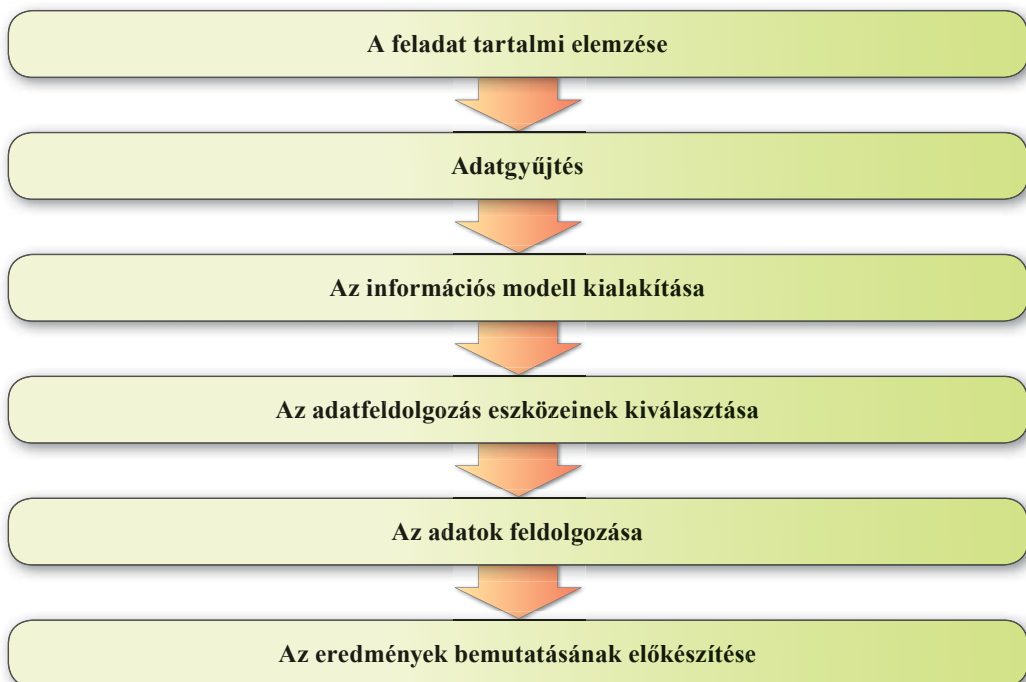
1. Mi a kompetenciafeladat?
2. Milyen algoritmusok szerint oldanak meg problémákat különböző emberi tevékenységi területeken?
3. Milyen módon mutathatjuk be a kompetenciafeladat megoldásának eredményeit?

A KOMPETENCIAFELADATOK MEGOLDÁSÁNAK LÉPÉSEI

A 7. osztályban már megismertétek, milyen feladatokat nevezünk kompetenciafeladatoknak. Idézzük emlékezetünkbe, hogy a **kompetenciafeladat** olyan feladat, amelynek megoldása során a megszerzett tudás gyakorlati alkalmazásának elsajátítása a cél. Az ilyen feladat megoldása a különböző tantárgyak elsajátítása során megszerzett tudás, képesség és tapasztalat alkalmazása egy, a mindennapi életből vett probléma kezelésére. A kompetenciafeladatok ebből adódóan ritkán korlátozódnak egyetlen tantárgy keretein belülre, többnyire tantárgyközi jellegűek.

Már megismerkedtetek a kompetenciafeladatok megoldásának a lépéseivel (8.1. ábra). Vizsgáljuk meg újra ezeket a lépéseket részletesebben.

A **feladat tartalmi elemzése** azt jelenti, hogy kiemeljük a feladat megfogalmazását, a kiindulási adatokat, a megoldáshoz szükséges még rendelkezésre nem álló adatokat, körülírjuk, mit tekintünk majd a feladat megoldásának.



8.1. ábra. A kompetenciafeladatok megoldásának lépései



Adatgyűjtés, ami lényegében két részfeladatot jelent:

- Az **adatgyűjtés stratégiájának** kidolgozása, amelynek során:
 - kitűzzük az információgyűjtés célját és feladatát;
 - a keresés során használt forrásokat;
 - a kulcsszavakat;
 - a talált információk értékelésének módját.
- Az adatgyűjtés végrehajtása a kidolgozott stratégia alapján, többek között internetes forrásokból, ezek összevetése a kitűzött céllal, és értékelése a kidolgozott kritériumok szerint.

Stratégia (görög eredetű szó, jelentése: *hadvezetés művészete*) – általános, nem részletezett távlati terv és a megvalósításának módja.

Az **információs modell kialakítása** a feladat megoldási tervének összeállítását jelenti, amiben meghatározzuk a feladat megoldásához vezető tevékenységek sorát.

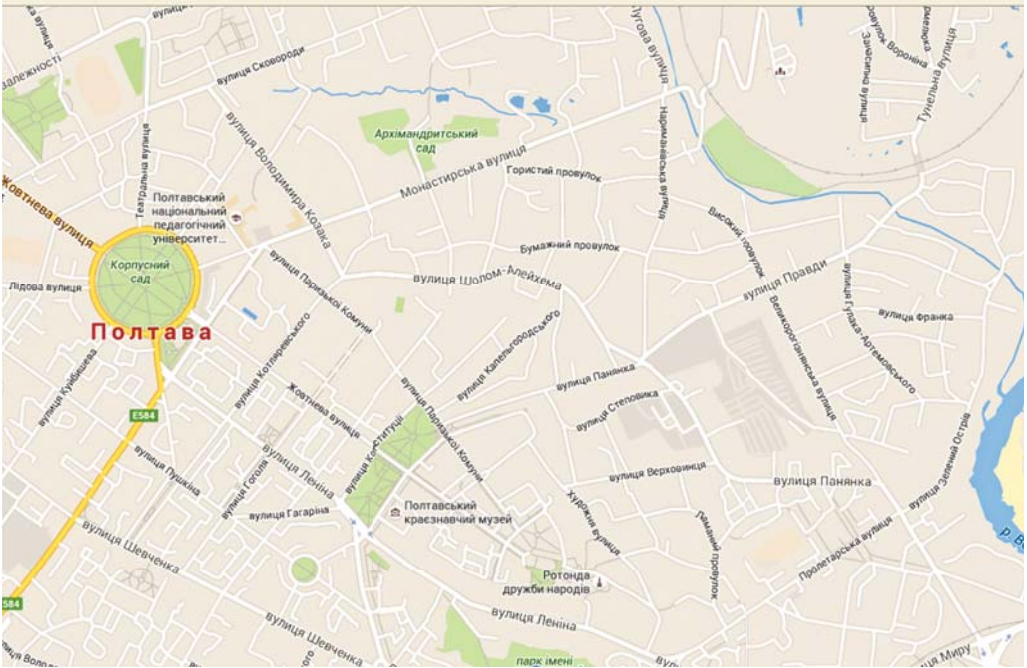
Az **adatifeldolgozás eszközeinek kiválasztása** a kitűzött feladat és az összegyűjtött információk függvénye. A lehetséges eszközök a szövegszerkesztő, a képszerkesztő, a prezentáció-szerkesztő, a táblázatkezelő, valamilyen algoritmus-végrehajtó eszköz, multimédia-szerkesztő stb. Sok esetben több eszköz alkalmazására lehet szükség.

Az **adatok feldolgozása** a kiválasztott eszközök alkalmazását jelenti az információs modellben megfogalmazott feladat megoldására. Ennek során történik meg a különböző forrásokból származó adatok összevetése, a nem adekvát és nem lényeges információk kiszűrése, az adatok strukturálása, új adatok előállítás a rendelkezésre állók alapján, valamint a következmények megfogalmazása.

