



Gábor Dénes
Főiskola

Baczoni Pál

**AZ EXCEL GAZDASÁGI
ALKALMAZÁSAI ÉS
PROGRAMOZÁSA**

238

FŐISKOLAI JEGYZET

2005/2006

I. félév

Tartalomjegyzék

BEVEZETÉS	5
1. RÉSZ: ALAPVETÉS.....	5
1. FEJEZET: FÜGGVÉNYEK A GYAKORLATBAN	5
<i>Dátum- és Idő függvények.....</i>	5
<i>Műszaki függvények.....</i>	5
<i>Statisztikai függvények.....</i>	6
<i>Mátrix függvények</i>	10
<i>Logikai függvények.....</i>	12
<i>Matematikai és trigonometrikus függvények.....</i>	13
<i>Műveletek szövegekkel.....</i>	14
<i>Információs függvények.....</i>	16
<i>Pénzügyi függvények.....</i>	17
<i>Trend függvény.....</i>	18
2. RÉSZ: AZ ALAPOKON TÚL.....	23
1. FEJEZET: IRÁNYÍTOTT BEILLESZTÉS	23
<i>Mintafeladat: Érték beillesztése.....</i>	23
<i>Mintafeladat: Sorok és oszlopok cseréje.....</i>	23
<i>Mintafeladat: Matematikai műveletvégzés</i>	24
<i>Mintafeladat: Csatolás</i>	24
2. FEJEZET: SZÁMFORMÁZÁS	27
<i>A számformázás alapfogalmai.....</i>	27
<i>Kategóriák.....</i>	27
<i>A formátumkódok értelmezése</i>	30
<i>Mintafeladat: Számok formázása.....</i>	31
<i>Mintafeladat: Feltételes számformátum kialakítása számformátum kóddal.....</i>	31
<i>Mintafeladat: Feltételes számformátum létrehozása menüparanccsal.....</i>	32
<i>Mintafeladat: Feltételes számformátum létrehozása függvény eredménye alapján, menüparanccsal.....</i>	32
<i>Mintafeladat: Feltétel nélküli formátumkód kialakítása.....</i>	33
3. FEJEZET: RAJZOLÁS	34
<i>Rajzolás eszköztár.....</i>	34
<i>A rajzolt objektum helye és mérete</i>	34
<i>Mintafeladat: Rajzolás.....</i>	34
4. FEJEZET: STÍLUSOK.....	36
<i>Alapismeretek</i>	36
<i>Munka stílusokkal.....</i>	36
5. FEJEZET: AUTOMATIKUS FORMÁZÁS	37
<i>Alapok.....</i>	37
<i>A formázás</i>	37
6. FEJEZET: VÉDELEM	38
<i>Alapok.....</i>	38
<i>Cellavédelem bekapcsolása.....</i>	38
<i>Cellavédelem megszüntetése.....</i>	39
<i>Állományvédelem.....</i>	39
7. FEJEZET: NEVEK	41
<i>Nevek létrehozása.....</i>	41
<i>Hivatkozás a névre.....</i>	41
<i>Cellanevek listája.....</i>	41
<i>Cellák kijelölése név alapján</i>	41
<i>Mintafeladat: Nevek.....</i>	42
<i>Cellák nem címkék alapján történő elnevezése.....</i>	42

8. FEJEZET: MUNKALAPOK	43
<i>Háttérinformáció</i>	43
<i>Minta feladat: Munkalapok</i>	43
9. FEJEZET: ÖSSZESÍTÉS	45
<i>Minta feladat: Az Összesítés menete</i>	45
<i>Minta feladat: Az Összesítés menete (nevek felhasználásával)</i>	46
<i>Az Összesítés párbeszédablak kezelőszervei</i>	47
10. FEJEZET: HÁROMDIMENZIÓS HIVATKOZÁSOK	48
<i>Alapfogalmak</i>	48
<i>Gyakorlat: Háromdimenziós hivatkozások</i>	48
11. FEJEZET: MEGJEGYZÉS	49
<i>Megjegyzés készítése</i>	49
<i>A megjegyzések beállítása</i>	49
<i>A megjegyzések nyomtatása</i>	49
<i>Megjegyzések szerkesztése</i>	49
<i>Minta feladat: Megjegyzés készítése</i>	49
<i>Minta feladat: Töröljük a A2-es cellához fűzött megjegyzést</i>	50
<i>Minta feladat: Jelöltessük ki az Excellel az összes olyan cellát, amelyhez megjegyzés van fűzve</i>	50
12. FEJEZET: ESZKÖZTÁRAK	51
<i>Minta feladat: Eszköztárak</i>	51
<i>Egyéni eszköztár válogatás</i>	52
13. FEJEZET: HIBAKERESÉS	54
<i>Hibák behatárolása menüparanccsal</i>	54
<i>Hibák behatárolása a képlet (függvény) egy részének kiértékelésével</i>	54
<i>Hibák behatárolása a képlet (függvény) menüparanccsal történő kiértékelésével</i>	54
<i>Mit tehetünk, hogy ne lássuk a hibajelzést?</i>	55
<i>Milyen eszközzel kereshetünk még hibát?</i>	55
14. FEJEZET: ŰRLAPKEZELÉS	56
<i>Minta feladat: Jelölőnégyzet</i>	56
<i>Minta feladat: Választókapcsoló</i>	57
<i>Minta feladat: Listapanel</i>	58
<i>Minta feladat: Léptető</i>	59
<i>Minta feladat: Görgetősáv</i>	60
15. FEJEZET: IRÁNYÍTOTT KIJELOLÉS	62
<i>Minta feladat: Irányított kijelölés</i>	62
<i>Hibás tartalmú celláink megkeresése</i>	63
16. FEJEZET: BEÁLLÍTÁSOK	64
17. FEJEZET: ADATBÁZIS-KEZELÉS AZ EXCELBEN	65
<i>Alapfogalmak</i>	65
<i>Fontosabb adatbázis parancsok</i>	65
<i>Sorba rendezés parancs</i>	65
<i>Szűrés</i>	68
<i>Űrlap parancs</i>	72
<i>Részösszegek parancs</i>	74
<i>Adatbázis függvények</i>	75
3. RÉSZ: HALADÓ EXCEL ISMERETEK	77
1. FEJEZET: KIMUTATÁS	77
<i>Minta feladat a Kimutatás készítés szemléltetésére</i>	77
2. FEJEZET: KÜLSŐ ADATOK IMPORTJA	82
<i>Külső adatok importja a Query program segítségével</i>	82
3. FEJEZET: ELEMZŐ ESZKÖZÖK	87
<i>Egyváltozós adattábla</i>	87
<i>Kétváltozós adattábla</i>	88
<i>Egyváltozós célérték keresés</i>	89

Solver.....	90
Az Esetvizsgáló.....	95
4. FEJEZET: SZEMELVÉNYEK A HALADÓ DIAGRAMKÉSZÍTÉSI LEHETŐSÉGEK TÁRGYKÖRÉBŐL.....	98
Értékjelzők grafikára cserélése.....	98
Trendvonal diagramba történő beszúrása.....	100
Mi van akkor, ha?.....	103
5. FEJEZET: ANALITIKAI ESZKÖZKÉSZLET.....	105
Mozgóátlag.....	105
Hisztogram.....	108
4. RÉSZ: AZ EXCEL PROGRAMOZÁSÁNAK ALAPJAI.....	111
1. FEJEZET: ALAPOK.....	111
Mi az objektum, objektumgyűjtemény (collection)?.....	111
Mi a tulajdonság?.....	111
Mi a metódus?.....	111
Objektummodell.....	111
Direkt kódbevitel.....	112
Makró rögzítés.....	113
Az Excel helyzetérzékeny Súgója (Help).....	115
Adattípusok deklarálása.....	115
Változók definiálása.....	115
Típus deklarálás kezdőbetűk alapján a DefType paranccsal.....	116
Típusdeklaráló jelekkel történő adattípus megadás.....	117
Munka változókkal, Szöveg (String) típusú változók.....	117
Dátum típusú változók.....	118
A felhasználó által definiált adattípus (Type).....	118
Tömbökkel végzett munka.....	118
Konstansok használata.....	119
Műveletek.....	120
Kommentek (magyarázatok) használata a kódban.....	122
Végtelen ciklus és hibamegállás.....	122
Üzenetküldés.....	122
Információkérés (InputDialog).....	124
Vezérlési struktúrák.....	126
Ciklusképzés.....	128
2. FEJEZET: ESEMÉNYEK.....	132
BeforeDoubleClick.....	132
Change.....	132
Munkalap eseményt (Worksheet) kezelő eljárás készítése.....	132
Mintafeladat: példa egy munkafüzetesemény (Workbook) kezelő eljárásra.....	133
Az Alkalmazás (Application) eseményei.....	133
Egyéb események.....	133
3. FEJEZET: AZ APPLICATION OBJEKTUM (KIEGÉSZÍTŐ INTERFACE ELEMÉK).....	135
4. FEJEZET: ŰRLAPOK.....	137
Mintafeladat: Beviteli mező (TextBox).....	137
Kombinált lista (URLAP3.XLS Munka1) /ComboBox/.....	137
Kétoszlopos lista.....	140
Listaablak.....	140
Lista (ListBox), mint választócsoport /USERFORM2, URLAP5.XLS/.....	142
Összefoglaló feladat.....	143
A Tab sorrend beállítása (Tab Order).....	148
Beépített Excel párbeszédablakok kijelzése.....	149
5. FEJEZET: BŐVÍTMÉNYEK.....	152
Bővítmények.....	152

6. FEJEZET: EGYÜTTMŰKÖDÉS.....	155
<i>Más munkafüzetben található adatok elérése.....</i>	<i>155</i>
<i>Más munkafüzetben található programok elindítása.....</i>	<i>155</i>
7. FEJEZET: HIBAKEZELÉS.....	156
<i>Alapvető ismeretek a hibafeldolgozásról (hibafeldolgozó függvények és parancsok).....</i>	<i>156</i>
8. FEJEZET: FÜGGVÉNYEK.....	158
<i>Munkalapfüggvények.....</i>	<i>158</i>
<i>A fontosabb VBA függvények (Function).....</i>	<i>158</i>
<i>Felhasználó által definiált függvények.....</i>	<i>158</i>
<i>Mit érdemes tudni a felhasználó által készített függvényekről.....</i>	<i>159</i>
<i>Példa egy függvénymakróra.....</i>	<i>159</i>
9. FEJEZET: GYAKORLÓ FELADATOK (GYAKORLAS.XLS).....	162

Bevezetés

Ezen Jegyzet a Gábor Dénes Főiskola “Az Excel közgazdasági alkalmazása és programozása” című tantárgyához készült. A jegyzet anyaga alapvetően az otthoni felkészülést szolgálja. **A gyakorlatokon való megjelenést és az ottani becsületes munkát nem pótolja.**

A gyakorlatokon feladatsorok közös végrehajtásával szándékozunk a tudást elmélyíteni.

A Jegyzet négy részre oszlik. Az 1. rész az alapok összefoglalását szolgálja, míg a 2. rész túlmutat az alapokon. A Jegyzet 3. része röviden, mintafeladatok segítségével, haladó Excel ismereteket nyújt. Jegyzetünk utolsó része a napjainkban egyre nélkülözhetetlenebb Excel programozási alapokat ismerteti. Ezen alapok megszerzése után bárki bátran belevághat komolyabb VBA-ról szóló könyvek felfedezésébe, ezek közül néhánynak a címét az Olvasó megtalálhatja a Jegyzet végén található Irodalomjegyzékben.

A Jegyzetben hivatkozott állományok, legtöbbször megoldott állapotban a főiskola honlapján kerülnek elhelyezésre. Ez utóbbi ne zavarja az Olvasót, az otthoni felkészülés során lehetőleg az összes feladatot oldja meg.

1. rész: Alapvetés

1. fejezet: Függvények a gyakorlatban

Jegyzetünk ezen fejezetében bemutatunk néhány hasznos függvényt.

Dátum- és Idő függvények

A következő feladat kapcsán bemutatunk néhány Dátum- és idő függvényt. Feladatunk minél többet megtudni függvények révén a mai napról.

1. Nyissuk meg a **Datum.xls** állományt.
2. Feladatunk az, hogy az **Informatikai alapszoftverek 2** tárgyban tanult függvényekkel megadjuk a válaszokat a címkével jelzett cellákba.
3. A felhasznált függvényeket a következő ábra mutatja.

Ma van:	=MOST()
Az év hányadik hónapja?	=HÓNAP(C8)
Az év hányadik napja?	=SZÖVEG(MA()-"2004.01.01";0)
A hónap hányadik napja?	=NAP(C8)
Óra:	=ÓRA(C8)
Perc:	=PERC(C8)
Másodperc:	=MPERC(C8)
Hét melyik napja?	=SZÖVEG(HÉT.NAPJA(C8);"nnnn")

1. ábra Adatok a mai napról

Műszaki függvények

A műszaki függvények egy része a komplex számokkal végez számításokat, míg a kategóriában néhány függvény a számrendszerek közötti átváltást támogatja. A mintául vett függvény is ez utóbbiak közül került ki.

A BIN2HEX függvény

A függvény egy bináris számot hexadecimálissá alakít (konvertál) át.

Ha a függvény nem áll rendelkezésre, akkor telepítsük az Analysis ToolPak beépülőt. A telepítés lépései:

1. Adjuk ki az **Eszközök** menü **Bővítmenykezelő** parancsát.

2. A megjelenő **Bővítménykezelő** párbeszédablakban kapcsoljuk be az Analysis ToolPak jelölőnégyzetet.

Szintaxis: **BIN2HEX(szám;jegyek)**

Ahol a szám: egy maximum 10 karakterből álló bináris szám. A bal oldali bit az előjelbit. A jegyek paraméter a használandó karakterek száma. Amennyiben nem adjuk meg, akkor az Excel a megjelenítéshez minimálisan elegendő számú karaktert alkalmazza. A 2. ábra példákat mutat a függvény alkalmazására.

	A	B	C
2		A	B
3		Képlet	Leírás (eredmény)
4	=BIN2HEX(11111011;4)	00FB	Az 11111011 bináris érték konvertálása 4 karakteres hexadecimálissá (00FB)
5	=BIN2HEX(1110)	E	Az 1110 bináris érték konvertálása hexadecimálissá (E)
6	=BIN2HEX(1111111111)	FFFFFFFF	Az 1111111111 bináris érték konvertálása hexadecimálissá (FFFFFFFF)

2. ábra Példák a BIN2HEX függvény alkalmazására (BIN2HEX.XLS)

Statisztikai függvények

Az Excel nagyszámú függvénnyel rendelkezik e kategóriában. Ezek közül (nem teljesen véletlenszerűen) kiragadtunk kettőt.

Gyakorisági tábla: a GYAKORISÁG függvény

Gyakori statisztikai feladat egy bizonyos sokaság osztályozása egy vagy több ismérv szerint. Az osztályozás célja, hogy az adott sokaságot valamilyen szempont alapján egységesebb (homogénebb) részekre bontsuk.

A jó osztályozás főbb ismérvei:

- ✦ egy adott egység, csak egy osztályhoz tartozhat
- ✦ a sokaság összes egysége tartozzon valamelyik osztályba
- ✦ a hasonló egységek azonos, míg a nem azonos egységek más osztályba kerüljenek.

Az osztályozás általában intervallumok létrehozását igényli.

Ha a sokaság viszonylag kevés értéket vehet fel, akkor az intervallumok helyett adható egy felsorolás is. Például a hallgatók jegyeit vizsgálva: 1, 2, 3, 4, 5.

Ha a sokaság sokféle értéket vehet fel, akkor érdemes **osztályközöket** (intervallumok) megadni. Arra azonban vigyázzunk, hogy ezek egymást nem fedhetik át.

Egy bizonyos ismérv szerinti osztályozás végeredménye a **gyakorisági tábla**. Az ebben kapott gyakoriságok (eredmények) megmutatják, hogy az adatok egy csoportjából mennyi adat tartozik az adott ismérv alapján képzett osztályba. A gyakorisági táblát a továbbiakban a GYAKORISÁG függvénnyel hozzuk létre.

A GYAKORISÁG függvény egy **adattömbben** az előfordulások számát határozza meg a **csoport_tömbben** adott intervallumhatárok alapján. Vigyázzunk a **csoport_tömb** létrehozásánál arra, hogy abban az osztályközök felső határát adjuk meg (tehát csak számokat tartalmazhat) valamint, hogy ezek a számok növekvő sorrendben kerüljenek bevitelre.

Szintaxis: **GYAKORISÁG(adattömb;csoport_tömb)**

A függvénynek több érdekessége van. Az egyik mindjárt a bevitel módja. Mivel a függvény eredménye tömb, ezért a függvényt tömbképletként kell bevinnünk. Ez annyit jelent, hogy a függvényt nem egy cellába, hanem egy cellatartományba kell bevinnünk a SHIFT+CTRL+ENTER billentyűkombinációval.

A másik érdekessége a GYAKORISÁG függvénynek az, hogy az eredménytömb egy cellával „hosszabb”, mint a **csoport_tömb**. Itt kapjuk meg a legfelső intervallumhatár feletti értékek számát. Amennyiben biztosak vagyunk benne, hogy az **adattömb**ben nincs nagyobb érték mint a **csoport_tömb**ben, akkor a függvényt csak annyi cellába kell bevinni mint a **csoport_tömböt**.

Mintafeladat a GYAKORISÁG függvény alkalmazására

Feladatunk egy teszt eredményeinek csoportosítása. Az eredmények nem osztályzatban, hanem pontokban adóttak. A tesztben maximum 10 pontot lehetett szerezni. Az érdekel minket, hogy hány hallgató szerzett 0-tól 5 pontot (ők sajnos egyest kapnak), mennyien kaptak 6 pontot, mennyi azon hallgatók száma, akik 7 pontot szereztek, hány hallgató gyűjtött 8 vagy 9 pontot és végül mennyien érték el a maximális 10 pontot.

1. Készítsük el az alábbi táblázatot (3. ábra). Ne gépeljük még be a függvényt.

	A	B	C	D	E	F	G	H	I
1									
2	Sorszám	Név	Pontszám	Adattömb				Csoport tömb	Gyakoriság
3	1.	Almási Pál	4					5	1;H3:H7)
4	2.	Bendegúz Péter	5					6	
5	3.	Csapodár Vilmos	6					7	
6	4.	Keres Imre	7					9	
7	5.	László Péter	8					10	
8	6.	Másik Éva	9						
9	7.	Piros Katalin	10						
10	8.	Rémes Zsófia	3						
11	9.	Veremi Gáspár	10						
12									

3. ábra A mintatáblázat (GYAKORISAG.XLS)

2. Jelöljük ki az I3:I8 cellatartományt.
3. Gépeljük be a Szerkesztőlécen látható függvényt.
4. Nyomjuk le egyszerre a SHIFT+CTRL+ENTER billentyűket, ennek hatására egy úgynevezett tömbképlet (array function) jön létre.

⚡ Igen érdekes, hogy bármelyik a tömbképletet tartalmazó cellában azonos képlet látható. A tömbképletben a képlet francia zárójelpárok közé van zárva! Fontos tudni, hogy tömbképlet csak a **SHIFT+CTRL+ENTER** billentyűkombináció hatására jön létre és nem hozható oly módon létre, hogy a képlet köré francia zárójelpárt gépelünk.

	H	I
2	Csoport_tömb	Gyakoriság
3	5	3
4	6	1
5	7	1
6	9	2
7	10	2
8		0

4. ábra A GYAKORISÁG tömbképlet eredménye

Az eredmény (4. ábra) mint látható az, hogy 3 hallgató szerzett 5 pontot vagy annál kevesebbet, 1-1 hallgató szerzett 6 illetve 7 pontot. Két hallgató gyűjtött be 8 vagy 9 pontot. És végül két hallgató érte el a maximális 10 pontot. Senki sem szerzett több pontot 10-nél (I8-as cella). Nem is tudott volna, mert csak 10 volt a maximálisan adható.

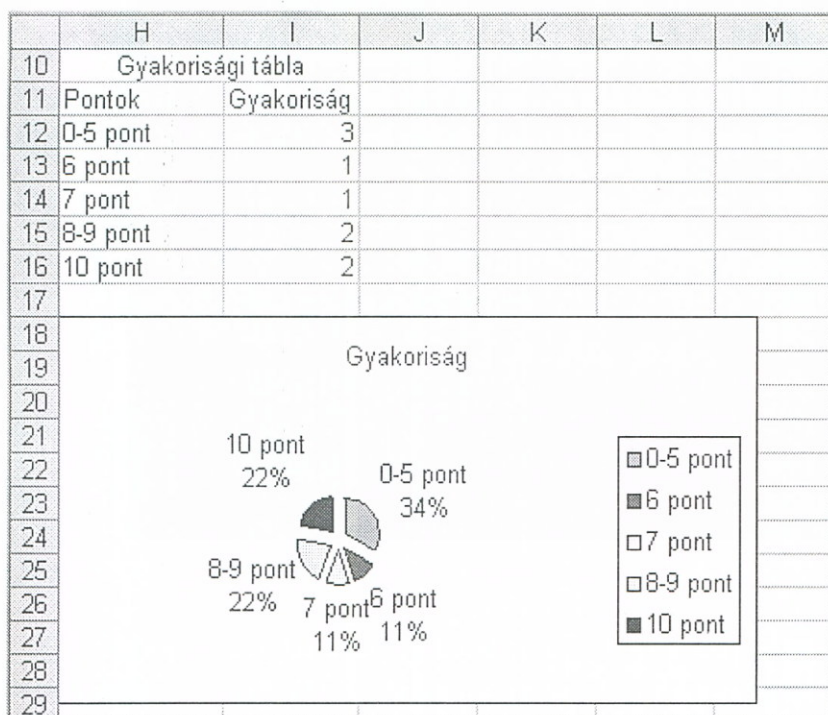
A megoszlásokat (a gyakorisági tábla értékeit) érdemes kör- vagy XY diagramon ábrázolni. A példa következő részében kördiagrammal szemléltetjük az eredményt. Ehhez előbb kiegészítjük a táblázatot, majd a Diagram varázslóval elkészítjük a diagramot.

1. Készítsük el az alábbi táblázatrészt (GYAKORISAG.XLS 5. ábra).

	G	H	I
10		Gyakorisági tábla	
11		Pontok	Gyakoriság
12		0-5 pont	3
13		6 pont	1
14		7 pont	1
15		8-9 pont	2
16		10 pont	2

5. ábra Táblázat, amelyből kördiagram készül

2. Jelöljük ki a H11:I16 cellatartományt (5. ábra).
3. Indítsuk el a Diagram varázslót, válasszuk annak első lépésben a típusok közül a kört, az altípusok közül a robbantott kört (Excel 2003).
4. Léptessük el a Diagram varázslót a 3. lépésig. Ezen összetett párbeszédablakban kattintsunk a **Feliratok** szegélyre és az így nyert párbeszédablakban kapcsoljuk be a **Kategória neve** és a **Százalék** nevű jelölőnégyzeteket.
5. A Diagram varázslót léptessük végig, készítsünk beágyazott diagramot. Az elkészült diagramot a 6. ábra mutatja.



6. ábra Táblázatunk a beágyazott diagrammal

Medián

Megfigyelt értékeinket sorba rendezve kiválaszthatjuk azt, amelynél ugyanynyi kisebb érték fordul elő, mint ahány nagyobb. A medián a nagyság szerint rendezett sor közepén lévő érték. Ebből adódón, ha az értékek száma páros, akkor a két középső érték számtani átlaga a medián. A mediánt az alábbiakban a MEDIÁN függvénnyel kerestjük ki.

Szintaxis: **MEDIÁN(lista)**

A függvény megkeresi egy adott számhalmaz mediánját. Az argumentumoknak tömböknek, számoknak, hivatkozásoknak vagy cellaneveknek kell lenniük. A függvény figyelmen kívül hagyja a logikai értékeket, szövegeket, üres cellákat.

Mintafeladat a MEDIÁN függvény alkalmazására

Készítsük el az alábbi táblázatokat (MEDIAN.XLS 7. ábra). Mint látható mindjárt két táblázatot kell létrehoznunk. A táblázatokban hallgatók szerzett pontszámai vannak. A bal oldali adatbázis-tábla, a pontszámok szerint nem rendezett. Ugyanakkor a jobb oldali adatbázis-tábla a pontszámok alapján növekvő sorrendbe rendezett.

C14 fx =MEDIÁN(C4:C12)

	A	B	C	D	E	F	G
1	Rendezetlen				Rendezett		
2							
3	Sorszám	Név	Pontszám		Sorszám	Név	Pontszám
4	1.	Almási Pál	4		8.	Rémes Zsófia	3
5	2.	Bendegúz Péter	5		1.	Almási Pál	4
6	3.	Csapodár Vilmos	6		2.	Bendegúz Péter	5
7	4.	Keres Imre	7		3.	Csapodár Vilmos	6
8	5.	László Péter	8		4.	Keres Imre	7
9	6.	Másik Éva	9		5.	László Péter	8
10	7.	Piros Katalin	10		6.	Másik Éva	9
11	8.	Rémes Zsófia	3		7.	Piros Katalin	10
12	9.	Veremi Gáspár	10		9.	Veremi Gáspár	10
13							
14		Medián:	7			Medián:	7

7. ábra Táblázatunk medián számításához (MEDIAN.XLS)

1. A C14-es cellába gépeljük be a MEDIÁN függvényt (7. ábra).
2. A G14-es cellába másoljuk át a C14-esben lévő függvényt. Természetesen az eredmény egyforma, ez azt bizonyítja, hogy a listának nem kell a pontszám alapján rendezettnnek lennie.

A következő ábrán összefoglaljuk a sűrűbben használt statisztikai függvényeket (8. ábra).

Név	Funkció
MIN	A tartomány értékei közül a legkisebb számot adja eredményül
ÁTLAG	A tartomány értékeinek átlagát adja eredményül
MAX	A tartomány értékei közül a legnagyobb számot adja eredményül
DARAB	A tartomány számot tartalmazó celláinak darabszáma
DARAB2	A tartomány kitöltött celláinak darabszáma
KICSI	Egy adathalmaz k-adik legkisebb elemét adja eredményül
NAGY	Egy adathalmaz k-adik legnagyobb elemét adja eredményül

8. ábra Statisztikai függvények

Mátrix függvények

Ha értékeket keresünk adatbázisban vagy táblázatban, illetve egy cellára való hivatkozást keresünk, alkalmazhatjuk a mátrix függvényeket. A következő ábrán bemutatunk három függvényt a kategóriából. Példaképpen csak az **FKERES** függvényt mutatjuk be.

Név	Funkció
FKERES	A függvény egy táblázat bal szélső oszlopában keres egy megadott értéket, és az így kapott sorból veszi az oszlop_számmal kijelölt cellát, és ennek tartalmát adja eredményül.
VÁLASZT	A függvény az érték argumentumok közül az index sorszámút adja vissza
KUTAT	Egy sorból vagy egy oszlopból álló tartományban vagy tömbben lévő értéket keres meg

9. ábra Mátrix függvények

Mintafeladat az FKERES függvény bemutatására

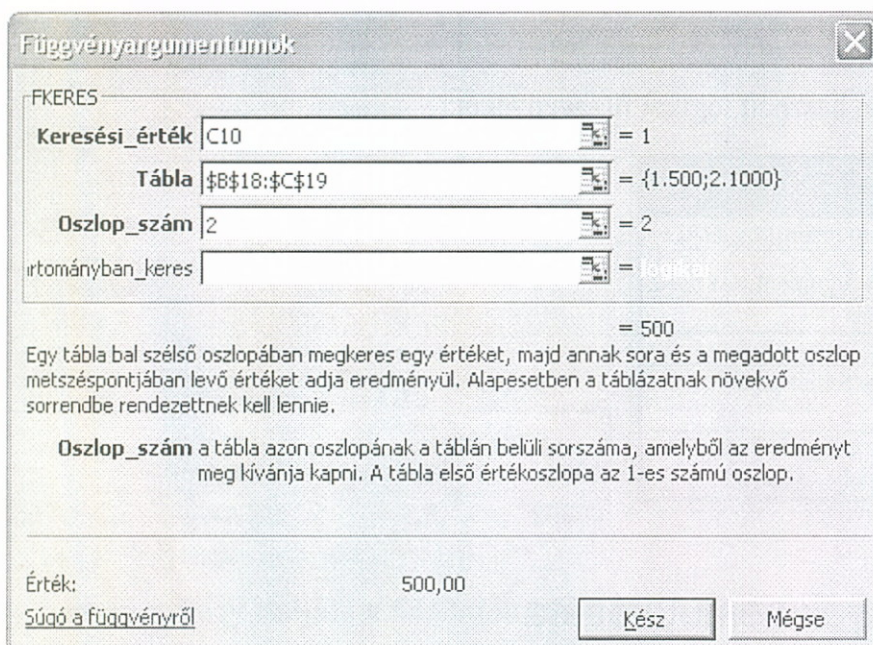
Az **FKERES** függvény felhasználásával számítsuk ki a táblázatban található személyek fizetését. Létezik egy segédtáblázat is, amely a kategóriákhoz tartozó órabéreteket adja meg.

1. Nyissuk meg az **fkeres.xls** állományt.

	B	C	D	E
9	Név	Kategória	Ledolgozott idő	Kereset
10	Kiss	1	40:00:00	
11	Kis	2	40:00:00	
12	Nagy	1	50:00:00	
13	Orsós	2	50:00:00	
14				
15				
16				
17	Kategória	Órabér		
18	1	500		
19	2	1000		

10. ábra Mintatáblázat az FKERES függvény bemutatásához

2. Legyen aktív az E10-es cella. Mivel a függvény a nehezebbek közé tartozik, hívjuk segítségül a Függvény varázslót. Kattintsunk a Szerkesztőléctől balra elhelyezkedő **Függvény beszúrása** nevű (fx) mini ikonra.
3. Miután a Függvény varázsló elindul, megjelenik annak első párbeszédablaka. A felső listaablakban válasszuk a **Mind** kategóriát. Kattintsunk az alsó listaablakban egy függvénynévre azért, hogy a listaablak aktívvá váljon.
4. A függvény nevének megkereséséhez gépeljük be a keresett függvény nevének első betűjét, ez most az "F" betű. Az Excel megkeresi a listában az első f betűvel kezdődő függvényt és ráviszi fénykurzort.
5. A továbbiakban a lapozó nyíllal lefelé haladva a függvények listájában, a fénykurzort vigyük az FKERES-re. Nyomjunk OK-t.
6. Adjuk meg a függvény paramétereit (aktív az E10-es cella).



11. ábra A függvény paramétereit

7. A **Keresési_érték** az illető személy kategóriája, a **Tábla** paraméter a segédtábla címe (abszolút címmel) , míg az **Oszlop_szám** azért 2, mert az eredmény a segédtábla 2. oszlopában van. Kattintva a **Kész** nyomógombra az Excel beviszi a függvényt az E10-es cellába.
8. Még csak az órabért számítottuk ki, ezt meg kell szorozni az illető által ledolgozott idővel és még 24-gyel is (öröknaptár): =FKERES(C10;\$B\$18:\$C\$19;2)*D10*24
9. Az így kapott képletet másoljuk be az E11:E13 cellatartományba. Mivel az eredményt az Excel időre formázza, azt vissza kell formáznunk valamilyen érvényes számformára. Ehhez például kattintsunk az **Ezres csoport** nevű ikonra a **Formázás** eszköztáron.

E10	=FKERES(C10;\$B\$18:\$C\$19;2)*D10*24					
	B	C	D	E	F	G
9	Név	Kategória	Ledolgozott idő	Kereset		
10	Kiss	1	40:00:00	20 000,00		
11	Kis	2	40:00:00	40 000,00		
12	Nagy	1	50:00:00	25 000,00		
13	Orsós	2	50:00:00	50 000,00		
14						
15						
16						
17	Kategória	Órabér				
18	1	500				
19	2	1000				

12. ábra A Mintafeladat eredménye

Logikai függvények

A következő ábrán bemutatjuk a sűrűbben használt logikai függvényeket:

Név	Funkció
ÉS	Eredménye IGAZ ha minden argumentuma IGAZ
VAGY	Eredménye IGAZ ha bármelyik argumentuma IGAZ
HAMIS	Eredménye HAMIS logikai érték
IGAZ	Eredménye IGAZ logikai érték
NEM	Az argumentum értékének ellentettjét adja eredményül
HA	Logikai vizsgálat eredményének függvényében két lehetséges eredmény egyikét adja válaszul

13. ábra Logikai függvények

Mintafeladat a logikai függvények bemutatására

Vizsgáljuk meg gépjárművek mért sebességeit, az eredménytől függően “büntessük” meg azok vezetőit, ha a sebesség határt elérték vagy átlépték. Büntetésük legyen 15000 Forint. Amennyiben a vezető nem gyorsajtó, a büntetés oszlopban a „Rendben” szöveg legyen olvasható.

1. Nyissuk meg a **ha.xls** állományt.
2. Az E10-es cellába gépeljük be az alábbi függvényt: **=HA(D10>=\$C\$7;\$E\$4;"Rendben")**

E10	fx =HA(D10>=\$C\$7;\$E\$4;"Rendben")				
	B	C	D	E	F
4		>=55	Büntetés 1	15 000 Ft	
5		>=100	Büntetés 2	90 000 Ft	
6					
7	Sebesség határ 1.:	55			
8	Sebesség határ 2.:	100			
9		Rendszám	Sebesség	Büntetés	
10		AAA-123	51	Rendben	
11		AAB-321	56		
12		AAA-222	127		

14. ábra Mintafeladat a Ha függvény megismerésére

3. A függvényt másoljuk át az E11:E12 cellatartományba, a feladatot ezzel megoldottuk. Vigyázzunk, abszolút cellacímzéssel kell címeznünk mind a sebesség határt, mind a büntetést, tartalmazó cellákat.
4. Alakítsuk úgy át a HA függvényt, hogy az tudjon háromirányú döntést hozni. Ezt csak úgy oldhatjuk meg, amennyiben a HA függvénybe egy újabb függvényt ágyazunk be (írunk be).

A HA nevű logikai függvénnyel oldjuk meg a háromirányú döntést, egészüljön ki vizsgálatunk egy olyan feltétellel, hogyha a sebesség nagyobb vagy egyenlő, mint 100 Km/ó, akkor a büntetés mértéke legyen 90000 Ft.

1. A C8-as cellában található ez a második sebesség határ, az E5-ös cellában, pedig a büntetés mértéke.
2. A feladat megoldható például ezzel a függvénykombinációval (az E10-es cellába gépeljük be): **=HA(D10>=\$C\$8;\$E\$5;HA(D10>=\$C\$7;\$E\$4;"Rendben"))**

E10		=HA(D10>=\$C\$8;\$E\$5;HA(D10>=\$C\$7;\$E\$4;"Rendben"))					
	B	C	D	E	F	G	H
4		>=55	Büntetés 1	15 000 Ft			
5		>=100	Büntetés 2	90 000 Ft			
6							
7	Sebesség határ 1.:	55					
8	Sebesség határ 2.:	100					
9		Rendszám	Sebesség	Büntetés			
10		AAA-123	51	Rendben			
11		AAB-321	56	15000			
12		AAA-222	127	90000			

15. ábra Ha függvények egymásba ágyazása

Lényeges, hogy tudjuk, a feladatok jelentős részének több helyes megoldása is van.

Matematikai és trigonometrikus függvények.

A **mat.xls** nevű állományban bemutatunk néhány hasznos matematikai függvényt. Érdekeség, hogy a **KERÉK** nevű függvény új nevet kapott a függvény új neve: **KERÉKÍTÉS**, míg két "testvér" függvényének neve maradt **KERÉK.FEL** és **KERÉK.LE**

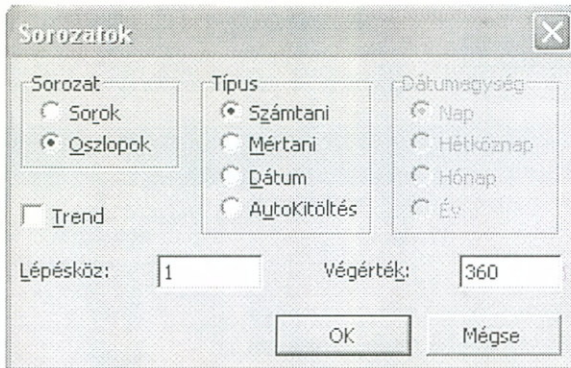
	A	B
1	5	=ABS(-5)
2	2,718282	=KITEVŐ(1)
3	6	=FAKT(3)
4	126	=INT(126,8)
5	1	=LN(2,7182818)
6	3	=LOG(8;2)
7	2	=MARADÉK(8;3)
8	3,141593	=PI()
9	6	=SZORZAT(1;2;3)
10	81,82439	=VÉL()*100
11	2,2	=KERÉKÍTÉS(2,15;1)
12	100	=KERÉKÍTÉS(121,5;-2)
13	-1	=ELŐJEL(-100)
14	8	=GYÖK(64)
15	36	=SZUM(12;24)
16	8	=CSONK(8,9)

16. ábra Matematikai függvények

Mintafeladat egy trigonometrikus függvény bemutatására

Feladatunk legyen a sinus függvény ábrázolása fokenkénti felbontásban.

1. Nyissuk meg a **mat.xls** állományt.
2. Tanulmányozva a Súgót, megtudjuk azt, hogy miként számíthatjuk ki 30 fok szinusztát.
3. Készítsük el a fokokat ábrázoló oszlopot (19. ábra). Gépeljünk a B4-es cellába 0-át. Ez a cella maradjon aktív.
4. Adjuk ki a **Szerkesztés** menü **Kitöltés**, majd a **Sorozatok** parancsot.
5. A párbeszédablakot állítsuk be a 17. ábra szerint, majd kattintsunk az **OK** gombra.



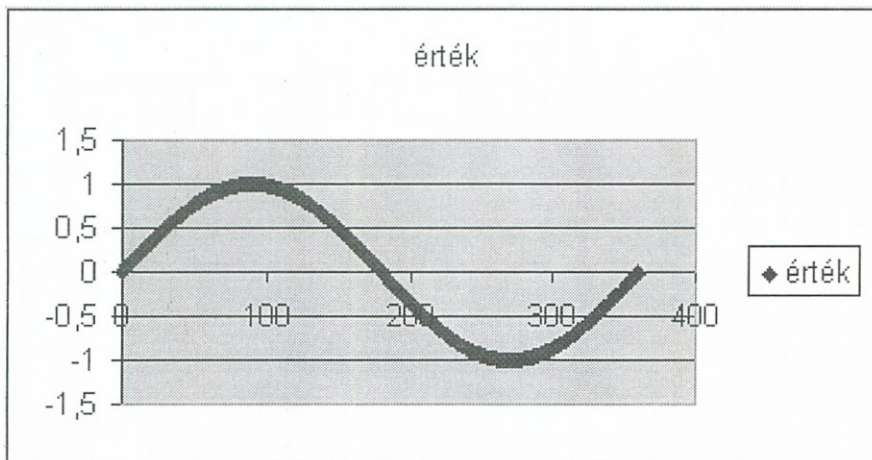
17. ábra Sorozattal történő kitöltés

6. Gépeljük be a SIN függvényt a C4-es cellába, majd a C4-es cella alapján töltsük ki a C5:C364 cellatartományt.

	C4	=SIN(B4*PI()/180)		
	A	B	C	D
3		fok	érték	
4		0	+0	

18. ábra A SIN függvény

7. Jelöljük ki a B3:C364 cellatartományt. Készítsünk X-Y diagramot (19. ábra).



19. ábra A SIN függvény grafikus ábrázolása

Műveletek szövegekkel

Szöveg függvények

A szövegfüggvényekkel karaktersorozatokat kezelhetünk.

A következő ábrán bemutatjuk a sűrűbben használt szöveg függvényeket.

Név	Funkció
BAL	Egy karaktersorozat adott számú bal szélső karakterét adja eredményül
JOBB	Egy karaktersorozat adott számú jobb szélső karakterét adja eredményül
KÖZÉP	A karaktersorozat meghatározott számú karakterét adja eredményül, adott pozíciójú karaktertől kezdve
HOSSZ	Egy szöveg karakterekben mért hosszát adja eredményül
SZÖVEG.KERES	A függvény egy szövegben balról jobbra haladva a keresett karakter vagy karaktersorozat első előfordulásának karakterpozícióját adja eredményül
ÖSSZEFŰZ	Több szövegdarabot egyetlen szöveggé fűz össze
TRIM	Minden szóközt töröl a karaktersorozatból, kivéve azokat, amelyek két szót választanak el egymástól
ÉRTÉK	A számot képviselő szöveget számmá alakítja
NAGYBETŰS	Szöveget csupa nagybetűsre alakít át
KISBETŰ	Szöveget csupa kisbetűsre alakít át
KARAKTER	A szám argumentummal megadott karaktert adja eredményül
KÓD	Egy szöveglánc első karakterének numerikus kódját adja eredményül

20. ábra Szöveg függvények

Mintafeladat a szöveg függvények bemutatására

Feladat: állítsuk elő függvények felhasználásával a vezeté- és keresztneveket.

1. Nyissuk meg a **nev.xls** állományt.
2. A **Név** oszlop celláiból vágjuk ki a Vezetéknveket. Azért kell egyet kivonni a szöveg karakterben mért hosszából, mert ha nem tennénk, akkor az Excel kivágná a vezeté- és keresztnvet elválasztó szóközt is.

	B	C
7	Név	Vezetéknév
8	Kiss Tyutyu	=BAL(B8;SZÖVEG.KERES(" ";B8)-1)
9	Nagy Zoárd	=BAL(B9;SZÖVEG.KERES(" ";B9)-1)
10	Orsós Vendel	=BAL(B10;SZÖVEG.KERES(" ";B10)-1)

21. ábra Vezetéknév kivágása

3. A **Név** oszlop celláiból vágjuk ki a Keresztnveket.

	B	D
7	Név	Keresztnév
8	Kiss Tyutyu	=JOBB(B8;HOSSZ(B8)-SZÖVEG.KERES(" ";B8))
9	Nagy Zoárd	=JOBB(B9;HOSSZ(B9)-SZÖVEG.KERES(" ";B9))
10	Orsós Vendel	=JOBB(B10;HOSSZ(B10)-SZÖVEG.KERES(" ";B10))

22. ábra A keresztnév kivágása

Összefűzés képlettel és függvénnyel

Az egyik leggyakoribb szöveg művelet két vagy több szöveg és/vagy számtartalmú cella összefűzése egyetlen cellába. Az összefűzést (konkatenáció) megvalósíthatjuk képlettel (műveleti jele: &) és függvénnyel (ÖSSZEFŰZ) egyaránt.

Mintafeladat a képlettel történő összefűzés műveletének bemutatására

Feladat: állítsuk elő a vezeték- és keresztnévek felhasználásával, képlettel történő összefűzés művelettel a neveket.

1. Nyissuk meg a **szoveg2.xls** állományt.
2. Gépeljük be következő ábrán látható képletet (23. ábra).

	B	C	D
2	Vezetéknév	Keresztnév	Név
3	Kis	Imre	Kis Imre
4	Kiss	Péter	Kiss Péter
5	Nagy	József	Nagy József

23. ábra Összefűzés képlettel

3. A képletet másoljuk át a D4:D5 cellatartományba.

Mintafeladat a függvénnyel történő összefűzés műveletének bemutatására

1. Nyissuk meg a **szoveg3.xls** állományt.
2. Gépeljük be a következő ábrán látható függvényt a D3-as cellába.

	B	C	D	E
2	Vezetéknév	Keresztnév	Név	
3	Kis	Imre	Kis Imre	
4	Kiss	Péter		
5	Nagy	József		

24. ábra Összefűzés függvénnyel

3. A függvényt másoljuk át a D4:D5 cellatartományba.

Információs függvények

Mint azt a kategórianév mutatja, ezen függvények segítségével információkat szerezhetünk celláink állapotáról, beállításokról.

A függvények jobb megismerése céljából érdemes tanulmányozni az **inf.xls** állományt.

	A	B	C
10	1,2231	=INDIREKT(\$A\$1)	A függvény eredménye a szöveggé megadott hivatkozás.
11	Automatikus	=INFÓ("recalc")	A rendszer- és munkakörnyezet aktuális állapotáról ad felvilágosítást.
12	HAMIS	=ÜRES(\$A\$1)	igaz, ha a mező üres
13	HAMIS	=HIBA(\$A\$2)	igaz, ha hiba van

25. ábra Részlet az inf.xls állományból

Mintafeladat az Információs függvények tanulmányozásához

Feladatunk egy munkaidő nyilvántartás elkészítése. Arról is gondoskodnunk kell, hogyha valamelyik adatot tartalmazó cella üres, ne történjen meg a napi munkaidő kiszámítása. A feladat megoldásához felhasználjuk az **ÜRES** nevű Információs függvényt.

1. Nyissuk meg az **inf2.xls** állományt (26. ábra).

	B	C	D	E
2		érkezett	eltávozott	dolgozott
3	hétfő	8:30	14:15	
4	kedd	8:20	18:15	
5	szerda	8:25	19:15	
6	csütörtök	8:30	20:15	
7	péntek	9:30	21:15	
8				
9			Össz.:	
10			órabér:	1000
11			Jár.	

26. ábra Táblázat a napi munkaidő és a járandóság kiszámításához

- Mint az a fenti ábrán is látható, tegyük aktívvá az E3-as cellát és gépeljük be a következő függvénykombinációt: **=HA(VAGY(ÜRES(C3);ÜRES(D3));"Üres cella";D3-C3)**
- A függvénykombináció működését jobban megértjük, ha részenként megvizsgáljuk annak működését. Ehhez jelöljük ki az egyik **ÜRES** függvényt, majd nyomjuk meg a F9-es funkciógombot. Ezzel kiértékelítettük (az Excellel) az egyik **ÜRES** függvényt. Nyomjunk **ESC**-t, ezzel kilépünk a kiértékelésből.
- Most a 3. pontban leírtak szerinti technikával értékeltsük ki a másik **ÜRES** függvényt.
- Most a 3. pontban leírtak szerinti technikával értékeltsük ki a **HA** függvényt.
- Abból a célból, hogy megnézzük a függvénykombináció helyes működését, töröljük ki, majd állítsuk vissza a C3-as, ezt követően pedig a D3-as cella tartalmát.
- Miután tapasztaltuk, hogy a függvénykombináció jól működik, az E3-as cellában lévő függvénykombinációt másoljuk át az E4:E7 cellatartományba. A táblázat képe ekkor a 27. ábra szerinti.

	B	C	D	E
2		érkezett	eltávozott	dolgozott
3	hétfő	8:30	14:15	5:45
4	kedd	8:20	18:15	9:55
5	szerda	8:25	19:15	10:50
6	csütörtök	8:30	20:15	11:45
7	péntek	9:30	21:15	11:45

27. ábra A táblázat megjelenése

- A feladat befejezését az Olvasóra bízunk, a következő ábra mutatja a táblázat fennmaradt részének kinézetét.

	D	E
9	Össz.:	50:00:00
10	órabér:	1000
11	Jár.	50 000 Ft

28. ábra Az összes ledolgozott idő és a járandóság

Pénzügyi függvények

A pénzügyi függvények igen sokrétűek. A kategóriában megtalálhatjuk a hitelnyújtás, a beruházás, az értékpapír ügyletekkel kapcsolatos függvényeket egyaránt. Mintául a **RÉSZLET** nevű függvényt mutatjuk be, azzal a nem is titkolt céllal, hogy a jegyzetben még többször fel szándékozunk használni azt.

A RÉSZLET függvény

A függvény segítségével kiszámíthatjuk egy kölcsön törlesztő részleteit. Azaz megadja a törlesztő rész és a kamat összegét egy adott időszakra.

Szintaxis: **RÉSZLET(ráta;időszakok_száma;mai_érték;jövőbeli_érték;típus)**

A paraméterek elemzése helyett bemutatunk egy mintafeladatot.

Mintafeladat

Feladatunk egy 10 évre felvett, évi 7,9% kamatozású kölcsön havi törlesztő részleteinek kiszámítása, a törlesztések a hónapok végén történnek. Vegyük észre, hogy mivel havonta törlesztünk, ezért az éves kamatlábat elosztottuk 12-vel.

Megoldás: gépeljük be egy munkalapra a 29. ábra adatait (*reszlet.xls*) és a **RÉSZLET** függvényt a C5-ös cellába. Ezzel máris választ kaptunk kérdésünkre.

C5		fx =RÉSZLET(\$C\$4;\$C\$2;\$C\$1)		
	A	B	C	D
1	Kölcsön		3 000 000 Forint	
2	Futamidő		120 hónap	
3	Éves kamat		7,90%	
4	Havi kamat		0,66%	
5	Havi törlesztés		-36 240 Forint	

29. ábra Havi törlesztés

Trend függvény

Az idősorokban (esetleg) meglévő tartós irányzat meghatározására, elemzésére, valamint előrejelzésre trendszámításokat használunk.

Akkor végzünk trendszámítást, ha a megfigyelt adatok alapján feltételezhető (például diagramon ábrázoljuk és látjuk, hogy az adatok jelentős része egy egyenes közvetlen közelében helyezkedik el), hogy egy ismert analitikus függvény nagy pontossággal leírja azt.

A TREND függvény nagyszerű lehetőséget kínál egy lineáris trend értékeinek kiszámítására.

Szintaxisa: **TREND(ismert_y;ismert_x;új_x;konstans)**

A függvény a legkisebb négyzetek elvét alapul véve, az ismert koordinátájú pontok alapján meghatározza a legjobban illeszkedő egyenes egyenletét. Egyúttal az **új_x** értékekhez tartozó, az egyenesre illeszkedő **y** értékeket adja eredményül. Az **ismert_y** az $y=mx+b$ egyenes egyenletéből a már ismert **y** értékeket magába foglaló tartomány, az **ismert_x** pedig ezen egyenlet független változóinak tartománya.

A konstans egy logikai érték. Ha a konstans értéke IGAZ (vagy nem adjuk meg ezt a paramétert), akkor a függvény a **b** értékét a megszokott módon számítja ki. Ha viszont a konstans értéke HAMIS, akkor a **b** értéke 0 lesz. Ekkor természetesen az **y** értékét az $y=mx$ egyenlet adja meg.

Mintafeladat a TREND függvény alkalmazására (trend.xls)

Készítsük el a következő ábrán látható táblázatot, ne gépeljük még be az F3-as és az F4-es cellákba a függvényeket. Feladatunk az, hogy egy képzeletbeli bolt nyereségét, jelezzük előre a 2004-es évre (**új_x**), lineáris trendet feltételezve. Ehhez rendelkezésre állnak (millió forintban) az elmúlt években (**ismert_x**) elért nyereségek (**ismert_y**).

F3		fx =TREND(\$B\$3:\$B\$9;\$C\$3:\$C\$9;\$D\$3;E3)					
	B	C	D	E	F	G	H
2	ismert_y	ismert_x	új_x	konstans	Trend		
3	10,1	1997	2004	HAMIS	13,11386		
4	11	1998		IGAZ	17,04286		
5	12,2	1999					
6	13,1	2000					
7	14,2	2001					
8	15	2002					
9	16	2003					
10							

30. ábra Trend függvény

Gépeljük be az F3-as cellába a Szerkesztőlécen látható függvényt, majd azt másoljuk át az alatta lévő cellába is (**trend.xls**, Minta feladat munkalapja).

Mint az látható, a várható nyereség (F4-es cella) 17,04 millió Forint. A függvényt kiszámítottuk a konstans paraméter HAMIS értéke mellett (ekkor a b értéke az egyenes egyenletében nulla lesz) is. Számításunk helyességéről diagramba beszúrt lineáris trendvonalal meg tudnánk győződni.

Mintafeladat trendvonal beszúrására

Feladatunk legyen a fentiekben kidolgozott TREND függvényen alapuló példa (**trend.xls**, Minta feladat munkalapja). A 30. ábra mutatja a táblázat jelenlegi állapotát.

Ha az ábrán látható információkat szavakba akarjuk önteni és egyúttal „ideológiát” is akarunk azokhoz gyártani, akkor azt mondhatjuk, hogy kiszámoltuk lineáris trendet feltételezve azt, hogy cégünk a táblázat adatai alapján 2004-ben 17,04 millió Forint nyereséget ér majd el. A számításhoz figyelembe vettük cégünk nyereségi adatait 1997-től kezdve 2003-ig bezárva.