Az **eredmények bemutatásának előkészítése** azt jelenti, hogy a hallgatóság számára előkészítünk egy összefoglaló dokumentumot, amelyben bemutatjuk a feladat megoldását és az elért eredményeket. Ennek elkészítése során fontos szempont, hogy a bemutatás logikus és tömör legyen, a következtetések logikusak legyenek és a bemutatás során ne veszítünk szem elől a dokumentum elkészítésének célját.

KOMPETENCIAFELADAT MEGOLDÁSÁNAK PÉLDÁJA

Tekintsük át a kompetenciafeladat megoldásának lépéseit egy konkrét példa alapján: *Osztályotok a tanulmányi gyakorlat idején egynapos kirándulást tervez Ukrajna egyik megyei központjába. A kirándulás célja, hogy megismerkedjete a 8. osztályban a történelem tanulása során megismert eseményekhez kapcsolható helyszínekkel, emlékművekkel. Az úti cél kiválasztása során ki kell deríteni, milyen, a tanult korhoz kapcsolható történelmi emlékek lehetnek ezekben a városokban, majd ki kell dolgozni az egynapos kirándulás útvonalát. Az útvonalat kiegészíthetjük történelemből vagy az irodalomból ismert személyiségekhez kötődő emlékhelyek meglátogatásával. Az osztály minden tanulója részt vesz az adatgyűjtésben. A te feladatod a Poltava városára vonatkozó információk összegyűjtése. Ehhez rendelkezésedre áll Poltava városának térképe (8.2 ábra).*



8.2. ábra. Poltava térképe

1. A feladat tartalmi elemzése során meg kell határozni:

a) Mit tekintünk a feladat megoldásának?

- A Poltaván teendő kirándulás útvonalának feltüntetése a térképen.
- Biztosítani, hogy a kitűzött útvonal teljesíthető legyen az egynapos kirándulás során.
- Bemutatni az útvonalon elhelyezkedő történelmi emlékhelyeket.

b) Milyen, a feladat megoldásához szükséges adatokat tartalmaz a feladat feltétele?

- A kiválasztott helyszíneknek kapcsolódniuk kell az Ukrajna történelmének a 8. osztályban tanulmányozott korszakához (XVI–XVII. század), valamint ismert történelmi vagy irodalmi személyiségekhez.
- Az útvonalnak egy nap alatt bejárhatónak kell lenni.

c) Milyen régebbi ismeretek felidézésére van szükség a feladat megoldásához?

- Ukrajna XVI–XVIII. századi történelmének mely eseményei kapcsolódnak Poltavához? (*Poltavai csata – 1709 – a történelem tankönyv alapján; H. Szkovoroda, T. Sevcenko, M. Csuhray, I. Kotljarevszkij – a történelem és irodalom tankönyv adatai alapján.*)
- A mai Poltava mely emlékhelyei kapcsolhatók a történelmi eseményekhez és személyiségekhez?
- Mely, a történelmi eseményekhez és közismert személyiségekhez kapcsolható helyszínek járhatók be egynapos gyalogtúra során?



2. Az adatgyűjtés során **információkat kell keresni** Poltava emlékhelyeiről, valamint azok elhelyezkedéséről a város területén.

a) Határozzuk meg a keresés stratégiáját:

- A **keresés célja**: a Poltava területén elhelyezkedő, Ukrajna XVI–XVIII. századi történelméhez kapcsolódó történelmi emlékhelyek adatainak összegyűjtése.
- **Keresési feladat**: megkeresni az emlékhelyek leírását, címét, fényképét, az emlékhelyeket tartalmazó térképeket.
- A történelmi emlékhelyeket leíró **források**: enciklopédiák, útikönyvek, tájékoztató könyvek, internet. A történelmi emlékek leírásait a város honlapjáról, a városi múzeumok, galériák webhelyeiről, az enciklopédiák, turisztikai szolgáltatók honlapjairól, a város látogatóinak személyes oldalairól, blogjairól gyűjtjük össze.
- **Kulcsszavak** a történelmi emlékhelyek kigyűjtéséhez:
 - Poltava a XVI–XVIII. században;
 - Poltava történelmi emlékhelyei;
 - Poltava történelmi szerepe;
 - irodalmi Poltava;
 - kirándulás Poltavába;
 - a poltavai csata emlékhelyei;
 - a poltavai csata múzeuma.
- Az összegyűjtött adatok **értékelésének szempontjai**:
 - az adatok legyenek aktuálisak, ne elavultak;
 - legyen lehetőségünk összevetni más források adataival;
 - a képek legyenek jó minőségűek.

b) Az adatgyűjtés és értékelés a következő eredményekhez vezetett:

- A XVI–XVIII. századi poltavai események emlékhelyei:
 - *A poltavai csatamező* történelmi–kulturális emlékpark (battle-poltava.org);
 - A poltavai csata múzeuma (tourism.poltava.ua/muzei/Muzej_istorii_Poltavskoi_bitvi);
 - Poltava történelmi emlékhelyei a *poltavai csatamezőn* kívül (histpol.pl.ua/ru/poltava-istoricheskije-ocherki/poltavskaya-bitva-i-ejo-pamyatniki).
- Történelmi és irodalmi személyiségekhez köthető emlékhelyek Poltava területén:
 - I. Kotljarevszkij emlékmúzeuma;
 - T. Sevesenko, M. Csuhráj, I. Kotljarevszkij emlékművei.

3. Az összegyűjtött adatok alapján a következő információs modellt állíthatjuk össze:

Az Ukrajna XVI–XVIII. századi történelméhez kapcsolható emlékhelyek Poltava területén *A poltavai csatamező* történelmi–kulturális emlékpark, valamint a mellette el-



8.3. ábra. Prezentáció Poltava történelmi emlékeinek bemutatásáról

helyezkedő *A poltavai* csata múzeumában található, ezek mind a város északi peremén helyezkednek el. Itt tíz gránit obeliszket találunk az összecsapás helyén, a svédeknek az oroszok, illetve a svédeknek a svédok által állított emlékmű, a szampszonijevói templom stb. A történelmi-kulturális emlékparkba a kirándulást *A poltavai* csata múzeumában rendelhetjük.

A város központja a Dicsőség emlékművétől az Elesett ukrán kozákok emlékművén, a szpasszkij templom, a Sevcsenko emlékmű, a Kotljarevskij-múzeum, a fehér rotunda útvonalon gyalogosan bejárható, utóbbiról szép kilátás nyílik a Szent Kereszt kolostorra.

Az útvonal hossza kb. 2 kilométer, ami az egynapos kirándulás során kényelmesen bejárható.

4. A feladat eredményeinek bemutatásához feltétlenül szükséges az **adatfeldolgozás eszközeinek kiválasztása**.

A térképre képszerkesztő segítségével vihetjük fel az útvonalat. Az útvonalon elhelyezkedő emlékhelyek bemutatása prezentációszerkesztőben, szövegszerkesztőben vagy videoszerkesztőben végezhető el.

5. Az **adatok feldolgozása**, vagyis az összefoglaló dokumentum elkészítése a kiválasztott szoftvertől függ.

6. Az eredmények megjelenítésének két lehetséges változatát a 8.3. ábrán, illetve a 8.4. ábrán mutatjuk be.



8.4. ábra. Poltava térképe a tervezett útvonallal

! Összefoglalás

A feladat megoldása a következő lépésekből áll:

1. A feladat tartalmi elemzése.
2. Adatgyűjtés.
3. Az információs modell felállítása.
4. Az adatok feldolgozása.
5. A szükséges eszközök kijelölése.
6. Az eredmény bemutatása.

Az **adatgyűjtés stratégiájának** kidolgozása, amelynek során:

- kitűzzük az információgyűjtés célját és feladatát;
- a keresés során használt forrásokat;
- a kulcsszavakat;
- a talált információk értékelésének módját.