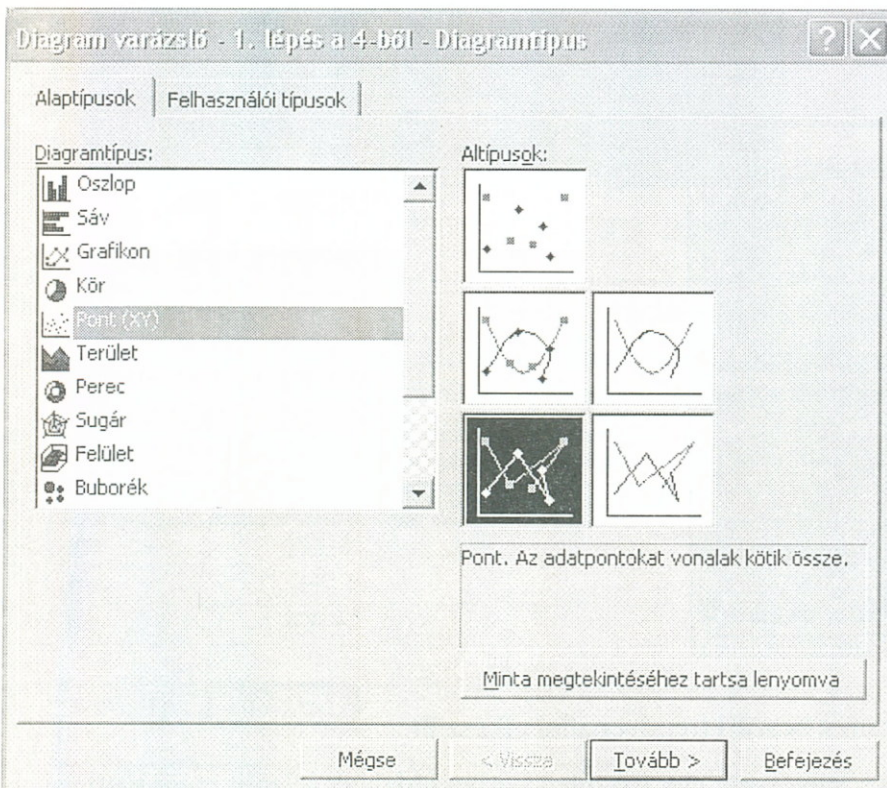
Most azt szeretnénk bizonyítani, hogy mindez a diagramok trendvonal beszúrási lehetőségével is kiszámítható. Ne feledjük, trendvonal térbeli diagramba nem szúrható be.

1. Készítsük el az alábbi táblázatot (**trend.xls**):

	C	D
15	ismert_x	ismert_y
16	1997	10,1
17	1998	11
18	1999	12,2
19	2000	13,1
20	2001	14,2
21	2002	15
22	2003	16

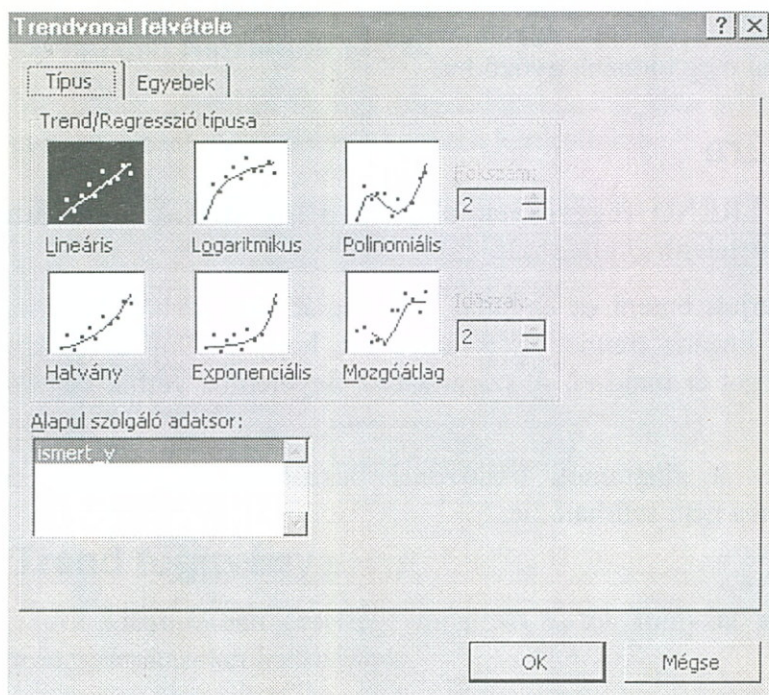
31. ábra Táblázat a diagramkészítéshez

2. Jelöljük ki a C15:D22 cellatartományt.
3. Készítsünk X-Y diagramot.



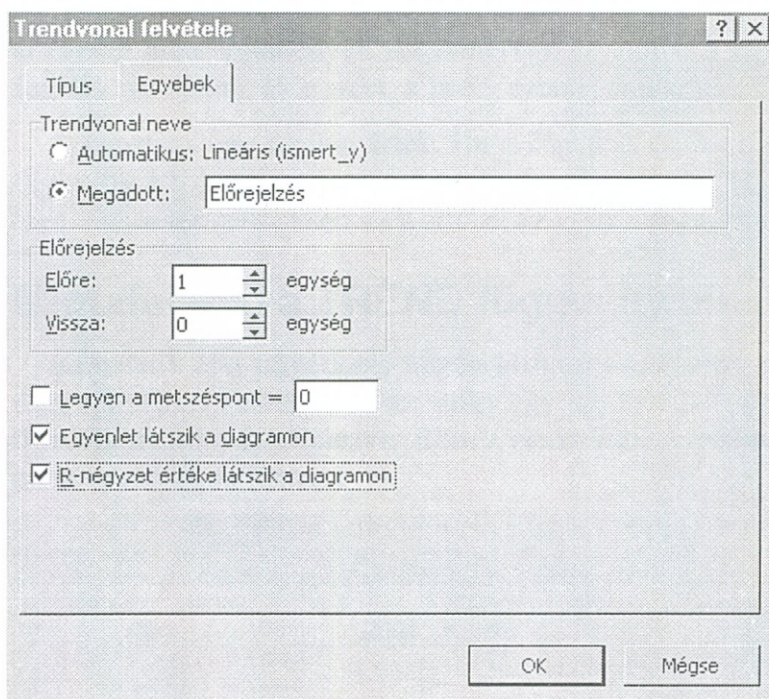
32. ábra A Diagram varázsló első lépése

4. Az elkészült diagramban kattintsunk valamelyik értékjelzőre jobb gombbal (az összes kijelölt lesz), erre az adatsorra illeszti majd az Excel a trendvonalat. A megjelenő rövid menüből válasszuk a **Trendvonal felvétele** parancsot.



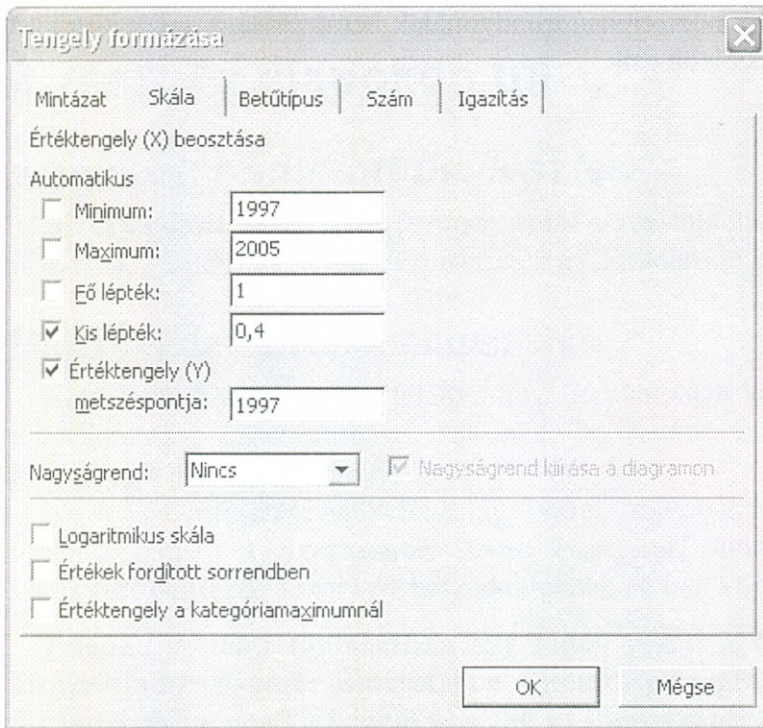
33. ábra Trendvonal típusa

5. Olyan trendvonal típust válasszunk, amely szerintünk a lehető legjobban megközelíti az adataink között lévő tendenciát. Ezért a **Trendvonal felvétele** párbeszédablakban válasszuk a **Lineáris** ablakot.
6. Mivel egyéves előrejelzést akarunk adataink alapján adni, kattintsunk a párbeszédablak **Egyebek** fülére. A most megjelenő párbeszédablakot állítsuk be a következő ábra alapján. Mint az ábrán látható (**Előrejelzés** keret, **Előre** léptetőjének 1 értékével) egyéves előrejelzést kérünk, valamint az **Egyenlet látszik a diagramon** és az **R-négyzet értéke látszik a diagramon** jelölőnégyzeteket is bekapcsoltuk.



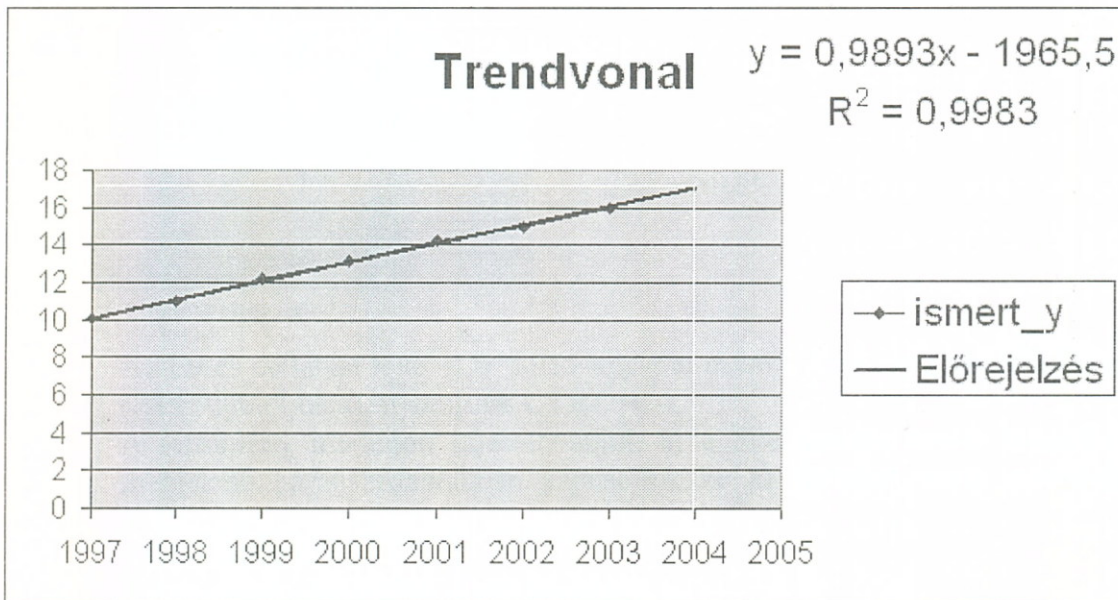
34. ábra Beállítások

7. Kattintsunk az **OK** gombra. A diagram a beszúrt trendvonallal elkészült.
8. Duplán kattintva a diagram vízszintes tengelyére egy összetett párbeszédablak jelenik meg, kattintsunk annak **Skála** nevű fülére, és állítsuk be a párbeszédablakot az alábbi ábra szerint:



35. ábra Skála párbeszédablak

9. Feladatunkat megoldottuk. A diagramba két kiegészítő fehér vonalat húztunk, erről elég pontosan leolvasható a 2004. évi előrejelzés, amelynek értéke 17-nek látszik.



36. ábra Trendvonal

10. A diagram jobb felső sarkában látható képlet alapján számítsuk ki a 2004. évi előrejelzést. Ehhez gépeljük be a következő ábrán látható adatokat és a képletet.

H46		fx =H43*H42+H44		
	G	H	I	J
42	x=	2004		
43	m=	0,9893		
44	b=	-1965,5		
45				
46	y=mx+b	17,0572		
47				

37. ábra Egyenlet

Ne feledjük, lehetőségünk van egy diagramba különböző típusú trendvonalak berajzolására. Ezek közül az a legjobb, amelyiknek R-négyzet értéke az 1-hez legközelebb esik.

2. rész: Az alapokon túl

1. fejezet: Irányított beillesztés

Az **Irányított beillesztés** parancs, mint neve mutatja igen sokrétű, segítségével megadhatjuk, hogy mit illesszünk a céltartományba. A műveletek a Clipboardon keresztül történnek.

Mintafeladat: Érték beillesztése

A gyakorlatban néha az a feladat, hogy képleteinket vagy függvényeinket úgy kell kitörölni, hogy azoknak maradjon meg az eredménye. Az érték beillesztés azért is speciális másolás, mivel a forrás tartomány megegyezik a céltartománnyal.

Az érték beillesztése művelet is úgy indul, mint minden másolás. A beillesztés viszont nem **CTRL+V**-vel történik, hanem a **Szerkesztés** menü **Irányított beillesztés** parancsával. A parancs végrehajtáshoz az **Irányított beillesztés** nevű párbeszédablakban be kell kapcsolnunk az **Értéket** választókapcsolót.

Feladat: A lenti táblázatban egy autós vezeti a tankolt benzin mennyiségét, a tankolás dátumát, kiszámítja az egységár ismeretében a költött összeget. Igen ám, de amikor 2004.01.21-én tankolni ment azt tapasztalta, hogy a benzin ára 280 Ft literenként. A feladat, átírni az egységárat, de úgy, hogy ennek hatására a régebbi tankolásokat mutató képletek eredménye ne változzon meg. Azaz a feladat a régebbi képletek eredményének megőrzése.

1. Nyissuk meg az **iranyitott.xls** állományt (38. ábra).

	C	D	E
5		Ár [Ft/liter]	250
6			
7	Dátum	Liter:	Összesen:
8	2004.01.02	40	10000
9	2004.01.13	30	7500
10	2004.01.21	40	

38. ábra A kiinduló táblázat

2. Az E9-es cellában lévő **=D9*\$E\$5** képletet másoljuk át a E10-es cellába. A régi képletek tartományát (E8:E9) jelöljük ki, nyomjunk **CTRL+C**-t.
3. A tartomány maradjon kijelölt, adjuk ki a **Szerkesztés** menü **Irányított beillesztés** parancsát, a megjelenő párbeszédablakban kapcsoljuk be az **Értéket** nevű választókapcsolót, majd nyomjunk **OK**-t.
4. Nyomjunk **ESC**-t.
5. Az árat módosítsuk 280-ra (E5-ös cella).

A feladatot, ezzel (a képletek eredményének megőrzése) megoldottuk.

Mintafeladat: Sorok és oszlopok cseréje

Az **Irányított beillesztés** parancsra van szükségünk akkor is, ha egy táblázat összes sorát és oszlopát fel kell egymással cserélni. A mintatáblázatot a 39. ábra mutatja.

A parancs végrehajtásához az **Irányított beillesztés** párbeszédablakban be kell kapcsolni a **Transzponálás** jelölőnégyzetet.

Feladatunk egy táblázat másolatának előállítás a sorok és oszlopok cseréjével.

1. Nyissuk meg az **iranyitott.xls** állományt (39. ábra).

	C	D
16		Árak
17	TV	100000
18	HIFI	80000
19	Rádió	10000

39. ábra Mintatáblázat a sorok és oszlopok cseréjéhez

2. Jelöljük ki a táblázatot (C16:D19), nyomjunk CTRL+C-t.
3. Adjuk meg a céltartomány bal felső sarkát (C21).
4. Adjuk ki a **Szerkesztés** menü **Írányított beillesztés** parancsát, annak párbeszédablakában kapcsoljuk be a **Transzponálás** jelölőnégyzetet.
5. Kattintsunk az **OK** gombra
6. Nyomjunk **ESC**-t.

Mintafeladat: Matematikai műveletvégzés

Akkor is az **Írányított beillesztés** parancsot alkalmazzuk, ha a másolás művelettel egyúttal matematikai műveletet is végre szeretnénk hajtani. Magát a műveletet az **Írányított beillesztés** párbeszédablakban lehet beállítani a **Művelet** keret választókapcsolóival.

Feladat: a C16:D19 tartományban (iranyitott.xls, 40. ábra) található táblázatban egy árulista részlet látható. Feladatunk az árak 20%-os emelése (azaz 1,2-del való szorzás). Az 1,2 az F15-ös cellában található.

	C	D
16		Árak
17	TV	100000
18	HIFI	80000
19	Rádió	10000

40. ábra Mintatáblázat a matematikai műveletvégzéshez

1. Legyen aktív az F15-ös cella nyomjunk CTRL+C-t.
2. Jelöljük ki a céltartományt (D17:D19). Adjuk ki a **Szerkesztés** menü **Írányított beillesztés** parancsát, annak párbeszédablakában a **Művelet** keretben, válasszuk a **Szorzás** választókapcsolót.
3. Kattintsunk az OK gombra. Az áremelés megtörtént.
4. Nyomjunk **ESC**-t.

Mintafeladat: Csatolás

Csatolás (dinamikus másolat): a felhasználások jelentős részében olyan másolatra van szükség, ami megváltozik akkor, ha a forrástartomány is megváltozik.

Csatolás létrehozása menü paranccsal: a dinamikus másolat létrehozásának egyik eszköze az **Írányított beillesztés** ablak **Csatolva** nevű nyomógombja.

Csatolás létrehozása az egyenlőség jelével: ez egy igen kényelmes technika. Azon munkalapokat, melyekhez az adott cellát csatoljuk, egyszerre láthatóvá kell tenni. Be kell gépelni az egyenlőségjelet és sorban kattintani azon cellákra, amelyekhez csatolunk. Természetesen nem elfeledkezve arról, hogy a cellacímek közé műveleti jeleket kell gépelni.

A csatolások utóélete: a csatolásnál speciális csatolt képlet jön létre. Amikor egy állományt, amelyben csatolt képlet van, megnyit az Excel, egy kérdést intéz hozzánk, hogy frissítse-e a csatolásokat. Válaszoljunk igennel. Nagyon vigyázzunk csatolt állományainkra, ne nevezzük és ne helyezzük át azokat. Amennyiben óvatlanok vagyunk, még segítségünkre lehet a **Szerkesztés** menü **Csatolások** parancsa, amely a csatolások karbantartására szolgál.

Mintafeladat: Csatolás létrehozása menü paranccsal egy Excel munkalapon belül

Feladat: az *iranyított.xls* állomány *Mintafeladat* nevű munkalapjának D32-es cellájában állítsuk elő a D26-os cella dinamikus másolatát (csatolás).

1. Nyissuk meg az *iranyított.xls* állományt, ha az nem lenne megnyitva (41. ábra).

D26	=SZUM(D23:D25)		
	C	D	E
22	Hónapok	Mennyiség	
23	január	11	
24	február	22	
25	március	33	
26	Összesen:	66	

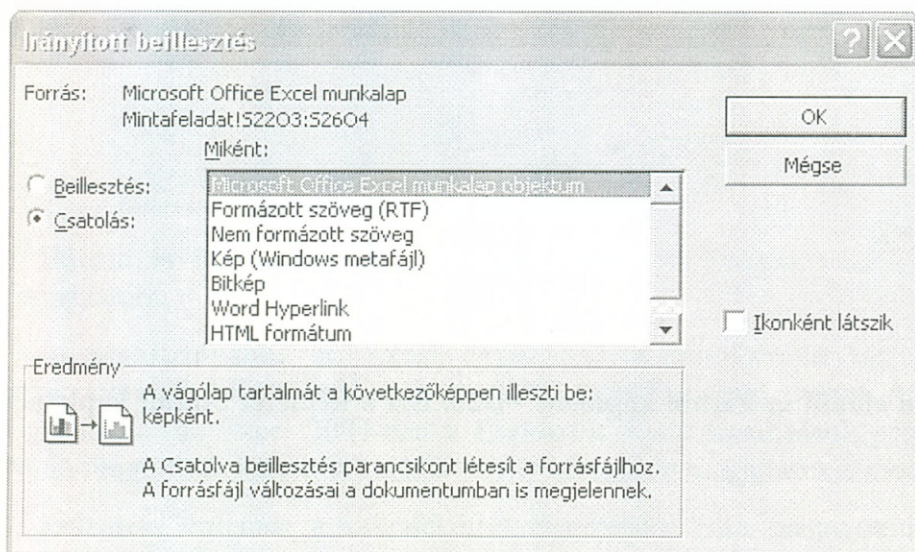
41. ábra Csatolás

2. Tegyük aktívvá a D26-os cellát (41. ábra). Nyomjunk **CTRL+C**-t.
3. Legyen aktív a D32-es cella. Adjuk ki a **Szerkesztés** menü **Irányított beillesztés** parancsát, kattintsunk a párbeszédablak **Csatolva** nyomógombjára.
4. Nyomjunk **ESC**-t a másolásból történő kilépésre.
5. **A csatolás próbája.** Írjuk át a D23-as cella tartalmát 2-re, ennek hatására megváltozik az első negyedév összesen értéke (D26) és meg kell, hogy változzon a csatolt cella (D32) tartalma is. Javítsuk vissza a D23-as cella tartalmát 11-re.

Mintafeladat: Csatolás létrehozása menü paranccsal (Excel és Word között)

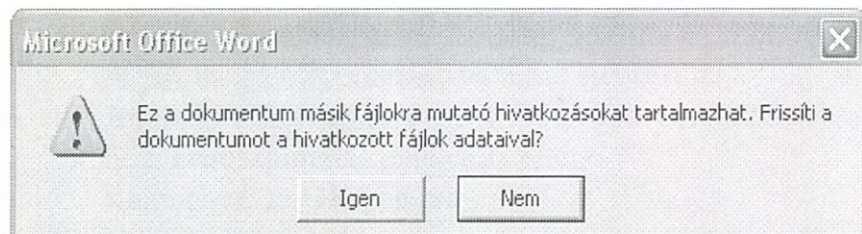
Feladatunk egy Excel táblázat másolása csatolással egy Word állományba.

1. Nyissuk meg az *iranyított.xls* állományt, ha az nem lenne megnyitva (41. ábra).
2. Jelöljük ki a táblázatot (C22:D26), adjuk ki a másolás parancsot (**CTRL+C**).
3. Nyissunk meg a Wordben egy már előzőleg elmentett (az sem probléma, ha üres) állományt, kattintsunk azon részére, ahol a táblázat másolatát látni szeretnénk és adjuk ki (Wordben) a **Szerkesztés** menü **Irányított beillesztés** parancsát.
4. A megjelenő párbeszédablakban válasszuk a **Csatolás** választókapcsolót és a listaablakból a munkalap objektumként történő beillesztést.



42. ábra Irányított beillesztés

5. **A csatolás ellenőrzése.** A Word állományt mentjük le és zárjuk azt be. Az Excelben változtassuk meg az **iranyított.xls** állományban a D23:D25 cellatartomány valamelyik cellájának tartalmát.
6. Mentjük le az Excel állományt és lépünk ki az Excelből.
7. Nyissuk meg a Word állományt, a következő párbeszédablakot kapjuk, kattintsunk az **OK** gombra:



43. ábra Kattintsunk az OK gombra

8. Az Excel állományból a Word felfrissíti a csatolási információkat.

Mikor érdemes a csatolást használni beillesztés helyett? Abban az esetben, ha nem akarjuk az Excel állomány méretét megnövelni. Veszélye a csatolásnak, hogy ha az Excel (Word) állományt valakinek odaadjuk, oda kell neki adni a csatolt állományokat is.

Milyen információk állnak rendelkezésünkre a csatolásról? Mivel példánkban Word állományhoz csatoltunk egy Excel állományt ezért, a Wordben kiadva a **Csatolások (Szerkesztés)** menü) parancsot, megkapjuk a **Csatolások** ablakot. Ne feledjük, ha az ablak valamelyik kezelőszervének nem értjük a funkcióját, elegendő arra egérrel, jobb gombbal kattintani és a megjelenő **Mi ez?** szövegre ballal és máris segítséget kapunk.

Csatolás létrehozása az egyenlőség jelével

1. Nyissuk meg az **iranyított.xls** állományt, ha az nem lenne megnyitva (44. ábra).

	C	D
22	Hónapok	Mennyiség
23	január	11
24	február	22
25	március	33
26	Összesen:	66
27		
28		
29		
30		
31		
32	I. n. év összesen:	
33	I. n. év összesen (képletel):	

44. ábra Csatolás az egyenlőség jelével

2. Legyen aktív a D33-as cella.
3. Gépeljük be az egyenlőség jelet (=).
4. Kattintsunk a D26-os cellára.
5. Nyomjunk **ENTER**-t.

Ezzel elértük, hogy a D33-as cellában előállt az alábbi képlet: **=D26**. Ezt a képletet csatolt képletnek nevezik.

2. fejezet: Számformázás

A számformázás alapfogalmai

Kijelzési forma: a számok kijelzési formája megváltoztatható. A számok formátumának könnyebb kézben tarthatósága érdekében a Microsoft számkategoriókat alakított ki, a felhasználók ezek közül választhatnak. Fontos tudni, hogy az Excel a szám tárolt formájával számol.