? Felelj a kérdésekre!

1. Mi a kompetenciafeladat?
2. Milyen lépésekből áll a kompetenciafeladat megoldása?
3. Milyen tevékenységekből áll a feladat tartalmi elemzése?
4. Minek az alapján döntjük el, hogy az adott információ mennyire hasznos a feladat megoldása szempontjából?

- 5°. Miben áll az információkeresés stratégiájának kidolgozása?
- 6°. Milyen formában készíthetjük el a feladat információs modelljét?
- 7°. Milyen tevékenységeket végzünk el az adatfeldolgozás során?
- 8°. Mitől függ a megoldás bemutatásának módja?
- 9°. Milyen számítógépes feladat-feldolgozási eszközökkel ismerkedtetek meg a 8. osztályban?



Végezd el a feladatokat!

- 1°. Készítsd el Poltava történelmi emlékhelyeinek bejárását bemutató térképet a 8.4. ábra alapján! A szükséges szöveges és képfájlokat a **Rozdil 8\pukt 8.1\zavdannya 8.1.1** mappában találod. A térkép készítéséhez használd:

a*) a **Paint** képszerkesztőt és **PowerPoint** prezentációszerkesztőt;

b*) a vázlatot **Word** szövegszerkesztőben, az emlékhelyek videós bemutatását pedig a videoszerkesztőben;

c*) az útvonal animált bemutatását **Scratch**-ben, az emlékhelyek bemutatását **Word**-ben.

- 2°. Készítsd el a megyeszékhely XVI–XVIII. századi történelmi emlékhelyeinek egy napos gyalogtúra kereteiben történő bemutatását! Az útvonal hosszának meghatározására használj táblázatkezelőt, a többi alkalmazott szoftvert válaszd ki a saját elképzeléseid szerint!



- 3*. Készítsd el egy tetszőlegesen választott megyeszékhely XVI–XVIII. századi ukrán történelmi emlékhelyeinek egy napos gyalogtúra keretében történő bemutatását! Akirándulás költségeinek (szállás, utazás, múzeumi belépőjegyek) meghatározására használj táblázatkezelőt, a többi alkalmazott szoftvert válaszd ki a saját elképzeléseid szerint!



- 4°. Keresd meg a szükséges információkat, és készíts egy összefoglalót a mosás helyes módja kiválasztásának bemutatására a ruhán feltüntetett jelzések alapján; prezentációkészítőben készíts egy bemutatót, amelyből kiderül, hogyan választunk vasalási módot az anyag minőségétől és a feltüntetett jelölésektől függően!

- 5°. Keresd meg a szükséges információkat, és mutasd be földrajzórán a lakosság megoszlását Ukrajnában! A bemutatást segítő diagramokat készítsd el a táblázatkezelőben!

- 6°. Keress képeket, amelyek segítségével bemutatható, milyen ételeket készítenek Ukrajna egyes régióiban a húsvéti ünnepek alkalmából! Készíts a szövegszerkesztőben egy bemutatót ezekből a munkaórára! Készíts **Lazarus**-projektet, amelyben meghatározod, mennyibe kerül a tradicionális magyar húsvéti ételek elkészítése!



- 7°. Keresd meg a tradicionális húsvéti ételek receptjeit! Mutasd be rövid videón az ételek elkészítését! Az ételek bekerülési költségének meghatározására használd a táblázatkezelőt!





- 8*. Osztályoddal erdei kirándulást terveztek. Keres információkat arról, hogyan tájékozódhatunk ez erdőben iránytű nélkül, illetve milyen túlélési praktikákat alkalmazhatunk nehéz időjárási körülmények között! Készíts ezekről a szövegszerkesztőben rövid emlékeztetőt! A táblázatkezelő segítségével határozd meg a kiránduláshoz szükséges ivóvíz mennyiségét a résztvevők számának és a tervezett időtartamának függvényében! Határozd meg a víz költségét is a víz mennyisége és a kiválasztott csomagolás függvényében!
- 9*. A családod elhatározta a szobád felújítását. A te feladatod a tapéták kiválasztása, a szükséges mennyiség meghatározása, ezek költségének kiszámítása, a szobád leendő külalakjának bemutatása a szüleidnek. A megoldás eszközeit és az elképzeléseid bemutatásának eszközeit válaszd ki a belátásod szerint!



8.2. Tanulmányi projektek



1. Milyen órák keretében végeztetek tanulmányi projekteket?
2. Milyen lépései vannak a projekt megvalósításának?
3. Milyen formában mutathatjuk be a projekt eredményeit?

A PROJEKT MEGVALÓSÍTÁSÁNAK LÉPÉSEI

Ahogy már tanultátok, a projektmunka a diákok önállóan végzett tevékenysége, melynek során az érintettek valamely probléma megoldásához keresnek információkat, feldolgozzák azokat és bemutatják a munkájuk eredményeit.

Idézzük fel, milyen lépésekből áll a projektmunka:

- a feladat feltételének elemzése (a célok és feladatok kitűzése);
- az információs modell megalkotása (a megvalósítás tervének kidolgozása);
- a szükséges információk, adatok összegyűjtése;
- a feldolgozás eszközeinek kiválasztása;
- az információk feldolgozása, elemzése;
- a projekt bemutatásához szükséges eszközök kiválasztása;
- a projekt eredményeinek összegzése;
- a projekt védelme.

A projekten dolgozhatunk önállóan, de a jellemző az, amikor egy projekten több ember dolgozik kisebb csoportban.

Az információs modell felállítása, az információ keresése, a beszámoló elkészítése során a csoport tagjai különböző információs technológiákat használhatnak a kommunikáció során.

A csoport tagjai a munkájuk eredményét elhelyezhetik a weboldalaikon, blogjaikban, ilyen módon segítve a közös munka elkészültét.

Tanulmányi projekteket több tantárgy tanulása során készítettetek. Ezek egy része egy konkrét tantárgyhoz köthető, más része több tantárgyat is érint. Az idei tanévben egyéni és csoportos projektmunkákat ajánlunk az informatika tudományterületről.

INFORMATIKAI TANTÁRGYI PROJEKT

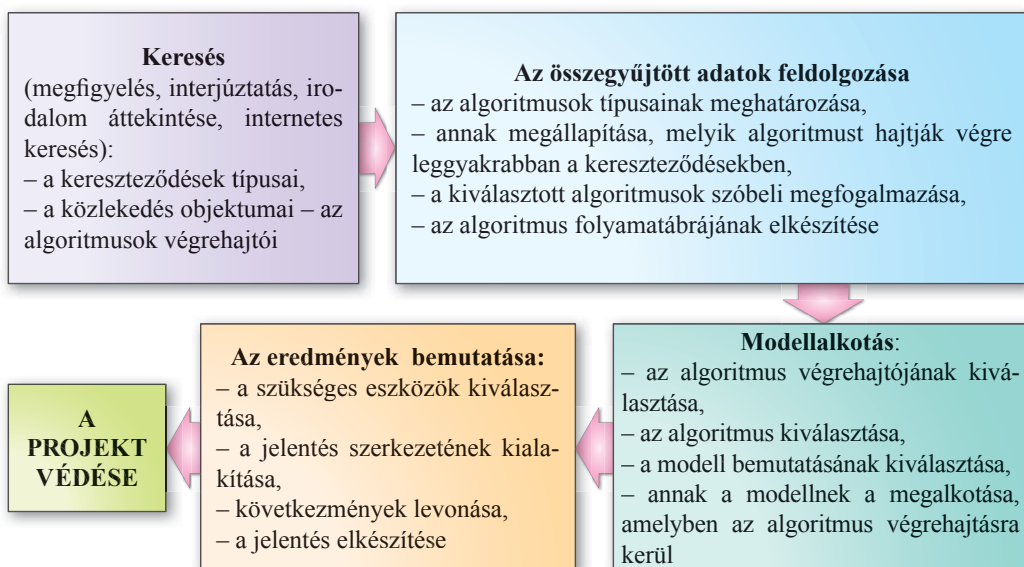
Vizsgáljuk meg a projektmunka készítésének folyamatát az *Algoritmusok körülöttünk* projekt kapcsán. A projekt a hatodik, *Objektumokkal és mennyiségekkel dolgozó algoritmusok* fejezethez kapcsolódik.

A projekt célja: algoritmusok keresése a környezetünkben, és számítógépes modellek készítése az algoritmus szereplői számára.

Az egyik csoport megvizsgálhatja az *Algoritmusok kereszteződése* témát. A csoport feladatai a következők lesznek:

- a keresztezések típusainak áttekintése, valamint annak meghatározása, milyen algoritmus-végrehajtók kaphatnak a keresztezésekben szerepet;
- az egyes végrehajtók által követendő algoritmusok meghatározása;
- egy végrehajtó kiválasztása, amelyeknek algoritmusát számítógépen elkészítik;
- a projekt eredményeinek bemutatása.

A feladat alapján elkészítjük a **megoldás tervét**. Ebben a tervben konkretizáljuk azokat a tevékenységeket, amelyeket a munka végrehajtása során el kell végezni (8.5. ábra).



8.5. ábra. A projekt végrehajtásának terve

A szükséges információk összegyűjtése sokféleképpen történhet: megfigyelés, interjúztatás, keresés az interneten és nyomtatott anyagokban. Projektünk eredménye a keresztezések és az abban közlekedők, illetve az általuk követett algoritmusok (leírások, képek, vázlatok) bemutatása lesz.

A projektben részt vevő diákok és a projekt vezetője az elektronikus postát fogják használni az együttműködésük során.

Az összegyűjtött információk feldolgozásának eredménye lehet egy vázlat vagy egy táblázat, ahol strukturáltan mutatjuk be ezeket. Ez lehet például a szövegszerkesztőben elkészített táblázat (8.1. táblázat).



Algoritmusok kereszteződése

<i>Kereszteződés</i> <i>Résztevők</i>	<i>Szabályozott kereszteződések</i>			<i>Szabályozatlan kereszteződések</i>
	<i>Közlekedési lámpa által szabályozott</i>	<i>Fénysorompóval és sorompóval szabályozott</i>	<i>Közlekedési lámpával, fénysorompóval és félsorompóval szabályozott</i>	
Gyalogos	Elágazásos algoritmus	Elágazásos algoritmus	Elágazásos algoritmus	Elágazásos algoritmus
Autóvezető	Elágazásos algoritmus	Elágazásos algoritmus	Elágazásos algoritmus	Elágazásos algoritmus
Mozdonyvezető	–	Elágazásos algoritmus	Elágazásos algoritmus	–
Közlekedési lámpa	Ciklikus algoritmus	–	Ciklikus algoritmus	–
Fénysorompó		Elágazásos ciklikus algoritmus	Elágazásos ciklikus algoritmus	
Félsorompó	–	–	Elágazásos algoritmus	–

A táblázatba foglalt adatok arra engednek következtetni, hogy a közlekedés résztvevői a kereszteződésekben leginkább elágazásos algoritmusokat hajtanak végre. Ez a közlekedés biztonságát szavatoló közlekedési szabályok követelményei miatt van így.

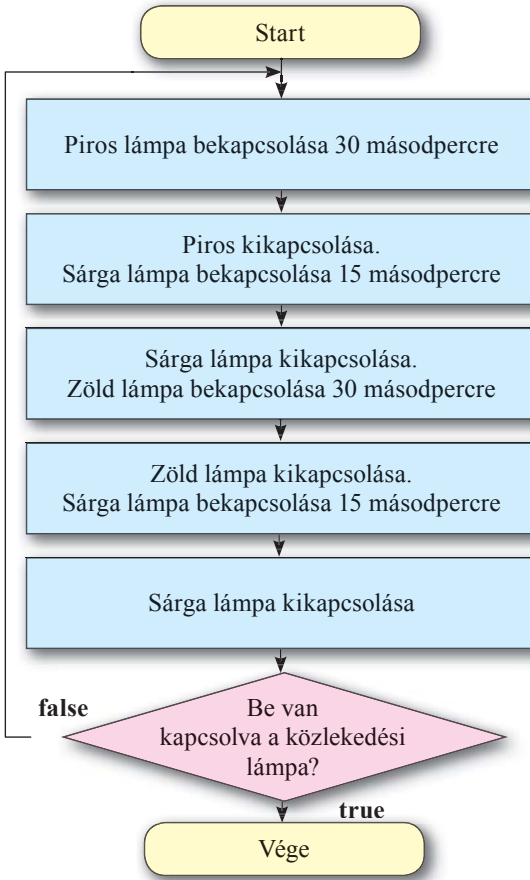
A projekt valamennyi résztvevője választhat magának egy algoritmus-végrehajtót, és modellezheti annak működését. Mi most a *közlekedési lámpát* fogjuk megvizsgálni.

A közlekedési lámpa a bekapcsolása után egy ciklikus algoritmust végez:

1. Piros lámpa bekapcsolása 30 másodpercre.
2. Piros kikapcsolása.
3. Sárga lámpa bekapcsolása 15 másodpercre.
4. Sárga lámpa kikapcsolása.
5. Zöld lámpa bekapcsolása 30 másodpercre.
6. Zöld lámpa kikapcsolása.
7. Sárga lámpa bekapcsolása 15 másodpercre.
8. Sárga lámpa kikapcsolása.
9. 1–8. lépések ismétlése mindaddig, amíg a közlekedési lámpa be van kapcsolva.

A 8.6. ábrán az algoritmus folyamatábrája látható.

A közlekedési lámpa modelljét **Lazarus**ban készítjük el. A program szövegét a 8.7. ábra mutatja.

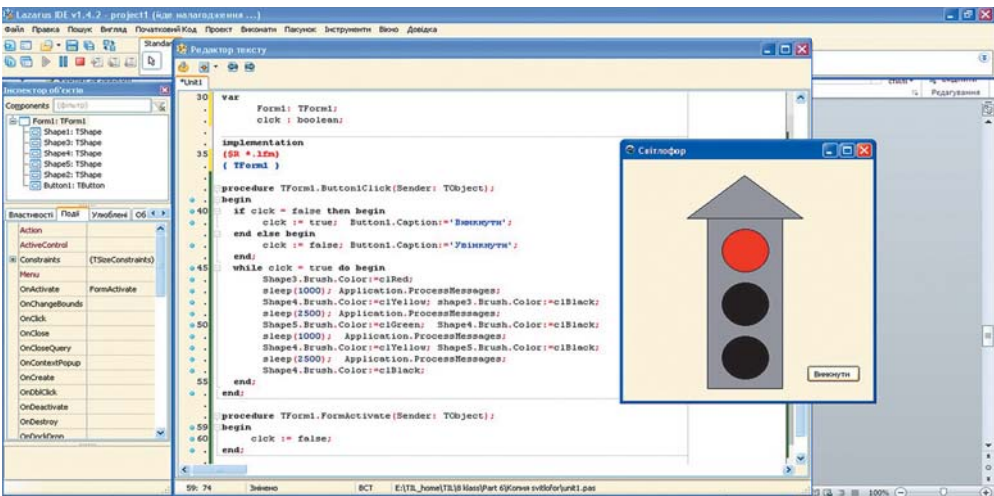


8.6. ábra. A közlekedési lámpa működésének algoritmus

Az **eredményeket elkészíthetjük** prezentáció vagy szöveges dokumentum formájában, vagy elhelyezhetjük azokat egy weboldalon vagy egy blogon. A prezentációban meg kell jeleníteni a munka menetét: a feladat kitűzését, az információkeresés módszereit, a feldolgozás módját, az elért eredményeket és a következtetéseket. A prezentáció egy lehetséges kialakítását a 8.8. ábra mutatja.