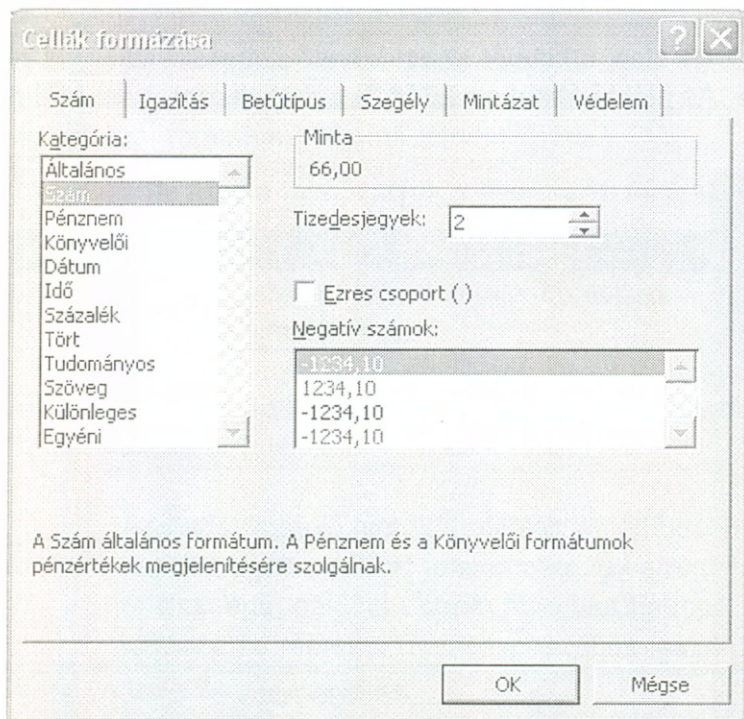
Egyéni számformák: mód és lehetőség van egyéni formátumkódok kialakítására. Az egyéni formátumkódokat az adott állomány őrzi meg, ez azt jelenti, hogy egy másik állományban, ha arra szükség van ismét létre kell hozni.

Súgó: a Súgó egyik legfontosabb kulcsszava a számformátum, ezt begépelve több irányú elágazásra nyílik lehetőségünk mind a szabványos, mind az egyéni számformátumok kategóriájában.

Parancsok: a számformázás végrehajtható a **Formátum** menü **Cellák** parancsa párbeszédablakának **Szám** fülére kattintva. Ugyanakkor a jobb gombos kattintás után kapott rövidmenü, **Cellaformázás** parancsa összetett párbeszédablakának **Szám** fülére kattintva is formázhatjuk celláinkat.

Kategóriák

Szám kategória: a **Formátum** menü **Cellák** parancsa összetett párbeszédablakának **Szám** fülére kattintva kapott párbeszédablakban a **Szám** kategória szolgál számaink adott hosszúságú megjelenítésére, ha kell ezres csoportosítására. A párbeszédablak **Minta** keretében láthatjuk az aktív cella várható formátumát.

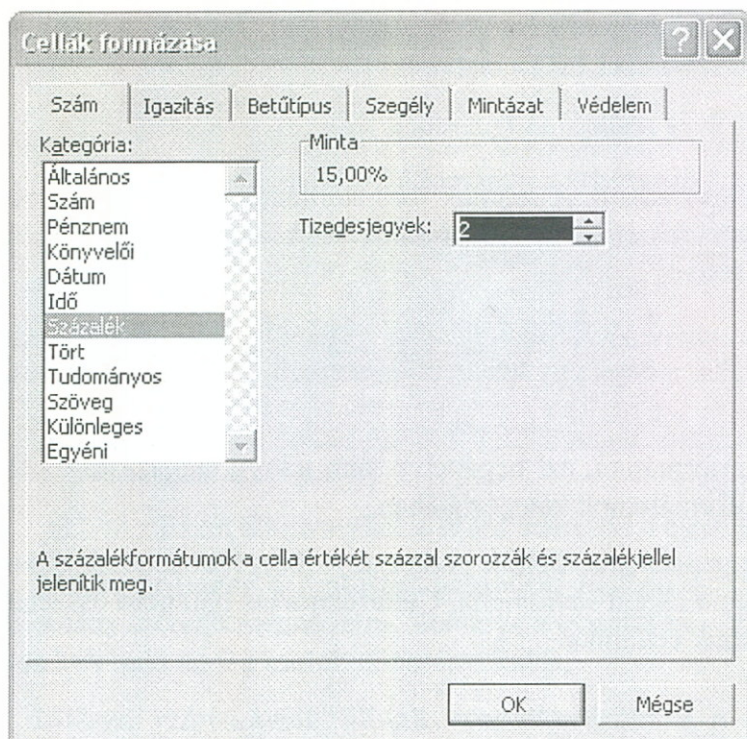


45. ábra A Minta keretben láthatjuk az aktív cella formátumát

Dátum és Idő kategória: a Dátum- illetve Idő kategória szolgál a cellák dátumra illetve időre történő formázására.

Automatizmusok: az Excel egyik u.n. automatizmusa az, hogy begépeléskor a számokban lévő szeparátorokat felismeri és ezek alapján meg is formázza azokat. A szeparátorok a Windowsból származnak. Ugyanakkor az Excel 2003-ban a **Eszközök** menü **Beállítások** parancsának összetett párbeszédablakában a **Nyelvfüggő** fülre kattintva ezektől eltérő szeparátor megadásra is mód van.

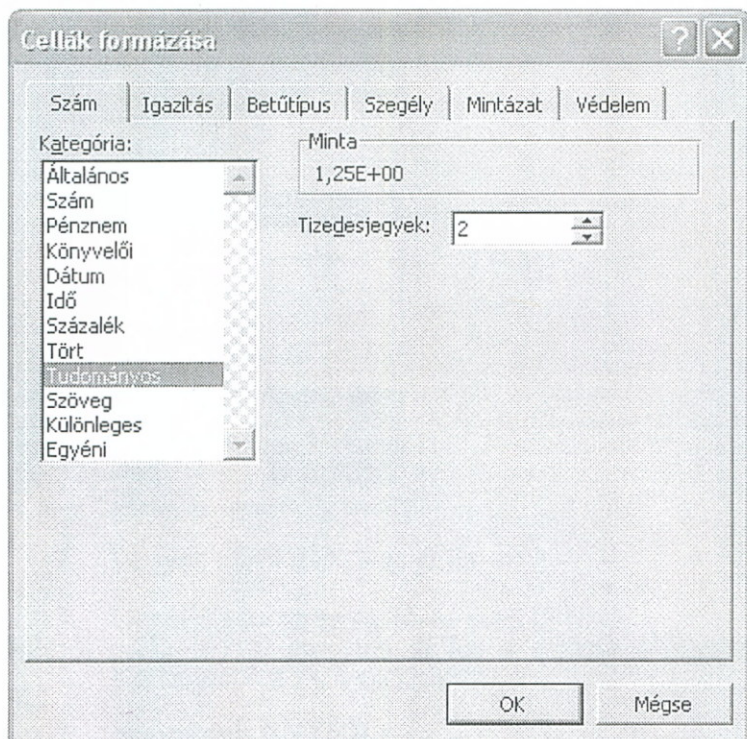
Százalék: érdemes a százalékkérték begépelése után megnyomni a % billentyűt. Amennyiben ezt nem tesszük, akkor a százalékkérték 100-ad részét kell begépelnünk. Amikor a számot megformázzuk százalékként, az Excel azt megszorozza 100-zal.



46. ábra Százalékra formázunk

Törtek: e kategória szolgál a törtek megjelenítésének megadására. Igen sokféle típus létezik a kategóriában, mégpedig azért, mert elég nagyszámú variáció létezik, már ami a törtek számlálója és nevezője számjegyeinek számát illeti.

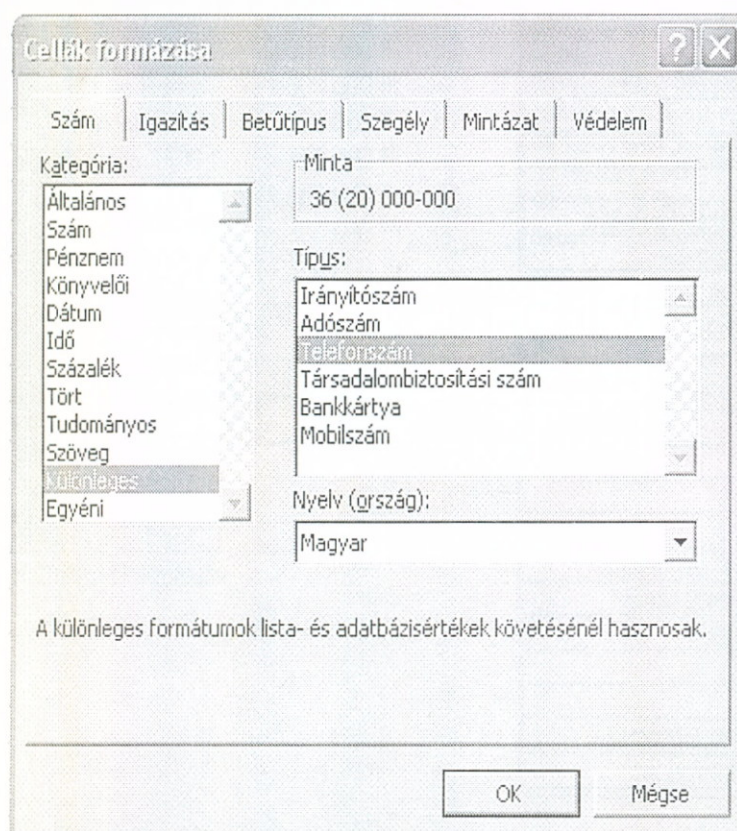
Tudományos: a Tudományos kategóriában az E a 10-es alap, a kitevőt az ezt követő szám(ok) adják, és ezt a hatványt az E-t megelőző számmal kell megszorozni. Az alábbi ábrán az $1,25 \cdot 10$ a nulladikont, azaz 1,25-öt látunk.



47. ábra Tudományos számforma

Szöveg: mivel az Excel automatizmusai sok bosszúság forrásai lehetnek, ezért különösen hasznos e kategória léte. Igen sokszor az a feladat, hogy a begépelte számokkal ne lehessen számolni. Ekkor megoldás e kategória használata. Ugyanakkor, ha a cellát vissza akarjuk formázni újból számformátumúra, akkor nem elegendő ezt megtenni, hanem a számokat is újból be kell gépelni vagy meg kell azokat szorozni eggyel.

Különleges: mint a mellékelt ábra mutatja, a Microsoft kidolgozott néhány speciális esetre formátkódokat, az élet ennél bonyolultabb és, ha nekünk ezektől eltérő formátum kódokra van szükségünk, akkor Egyéni formátum kódokra (**Egyéni** kategória) van szükségünk.



48. ábra Különleges számformátum kódok

Egyéni kategória: e kategória szolgál saját formátkódok kialakítására. A **Formátum** menü **Cellák** parancsa összetett párbeszédablakának **Szám** szegélyére kattintva és választva az **Egyéni** kategóriát, módunk nyílik saját formátkódjaink létrehozására.

Az alábbi ábrán bemutatjuk a fontosabb formátkódokat.

Kód	Hatása
#	Csak az értékes jegyeket jeleníti meg, a fölösleges nullákat nem.
0	Nullákkal egészíti ki a számot, ha az kevesebb számjegyet tartalmaz, mint ahány 0 szerepel a formátumban.
?	A tizedesvessző mindkét oldalán az értéktelen nullák helyén szóközt jelenít meg, így a rögzített szélességű, például Courier New betűtípussal formázott számok a tizedesvesszőhöz igazodnak.

49. ábra A fontosabb formátkódok

Természetesen ennél sokkal több formátkód van, a **Súgó** (számformátumok c. témaköre) a támpont ezek használatára.

Színkódok: A formátkódok színkódokat is tartalmazhatnak (lásd Súgó).

Az alábbi ábrán bemutatunk néhány, a gyakorlatban sűrűbben használt formátumkódot (50. ábra).

Formátumkód	Begépett szám (szöveg, függvény)	Hatása	Megjegyzés
0	4,6	5	Kerekít
0,00	4,375	4,38	Kerekít
####	0044	44	Nem látszik a vezető nulla
0000	0044	0044	Látszik a vezető nulla
# ##0	1546	1 546	Ezres szeparálás
0 F t	12	12	Hely kihagyás
0 " ezer Kg"	6000	6 ezer Kg	Ezerrel osztott érték kijelzése
???,???	12,123	12,123	Igazított tizedesjegyek
@ " bevétel"	Összes	Összes bevétel	Szöveg befűzése
0*	0*	12-----	Karakter ismétlése
...	123		Rejtett számforma
::	Maci	Maci	Csak a szöveg látszik
0,00[fekete]	123	123,00	Színkód
h	2003.05.04	5	Hónapok: 1-12
hh	2003.05.04	05	Hónapok: 01-12
hhh	2003.05.04	máj	Hónapok: jan-dec
hhhh	2003.05.04	május	Hónapok: január-december
hhhhh	2003.05.04	m	Hónapok: első betű szerint
éééé	2003.05.04	2003	Évek: 1900-9999
ó	2:26:34	2	Óra: 0-23
óó	2:26:34	02	Óra: 00-23
[ó]:pp	48:12:00	48:12	Eltelt idő órákban kifejezve
ó:pp:mm,00	=most()	9:15:56,04	Másodperc törtreszei is

50. ábra Gyakrabban használt formátumkódok

A formátumkódok értelmezése

Formátumkód mező hossza: a formátumkód mező (ha nincs benne feltétel) négy mezőrészből (szakaszból) állhat. Ezeket egymástól pontosvessző választja el. Az első kódrész adja a pozitív, a második a negatív számok, a harmadik a nulla, míg a negyedik a szöveg formátumát. A kódrészek pozícionáltak, azaz, ha például két pontosvesszőt követ egy formátumkód, akkor az a nulla formátumát határozza meg.

Érdeemes-e kísérletezni? Kísérletezni érdemes, az egyik legnagyobb segítség (szó volt már erről) az, hogy a Minta dobozban az Excel mindig mutatja az aktuális cella várható formátumát. Amennyiben az általunk begépett formátumkód értelmetlen, azt az Excel nem fogadja el.

Feltételes formázás formátumkóddal: a formátumkódban feltételeket is megadhatunk, ekkor a formázás a formátumkód függvénye. Szögletes zárójelek között egymást nem kizáró feltételek, és az azok teljesüléséhez tartozó formátumkódok is megadhatók.

Lényeges még az is, hogy a feltételes formátumkód maximum 4 mezőből (szakaszból) állhat. Ekkor a négy szakaszból maximum kettő a feltétel függvényében szabja meg a szám kijelzési módját. A harmadik szakasz a két feltételnek nem megfelelő szám, míg a negyedik a szöveg kijelzési módját adja meg.

Az alábbi feltételes formátumkód hatására a megformázott tartományban azok a számok, melyeknek értéke kisebb vagy egyenlő, mint száz piros, míg azok, amelyek értéke nagyobb, mint 100 kék színűek lesznek:

```
[Piros][<=100];[Kék][>100]
```

51. ábra Feltételes formátumkód

Feltételes formázás menüparanccsal: a **Formátum** menü **Feltételes formázás** parancsával maximum három feltétel adható meg egyszerre. Ez a formázási mód jóval kényelmesebb, mint a formátumkóddal megvalósuló. E formázási módszer "többet tud", mivel megadható, mint formázási feltétel függvények értéke is.

Mintafeladat: Számok formázása

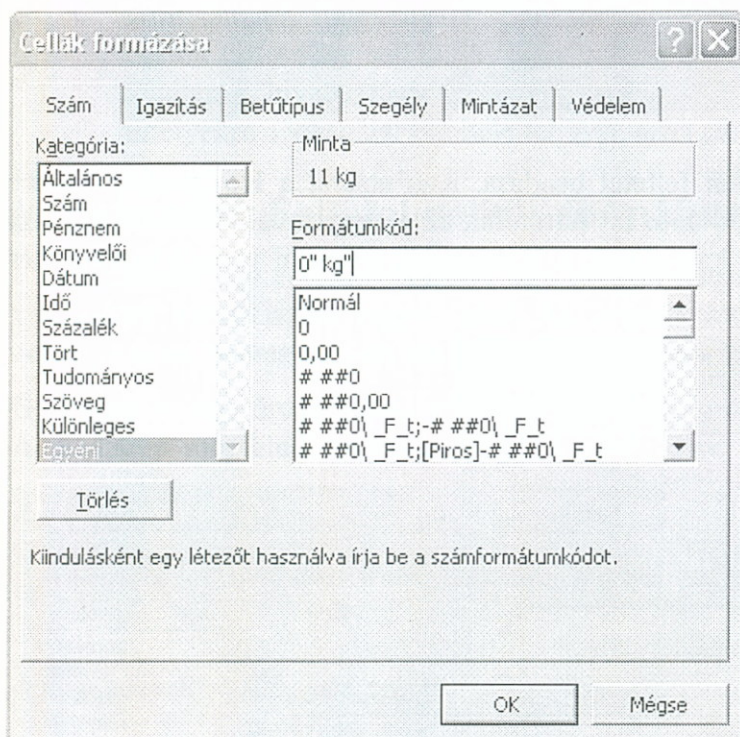
Formázzuk meg úgy a számokat, hogy mögöttük a **Kg** rövidítés legyen látható.

1. Nyissuk meg az **iranyitott.xls** állományt, ha az nem lenne megnyitva (52. ábra).

	C	D
22	Hónapok	Mennyiség
23	január	11
24	február	22
25	március	33
26	Összesen:	66

52. ábra A formázandó táblázat

2. Miután kijelöltük az D23:D26 cellatartományt, adjuk ki a **Formátum** menü **Cellák** parancsát, annak párbeszédablakában kattintsunk a **Szám** fülre és a **Kategóriák** közül, válasszuk az **Egyéni**.
3. Mivel a számok mögött a **kg** szót szeretnénk látni, ezért feladatunk ezen szöveg szám mögé helyezése. Ennek módja (ez akár a Súgóból is kiderül, a **"Számformátum"** témakörből), a szöveg macskakörmök közé helyezése, a számformátum kódok közül válasszuk a 0-át (53. ábra).



53. ábra Egyéni formátumkód

4. A kijelölt tartományt formázzuk vissza az alapformátumra, az **Általánosra**.

Mintafeladat: Feltételes számformátum kialakítása számformátum kóddal

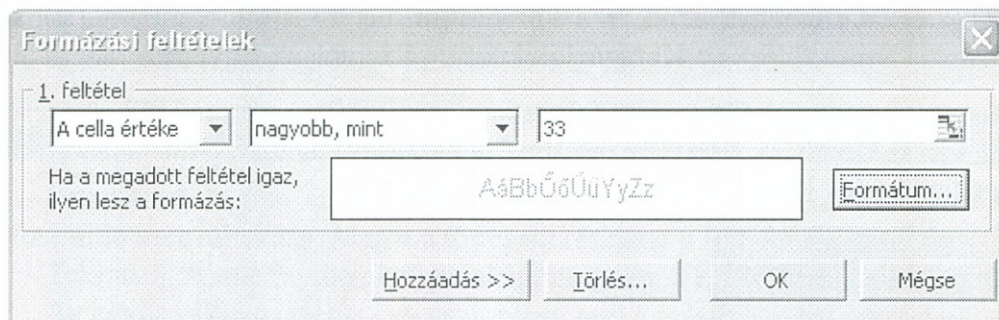
Feladat: feltételes számformátummal oldjuk meg azt, hogy ha egy adott cellatartományban a számok értéke nagyobb, mint 33, akkor azok színe legyen zöld, ha kisebb, mint 22, akkor kék és a többi szám színe bíbor legyen.

1. Nyissuk meg az **iranyitott.xls** állományt, ha az nem lenne megnyitva (52. ábra).
2. Legyen kijelölt a számokat tartalmazó tartomány (52. ábra), a fentiek szerint válasszuk az **Egyéni** kategóriát. Az alábbi számformátum kódot gépeljük be: `[Zöld][>33]0;[Kék][<22]0;[Bíbor]0`
3. A fenti formátumkódban a 0-ák biztosítják az egészszám kijelzést. A feltételeket, mint látjuk szögletes zárójelek közé kell helyezni.

Mintafeladat: Feltételes számformátum létrehozása menüparanccsal

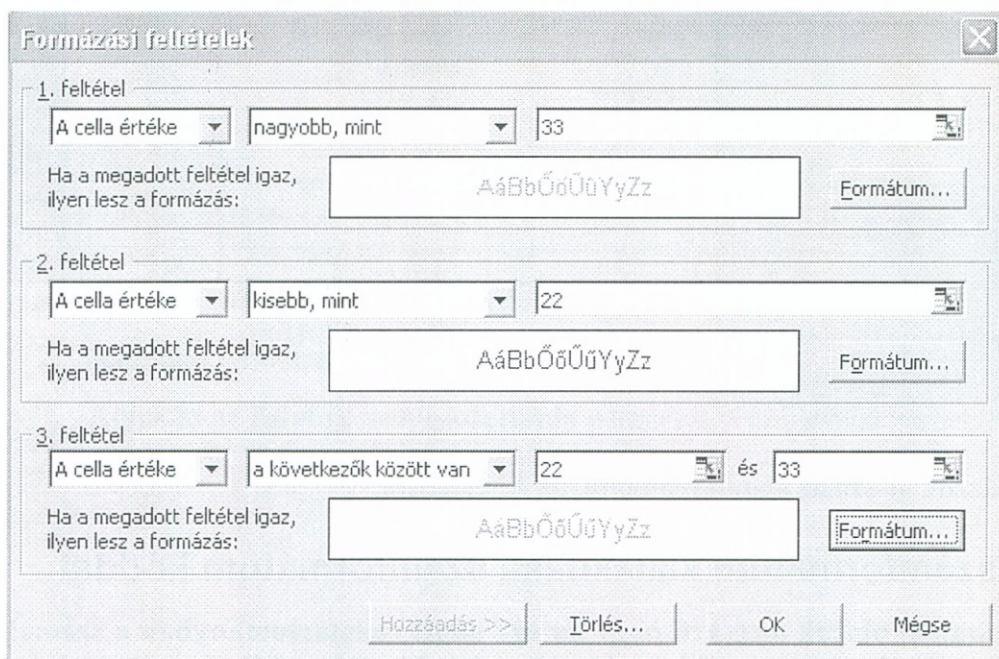
Feladat: a **Formátum** menü **Feltételes formázás** parancsával oldjuk meg a fenti feladatot, azzal a különbséggel, hogy a fent bíbor színűre formázott számok színe narancs legyen. Ennek az oka az, hogy a palettán nincs bíbor szín. Mielőtt a feladatot megoldanánk az D23:D26 cellatartomány celláit formázzuk meg **Általános** kategóriára.

1. Nyissuk meg az **iranyított.xls** állományt, ha az nem lenne megnyitva (52. ábra).
2. Jelöljük ki a D23:D26 cellatartományt. Adjuk ki a **Formátum** menü **Feltételes formázás** parancsát.
3. Mint az alábbi párbeszédablakban látszik, beadtuk az első feltételt. Nem látszik, hogy utána viszont kattintani kell a **Formátum** nyomógombra és a megjelenő ablak **Betűtípus** szegélyére és az ekkor megjelenő ablakban a **Szín** lenyílóra, ahonnan egy zöld színt kell választani.



54. ábra Az első feltétel megadása

4. Nincs is más feladatunk már, mint további két feltétel beadása. Kattintsunk a **Hozzáadás** (régebbi verziókban **Bővítés**) nyomógombra, ekkor a **Formázási feltételek** ablak egy további feltétel ablakkal bővül. A harmadik feltétel beadásához újfent kattintsunk a **Hozzáadás** (régebbi verziókban **Bővítés**) gombra.



55. ábra Megadtuk az összes feltételt

Mintafeladat: Feltételes számformátum létrehozása függvény eredménye alapján, menüparanccsal

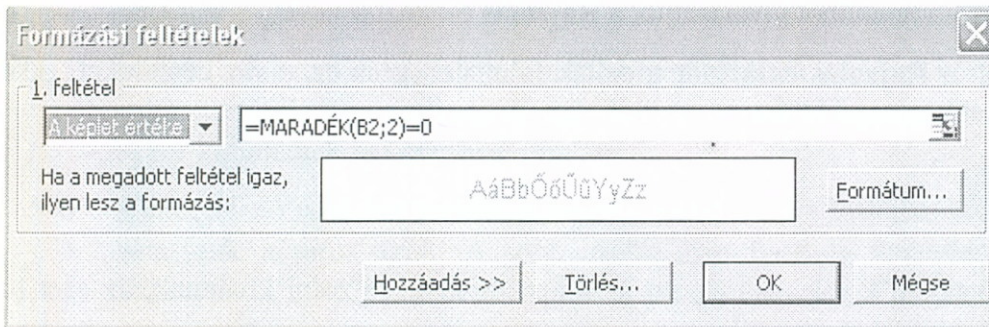
Feladat: a **Formátum** menü **Feltételes formázás** parancsával oldjuk meg azt, hogy egy cellatartomány páros értéket tartalmazó celláiban a számok színe legyen zöld.

1. Egy üres cellatartományba gépeljük be két páros és két páratlan számot.

	B	C
2	11	33
3	22	44

56. ábra Mintatáblázat a formázáshoz

2. A cellatartományt jelöljük ki.
3. Adjuk ki a **Formátum** menü **Feltételes formázás** parancsát.
4. A párbeszédablakot állítsuk be az alábbiak szerint:



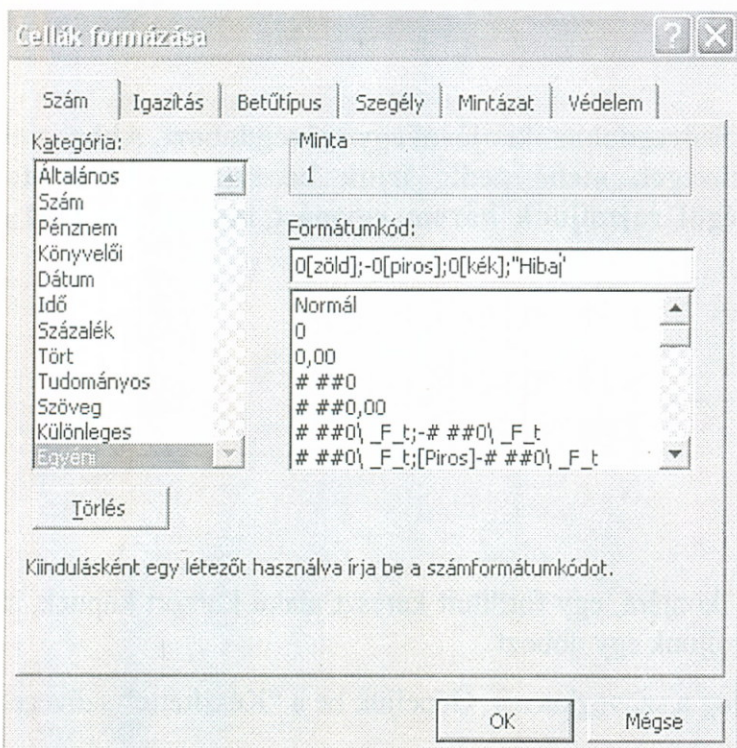
57. ábra Függvény, mint formázási feltétel

5. Mint látható, függvény is lehet formázási feltétel. A **MARADÉK** nevű függvénnyel vizsgáljuk a kijelölt tartomány celláit. A függvényben nem kell megadni a tartomány címét, hanem csak a legfelső cellájának hivatkozását (relatív hivatkozással, ez most B2). Csak azon cellák kerülnek formázásra, amelyekre a vizsgálat feltétele teljesül, azaz páros szám található bennük.