A projektmunka záróakkordja annak **védése**. A védeés során a résztvevők bemutatják az elért eredményeket. A létrehozott prezentáció egyben a védeés terve. Nagyon fontos, hogy ne csak a projektfeladatot mutassuk be, hanem azt is, milyen módon működtek közre az egyes csoporttagok annak megvalósításában. A védeés során nevezzük meg:

- a projekt nevét és célját;
- a projekt megvalósítása során alkalmazott módokat és utakat;
- milyen nehézségekkel találkoztak a csoporttagok a projektmunka során;



8.7. ábra. A közlekedési lámpát modellező projekt szövege



- az elért eredményeket;
- az elvégzett munka alapján elért eredményeket;
- milyen mértékben sikerült elérni a kitűzött célt.

ПЕРЕХРЕСТЯ АЛГОРИТМІВ

Виконали учні 8 класу

Наше завдання

- Визначити види перехрестя, які зустрічаються на дорогах, та об'єкти на перехрестях, які є виконавцями алгоритмів.
- Описати алгоритми, які виконують кожен з виконавців.
- Порівняти, алгоритми якого типу найчастіше виконуються учасниками дорожнього руху на перехрестях.
- Вибрати виконавця та один з алгоритмів, розробити комп'ютерну модель реалізації вибраного алгоритму.

Хід виконання

- У нашому населеному пункті ми побачили перехрестя, на яких рух:
 - не регулюється
 - регулюється світлофором
- Ми дізналися, що існують перехрестя, на яких рух регулюється:
 - semaфором і шлагбаумом
 - регулявальником

Виконавці алгоритмів на перехресті

- Виконавцями алгоритмів є люди:
 - пішоходів
 - водії автомобілів
 - машиністи літаків
 - регулявальники руху
- Виконавцями алгоритмів є автоматичні пристрої:
 - світлофори
 - semaфори
 - шлагбауми

Види алгоритмів на перехресті

Перехрестя	Перехрестя з регулюванням руху			
Рух дорожнього руху	Рух регулюється світлофором	Рух регулюється semaфором і шлагбаумом	Рух регулюється світлофором та semaфором	Перехрестя з регулюванням руху
Пішоходи	Регулювання	Регулювання	Регулювання	Регулювання
Водії автомобілів	Регулювання	Регулювання	Регулювання	Регулювання
Машиністи літаків	---	Регулювання	Регулювання	---
Семафор	Попередження	---	Попередження	---
Світлофор	---	Регулювання та попередження	Регулювання та попередження	---
Шлагбаум	---	---	Регулювання	---

Алгоритм роботи світлофора

1. Увімкнути лампу червоного кольору на 30 секунд.
2. Вимкнути лампу червоного кольору
3. Увімкнути лампу жовтого кольору на 15 секунд.
4. Вимкнути лампу жовтого кольору
5. Увімкнути лампу зеленого кольору на 30 секунд.
6. Вимкнути лампу зеленого кольору
7. Увімкнути лампу жовтого кольору на 15 секунд.
8. Вимкнути лампу жовтого кольору
9. Повторити дії 1-8, доки світлофор увімкнено.

Блок-схема алгоритму

```

    graph TD
      Start([Початок]) --> Step1[Вимкнути лампу червоного кольору на 30 сек.]
      Step1 --> Step2[Вимкнути лампу червоного кольору на 15 сек.]
      Step2 --> Step3[Вимкнути лампу жовтого кольору на 15 сек.]
      Step3 --> Step4[Вимкнути лампу жовтого кольору на 15 сек.]
      Step4 --> Step5[Вимкнути лампу зеленого кольору на 30 сек.]
      Step5 --> Step6[Вимкнути лампу зеленого кольору на 15 сек.]
      Step6 --> Step7[Вимкнути лампу жовтого кольору на 15 сек.]
      Step7 --> Step8[Вимкнути лампу жовтого кольору на 15 сек.]
      Step8 --> Decision{Світлофор увімкнено?}
      Decision -- Ні --> End([Віконець])
      Decision -- Так --> Step1
    
```

Наша модель роботи світлофора

Фрагмент програми увімкнення світлофора:

```

if ock = false then begin   ock = true;
  Button1.Caption="Вимкнути";
end else begin   ock = false;
  Button1.Caption="Увімкнути"; end
while ock = true do begin
  Shape3.Brush.Color:=clRed; sleep(1500);
  Application.ProcessMessages;
  Shape3.Brush.Color:=clWhite;
  Shape3.Brush.Color:=clBlack; sleep(3000);
  Shape3.Brush.Color:=clGreen;
  Shape3.Brush.Color:=clBlack; sleep(1500);
  Shape4.Brush.Color:=clYellow;
  Shape4.Brush.Color:=clBlack; sleep(3000);
  Shape4.Brush.Color:=clBlack;
end.
    
```

Висновки

- На перехрестях виконавцями алгоритмів є люди та автоматичні пристрої
- Більша частина алгоритмів на перехресті, які виконуються:
 - люди – алгоритми з регулюваннями
 - автоматичні пристрої – алгоритми з повтореннями
- Виконувати алгоритми на перехресті потрібно, щоб забезпечити безпеку руху
- Алгоритми роботи світлофора є алгоритмом з повтореннями

Використані джерела:
<https://vody.ua/pdd/>
<http://kibus255.shpeedu.com/uk/news-51>

8.8. ábra. A projektről készült prezentáció egy lehetséges kialakítása



Felelj a kérdésekre!

- 1°. Milyen lépésekből áll a projekt megvalósítása?
- 2°. Mi a célja a projekttervnek?
- 3°. Milyen forrásokból gyűjthetünk információkat a projektmunka során?
- 4°. Mi képezheti az anyaggyűjtés eredményét a projektmunka során?
- 5°. Milyen formában készíthetjük el a projekt beszámolóját?
- 6*. Milyen tényezők alapján választjuk ki a projektmunka során alkalmazott szoftvereket?
- 7°. Mit kell bemutatni a projektmunkát összefoglaló prezentációban?



Végezd el a feladatokat!

Készítsd el a tanulmányi projekt tervét, válassz ki egy objektumot a kitűzött cél alapján, majd az adatfeldolgozás eszközeit, és hajtsd végre az informatikai projektet:

1. Téma: **Büszkék vagyunk arra, hogy ukrán.** Cél: *kutatni az informatika ukrainai történetét és meghatározni, milyen eredményeket értek el ezen a területen az ukránok és az Ukrajna területén alkotó tudósok.*
2. Téma: **A számítógépek megítélése.** Cél: *kideríteni, milyen káros hatással bírnak a számítógépek az emberi szervezetre, és ajánlásokat megfogalmazni ezek csökkentése céljából.*
3. Téma: **Társam, a számítógép.** Cél: *a régiótok piacán elérhető modern számítógépek típusainak és főbb paramétereinek elemzése, valamint annak meghatározása, milyenek az adott alkalmazási területeknek megfelelő számítógép tulajdonságai.*
4. Téma: **Szaktám – IKT szakértő.** Cél: *olyan szakmák gyűjtése, amelyek az információs technológiák szakterületéhez tartoznak, és megállapítani, milyen ismeretekkel és tapasztalatokkal rendelkeznek ezen szakmák művelői.*
5. Téma: **Web 2.0 az iskolában.** Cél: *meghatározni, mit takar a Web 2.0 fogalma, milyen szolgáltatások tartoznak ebbe a fogalomkörbe, és annak felvázolása, hogy lehet ezeket alkalmazni az oktatásban.*
6. Téma: **Információs technológiák górcső alatt.** Cél: *az adatfeldolgozás (grafikai, szöveges, számadatok, multimédiás adatok) során alkalmazott információs technológiák fogalmának általánosítása, a feldolgozott alapobjektumok, alkalmazási területek, szoftverek, történelmi fejlődés áttekintése.*