Mintafeladat: Feltétel nélküli formátumkód kialakítása

Feladatunk egy cellatartományt úgy megformázni, hogyha annak bármelyik cellájában pozitív érték van akkor annak színe legyen zöld, ha negatív akkor piros, ha nulla akkor kék, ha szöveg akkor a cellában ezen szöveg helyett azt a szót lássuk, hogy hiba. A számok egész számmal legyenek kijejezve.

Gépeljünk be egy üres munkalap B20:E20 tartomány celláiba balról jobbra 1-et, -1-et, nullát, és egy szöveget. A feladatnak megfelelő formátumkódot az alábbi ábrán láthatjuk.



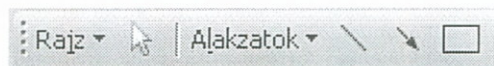
58. ábra Feltétel nélküli formátumkód

3. fejezet: Rajzolás

Rajzolás eszköztár

Táblázatok rajzzal történő szépítése: táblázataink szépítésének egyik eszköze, ha mi magunk rajzolunk a cellák fölé. A rajzolásra az Excelben egy külön eszköztár szolgál, neve: **Rajzolás**. Amennyiben nem látható az eszköztár, akkor megjelenítésének egyik kényelmes módja, ha jobb gombbal kattintunk bármelyik eszköztár tetszőleges helyére és a megjelenő rövidmenüből kiválasztjuk a **Rajzolás** eszköztár nevét.

A Rajzolás eszköztár helye: ha a **Rajzolás** eszköztár a cellák felett lebeg és ez zavar bennünket, akkor egérrel ragadjuk meg és húzzuk azt a képernyő aljára.

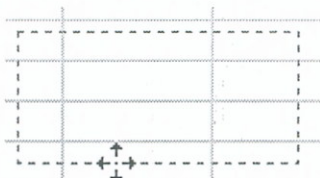


59. ábra A dokkolt eszköztár (részlet)

A rajzolás módja: miután kattintással kiválasztottuk azt az ikont amellyel rajzolni kívánunk, az egér bal gombját nyomva nekünk kell az objektumot megrajzolni.

A rajzolt objektum helye és mérete

Az objektumot mikor az kijelölt (téglalap, 60. ábra) a kereténél fogva lehet a képernyőn odébb helyezni. Vigyázzunk induláskor ne álljunk egérrel a kereten lévő kis négyszögekre, mert azok az objektum méretváltogatására szolgálnak. Amennyiben az objektumot a kis négyszögek húzásával át szeretnénk méretezni, akkor ezen folyamat közben tartsuk lenyomva a **SHIFT** billentyűt. Ez utóbbi billentyű a titka az arányos méretezésnek.



60. ábra Az objektum áthelyezése folyamatban van

Mintafeladat: Rajzolás

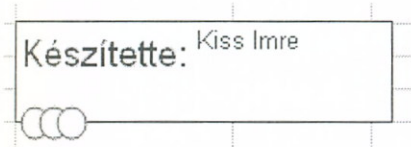
Rajzoljunk egy táblázat alá a Rajzolás eszköztár Szövegdoboz ikonjával egy szövegdobozt. Abba írjuk bele 14 pontos karakterekkel a “készítette” szöveget, mellé pedig írjuk be szintén 14 pontos karakterekkel felső indexbe, Kiss Imre nevét. Végül rajzoljunk három egymást metsző kis kört a szövegdoboz aljára.



61. ábra Kiss Imre neve felső indexben

1. Kattintva a **Rajzolás** eszköztár **Szövegdoboz** ikonjára, egy fordított kereszt alakú kurzort kapunk, az egér bal lenyomott gombjának kíséretében rajzoljunk egy dobozt.
2. Kattintsunk annak bal felső sarkába, ekkor villog a szövegkurzor. Gépeljük be a “Készítette” szöveget, jelöljük ki azt és formázzuk meg 14 pontosra.
3. A szöveg mögé gépeljük be Kiss Imre nevét és helyezzük a nevet felső indexbe (**Formátum/Szövegdoboz/Különleges hatások/Felső index**).

- Rajzoljunk a szövegdohoz aljára, átfedéssel három kis kört. Fontos tudni, hogy az **Ellipszis** ikonnal úgy kell kört rajzolni, hogy rajzolás közben nyomva kell tartani a **SHIFT** gombot. A kijelölt kör szélein kijelölő négyzetek láthatóak. Ezek egyikére állva egérrel állíthatjuk a kör méretét. A köröket egyesével, egérrel történő rákattintással jelöljük ki.



62. ábra Íme a körök is elkészültek

- Miután a feladatkiírásban adott módon megrajzoltuk a három kört, álljunk egérrel a bal szélső fölé (ez akkor történik meg amikor, az egér kurzor vékony hegyes nyilakkal végződő kereszt alakú lesz) és ekkor kattintsunk bal gombbal duplán.
- A megjelenő **Alakzat formázása** párbeszédablakban kattintsunk a **Vonal** keretben a **Szín** lenyílóra és válasszuk a piros színt. A jobb szélső kört hasonló technikával formázzuk meg, de azt zöldre színezzük.



63. ábra A színes körök

4. fejezet:Stílusok

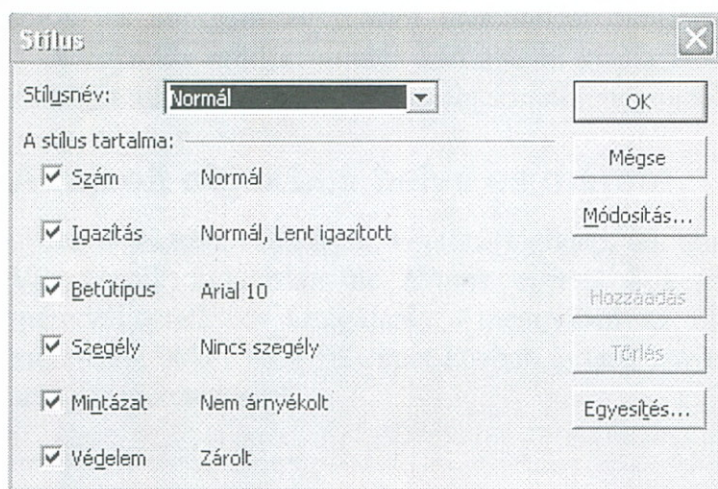
Alapismeretek

Mi a stílus? Névhez kötött formátumjellemzők összessége. Vigyázzunk a Normál stílusra. A Normál stílus az összes cella gyári alapformátumát tartalmazza, ezért nem illik megváltoztatni azt.

Munka stílusokkal

Stílus létrehozása

A stílusok létrehozására a **Formátum** menü **Stílus** parancsa szolgál. Mint az a következő ábrán látható, jelölőnégyzetekkel adható meg, hogy a stílus milyen formátumjellemzőkkel bírjon.



64. ábra Stílus jellemzők

1. A stílust általában egy már megformázott celláról lelopjuk. Ennek az a feltétele, hogy a parancs kiadásakor aktívnak kell lennie ennek a cellának.
2. Miután a **Stílus** párbeszédablak megjelenik, gépeljük be a létrehozandó stílus nevét.
3. Lehetőség van egyes formátumjellemzők megváltoztatására is, ebben az esetben kattintsunk a **Módosítás** gombra.
4. Miután a stílust létrehoztuk, az **OK** gombbal a párbeszédablakból lépünk ki.

A létrehozott stílus alkalmazása

1. Jelöljük ki a formázandó tartományt, adjuk ki a **Formátum** menü **Stílus** parancsát.
2. A **Stílusnév** lenyílóból válasszuk ki az alkalmazni kívánt stílus nevét.
3. Nyomjunk **OK**-t.

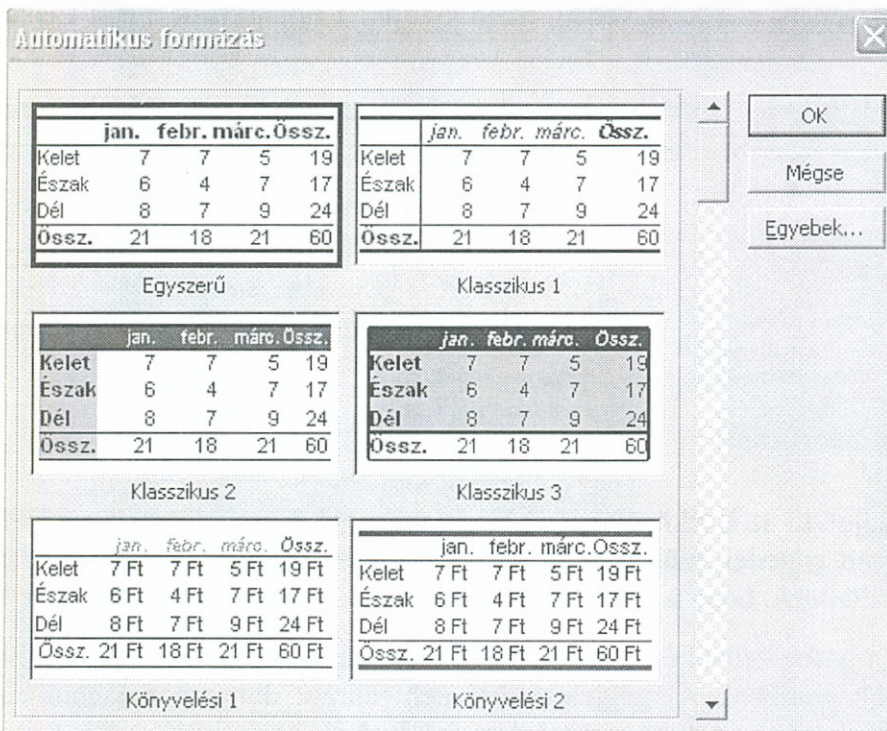
5. fejezet: Automatikus formázás

Alapok

Az Excel gyárilag készített formátumsablonokkal is rendelkezik. Ezeket munkáink formázásához kiválóan felhasználhatjuk.

A formázás

A **Formátum** menü **Automatikus formázás** parancsával a már kész táblázataink tetszetősen megformázhatók. A parancs egy sablonválasztékot nyújt segítségül ahhoz, hogy elkészült táblázataink profi módon nézhessenek ki.



65. ábra Automatikus formázás

Mi a teendő, ha az Excel az Automatikus formázáskor megváltoztatja az oszlopok szélességét, de ezt mi nem akarjuk? Az **Automatikus formázás** párbeszédablak **Egyebek** gombjának hatására az ablak jelölőnégyzetekkel bővül ki. A jelölőnégyzetek kikapcsolásával hatástalaníthatunk egyes formátumjellemezőket. Ennek kipróbálása végett érdemes kikapcsolni ki a **Széles/Magas** jelölőnégyzetet és végrehajtani akkor egy formázást.

Ne feledjük! Az **Automatikus formázás** parancs végrehajtása előtt jelöljük ki a formázandó tartományt.

6. fejezet: Védelem

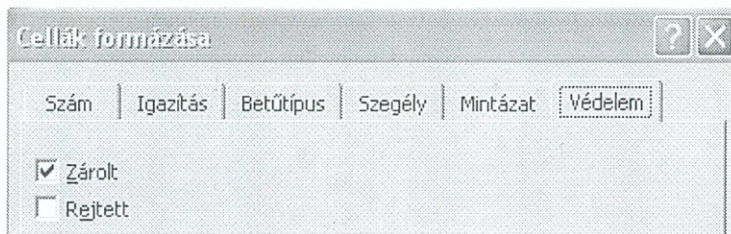
Alapok

Celláink és állományaink jelszavakkal védhetők az illetéktelen beavatkozás elől.

Az összes cella zárolt: celláink tartalma jelszavas védelemmel az illetéktelen hozzáféréstől megvédhető. Alapállapotban az összes cella zárolt állapotú, azaz bekapcsolva a védelmet, egyikbe sem lehet írni. Ami a mindennapi munkában általában nem szerencsés. A probléma megoldására jelöljük ki egy tartományt, ez nem lesz védett, és csak ezután indítjuk a cellák védelmét szolgáló utasítássorozatot.

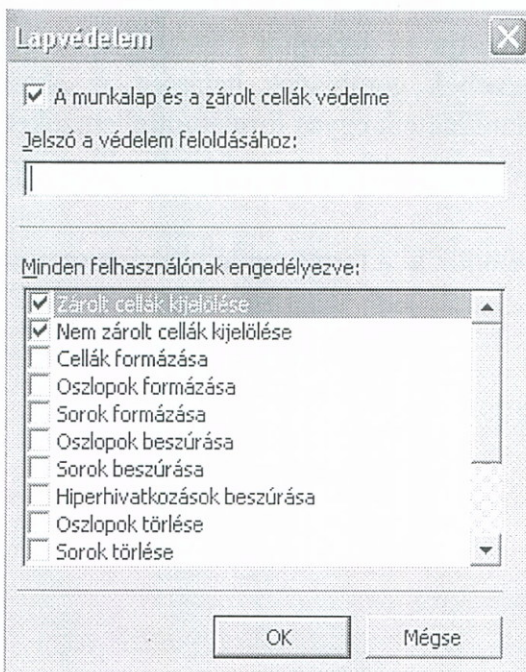
Cellavédelem bekapcsolása

1. Ha kell, kijelöljük azt a tartományt, amely nem lesz védett, majd kiadjuk a **Formátum** menü **Cellák** parancsát.
2. A megjelenő összetett párbeszédablakban kattintsunk a **Védelem** szegélyre, a **Zárolt** nevű jelölőnégyzetet kapcsoljuk ki. Ez utóbbi ténykedésünkkel a kijelölt tartomány védelmét megszüntetjük.



66. ábra A cellák védelme

3. A védelem bekapcsolása céljából adjuk ki az **Eszközők** menü **Védelem** majd, **Lapvédelem** parancsát. Állítsuk be a védelem hatókörét (mit engedélyezünk). Adhatunk jelszót is, ennek maximális hossza 255 karakter lehet. Ha egyszer úgy döntünk, hogy a védelmet megszüntetjük, ehhez a jelszót tudni kell.
 - **Segítség.** Ne feledjük, ha a párbeszédablakok valamelyik elemének szerepe nem egyértelmű, akkor kattintsunk arra jobb gombbal. A megjelenő **Mi ez?** feliratú dobozra bal gombbal kattintva, rövid emlékeztetőt kapunk az illető vezérlőelem feladatáról.



67. ábra A védelem bekapcsolása

Cellavédelem megszüntetése

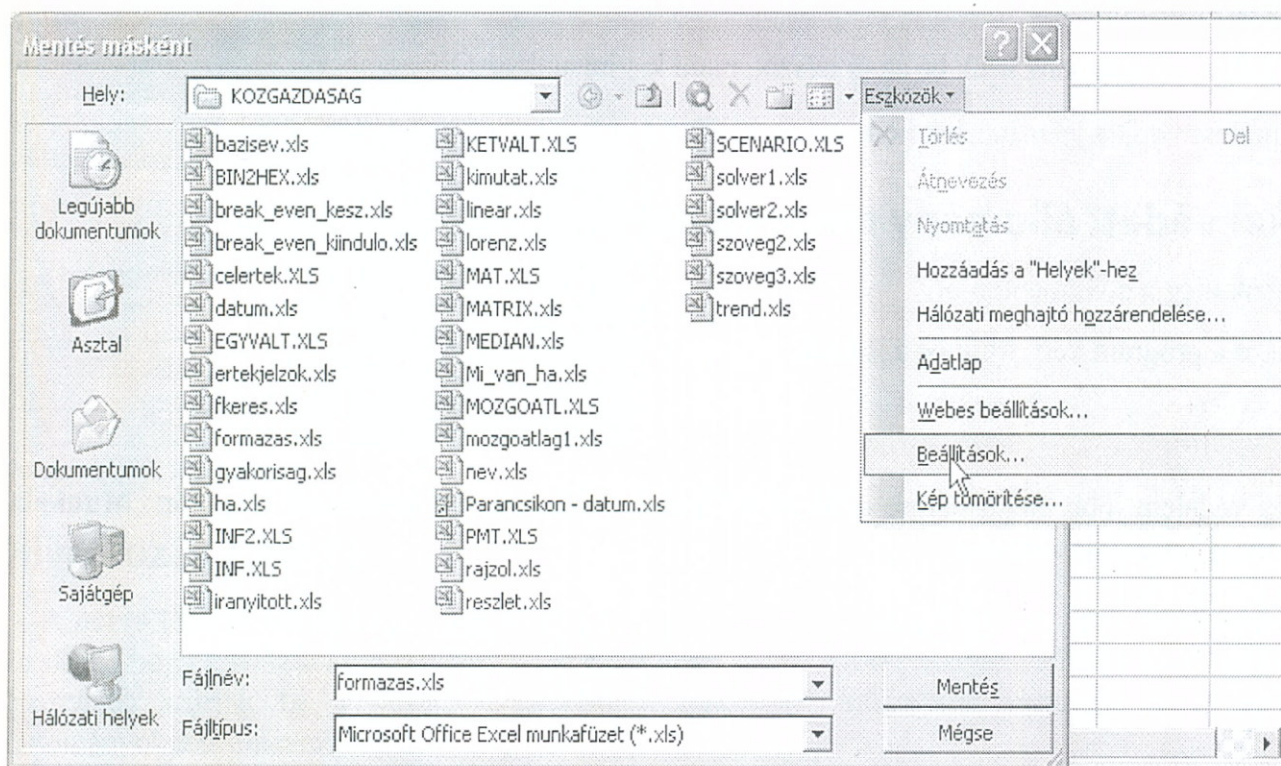
Ha a cellavédelmet meg szeretnénk szüntetni, adjuk ki az **Eszközök** menü **Védelem** parancsát, majd a **Lapvédelem feloldása** parancsot (ha lapvédelmet használtunk).

Állományvédelem

A **állományvédelem két szintje**: állományaink két különböző szituáció esetére védetté tehetők. A **Jelszó a betekintéshez** mező jelszavával az állományt megnyitástól, míg a **Jelszó a módosításhoz** mező jelszavával a nem kívánt módosítástól tehetjük védetté.

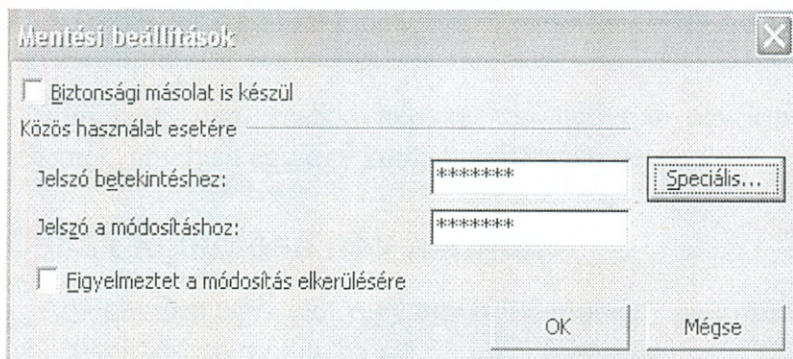
A védelem beállítása

- Adjuk ki a **Fájl** menü **Mentés másként** parancsát, ekkor megjelenik a **Mentés másként** párbeszédablak. Az ablak jobb felső sarkában kattintsunk az **Eszközök** (68. ábra) nevű lenyílóra, a legördülő parancsok közül válasszuk a **Beállításokat**.



68. ábra Válasszuk a Beállítások parancsot

- Az ekkor megjelenő **Mentési beállítások** (69. ábra) párbeszédablakban adhatjuk meg két jelszavunkat.



69. ábra Mentési beállítások

- A jelszavakat természetesen kétszer kell megadnunk.

Jelszók eltávolítása és megváltoztatása

Biztonsági figyelmeztetés: Célszerű erős (biztonságos) jelszavakat használni, amelyekben kisbetűk, nagybetűk, számok és szimbólumok egyaránt szerepelnek. Gyenge jelszó az, amely nem vegyíti ezeket az elemeket. Biztonságos jelszónak számít például az Y6dh%et5, gyengének a Kutya27. Ajánlatos megjegyezhető jelszót választani, így azt nem kell papíron tárolni.

1. Nyissuk meg a dokumentumot.
2. Amikor a program kéri, írjuk be a jelszót.
3. Kattintsunk az **Eszközök** menü **Beállítások** parancsára, majd a **Biztonság** fülre.
4. A **Jelszó a megnyitáshoz** vagy a **Jelszó a módosításhoz** mezőkben jelöljük ki a meglévő jelszavat képviselő helyettesítő karaktereket (általában csillag).
5. Az alábbi lehetőségek közül választhatunk:
 - A jelszó eltávolításához nyomjuk le a **DELETE** billentyűt, majd kattintsunk az **OK** gombra.
 - A jelszó módosításához írjuk be az új jelszót, majd kattintsunk az **OK** gombra.
 - Ha módosítottuk a jelszót, írjuk be ismét az új jelszót, majd kattintsunk az **OK** gombra.

Jelszókkal védett állomány megnyitása

1. Nyissuk meg a dokumentumot.
2. Amikor a program kéri, írjuk be a jelszót.

Súgó

A jelszavas védelem jobb megismerése céljából a Súgóban olvassuk el a következő tárgykört: **Munkalap vagy munkafüzet jelszavas védelme**

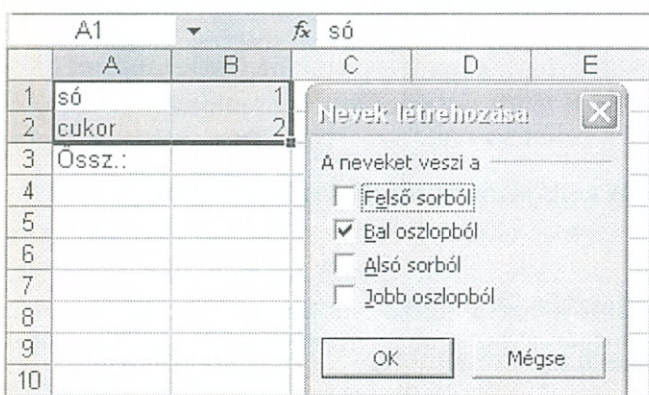
7. fejezet: Nevek

Az Excel lehetőséget ad cellák és tartományok elnevezésére. Az így létrejött nevekre számításainkban hivatkozhatunk.

Nevek létrehozása

Az alábbiakban bemutatjuk a nevek létrehozásának lépéseit.

1. Jelöljük ki azt a cella tartományt (**nevek.xls, A1:B2**), amelyben minden egyes B oszlopbeli cellának saját nevet akarunk adni.
2. Adjuk ki a **Beszűrés** menü **Név** majd **Létrehozás** parancsát.
3. Ha például (lásd a következő ábrát), a cellákhoz képest a nevek a bal szélső oszlopban találhatóak, hagyjuk bekapcsolva a **Bal oszlopból** jelölőnégyzetet, majd nyomjunk **OK**-t. Érdemes megemlíteni, hogy az Excel mindig detektálja, hogy a számokhoz képest a lehetséges nevek hol helyezkednek el.



70. ábra Nevek létrehozása

Hivatkozás a névre

A névre történő hivatkozás egyik módszere, ha a nevet beírjuk a függvényekbe (képletekbe). Egy másik módszer, hogy a **Beszűrés** menü **Név** majd **Beillesztés** parancsával (érdemes helyette az **F3**-as funkciógombot használni) szűrjük be a nevet. A megismert elnevezési módszerrel nemcsak egyes cellákat, hanem tartományokat is elnevezhetünk.

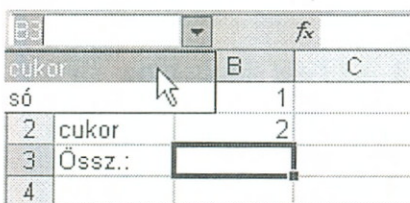
Cellanevek listája

A **Beszűrés** menü **Név** majd **Beillesztés** parancsát kiadva, a megjelenő párbeszédablakban található **Listát** nevű nyomógomb segítségével, a cellaneveket belistáztathatjuk egy cellatartományba.

Vigyázat: nem szabad képletekben tartománynevekre hivatkozni, mivel képletben csak olyan nevek lehetnek, amelyek egy-egy konkrét cellára hivatkoznak.

Cellák kijelölése név alapján

Amennyiben egy cella vagy egy cellatartomány elnevezett, úgy, az gyorsan kijelölhető a név alapján. Ennek legkényelmesebb módja, ha kattintunk a **Név mező** lenyílójára, ekkor megkapjuk a nevek listáját, amelyből bal gombbal választhatunk.



71. ábra Kijelölés név alapján

Mintafeladat: Nevek

Feladat: Nevezzük el a cellákat, a celláktól balra lévő nevekkal.

1. Nyissuk meg a **nevek.xls** állományt.
2. Jelöljük ki a táblázatot (A1:B2). Adjuk ki a **Beszűrés** menü **Név**, majd **Létrehozás** nevű parancsát. A megjelenő párbeszédablakban hagyjuk bekapcsolva a **Bal oszlopból** jelölőnégyzetet.
3. Miután a cellákat elneveztük, tegyük aktívvá a B3-as cellát és gépeljük be az alábbi képletet: **=só+cukor**.

Megjegyzés: ha sok nevet használunk, akkor érdemes azokat gépelés helyett, a **Beszűrés** menü **Név** parancsát majd a **Beillesztés** nevű parancsot kiadva (F3), bevinni a függvénybe (képletbe).

Feladat: A nevek felhasználásával jelöljük ki a só nevű cellát.

1. Kattintsunk a **Név mező** lenyílójára és legördülő cellanevek közül válasszuk a só-t.
2. Töröljük ki a cellanevek közül a só-t. Ehhez, adjuk ki a **Beszűrés** menü **Név** majd a **Név megadása** parancsát. A megjelenő párbeszédablakban a nevek listájában jelöljük ki a só-t és kattintsunk az **Eltávolítás** nyomógombra. Természetesen a képletünk hibát jelez (#NÉV).
3. A már megismert módszerrel újfent nevezzük el a B1-es cellát só-nak.

Feladat: a cellaneveket listáztassuk be a munkalap A15-ös cellájától kezdődő tartományba.

1. Legyen aktív az A15-ös cella.
2. Adjuk ki a **Beszűrés** menü **Név** parancsát, majd a **Beillesztés** parancsot.
3. A megjelenő párbeszédablakban kattintsunk a **Listát** nevű nyomógombra.

Cellák nem címkék alapján történő elnevezése

A cellák és cellatartományok elnevezhetők olyan nevek alapján is, amelyek nem szerepelnek a cellákban. Az egyik módszer erre a **Beszűrés** menü **Név** majd **Név megadás** parancs felhasználása, a másik a **Név mező** felhasználása. Mindkét megoldási módnál első lépésben a tartományt ki kell jelölni.

8.fejezet: Munkalapok

Háttérinformáció

Mint az tudvalevő, munkáinkat alapvetően a munkalapokon lévő cellákban végezzük. Egy üres Excel állomány megnyitásakor alapértelmezett számú (**Eszközök/Beállítások/Általános/Lapok száma új munkafüzetben**) munkalapot kapunk. A munkalapok minimális száma egy munkafüzetben 1, maximális számát a RAM mérete határozza meg.

Több munkalap kijelölése: A munkalapok kijelölésére két gyors módszer van. Amennyiben a kijelölendő *munkalapok szomszédosak*, akkor azok kijelöléséhez a bal szélső munkalap szegélyére kell kattintani az egérrel, majd a jobb szélső munkalap szegélyére, miközben végig nyomni kell a **SHIFT** gombot is. Ha a kijelölendő *munkalapok nem szomszédosak*, akkor a kijelölésükhöz a bal szélső munkalap szegélyére kell kattintani, majd a további munkalapok szegélyére egyesével és közben nyomni kell a **CTRL** gombot is.

Munkalap átnevezése: ha az átnevezendő munkalap szegélyére bal gombbal duplán kattintunk, akkor nevet adhatunk neki. A munkalapnév több szóból is állhat. Miután a nevet **begépeljük**, nyomjunk **ENTER**-t.

Munkalap(ok) törlése: a munkalapok törlésének legkényelmesebb módja, hogy azok kijelölése után, egy kijelölt munkalapszegélyre jobb gombbal kattintunk. A megjelenő rövidmenüből válasszuk a **Törlés** parancsot, majd engedélyezzük azt.

Munkalapok áthelyezése: az áthelyezni kívánt munkalap szegélyét az egér bal gombját nyomva tartva ragadjuk meg. Ekkor egy kis lap jelenik meg a munkalap szegélye fölött. A lap jelképezi a munkalapot. Ezt húzzuk végig a munkalapok fölött. Amikor munkalapok közé érünk, egy háromszög villan be. Ha ekkor felengedjük a bal gombot, akkor ezzel a munkalapot be is ejtettük ebbe a pozícióba. Ezzel a technikával, amennyiben több munkafüzet ablakait jelenítjük meg egy időben a képernyőn, a munkalap áthelyezhető akár egy másik munkafüzetbe (állományba) is.

Munkalapok másolása: amennyiben a fenti technika kiegészül a CTRL billentyű folyamatos nyomásával, nem áthelyezés, hanem másolás történik. (Ennek megtörténtét úgy látjuk, hogy a munkalap neve mellett egy szám látható zárójelben, jelezve, hogy az hányadik példány). **Fontos:** amikor beejtjük a "lapot" a munkalapok közé, előbb az egér bal gombját, majd csak ezt követően a CTRL billentyűt, kell felengednünk.

Csoport üzemmód: több munkalap egyszerre történő kiválasztásakor, megjelenik az állománynév mellett a Csoport szó, mivel átléptünk a Csoport üzemmódba. Addig, míg a Csoport üzemmódban vagyunk, a formázó parancsok az összes kijelölt munkalap azonos pozíciójú celláira azonosan hatnak. Ha valamelyik cellába beírunk egy szöveget, akkor az összes kijelölt munkalap azonos pozíciójú cellájába beíródik. Az eddigiek jól mutatják, hogy a Csoport üzemmód több munkalap azonosra formázásának eszköze.

Hogyan lépünk ki a Csoport üzemmódból? Kattintsunk valamelyik kijelölt munkalap szegélyére jobb gombbal. A megjelenő rövid menüből válasszuk a **Csoport törlése** parancsot. A Csoport üzemmódból úgy is kiléphetünk, ha kattintunk valamelyik nem kijelölt munkalap szegélyére.

Mintafeladat: Munkalapok

Feladat: a munkalapok.xls állományban, a Maci Laci nevű munkalapot nevezzük át márciusra.

1. Nyissuk meg a **munkalapok.xls** állományt.
2. Bal gombbal duplán kattintsunk a **Maci Laci** nevű munkalap munkalapszegélyére.
3. Gépeljük be: március
4. Nyomjunk **ENTER**-t.

Feladat: állítsuk elő a január nevű munkalap másolatát.

1. Nyissuk meg a **munkalapok.xls** állományt.

2. Tegyük aktívvá a **január** munkalapot.
3. Álljunk rá egérkurzossal a **január** munkalap szegélyére, ekkor az egérkurzor alakja balra álló nyíl lesz. Nyomjuk az egér bal gombját és a **CTRL** billentyűt és az egérrel vándoroljunk jobbra (egy kis lapocska vándorol).
4. A szegélyek között felvillan egy-egy háromszög. Amikor az utolsó munkalap szegélye mögé értünk, engedjük fel előbb az egér bal gombját, majd a **CTRL** billentyűt. A másolat létrejött. Az új munkalap neve **január (2)** lesz. Ahol a 2-es index jelzi, hogy ez a lap a **január** munkalap 2. példánya.

Megjegyzés: áthelyezni is hasonló technikával kellene a munkalapot, azzal a különbséggel, hogy nem kellene a CTRL billentyűt nyomni.

5. Töröljük a **január(2)** munkalapot, ehhez jobb gombbal kattintsunk a munkalap fülére. A megjelenő rövid-menüből válasszuk a **Törlés** parancsot.

Feladat a Csoport üzemmód jobb megismerésére

Ha kijelölünk legalább két munkalapot, akkor azzal belépünk a **Csoport** üzemmódba. Ezen üzemmód a csoportos formázás eszköze. Ha például, gépelünk egy cellába, akkor az összes kijelölt munkalapon ugyanebbe a cellába bekerül a gépelt információ. Amennyiben megformázunk egy tartományt, akkor ezen tartomány az összes kijelölt munkalapon megformázódik.

1. Nyissuk meg a **munkalapok.xls** állományt, ha az nem lenne megnyitva.
2. Ha a harmadik munkalap neve: **Maci Laci**, kattintsunk duplán a munkalap szegélyére, ennek hatására a név kijelölt lesz, gépeljük be, hogy március, majd nyomjunk **ENTER**-t.
3. Legyen aktív a január munkalap.
4. Jelöljük ki az összes munkalapot.
5. Jelöljük ki a B4:C6 cellatartományt,
6. Adjuk ki a **Formátum** menü **Automatikus formázás** parancsát, válasszuk a **Klasszikus 2** sablont. Ezzel elvégeztük a csoportos formázást.
7. A Csoport üzemmódból való kilépéshez kattintsunk valamelyik munkalap szegélyre jobb gombbal, a megjelenő rövid-menüből válasszuk a **Csoport szétbontása** parancsot.

9.fejezet: Összesítés

Mikor alkalmazunk összesítést? Gyakran felmerülő feladat két vagy több azonos szerkezetű táblázat azonos pozíciójú vagy nevű celláival történő műveletvégzés. A művelet maga az esetek többségében összeadás.

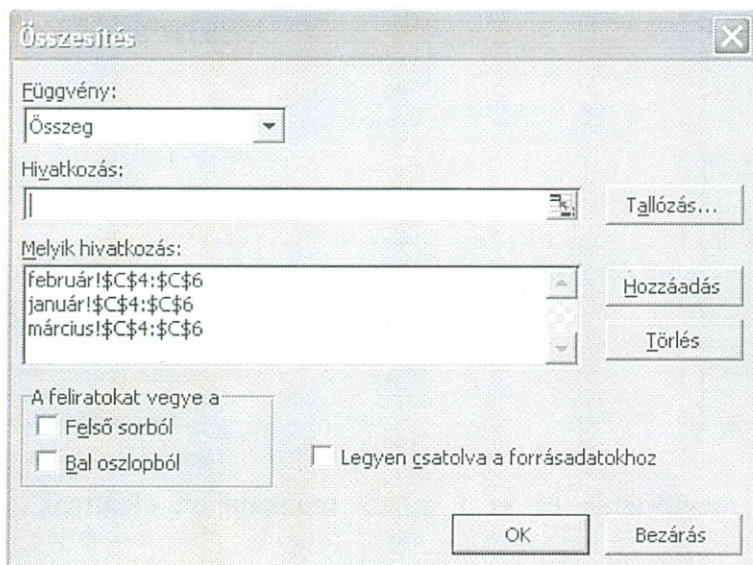
Az adatok összesítése történhet:

- háromdimenziós hivatkozások használatával (ajánlott módszer);
- az **Adatok** menü **Összesítés** parancsának használatával;
- az **Adatok** menü **Kimutatás** parancsával.

Mintafeladat: Az Összesítés menete

Tegyük fel, hogy feladatunk **Összesítés az Adatok menü Összesítés parancsának használatával**. Célunk három munkalap azonos felépítésű táblázatai azonos pozíciójú cellái összegének képzése egy negyedik munkalpra.

1. Nyissuk meg a **munkalapok.xls** állományt, ha az nem lenne megnyitva.
2. Legyen aktív az **I. N. év** nevű munkalap, jelöljük ki a **C4:C6** cellatartományt.
3. Adjuk ki az **Adatok** menü **Összesítés** parancsát.
4. Kattintsunk a **január** munkalpra és jelöljük ott ki a **C4:C6** cellatartományt (miközben a párbeszédablakban a **Hivatkozás** mezőben van a kurzor).
5. Kattintsunk a **Hozzáadás** gombra. Ennek hatására a hivatkozás a párbeszédablak **Melyik hivatkozás** listaablakába kerül. Itt majd azon hivatkozások gyűlnek össze, amelyek részt vesznek az **Összesítés** műveletében.
6. Kattintsunk a **február** nevű munkalap szegélyére, az Excel kijelöli ugyanazt a tartományt, amelyet a január munkalapon kijelöltünk. Lényeges tudni, hogy a táblázatoknak nem kell ugyanazon cellatartományban elhelyezkedniük. Mivel a táblázat az előző munkalapon lévővel megegyező cellákban helyezkedik el, nem kell adnunk új tartomány címet. Kattintsunk a **Hozzáadás** gombra.
7. Kattintsunk a **március** nevű munkalap szegélyére, majd a párbeszédablak **Hozzáadás** gombjára.
8. A **Függvény** lenyílóból válasszuk ki a szükséges függvényt, ez legyen most az **Összeg**.



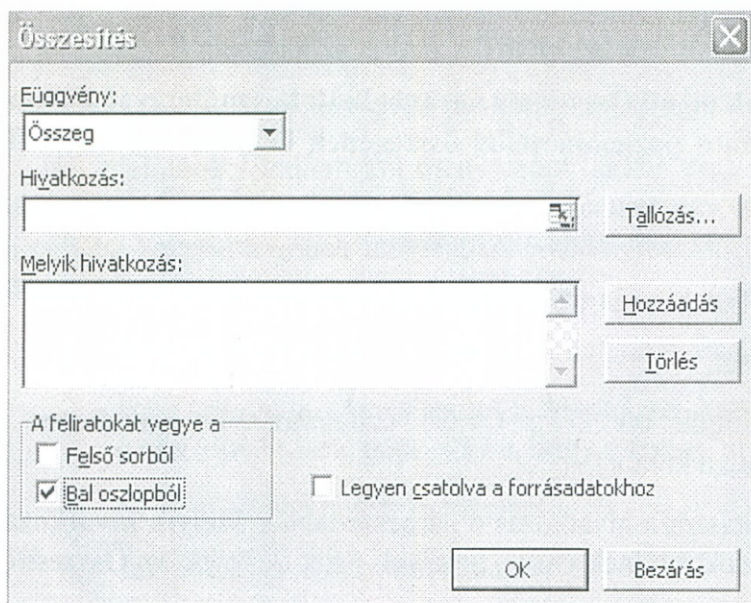
72. ábra Az Összesítés párbeszédablak

9. Kattintsunk az **OK** gombra. Ezzel az **Összesítés** megtörtént, az **I. n. év** munkalapon a három munkalap azonos pozíciójú celláinak összegei megjelennek.

Mintafeladat: Az Összesítés menete (nevek felhasználásával)

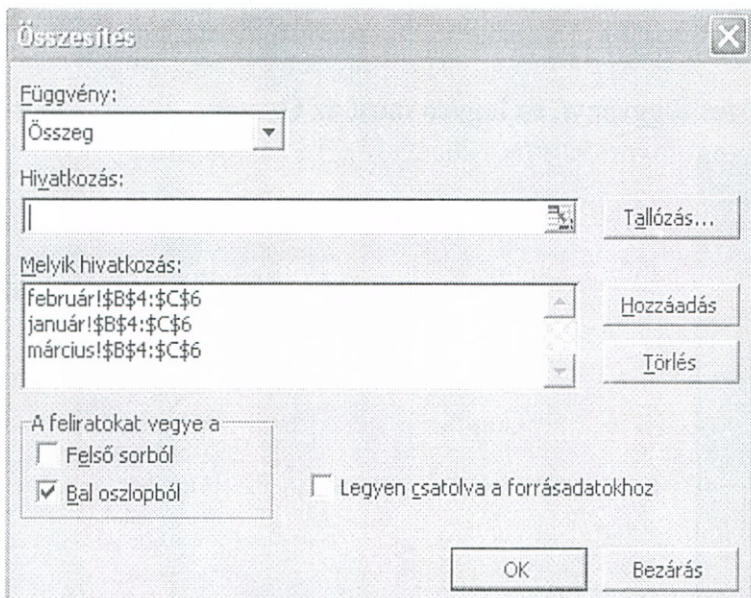
Feladat: három munkalappal rendelkezünk, amelyeken a cellák elnevezettek. Feladatunk a nevek felhasználása összesítéshez.

1. Nyissuk meg a **munkalapok.xls** állományt, ha az nem lenne megnyitva.
2. Győződjünk meg, hogy a január, február, március nevű munkalapon a cellák a tőlük balra címkék alapján elnevezettek-e. Ha nem, nevezzük el azokat.
3. Legyen aktív az I. N. év nevű munkalap, jelöljük ki a B4:C6 cellatartományt. Adjuk ki az **Adatok** menü **Összesítés** parancsát. A megjelenő párbeszédablakban kapcsoljuk be a **Bal oszlopból** jelölőnégyzetet (a nevek a cellákhoz képest balra vannak).



73. ábra Név alapján összesítünk

4. Ettől a ponttól teendőink megegyeznek az előző mintafeladat 4-9. lépéseivel, azzal a különbséggel, hogy nem a C4:C6, hanem a B4:C6 cellatartományt kell kijelölünk. Miután eljutottunk ideig a párbeszédablak a következőképpen néz ki:



74. ábra Név szerinti összesítéshez készülünk

5. Kattintsunk az OK gombra, az összesítés megtörténik és az I. n. év munkalapon előállnak az eredmények.

	B	C
4	só	3
5	cukor	6
6	Össz.	9

75. ábra Az Összesítés eredménye

Megjegyzés: úgy tűnik, hogy a kétféle összesítés között nincs különbség, pedig van, ehhez elegendő törölni valamelyik címkét, és újfent elvégezni az összesítést. Ehhez ekkor nem kell megint beadni a tartományokat.

Az Összesítés párbeszédablak kezelőszervei

Felső sorból (Bal oszlopból) jelölőnégyzetek (74. ábra): lehetőségünk van (nemrég tanultuk) a táblázatokban tartományokat elnevezni. Az **Összesítés** parancs képes detektálni az elnevezett tartományokat és azokkal műveleteket végezni. Nekünk ezzel a két jelölőnégyzettel kell megadni, hogy az Excel honnan vegye a neveket.

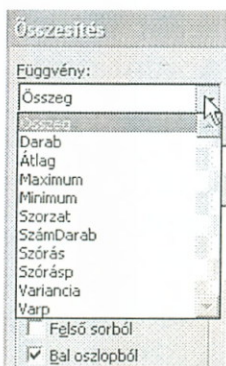
Legyen csatolva a forrásadatokhoz jelölőnégyzet: mivel az Excel úgy képzi a művelet eredményét, hogy az(ok) konstans(ok) ezért, ha megváltoztatunk valamelyik táblázatban egy számot, azt az eredmény nem követi. Amennyiben szeretnénk, hogy az esetleges változások egyből megjelenjenek az eredményben, akkor ezt a jelölőnégyzetet az **Összesítés** parancs végrehajtásakor be kell kapcsolnunk.

A következő ábra egy olyan esetet mutat, amikor a **Legyen csatolva a forrásadatokhoz jelölőnégyzetet** bekapcsoltuk az **Összesítés** parancs párbeszédablakában. Mint látható speciális képernyőképet kaptunk (tagolás, vázlat). Nekünk, ha például a vázlat legmagasabb szintjére van szükségünk az 1 jelű vázlat szint kapcsolóra kell kattintanunk (az oszlopnevektől kicsit balra). A tagolásról (vázlatról) az adatbázis-kezelésnél még hallunk.

E7		=SZUM(E4:E6)				
1	2	B	C	D	E	F
	1					
	2					
	3					
	4			Munkalapok	1	
	5			Munkalapok	1	
	6			Munkalapok	1	
	7	só	só		3	
	8			Munkalapok	2	
	9			Munkalapok	2	
	10			Munkalapok	2	
	11	cukor	cukor		6	
	12			Munkalapok	3	
	13			Munkalapok	3	
	14			Munkalapok	3	
	15	Össz.	Össz.		9	
	16					

76. ábra A tagolás 1. szintje

Függvény lenyíló: az adatokon végrehajtandó műveletet a függvény lenyílóból kell választanunk.



77. ábra Innen választhatunk függvényt

10.fejezet: Háromdimenziós hivatkozások

Alapfogalmak

Háromdimenziós hivatkozások használata adatok egybegyűjtésére: ha a források elrendezése és adatfelirata különböző, vagy egyedi elrendezést kívánunk létrehozni, vagy egyedi összegző képleteket szeretnénk használni az összesítéshez, akkor erre a célra használjunk háromdimenziós hivatkozásokat.

Adatok felvétele összesítésbe: amennyiben minden forrásmunkalap elrendezése azonos, a háromdimenziós képletekben lapnév tartományt is használhatunk.

Gyakorlat: Háromdimenziós hivatkozások

Feladat: lapnév tartomány nélküli Háromdimenziós hivatkozások

1. Nyissuk meg a **munkalapok.xls** állományt, ha az nem lenne megnyitva.
2. Tegyük aktívvá az I. n. Év munkalapot, kattintsunk a C12-es cellára.
3. Gépeljük be az = jelét, majd kattintsunk a január munkalap szegélyére.
4. Kattintsunk a C6-os cellára.
5. Gépeljük be egy + jelet.
6. Tegyük aktívvá a február munkalapot, kattintunk annak C6-os cellájára.
7. Nyomjunk **ENTER**-t. Ennek hatására az Excel visszavált az I. N. év munkalapra és elkészül a háromdimenziós képlet:

	C12	fx =január!C6+február!C6				
	A	B	C	D	E	F
12		jan+febr Össz.:	6			
13						

78. ábra Háromdimenziós hivatkozás lapnév tartomány felhasználása nélkül

Háromdimenziós hivatkozások Lapnév tartomány felhasználásával

1. Nyissuk meg a **munkalapok.xls** állományt, ha az nem lenne megnyitva.
2. Tegyük aktívvá az I. n. Év munkalapot, kattintsunk a C12-es cellára. Töröljük ki annak tartalmát, ha az nem üres.
3. Gépeljük be a következőket: **=szum(**
4. Kattintsunk a január munkalap munkalapszegélyére, majd a munkalap C6-os cellájára.
5. A SHIFT gombot nyomva tartva kattintsunk a február munkalap szegélyére, majd nyomjunk ENTER-t. Az Excel visszavált az I. n. év munkalapra és elkészíti a függvényt:

	C12	fx =SZUM(január:február!C6)				
	A	B	C	D	E	F
12		jan+febr Össz.:	6			

79. ábra Háromdimenziós hivatkozás lapnév tartomány felhasználásával

11. fejezet: Megjegyzés

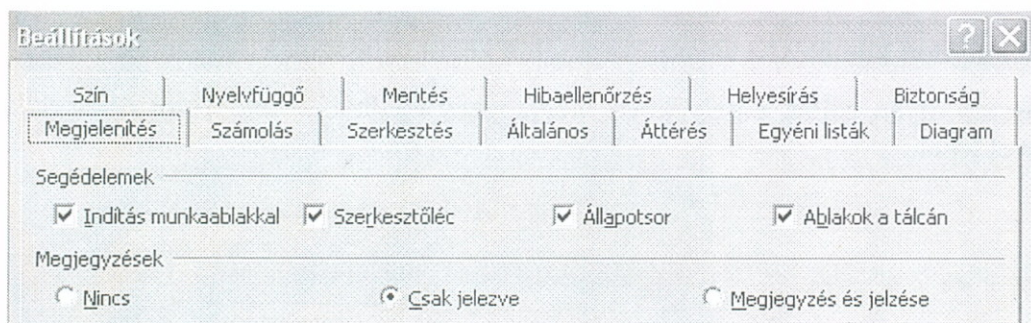
Fontos számításokat tartalmazó celláinkhoz megjegyzéseket fűzhetünk, amelyeket akár ki is nyomtathatunk.

Megjegyzés készítése

1. A **Beszúrás** menü **Megjegyzés** parancsával (elsősorban képlet vagy függvény tartalmú) celláinkhoz megjegyzést készíthetünk, mely alapbeállítás szerint akkor látszik, ha a cellára állunk az egér kurzorral.
2. Amikor a megjegyzés elkészült, kattintsunk egy cellára, ezzel lépve ki a megjegyzés szerkesztéséből.

A megjegyzések beállításai

Kiadva az **Eszközők** menü **Beállítások** parancsát, megjelenik egy összetett párbeszédablak, ebben kattintsunk a **Megjelenítés** fülre. A **Megjegyzések** keret kezelőszervei (80. ábra) adják a megjegyzések alapbeállításait. Érdeemes a **Csak jelezve** választókapcsolót választani. Ekkor az Excel egy kis piros háromszöget jelenít meg azokban a cellákban, amelyekben megjegyzés van. A megjegyzést úgy lehet visszanyerni, ha egérkurzorral a cella fölé pozícionálunk.



80. ábra Megjegyzések beállításai

A megjegyzések nyomtatása

Ha megjegyzéseinket ki szeretnénk nyomtatni, akkor győződjünk meg előtte a megjegyzések nyomtatási beállításáról.

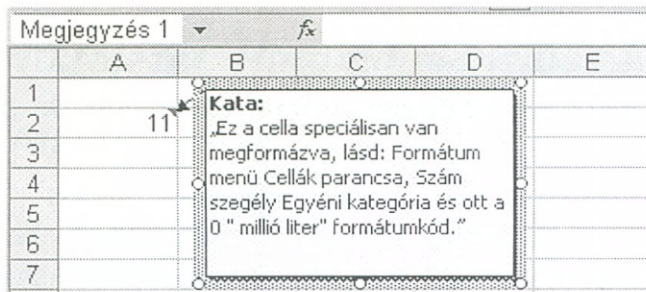
1. Adjuk ki a **Fájl** menü **Oldalbeállítás** parancsát, kattintsunk a **Lap** nevű fülre.
2. Állítsuk be a **Jegyzetek** nevű lenyíló, érdemes abból a **Lap végére** lehetőséget választani, ekkor az Excel egy (vagy több) külön lapra nyomtatja a megjegyzéseket.

Megjegyzések szerkesztése

Jobb gombbal kattintva a megjegyzést tartalmazó cellára, egy rövidmenüt kapunk. A rövidmenü, mint látható tartalmaz szerkesztést indító és megjegyzést törölő parancsot egyaránt.

Mintafeladat: Megjegyzés készítése

1. Egy üres munkalapra gépeljünk be egy számot az A2-es cellába, ez a cella maradjon aktív.
 - ✦ Az A2-es cellához készítsünk egy megjegyzést ezzel a szöveggel: „Ez a cella speciálisan van megformázva, lásd: Formátum menü Cellák parancsa, Szám szegély Egyéni kategória és ott a 0 " millió liter" formátumkód.”
2. Adjuk ki a **Beszúrás** menü **Megjegyzés** parancsát, megjelenik egy szövegdoboz, ebbe kell gépelni a megjegyzés szövegét. Miután a gépelést befejeztük, kattintsunk egy cellára a szerkesztésből való kilépés céljából.



81. ábra Az elkészült megjegyzés

Vigyázat: az, hogy miként látszik a megjegyzés szövegdoboza, beállításfüggő.

Mintafeladat: Töröljük a A2-es cellához fűzött megjegyzést.

Legyen aktív a A2-es cella, jobb gombbal kattintsunk rá, a megjelenő rövidmenüből válasszuk a **Megjegyzés törlése** parancsot.

Mintafeladat: Jelöltessük ki az Excellel az összes olyan cellát, amelyhez megjegyzés van fűzve.

1. Nyomjuk meg az **F5**-ös funkciógombot (Ugrás), kattintsunk az **Irányított** nevű nyomógombra.
2. A megjelenő ablakban (Irányított kijelölés) a választókapcsolók közül válasszuk a **Jegyzetek** nevűt.
3. Az Excel kijelöli az összes olyan cellát, amelyekben megjegyzés található, vándorolni ezen cellák között a **TAB** billentyűvel lehet.