SZÓMAGYARÁZÓ

A

Abszolút cellahivatkozás – a másolás során nem módosuló cellahivatkozás – **230**

Adatfeldolgozás – új adatok előállítása már meglévők alapján – **138**

Adattípus – meghatározza, hogy az adat:

- milyen értékeket vehet fel;
- milyen műveleteket végezhetünk ezekkel;
- mennyi helyet foglal el az adat a memóriában – **184**

Alakzat – vonalakból felépülő grafikai alapelem – **91**

Alkalmazói szoftver – konkrét adatfeldolgozási feladatok ellátására szolgáló szoftver, amelyeket a felhasználó a tevékenysége során használ – **51**

Archiválás – adatok másolása tömörítőprogramok segítségével – **62**

Archivátorok – adatok archiválására szolgáló programok – **63**

Archívum – tömörített vagy tömörítetlen fájlokat és mappákat tartalmazó fájl – **62**

Az üzenet bináris kódjának hossza (adatmennyiség) – az üzenet bináris kódjának hossza bájtokban kifejezve – **17**

B

Bájt – nyolc bitből álló bitsorozat – **17**

Belső memória – a processzor működését lehetővé tevő része a számítógépnek, operatív tárból (RAM), állandó memóriából (ROM) és cache-memóriából áll – **26**

Billentyűzet – szimbólumok és utasítások bevitelére szolgáló eszköz – **32**

Bináris kód – olyan kódrendszer, amely két értéket használ – **17**

Bit – a 0 és az 1 bináris kódolásban – **17**

C

Cirill betűk – a szláv nyelvek egy része (ukrán, orosz, belorusz, bolgár) által használt jelrendszer – **11**

Clip (vágni) – kisméretű (méretre és időre nézve) videorészlet – **138**

D

Digitális rajztábla – grafikai adatok beolvasására szolgáló eszköz. Felülete a speciális toll érintését érzékeli – **33**

Dokumentum struktúrája – a dokumentum részeinek hierarchikus rendje – **108**

E

Előtesztelt ciklus általános alakja **Object Pascalban** – **217**

Érintőképernyő – adatbeviteli- és kiviteli eszköz, amely az ujjal vagy a speciális tollal való érintésre érzékeny – **33**

F

Fejléc – a dokumentum „szolgálati” jellegű (például oldalszám) információkat tartalmazó ismétlődő része – **109**

Feltételes formázás – az elektronikus táblázat celláinak olyan formázása, amikor a cella külalakja bizonyos feltételektől függően változik – **260**

Forma – olyan objektum, amelyen komponenseket (vezérlőelemeket), például gombokat, mezőket, feliratokat, menüket, gördítősávokat helyezhetünk el. A végrehajtás során a formából lesz a program ablaka, amelyeken megjelennek a megfelelő vezérlők – **157**

G

Grabber (*ami eltulajdonít, rabol*) – lásd **rekorder** – **133**

Grafikai objektumok markerei – a grafikai objektum átméretezésére és forgatására szolgáló négyzet vagy kör alakú jelek – **94**

Grafikon – időbeli folyamatot, vagy két mennyiség közötti összefüggést bemutató diagramtípus – **224**

H

Hangszerkesztő – hangadatok módosítására szolgáló program – **138**

Információkeresési stratégia – a keresés céljának, forrásainak, kulcsszavainak, értékelési kritériumainak meghatározása – **267**

I

Információmennyiség többszörös mértékegységei:

1 kbájt = 2^{10} bájt = 1024 bájt

1 Mbájt = 2^{10} kbájt = 2^{20} bájt = 1 048 586 bájt

1 Gbájt = 2^{10} Mbájt = 2^{20} kbájt = 2^{30} bájt

1 Tbájt = 2^{10} Gbájt = 2^{20} Mbájt = 2^{30} kbájt = 2^{40} kbájt – **18**

Ingyenesen használható szoftver (freeware) – olyan licenelési mód, amelynél a felhasználó ingyenesen használhatja a szoftvert és nincs lehetősége a kód módosítására – **52**

K

Karaktertábla – a **Windows** segédprogramja, amelynek segítségével a billentyűzeten nem található szimbólumokat illeszthetünk a szövegbe, megtekinthetjük ezeket a szimbólumokat, válthatunk a karakterkódolási módok között – **13**

Kereskedelmi licenc – olyan licenelési mód, amely esetében a felhasználó megfizeti a szoftver használatát. Ebben a kategóriában megkülönböztetjük a dobozos és az OEM változatot – **52**

Kodek – multimédiás adatok tömörítésére és kitömörítésére szolgáló szoftver – **132**

Kódtábla – olyan táblázat, amiben minden felhasználandó szimbólumhoz számot rendelünk – **10**

Komment – a compiler által figyelmen kívül hagyott szöveg, amelyet a jobb érthetőség és a hibakeresés megkönnyítése céljából helyezünk el a programban. Általában {} kapcsos zárójelek, vagy (**) zárójel-csillag párok közé helyezzük, de // dupla dőlt vonallal is elválaszthatjuk a program szövegétől – **189**



Komponensek tulajdonságai:

1. **Name** (*név*) – a form objektumának azonosítója – **157**
 2. **Caption** (*felirat*) – tartalma a form ablakának címsorában látható szöveg – **157**
 3. **Color** (*szín*) – a form háttérszíne – **157**
 4. **Height** (*magasság*) – a form magassága pixelekben kifejezve – **157**
 5. **Width** (*szélesség*) – a form szélessége pixelekben kifejezve – **157**
 6. **Top** (*teteje*) – a képernyő felső széle és a form közötti távolság pixelekben – **157**
 7. **Left** (*bal*) – a képernyő bal széle és a form közötti távolság pixelekben – **157**
 8. **Enabled** – (*megengedett, hozzáférhető*) – értéke határozza meg, hogy az objektum tulajdonságai megváltoztathatók-e vagy sem. Értéke csak **true** vagy **false** lehet – **171**
 9. **Visible** (*látható*) – értéke határozza meg, látható-e az objektum a formon, vagy sem. Ennek a tulajdonságnak is csak két értéke (**true** vagy **false**) lehet – **171**
 10. **Font** (*betűtípus*) – értéke határozza meg, az objektumra kiírt szöveg külalakját. Összetett tulajdonság, részei a **Color**, a **Height**, a **Name**, a **Size** stb. – **171**
- Konverter** – fájlformátum-átalakító program – **135**
- Külső memória** – a számítógépnek az adatok hosszútávú tárolására szolgáló része. Ide tartoznak a merevlemezek, optikai lemez, flash-memória stb. – **27**

L

Link – a dokumentum speciális eleme, amely a dokumentum másik helyére, vagy egy másik dokumentumra mutat – **118**

Listák – a szöveges dokumentum speciálisan formázott bekezdései, objektumok felsorolására használjuk – **73**

Logikai függvény (HA, ÉS, VAGY, NEM) – olyan függvény, amely **true** vagy **false** értéket vehet fel – **237**

Logikai kifejezés – összehasonlítást tartalmazó matematikai kifejezés – **237**

Logikai változó – olyan változó, amelynek értéke **true** vagy **false** lehet. **Boolean**-ként deklaráljuk – **196**

M

Monitor – a személyi számítógép elsődleges kiviteli eszköze. A képkötés módja alapján megkülönböztetünk **LCD** (*folydékkristály*), **plazma**, **OLED** (szerves fénydióda), **elektronikus tinta** stb. monitorokat – **34**