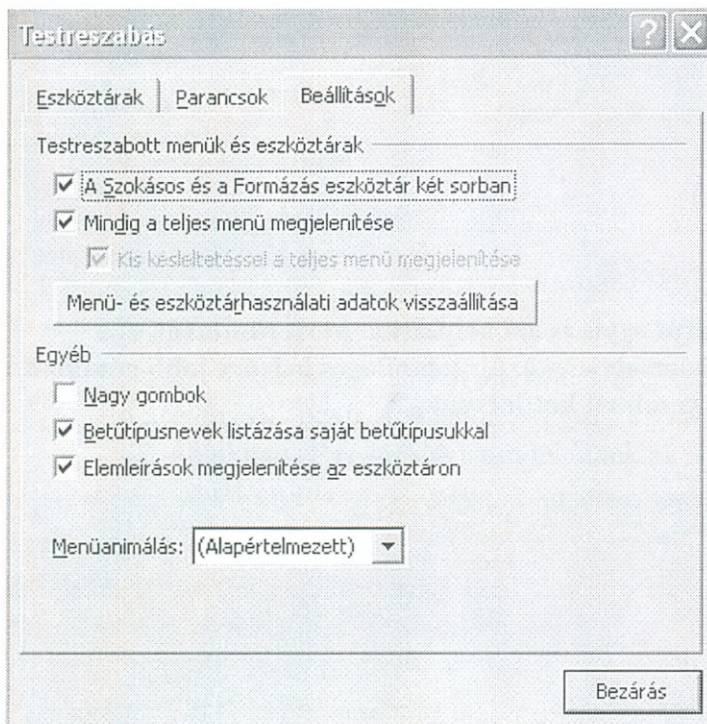
12. fejezet: Eszköztárak

Alapértelmezett eszköztárak: az Excel első indításakor megjelenik a **Szokásos** és a **Formázás** eszköztár.

Az eszköztárak egy része bizonyos parancsok végrehajtásakor automatikusan megjelenik (például: **Diagram**).

Eszköztár beállítások: a **Nézet** menü **Eszköztárak** parancsát, majd a **Testreszabás** parancsát kiadva egy ugyanolyan nevű összetett párbeszédablakot kapunk. Az **Eszköztárak** fülre való kattintás után megjelenő ablak feladata a megjelenítés szabályozása, az eszköztár nevek mellett elhelyezett jelölőnégyzetek segítségével. Ugyanezen ablakrész biztosítja a saját eszköztár válogatások létrehozását is.

A **Testreszabás** nevű összetett párbeszédablak **Beállítások** fülére kattintva, megszabhatjuk az eszköztárak megjelenésének módját. Az alábbi képen látható beállítások már nem alapbeállítások. Javasoljuk ezek használatát:



82. ábra Beállítások

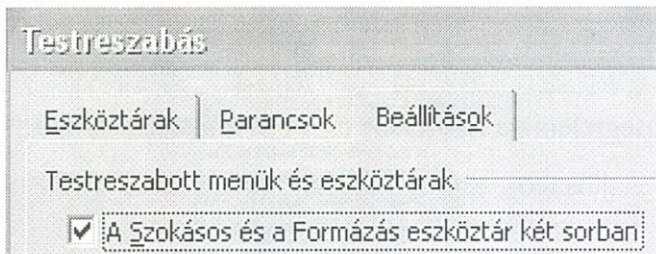
Megjegyzés: ezek a beállítások (82. ábra) biztosítják, hogy a két eszköztár egymás alatt jelenjen meg, illetve, hogy egy menüparancsra kattintva az összes parancs egyszerre jelenjen meg a legördülő listában.

Gyors eszköztár váltás, megjelenítés: ha bármelyik eszköztár tetszőleges helyére jobb gombbal kattintunk, egy legördülő listában megjelennek az eszköztárak nevei. Kattintva egy eszköztár nevére, az vagy megjelenik vagy eltűnik. Amelyik eszköztár neve előtt egy „pipa” található, az a képernyőn van.

Mintafeladat: Eszköztárak

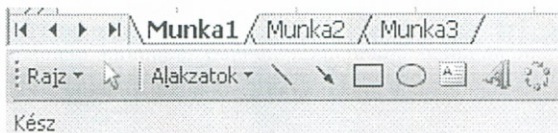
Feladat: Úgy állítsuk be az eszköztárakat, hogy fent a Szokásos és a Formázás nevű egymás alatt, a képernyő alján, pedig a Rajzolás eszköztár látszódjon.

1. Nyissunk meg egy új, üres állományt.
2. Jobb gombbal kattintsunk bármely eszköztár tetszőleges pontjára, a megjelenő rövidmenüből válasszuk a **Testreszabás** parancsot.
3. A **Beállítások** fülre kattintva megszabhatjuk az eszköztárak megjelenésének módját. A párbeszédablakot állítsuk be az alábbiak szerint:



83. ábra A két eszköztár két sorban fog megjelenni

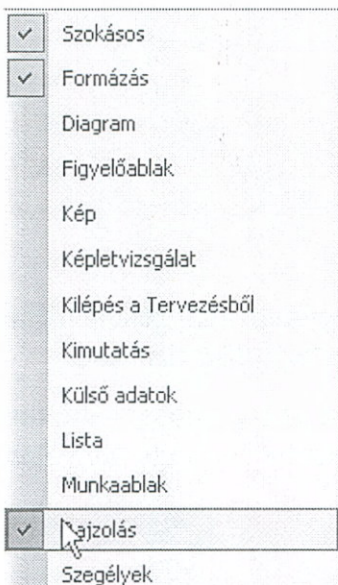
4. A fenti beállítással biztosítottuk, hogy a két eszköztár két sorba kerüljön.
5. Jobb gombbal kattintsunk bármely eszköztár bármely pontjára, a megjelenő rövidmenüben keressük ki a **Rajzolás** eszköztár nevét, ha “pipa” van előtte, akkor a képernyőn van és keressük meg, ha nincs előtte “pipa”, akkor kattintsunk rá.
6. A **Rajzolás** eszköztár megjelenik, ha még a telepítés óta nem jelenítettük meg, akkor előfordulhat, hogy lebegő pozícióban láthatjuk. Ekkor egérrel ragadjuk meg és húzzuk az Excel képernyő aljára, az állapotsor fölé, íme egy részlet a dokkolt **Rajzolás** eszköztárból:



84. ábra Részlet a Rajzolás eszköztárból

Feladat: a Rajzolás eszköztárat tüntessük el a képernyőről

Fenti ténykedésünkkel a Szokásos és a Formázás eszköztárat egymás alá helyeztük. Most tüntessük el a képernyőről a Rajzolás eszköztárat. Ehhez, kattintsunk bármely eszköztár tetszőleges helyére jobb gombbal, a megjelenő rövidmenüben (85. ábra) a Rajzolás szóra bal gombbal kattintsunk.



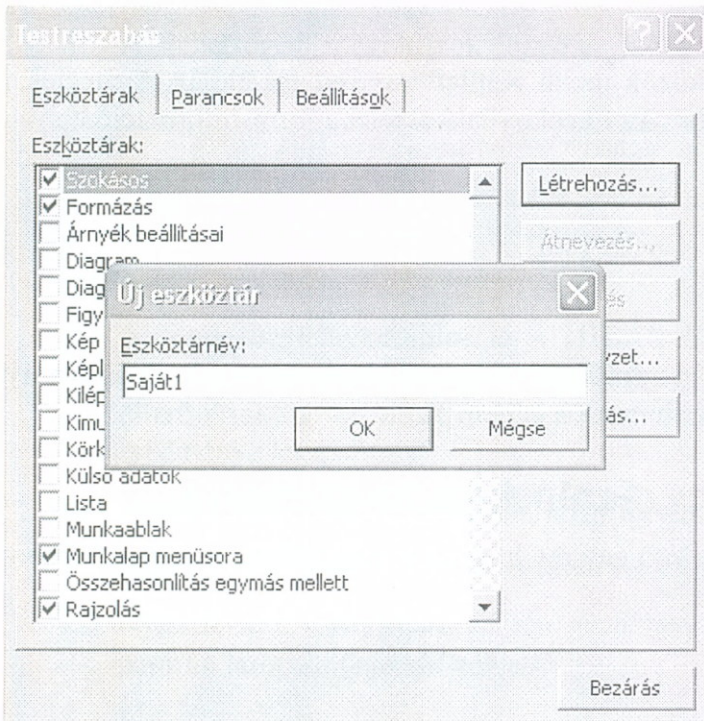
85. ábra Rövidmenü

Egyéni eszköztár válogatás

Elképzelhető, hogy az egyes eszköztárakban néhány hasznos ikonra találunk. Felmerül a kérdés, hogy nem lehetne-e egy eszköztárba integrálni azokat? Természetesen lehet.

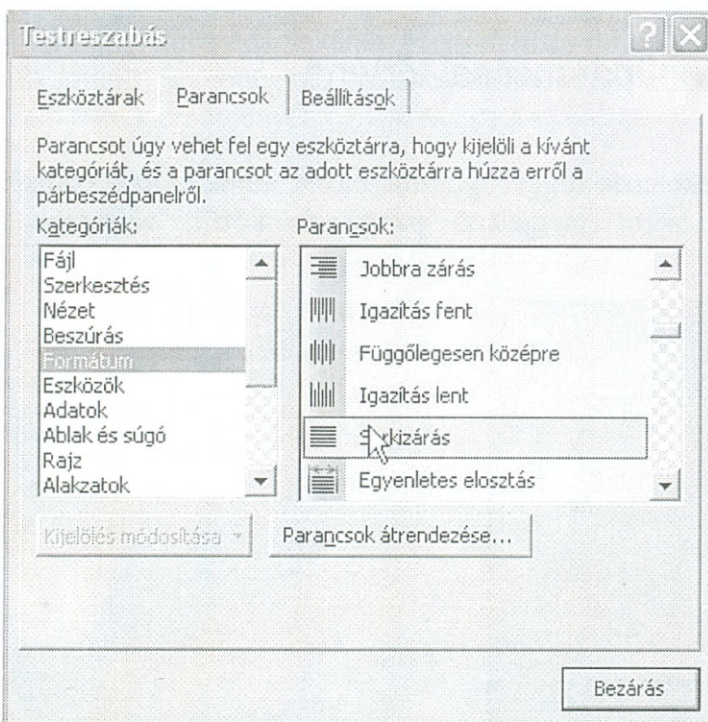
1. Adjuk ki a **Nézet** menü **Eszköztárak** parancsát, majd a **Testreszabás** parancsot. Kattintsunk az **Eszköztárak** szegélyre és a megjelenő párbeszédablak **Létrehozás** nyomógombjára.

2. A megjelenő **Új eszköztár** párbeszédablakba gépeljük be eszköztár válogatásunk nevét. Miután **OK**-t nyomtunk a **Testreszabás** párbeszédablak közvetlen közelében megjelenik saját kis eszköztárunk. Ebbe kell az egyes eszköztárakból egérrel az ikonokat áthúzni.



86. ábra Egyéni eszköztár létrehozása

3. Ehhez kattintsunk a **Parancsok** fülre, az ekkor megjelenő párbeszédablakban válasszuk ki a **Kategóriák** nevű listaablakból az eszköztár nevét (például: **Formátum**), ekkor a jobboldali listaablakban (**Parancsok**) megjelennek az ikonok.



87. ábra Sorkizárás ikon húzása

4. A nekünk szimpatikus ikonokat az egérrel megragadva, az egér bal gombját nyomva tartva, húzzuk át saját eszköztárunkba
5. Miután eszköztárunk elkészült, húzzuk azt a képernyő valamelyik szélső helyzetébe, például a **Rajzolás** eszköztár alá.

13. fejezet: Hibakeresés

Hibák behatárolása menüparanccsal

1. Miközben a hibás cella az aktív, adjuk ki **Eszközök** menü **Képletvizsgálat** parancsát. A megjelenő további parancsok közül, választuk a **Hibajelölést**. Az Excel nyilakkal jelzi a forrásinformáció helyét.

	B	C	D
6	5		#ERŐOSZTÓ!
7			

88. ábra Osztas nullával

2. **A hibajelölő nyilak eltávolítása:** az **Eszközök** menü **Képletvizsgálat** parancsát, majd a **Minden nyíl eltávolítása** parancsot kiadva a hibajelölő nyilak eltűnnek a képernyőről.

Hibák behatárolása a képlet (függvény) egy részének kiértékelésével

1. Amennyiben a hibásnak bizonyult képlet (függvény) egy értelmes részét kijelöltük, az **F9**-es billentyű megnyomásának hatására, az Excel azt kiértékeli.

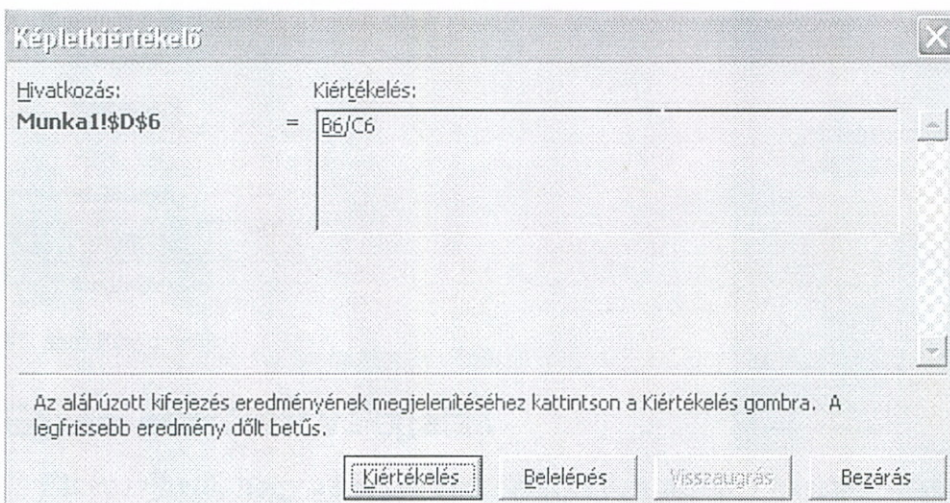
	B	C	D
6	5	0	=B6/D

89. ábra A nevezőben nulla található

2. Ne felejtjük el a kiértékelés végeztével az **ESC** billentyűt lenyomni. **ENTER** billentyűt ilyenkor véletlenül se nyomjunk, mert akkor a kiszámított értékkel az Excel felülírja a képlet (függvény) kijelölt részét.

Hibák behatárolása a képlet (függvény) menüparanccsal történő kiértékelésével

1. Miután aktívvá tettük azt a cellát, amely a kiértékelendő függvényt tartalmazza, adjuk ki az **Eszközök** menü **Képletvizsgálat** parancsát, majd az ekkor megjelenő parancsok közül, választuk a **Képletkiértékelő** parancsot.



90. ábra Képletkiértékelő

2. A párbeszédablak kezelőszerveit értelemszerűen (alapvetően a **Kiértékelést**) használva a függvényt lépésenként kiértékelhetjük.

Mit tehetünk, hogy ne lássuk a hibajelzést?

Mit tudnánk tenni annak érdekében, hogy amíg hibát hoz a egy képlet (például a D6-os cellában), addig ne lehessen látni a hibajelzést?

1. A lehetséges megoldások egyike, ha a D6-os cellába az alábbi függvénykombinációt írjuk:
`=HA(HIBÁS(B6/C6);"";B6/C6)`

	B	C	D	E	F
6	5	0			
7					

91. ábra Hibajelzés "eltüntetése"

2. A fenti függvénykombináció a **HIBÁS** függvény segítségével érzékeli, ha hibát okoz az osztás művelet, viszont ekkor a **HA** függvény eredménye üres cella. Azonban, ha a C6-os cellába nullától eltérő értékű szám kerül, a **HA** függvény nem teljesült ága hajtódik végre és az eredmény a osztás eredménye lesz.

Mit tudunk tenni annak érdekében, hogy nyomtatásánál ne lehessen látni a hibajelzést?

1. Adjuk ki a **Fájl** menü **Oldalbeállítás** parancsát.
2. Nyissuk le a **Cellahibák esetén** nevű lenyíló és válasszuk ki, hogy milyen karaktert szeretnénk a papíron látni a hibajelzés helyett.

Milyen eszközzel kereshetünk még hibát?

A hibakeresést támogatja még az **Irányított kijelölés** (F5) parancs is. A parancsról a későbbiekben még szólnunk.

A hibakeresésben szerepe lehet még egy beállításnak is. Az **Eszközők** menü **Beállítások** parancsának összetett párbeszédablakának **Megjelenítés** fülére kattintva lehetőségünk van a **Képletek** nevű (alaphelyzetben kikapcsolt) jelölőnégyzet bekapcsolására, ekkor az összes olyan cellában, amelyben képlet vagy függvény van, nem annak eredményét, hanem magát a számítási műveletet látjuk.

14. fejezet: Űrlapkezelés

Igen látványos munkákat készíthetünk, ha munkalapjainkra controlokat rajzolunk, majd azokat függvényekkel (ez nem programozás, bár ezen controlok programozhatók is) kiértékeljük. A következőkben egy feladatsor elvégzése révén kívánjuk bemutatni az űrlapok kezelését.

Mintafeladat: Jelölőnégyzet

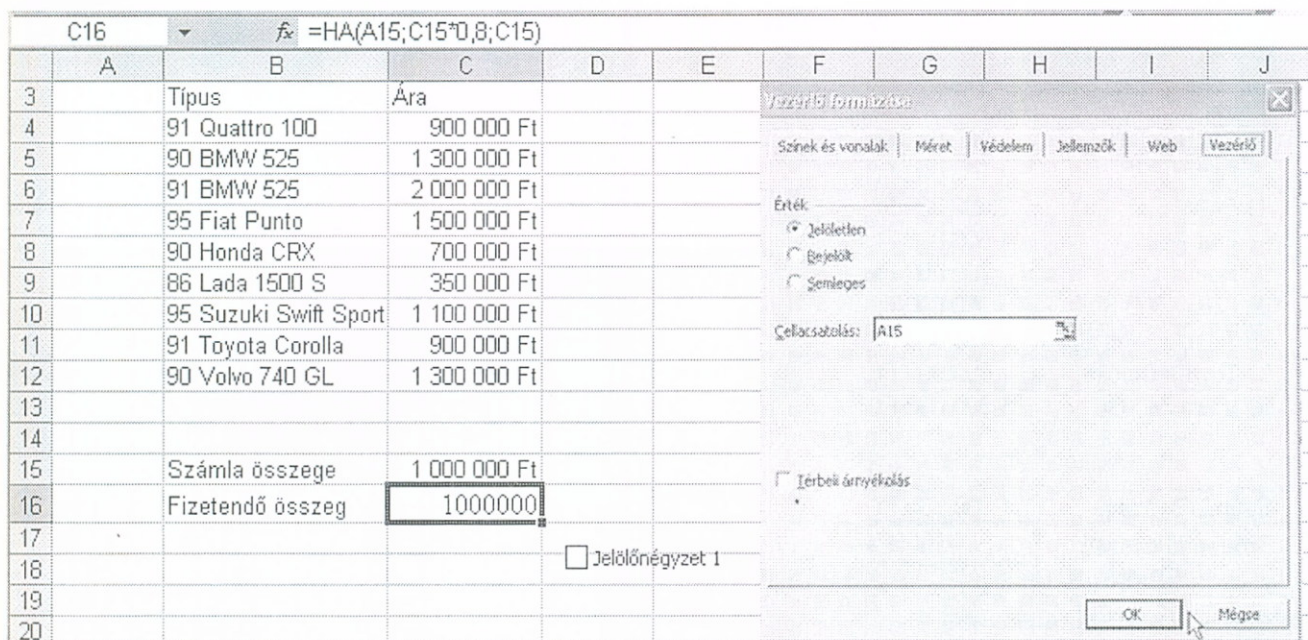
Feladatunk legyen az, hogy egy Jelölőnégyzet be és ki kapcsolásával egy árból kedvezményt adjunk (bekapcsolt a jelölőnégyzet) vagy ne adjunk (ha kikapcsolt a jelölőnégyzet). A kedvezmény mértéke legyen 20%.

1. Nyissuk meg az **urlap.xls** állományt.

	B	C
3	Típus	Ára
4	91 Quattro 100	900 000 Ft
5	90 BMW 525	1 300 000 Ft
6	91 BMW 525	2 000 000 Ft
7	95 Fiat Punto	1 500 000 Ft
8	90 Honda CRX	700 000 Ft
9	86 Lada 1500 S	350 000 Ft
10	95 Suzuki Swift Sport	1 100 000 Ft
11	91 Toyota Corolla	900 000 Ft
12	90 Volvo 740 GL	1 300 000 Ft
13		
14		
15	Számla összege	1 000 000 Ft
16	Fizetendő összeg	

92. ábra urlap.xls

2. Jelenítsük meg az **Űrlapok** nevű eszköztárat, a **Jelölőnégyzet** ikonra kattintva rajzoljunk egy jelölőnégyzetet.
3. A megrajzolt jelölőnégyzet keretére kattintsunk duplán.
4. A megjelenő **Vezérlő formázása** összetett párbeszédablak **Vezérlő** szegélyére kattintva (hacsak nem ez az aktív) állítsuk be annak állapotát (**Érték** keret), majd adjuk meg a **Cellacsatolás** lenyílóban az **A15**-ös cella címét. E cella ezek után mutatja a jelölőnégyzet állapotát, ha az bekapcsolt, akkor a logikai **IGAZ**, ha kikapcsolt, akkor a logikai **HAMIS** állítás látható abban.
5. Olyan függvényre van szükségünk, amely képes ezt a két értéket megvizsgálni és ezek függvényében két különböző eredményt előállítani. E célra igencsak alkalmas a **HA** függvény.



93. ábra Jelölőnégyzet

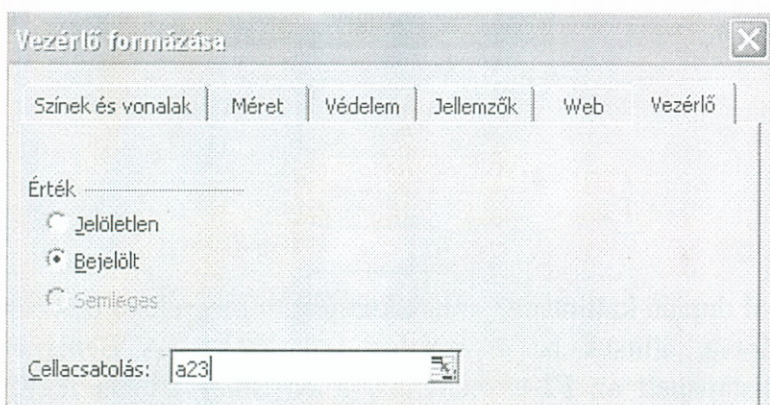
6. A C16-os cellába gépeljük be a következő függvényt: **=ha(A15;C15*0,8;C15)**

Megjegyzés: a továbbiakban, mint az a függvényből is látható a jelölőnégyzettel “vezéreljük” a C16-os cella értékét. Amennyiben a jelölőnégyzet bekapcsolt, akkor a **HA** függvény második paramétere (C15*0,8) az eredmény.

Mintafeladat: Választókapcsoló

Feladatunk háromféle kedvezmény nyújtásának vezérlése választókapcsolókkal (urlap.xls, 92. ábra).

1. Az **Űrlapok** nevű eszköztáron kattintsunk duplán a **Választókapcsoló** ikonra, majd rajzoljunk egymás alá négy választókapcsolót. Miután megrajzoltuk a négy választókapcsolót, nyomjunk ESC-t, a rajzolásból való kilépés céljából.
2. Tervünk az, hogy a négy választókapcsoló segítségével különböző mértékű kedvezményeket adunk a gépkocsi árából.
3. Kattintsunk jobb gombbal az elsőként rajzolt választókapcsolóra, a megjelenő keretre, pedig duplán ballal. Az összetett párbeszédablakban állítsuk be a választókapcsoló vezérlését, állapota legyen bejelölt, csatoljuk azt az **A23**-as cellához.



94. ábra A választókapcsoló vezérlése

4. Próbáljuk ki, hogy nem hibáztunk-e. Egyesével kattintsunk a választókapcsolókra, mindig győződjünk meg az A23-as cella értékéről, abban rendre 1, 2, 3, 4 kell, hogy látszódjon.
5. Az alábbi függvényt gépeljük be a C16-os cellába: **=VÁLASZT(A23;C15;C15*0,95;C15*0,9;C15*0,8)**

C16		fx =VÁLASZT(A23;C15;C15*0,95;C15*0,9;C15*0,8)				
	A	B	C	D	E	F
3		Típus	Ára			
4		91 Quattro 100	900 000 Ft		<input type="radio"/>	Választókapcsoló 3
5		90 BMW 525	1 300 000 Ft			
6		91 BMW 525	2 000 000 Ft			
7		95 Fiat Punto	1 500 000 Ft			
8		90 Honda CRX	700 000 Ft		<input checked="" type="radio"/>	Választókapcsoló 4
9		86 Lada 1500 S	350 000 Ft			
10		95 Suzuki Swift Sport	1 100 000 Ft			
11		91 Toyota Corolla	900 000 Ft		<input type="radio"/>	Választókapcsoló 5
12		90 Volvo 740 GL	1 300 000 Ft			
13						
14						
15		Számla összege	1 000 000 Ft		<input type="radio"/>	Választókapcsoló 6
16		Fizetendő összeg	950000			
17						

95. ábra A függvényt a C16-os cella tartalmazza

- A feladatot ezzel megoldottuk, ahhoz, hogy igazán mutatós legyen a megoldás érdemes lecserélni a választókapcsolók címkéjét, a legfelsőét például a következőre: 0%.
- Kattintsunk a legelső választókapcsolóra jobb gombbal, de ne parancsot válasszunk, hanem kattintsunk egérrel a keretbe, a kurzor máris a szöveg előtt villog. A címke szövegének átszerkesztése után kattintsunk egy cellára.