N

Nemteljes elágazás általános alakja **Object Pascal**ban – **201**

```
If<logikai változó>
```

```
Then begin
```

```
  <utasítássor>
```

```
end;
```

Nyomtató – kiviteli eszköz, amely papíron jeleníti meg a különböző adatokat, jellemzően A3-at meg nem haladó méretben – **35**

P

Plotter (rajzgép) – nagyméretű (A2 vagy nagyobb) grafikai objektumok kivételére szolgáló eszköz – **35**

Podcast (*ipod* – médialejátszó, broadcasting – éteri megjelenítés) – az interneten terjeszthető multimédiás fájl, általában a szerzők véleménynyilvánítását tartalmazza politikai, gazdasági, műszaki, irodalmi, zenei kérdésekben – **149**

Podcaster – podcastokat létrehozó személy – **149**

Pontdiagram – két mennyiség összefüggésének bemutatására szolgáló diagramtípus – **243**

Próbaverzió (*trial licenc*) – olyan licenclési mód, amikor a szoftver bizonyos ideig (kipróbálási periódus) használható, esetleg korlátozott funkcionalitású – **52**

Processzor – a számítógépben az adatok feldolgozását végző eszköz. **Vezérlőegységből** (biztosítja az utasítások végrehajtását) és **aritmetikai-logikai egységből** (biztosítja a műveletek elvégzését) áll – **25**

Programozási rendszer – új programok létrehozására szolgáló szoftver – **51**

Projekt tesztelése – logikai hibák keresésének folyamata tesztadatok alkalmazásával – **188**

R

Rekorder – hang- és videó rögzítésére szolgáló eszköz – **133**

Relatív cellahivatkozás – a másolás során módosuló cellahivatkozás – **230**

Rendezés – a sorok vagy oszlopok sorrendjének megváltoztatása bizonyos kritériumok alapján – **248**

Rendszerszoftver – a számítógép eszközeinek vezérlésére, az eszközök közötti adatforgalom lebonyolítására, az eszközök diagnosztizálására és javítására, a felhasználó és a számítógép közötti kapcsolat megvalósítására szolgáló szoftver. Rendszerszoftverhez soroljuk az operációs rendszereket és segédprogramokat – **51**

Részösszeg – az adatok előzetes csoportosítását feltételező feldolgozási mód a táblázatkezelőben – **259**

S

Sablon – külön fájlban tárolt előreformázott dokumentum, amely mintául szolgál az új dokumentum létrehozása során – **121**

Sávdiaagram – olyan diagramtípus, amelyet egymással összefüggésben nem álló adat-sorok ábrázolásakor használunk – **242**

Segédprogramok – olyan szoftverek, amelyek a szoftver tesztelésére, az operációs rendszer lehetőségeinek kiterjesztésére szolgálnak – **51**

Stílus – bizonyos objektum tulajdonságainak névvel ellátott összessége – **111**

Szabad (FSF) szoftver – olyan licenclési mód, amelynél a felhasználó ingyenesen használja a szoftvert és lehetősége van a programkód módosítására – **53**

Számítógép architektúra – a számítógép részeinek és szoftvereinek együttműködését leíró modell – **25**

Számítógépes program – a számítógép számára értelmezhető és végrehajtható nyelven leírt adatfeldolgozó algoritmus. Működése során a program adatokat dolgoz fel – **154**

Számlálós ciklus általános alakja **Object Pascalban** – **213**



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

Szkennert – grafikus beviteli eszköz. Működésük szerint megkülönböztetünk kéziszkennert, lapszkennert és szkencamerát – **33**

Szoftver – a számítógépes programok összessége – **51**

Szűrés – az elektronikus táblázat celláinak kiválogatása bizonyos feltételek alapján – **251**

T

Tabulátorok – a szövegszerkesztő azon eszköze, amelynek segítségével szöveges objektumokat pozícionálhatunk – **79**

Tárgymutató – szakkifejezések rendezett listája azoknak az oldalaknak a megjelölésével, ahol ezek a kifejezések előfordulnak – **118**

Tartalomjegyzék – a dokumentum szerkezeti elemeinek listája az oldalszámok feltüntetésével – **118**

Telepítés – szoftverek üzembe helyezése – **54**

Teljes elágazás általános alakja **Object Pascalban** – **201**

```
if<logikai változók>
Then begin
    <utasítássor1>
end
Else begin
Thenbegin
    <utasítássor2>
end;
```

Tesztadatok – olyan adatok, melyek segítségével meggyőződhetünk a projekt helyességéről, és amelyekhez előre ismerjük az elvárt eredményeket – **188**

Touchpad – notebookok, netbookok számítógép vezérlésére szolgáló adatbeviteli eszközei – **32**

Tömörítés – adatok újrakódolása az adatok méretének csökkentése céljából – **60**

Töréspont – programsor, amelynél a program futása megáll – **190**

TV-tuner – olyan eszköz, amelynek segítségével televíziós műsorokat tekinthetünk meg a számítógépen – **34**

U

UNICODE (*universal coded character set*) – karakterkódolási rendszer, amely 17, egyenként 65 536 elemű kódtáblát tartalmaz, ezáltal 1 114 112 karakter kódolását teszi lehetővé – **11**

Ü

Üzenet bináris kódja – az üzenet binárisan kódolt alakja – **17**

Üzenet dekódolása – az eredeti üzenet visszaállítása kódoltból – **8**



Üzenetek kódolása – az üzenet kódolására szolgáló jelrendszer felcserélése egy másikkra – 7

V

Vegyes cellahivatkozás – olyan cellahivatkozás, amely részben módosul a másolás során, vagy az oszlop neve, vagy a sor száma módosul – 231

Videostúdió – professzionális videofeldolgozás céljait szolgáló program – 138

Videoszerkesztő – videoadatok feldolgozására szolgáló program – 138

Vonal – alakkal, vastagsággal, sablonnal és színnel jellemezhető grafikai objektum – 91

Z

Zenei stúdió – professzionális hangszerkesztő szoftver – 138

A

ABS – a szám abszolút értékének meghatározására szolgáló matematikai függvény – 237

ASCII – az angol nyelv betűit, számjegyeket, elválasztójeleket tartalmazó, összesen 128 karakter kódolását lehetővé tevő karaktertábla – 10

F

FLAC (*Free Lossless Audio Codec* – szabad veszteségmentes audiokodek) – szabad felhasználású, veszteségmentes tömörítést alkalmazó audiokodek – 131

Flash videó (flv kiterjesztésű fájl) – videó-fájlformátum, leginkább interneten elhelyezett fájlloknál alkalmazzák – 132

G

GYÖK – számtani gyök kiszámítására szolgáló matematikai függvény – 237

H

HATVÁNY – a hatvány értékének kiszámítására szolgáló függvény a táblázatkezelőben – 237

K

KEREKÍTÉS – a kerekítés műveletének végrehajtására szolgáló matematikai függvény – 237

KOI8-U – cirill betűk kódolására szolgáló kódtábla, amely az ASCII kódokat is tartalmazza – 11

M

MIDI – (*Musical Instrument Digital Interface* – digitális zenei interfész) – speciális eszköz vagy program (szintetizátor) általi lejátszásra alkalmas fájlformátum – 131

MP3 – (*MPEG-1/2/2.5 Layer3* – (*Motion Picture Experts Group* – mozgóképek szakértői csoport, 3. réteg) – veszteségesen tömörített hangfájl, fájlcsereelőkkben, interneten elhelyezett hangfájlok esetében alkalmazzák – 131



MP4 (MPEG-4 Part 14) – videofilmek tárolására, továbbá digitális televíziózásban alkalmazott videó fájlformátum – **131**

P

PI – matematikai függvény a π értékének közelítő meghatározására – **237**

Q

QuickTime (gyors idő) – az Apple által kidolgozott, hang- és videoanyagok tárolására alkalmas fájlformátum – **132**

S

ShowMessage('<szöveg>') – szöveges üzenet megjelenítésére szolgáló utasítás – **167**

Y

YouTube (a te telefonkagylód) – videomegosztó szolgáltatás az interneten – **146**

W

WAV vagy **WAVE** (*wave forma udio format* – hullámforma audioformátum) – a **Windows** operációs rendszerek alapértelmezett hangformátuma – **131**

Windows-1251 – az **ASCII** kódtáblát tartalmazó, cirill betűk megjelenítésére alkalmas, a **Windows** operációs rendszerekben alapértelmezett kódolási tábla – **11**

WMA/WMV (*Windows Media Audio/Video*) – a **Windows** operációs rendszerben szabványos, hang- és videófájlok tárolására szolgáló fájlformátum – **132**

TARTALOM

Kedves nyolcadikosok!.....	3
----------------------------	---

1. FEJEZET. ADATOK KÓDOLÁSA

1.1. Az adatfeldolgozás, mint információs folyamat. Üzenetek kódolása és dekódolása.....	6
1.2. Karakterek kódolása.....	10
1.3. Bináris kód.....	16
1. számú <i>gyakorlati munka</i> . Feladatok megoldása szöveges kód hosszának meghatározására	21

2. FEJEZET. A SZÁMÍTÓGÉP HARDVERE ÉS SZOFTVERE

2.1. A számítógép architektúrája	24
2.2. Adatbeviteli és kiviteli eszközök.....	31
2.3. Az információs folyamatok megvalósításának története. A mai számítógépek típusai.....	38
2. számú <i>gyakorlati munka</i> . Számítógép-konfiguráció kialakítása igény szerint	50
2.4. A számítógépes szoftverek csoportosítása és általános jellemzői	51
2.5. Adatok tömörítése és archiválása.....	60
3. számú <i>gyakorlati munka</i> . Fájlok tömörítése és visszaállítása	69

3. FEJEZET. SZÖVEGFELDOLGOZÁS

3.1. Karakterek, listák és hasábok beszúrása szöveges dokumentumba	71
3.2. Táblázatok szöveges dokumentumokban	81
3.3. Grafikai elemek létrehozása, szerkesztése és formázása a szöveges dokumentumban	90
3.4. Speciális grafikai objektumok létrehozása, szerkesztése és formázása.....	101
4. számú <i>gyakorlati munka</i> . Különböző objektumokat tartalmazó szöveg szerkesztése	107
3.5. Szakaszok, fejlécek, stílusok. A dokumentum szerkezete.....	108
3.6. A szöveges dokumentum létrehozásának automatizálása	118
5. számú <i>gyakorlati munka</i> . A dokumentum szerkezete. Tartalomjegyzék és tárgymutató létrehozása	129

4. FEJEZET. MULTIMÉDIÁS OBJEKTUMOK FELDOLGOZÁSA

4.1. Multimédiás objektumok feldolgozása.....	131
4.2. Videók létrehozása	138
6. számú <i>gyakorlati munka</i> . Videoklip készítése. Effektusok alkalmazása, a videosáv és hangsáv időzítése.....	146
4.3. Multimédiás fájlok közzététele az interneten	146
7. számú <i>gyakorlati munka</i> . Hangfájlok és filmek elhelyezése az interneten	152

5. FEJEZET. AZ ESEMÉNY- ÉS OBJEKTUMORIENTÁLT PROGRAMOZÁS ALAPJAI

5.1. A számítógépes programok és programozási nyelvek	154
5.2. Események. Eseménykezelők.....	163



8. számú gyakorlati munka. Üzenetablakot megjelenítő objektumorientált program létrehozása.....	169
5.3. A gomb.....	170
5.4. A címke.....	174
9. számú gyakorlati munka. Gombokat és címkéket tartalmazó projekt létrehozása.....	178

6. FEJEZET. OBJEKTUMOKKAL ÉS MENNYISÉGEKKEL DOLGOZÓ ALGORITMUSOK

6.1. A beviteli mező. Mennyiségek. Állandó és változó mennyiségek. A változók tulajdonságai.....	181
10. számú gyakorlati munka. Szekvenciális algoritmus összeállítása mennyiségek feldolgozására.....	187
6.2. Projekt tesztelése és javítása.....	188
11. számú gyakorlati munka. Program tesztelése és javítása.....	195
6.3. Logikai kifejezések. Logikai változók. Logikai műveletek.....	195
6.4. Elágazást tartalmazó algoritmusok.....	200
6.5. A jelölőnégyzet. A kombinált lista. A választógomb.....	207
6.6. Számlálós ciklus.....	213
6.7. Előtesztelő ciklus.....	217
12. számú gyakorlati munka. Elágazásokat és ciklusokat tartalmazó projektek létrehozása.....	220
6.8. Grafikai objektumok megjelenítése a programozási nyelv eszközeivel.....	221
13. számú gyakorlati munka. Adatok grafikai megjelenítését tartalmazó algoritmusok összeállítása és végrehajtása.....	228

7. FEJEZET. SZÁMADATOK FELDOLGOZÁSA

7.1. Abszolút, relatív és vegyes cellahivatkozások. Oldalbeállítás és nyomtatás elektronikus táblázatokban.....	230
14. számú gyakorlati munka. Számítási feladatok megoldása.....	236
7.2. A táblázatkezelő matematikai, statisztikai és logikai függvényeinek rendeltetése és alkalmazása.....	236
7.3. Diagramok létrehozása és formázása.....	242
15. számú gyakorlati munka. A táblázatkezelő matematikai, logikai és statisztikai függvényeinek alkalmazása. Diagramok beszúrása.....	248
7.4. Adatok rendezése a táblázatokban. Automatikus és irányított szűrők.....	248
7.5. Részösszegek. Feltételes formázás.....	258
16. számú gyakorlati munka. Adatok rendezése a táblázatokban. Automatikus és irányított szűrés. Feltételes formázás.....	264

8. FEJEZET. KOMPETENCIAFELADATOK MEGOLDÁSA, PROJEKTMUNKÁK VÉGREHAJTÁSA

8.1. Kompetenciafeladatok megoldása.....	266
8.2. Tanulmányi projektek.....	273
Szómagyarázó	279

Навчальне видання

РИВКІНД Йосиф Якович
ЛИСЕНКО Тетяна Іванівна
ЧЕРНІКОВА Людмила Антонівна
ШАКОТЬКО Віктор Васильович

ІНФОРМАТИКА

Підручник для 8 класу
загальноосвітніх навчальних закладів
з навчанням угорською мовою

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Переклад з української мови

Перекладач *Гавриіл Гаврилович Семере*

Угорською мовою

Зав. редакцією *Адальберт Варга*
Редактор *Бертолон Ковач*
Обкладинка *Тетяни Куц*
Макет, художнє оформлення,
комп'ютерна обробка ілюстрацій *Василя Марущинця*
Коректор *Габрієлла Тирканич*

Формат 70×100/16.

Ум. друк. арк. 23,328. Обл.-вид. арк. 22,1.

Тираж 2160 пр. Зам. № 43П.

Державне підприємство „Всеукраїнське спеціалізоване видавництво „Світ”

79008 м. Львів, вул. Галицька, 21

Свідоцтво суб'єкта видавничої справи ДК № 4826 від 31.12.2014
www.svit.gov.ua; e-mail: office@svit.gov.ua; svit_vydav@ukr.net

Друк ТДВ „Патент”, 88006 м. Ужгород, вул. Гагаріна, 101

Свідоцтво суб'єкта видавничої справи ДК № 4078 від 31.05.2011