Mintafeladat: Listapanel

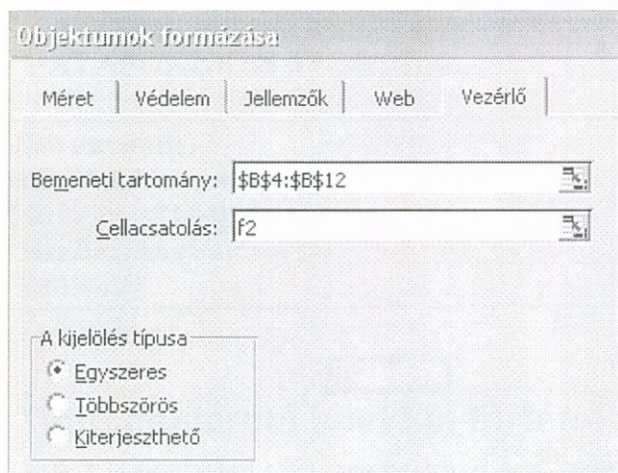
Feladat: feladatunk egy listából történő választás, egy kiválasztott autónak a típusát és az árát egy-egy cellában meg kell jeleníteni.

- Továbbra is az **urlap.xls** állományban dolgozzunk:

	B	C
3	Típus	Ára
4	91 Quattro 100	900 000 Ft
5	90 BMW 525	1 300 000 Ft
6	91 BMW 525	2 000 000 Ft
7	95 Fiat Punto	1 500 000 Ft
8	90 Honda CRX	700 000 Ft
9	86 Lada 1500 S	350 000 Ft
10	95 Suzuki Swift Sport	1 100 000 Ft
11	91 Toyota Corolla	900 000 Ft
12	90 Volvo 740 GL	1 300 000 Ft
13		
14		
15	Számla összege	1 000 000 Ft
16	Fizetendő összeg	

96. ábra urlap.xls

- Rajzoljunk egy listapanelt, majd bal gombbal duplán kattintsunk annak keretére, a megjelenő összetett párbeszédablak **Vezérlő** nevű fülére kattintva, állítsuk be a vezérlési jellemzőket. A **Bemeneti tartomány** legyen **B4:B12**. Csatoljuk a listapanelt az **F2**-es cellához, a **Kijelölés típusa** legyen egyszeres.



97. ábra A Listapanel vezérlése

3. Az OK gombot megnyomva, majd kattintva egy cellára, a listapanel működőképesé válik. Annak bizonyítására, hogy nem dolgoztunk feleslegesen, kattintsunk a lista bármelyik elemére. Ekkor a csatolt cellában (F2) a kiválasztott elem sorszámát kell látnunk (a legelső elem sorszáma az 1-es).
4. Nincs is más feladatunk, mint egy-egy olyan függvény kidolgozása, amelyek képesek detektálni a kiválasztott sort és ennek hatására kijelezni két egymás melletti cellában a kiválasztott gépkocsi típusát és árát. E feladat megoldása céljából gépeljük be az alábbiakat:

	B	C
13	Típus	Ara
14	=OFSZET(\$B\$3;\$F\$2;0)	=OFSZET(\$B\$3;\$F\$2;1)

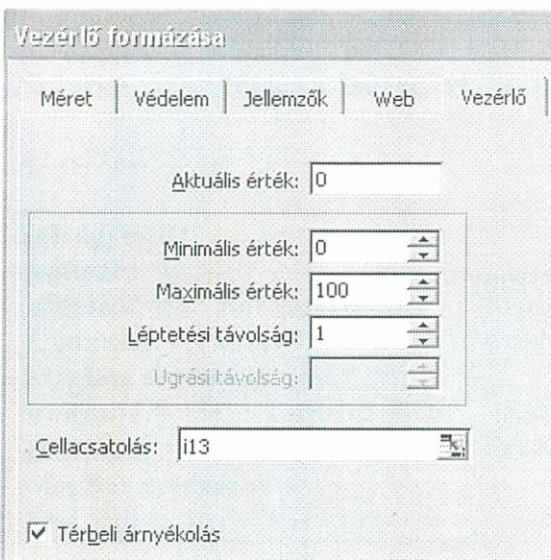
98. ábra Ofszet függvény

Ezzel a listapannellel kapcsolatos feladatunkat megoldottuk, de nem értünk a fontosabb controlok kipróbálását célul tűző feladatsorunk végére, most próbáljuk ki a Léptetőt.

Mintafeladat: Léptető

Feladat: rajzoljunk egy léptetőt (urlap.xls, 96. ábra), célunk egy cella értékének százalékos megváltoztatása, ezzel a technikával adjunk egy gépkocsi árából tetszőleges kedvezményt.

1. Állítsuk be a vezérlési jellemzőket, a **Minimális érték** legyen 0, a **Maximális érték** legyen 100, a **Léptetési távolság** legyen 1 (növekedés / csökkenés mértéke), a csatolás történjen az **I13**-as cellához.



99. ábra A léptető vezérlése

2. Végezetül a C16-os cellába (mivel százalék értékeket kívánunk megadni) a következő képletet gépeljük: **=C14*(1-I13/100)**

	B	C	D	E	F	G	H	I
3	Típus	Ára						
4	91 Quattro 100	900 000 Ft		91 Quattro 100				
5	90 BMW 525	1 300 000 Ft		90 BMW 525				
6	91 BMW 525	2 000 000 Ft		91 BMW 525				
7	95 Fiat Punto	1 500 000 Ft		95 Fiat Punto				
8	90 Honda CRX	700 000 Ft		90 Honda CRX				
9	86 Lada 1500 S	350 000 Ft		86 Lada 1500 S				
10	95 Suzuki Swift Sport	1 100 000 Ft		95 Suzuki Swift Sport				
11	91 Toyota Corolla	900 000 Ft		91 Toyota Corolla				
12	90 Volvo 740 GL	1 300 000 Ft		90 Volvo 740 GL				
13	Típus	Ára						50
14	91 BMW 525	2000000						
15	Számla összege	1 000 000 Ft						
16	Fizetendő összeg	1000000						
17								
18								
19								
20								

100. ábra Az elkészült léptető

Ezzel megoldottuk a gépkocsi árának százalékos megváltoztatását a Léptető nevű vezérlő elem segítségével. Ugyanezt a feladatot oldjuk meg a Görgetősáv nevű vezérlő elemmel is.

Mintafeladat: Görgetősáv

Rajzoljunk egy Görgetősávot, célunk egy cella (urlap.xls, 96. ábra) értékének százalékos megváltoztatása.

1. Állítsuk be a vezérlési jellemzőket, ez nem nehéz, mert többségük egyezik a számskáláéval. Az **Ugrási távolság** értékét állítsuk 10-re (akkor történik ugrás, ha a csúszka alá vagy fölé kattintunk), A csatolás történjen a **J7**-es cellához.

Vezérlő formázása

Méret | Védelem | Jellemzők | Web | Vezérlő

Aktuális érték: 0

Minimális érték: 0

Maximális érték: 100

Léptetési távolság: 1

Ugrási távolság: 10

Cellacsatolás: J7

101. ábra A görgetősáv vezérlése

2. Gépeljük be az alábbi képletet a C16-os cellába: **=C14*(1-J7/100)**.

C16		fx =C14*(1-J7/100)							
	B	C	D	E	F	G	H	I	J
3	Típus	Ára							
4	91 Quattro 100	900 000 Ft		91 Quattro 100					
5	90 BMW 525	1 300 000 Ft		90 BMW 525					
6	91 BMW 525	2 000 000 Ft		91 BMW 525					
7	95 Fiat Punto	1 500 000 Ft		95 Fiat Punto					50
8	90 Honda CRX	700 000 Ft		90 Honda CRX					
9	86 Lada 1500 S	350 000 Ft		86 Lada 1500 S					
10	95 Suzuki Swift Sport	1 100 000 Ft		95 Suzuki Swift Sport					
11	91 Toyota Corolla	900 000 Ft		91 Toyota Corolla					
12	90 Volvo 740 GL	1 300 000 Ft		90 Volvo 740 GL					
13	Típus	Ára						50	
14	91 BMW 525	2000000							
15	Számla összege	1 000 000 Ft							
16	Fizetendő összeg	1000000							
17									
18									
19									

102. ábra Görgetősáv

15. fejezet: Irányított kijelölés

Az Excel **Írányított kijelölés** parancsának segítségével lehetőségünk nyílik speciális tartalmú cellák kijelölésére.

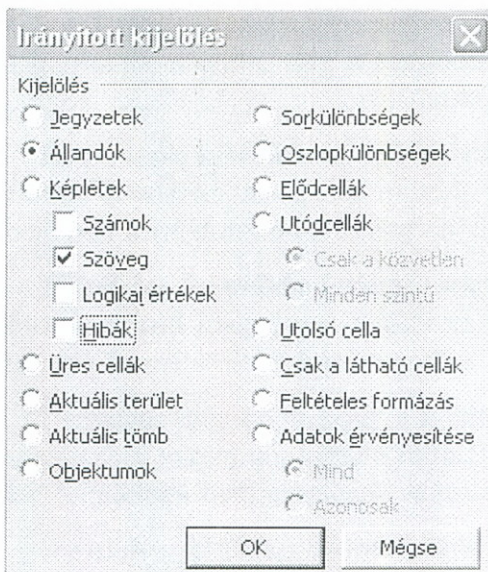
Mintafeladat: Irányított kijelölés

Feladat:szövegtartalmú cella (cellák) megkeresése egy tartományban. A gyakorlatban egy valós probléma (ami az alábbi ábrán is látszik) az, hogy egy oszloptartományban, melyben számolni akarunk, egy vagy több cellában elgépelés miatt szöveg van. Ilyenkor elsőrendű feladatunk az összes ilyen cella megkeresése.

	A
1	1
2	2
3	ö
4	4
5	ö
6	6
7	7

103. ábra A vizsgálandó tartomány

1. Jelöljük ki a tartományt és nyomjunk **F5**-öt.
2. A megjelenő párbeszédablakban kattintsunk az **Írányított** nyomógombra, ekkor megkapjuk az **Írányított kijelölés** párbeszédablakot. Állítsuk azt úgy be, mint az alábbi képen látható, majd nyomjunk **OK**-t.



104. ábra Szöveget keresünk

3. **Az Irányított kijelölés eredménye:** mint az alábbi képen látszik, az Excel kijelöli a szövegtartalmú cellá(ka)t. A kijelölt cellák bejárhatók a **TAB** billentyű nyomogatásával.

	A
1	1
2	2
3	ö
4	4
5	ö
6	6
7	7

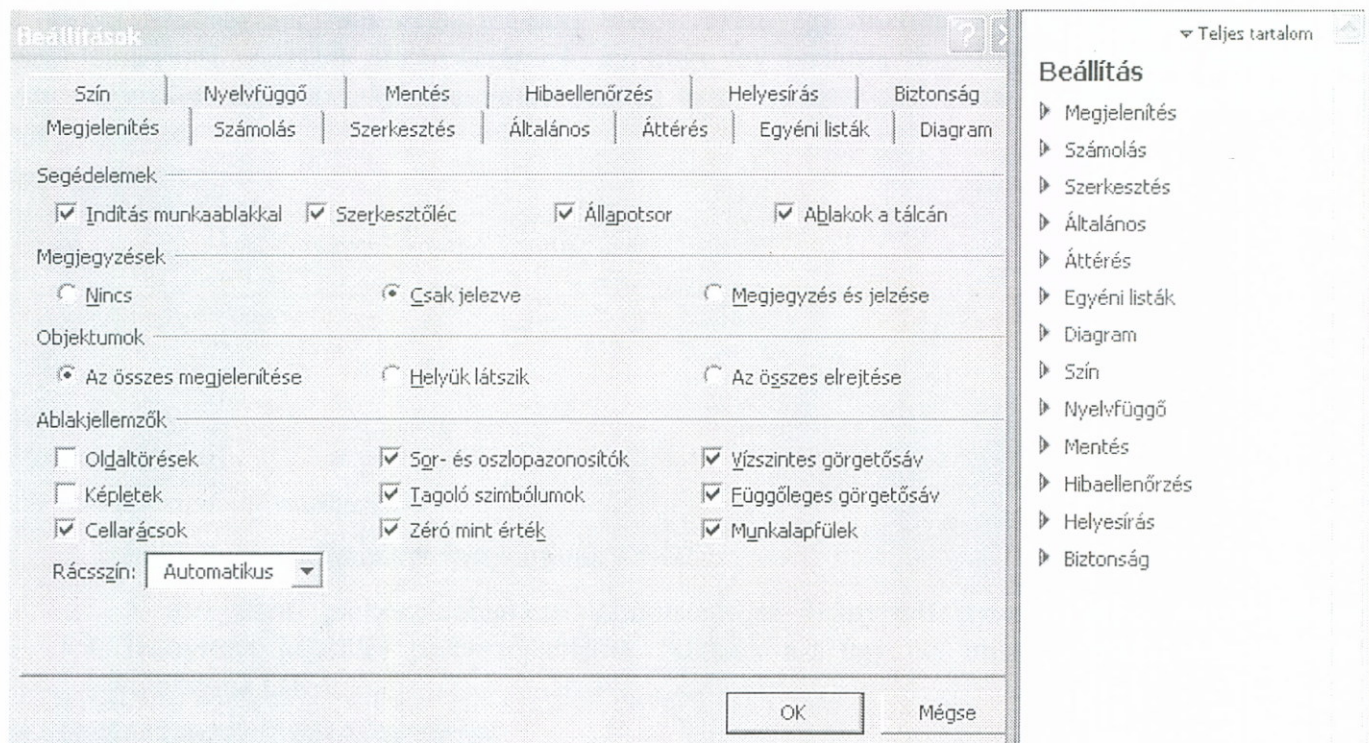
105. ábra A szöveg tartalmú cellák kijelöltek

Hibás tartalmú celláink megkeresése

Nagy méretű táblázatokkal való munkavégzésnél fenn áll a veszély, hogy egy hibás ténykedésünk következtében egy vagy több képletünk (függvényünk) hibát jelez. Az ilyen cellák is kijelölhetők az **Irányított kijelöléssel**. Ehhez a **Képletek** választókapcsolót kell választani és a jelölőnégyzetek közül csak a **Hibák** feliratút bekapcsolni.

16. fejezet: Beállítások

Az Excel alapbeállításaihoz az **Eszközők** menü **Beállítások** parancsát kiadva férhetünk hozzá. A parancsot kiadva egy összetett párbeszédablakhoz jutunk. **Vigyázat:** a beállítások egy részének megváltoztatása igencsak veszélyes. Ha a **Beállítások** összetett párbeszédablak kérdőjel alakú ikonjára kattintsunk, a **Súgó** megnyílik, és kattintva a megjelenő linkekre, az összetett párbeszédablak egy-egy párbeszédablakáról kaphatunk információkat.



106. ábra A Beállítások súgója (jobb oldalon)

17. fejezet: Adatbázis-kezelés az Excelben

Alapfogalmak

Adatbázis: egy speciális felépítésű táblázat, mivel a soroknak nem lehetnek címkéik, az oszlopok címkéit mező-neveknek hívjuk.

Az oszlopok adatbázisban: mezők.

A sorok adatbázisban: rekordok.

Az adatbázis-tábla első sorban a mezőnevek találhatóak.

A műveletek a rekordokon történnek.

Kritériumok: vizsgálati feltételek, amelyek teljesülése esetén, a rekordokon adatbázis műveletek hajtódhatnak végre.

Fontosabb adatbázis parancsok

Szűrés parancs: az adatbázisnak csak azon rekordjait jeleníti meg a képernyőn, amelyek kritériumainknak megfelelnek.

Sorba rendezés parancs: rekordjainkat általunk adott kritériumok alapján, sorba rendezve jeleníti meg.

Űrlap parancs: az adatbázis egy-egy rekordját, jól olvasható, űrlapszerű formában jelzi ki.

Részösszegek parancs: az adatbázis rekordcsoportjai alapján csoportonként részösszegeket képez.

Érvényesítés parancs: amennyiben nagyszámú adatot kívánunk bevinni egy oszlopba és a beviendő adatok köre jól meghatározott, akkor ez a parancs képes jelezni a hibás bevittet.

Automatikus Tagolás parancs: ha adataink adatbázisba szervezettek, az Excel egyetlen egérgattintásra tagolást végez, amellyel elrejtethetjük vagy megjeleníthetjük a különböző szinteket.

Függvények az adatbázisban: adatbázis-kezeléshez az Excel saját függvénykészlettel rendelkezik, a függvények neve AB.-tal kezdődik.

Sorba rendezés parancs

Feladata: az adatbázis rekordjait általunk adott (egyidejűleg, maximum három) feltétel alapján sorba rendezzi. Amennyiben háromnál több feltételünk van, a parancsot többször kell végrehajtani, vigyázva, hogy végére maradjon a három legfontosabb feltétel.

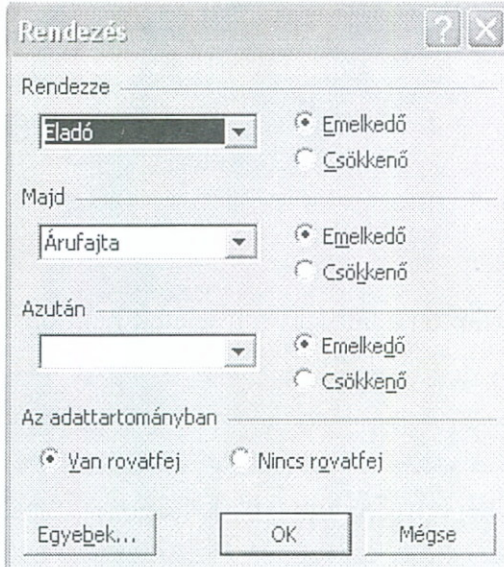
A sorba rendezés menete

1. Kattintsunk az adatbázis (**adatbaz.xls**, **Termékek** munkalap) valamelyik mezőjére (nem kell tartományt adni). Adjuk ki az **Adatok** menü **Sorba rendezés** parancsát.

	A	B	C	D
1	Eladó	Árufajta	Egység	Forgalom
2	Dorozsmai	Készétel	39	7 191 Ft
3	Barta	Készétel	744	2 666 Ft
4	Barta	Készétel	983	8 165 Ft
5	Barta	Készétel	1242	2 956 Ft
6	Dorozsmai	Tejtermék	1695	3 338 Ft
7	Barta	Tejtermék	1773	8 751 Ft
8	Dorozsmai	Készétel	1824	1 361 Ft

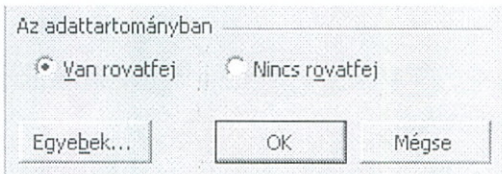
107. ábra Részlet egy adatbázisból

- Most adjuk meg a sorba rendezés feltételeit a három lenyílóból történő mezőnév választással. Az elsődleges (legfontosabb) rendezési feltételt a legfelső lenyílóban kell megadni, a legkevésbé fontosat az alsóban.
- A feladattól függ, hogy beállítjuk-e mind a három lenyílot. Ha például két sorba rendezési feltétel van, akkor csak a felső két lenyílot kell beállítani.



108. ábra Két rendezési kulcsot adtunk meg

- ✦ **A rendezés iránya:** A rendezés irányát a lenyílok melletti választókapcsoló párossal lehet meghatározni. Az emelkedő irányú rendezés betűk esetén A-tól Z-ig, míg számok esetén, kis számoktól a nagyok irányába történő rendezést jelent.
- ✦ **A párbeszédablak alsó része:** Az **adattartományban** nevű keret választókapcsoló párja jelzi, hogy az Excel talált-e az adatbázis-táblázat első sorában mezőneveket (rovatfej) vagy sem.



109. ábra Az adattartományban nevű keret választókapcsolói

- Miután a szükséges beállításokat megtettük, a **Rendezés** párbeszédablakban kattintsunk az **OK** nyomógombra és a rendezés (amely a rekordok sorrendjének megváltozását vonhatja maga után) megtörténik.

	A	B	C	D
1	Eladó	Árufajta	Egység	Forgalom
2	Barta	Készétel	744	2 666 Ft
3	Barta	Készétel	983	8 165 Ft
4	Barta	Készétel	1242	2 956 Ft
5	Barta	Készétel	2891	8 670 Ft
6	Barta	Készétel	3219	3 106 Ft
7	Barta	Készétel	3868	797 Ft
8	Barta	Készétel	3981	2 428 Ft

110. ábra A sorba rendezett rekordok (részlet)

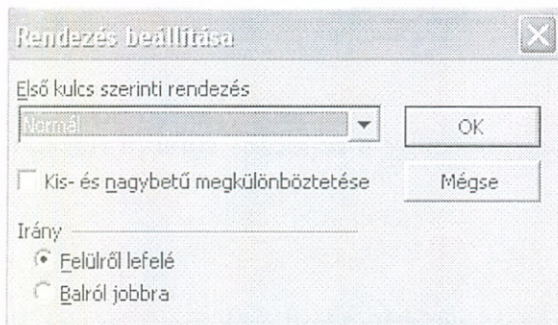
Speciális rendezési technikák

A gyakorlatban sűrűn felmerülő feladat, hogy a rendezést ne az ABC vagy számok alapján, hanem sorozatok elemei alapján hajtsuk végre. Az alábbi ábra egy ilyen esetet mutat. Itt a feladat a rekordok sorba rendezése a **Hónap** mező alapján. Ekkor nyilvánvalóan értelmetlen egy ABC szerinti rendezés.

	B	C
3	Hónap	Adatok
4	január	1000
5	február	2000
6	március	3000
7	január	4000
8	február	5000
9	március	6000

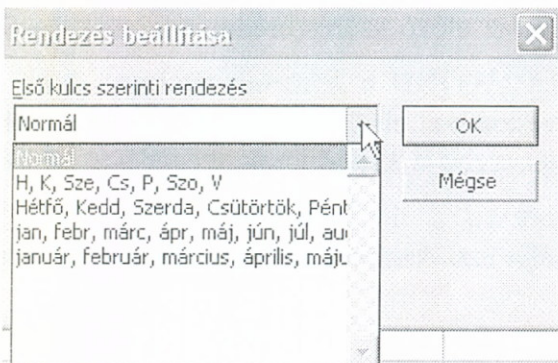
111. ábra Egy speciális adatbázis-tábla (részlet)

A **rendezés speciális beállításai**: a **Rendezés** párbeszédablak **Egyebek** gombja az, amelyre kattintva a **Rendezés beállítása** ablakot megkapjuk. Mint látható az ablak kezelőszerveinek révén megváltozathatjuk a rendezés irányát, megkülönböztethetjük a kis és nagybetűket, de talán a legfontosabb az **Első kulcs szerinti rendezés** lenyíló.



112. ábra Első kulcs szerinti rendezés

1. A lenyílót (112. ábra) lenyitva a rendezés alapjául bármelyik sorozatot választhatjuk, akár egy sajátot is.



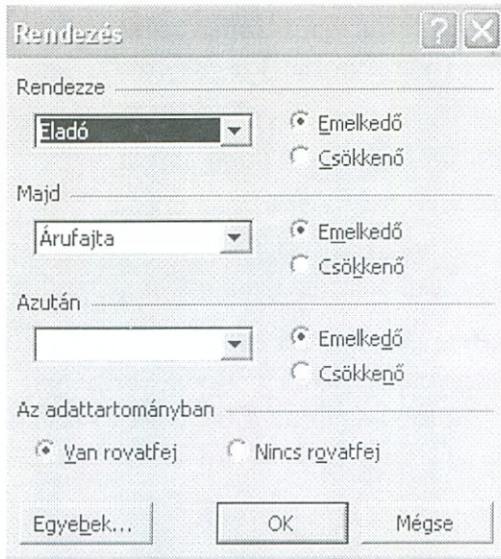
113. ábra Választhatunk rendezési szempontot

2. A mintául vett adatbázis-tábla **Hónap** mezeje alapján történő lekérdezéshez a fenti képen legördült lista 5. sorát kell választani (hónapok teljes neve).

Gyakorlat: Sorba rendezés

Feladat: az **adatbaz.xls** állomány **Termékek** nevű munkalapján lévő adatbázis-tábla rekordjait rendezzük sorba, elsődlegesen a **Eladó** mező alapján, a második rendezési kulcs az **Árufajta** legyen. A rendezés történjen emelkedő irányban.

1. Kattintsunk a (**adatbaz.xls**) **Termékek** nevű munkalap valamelyik mezőjére (nem kell tartományt adni). Adjuk ki az **Adatok** menü **Sorba rendezés** parancsát.



114. ábra A Rendezés beállításai

2. Az **OK** gombra kattintva végezzük el a rendezést, vegyük észre, hogy a rendezés során a rekordok helyet cserélnek.

Szűrés

Igen sűrűn előfordul, hogy az adatbázis adott szempontoknak megfelelő rekordjait kell csak megjeleníteni. Ez a szűrés műveletével valósulhat meg. Két művelet áll rendelkezésünkre szűrésre az Excelben, az egyik az **AutoSzűrő**, a másik az **Irányított szűrő** parancs.

AutoSzűrő: az adatokat helyben szűri (ez néha hátrány), a szűrési feltételeket a mezőnevek mellett megjelenő listanyilakra kattintva adhatjuk meg. Egy újabb szűrésnél, az előző szűrés eredményei elvesznek, hacsak ezen szűrés előtt az eredményrekordokat nem másoltuk ki.

Irányított szűrő: az adatokat képes más helyre szűrni, így nem vesznek el előző szűréseink. Bonyolultabb feltételeket is megadhatunk, mint az AutoSzűrőnél. Ugyanakkor nehezebb a kritériumok megadása, azokat nekünk kell gépeléssel (másolással) megadni.

Meg kell-e adni a tartományt szűrésnél? Nem, az Excel detektálja azt. Azonban a fénykurzor nem állhat tetszőleges helyen, hanem csak valamelyik adatbázis mezőben.

AutoSzűrő

A szűrés menete (egyszerű feltételek esetén)

1. Adjuk ki az **Adatok** menü **Szűrő** parancsát, majd az **AutoSzűrő** parancsot.
2. A mező nevek mellett megjelennek a listanyilak, melyek segítségével a szűrés feltételei megadhatóak.
3. Kattintva egy listanyílra, legördülnek egy listaablakban a szűrési feltételek és parancsok. A feltételek listáját a mezőértékek (de az azonosak csak egyszer) alkotják.

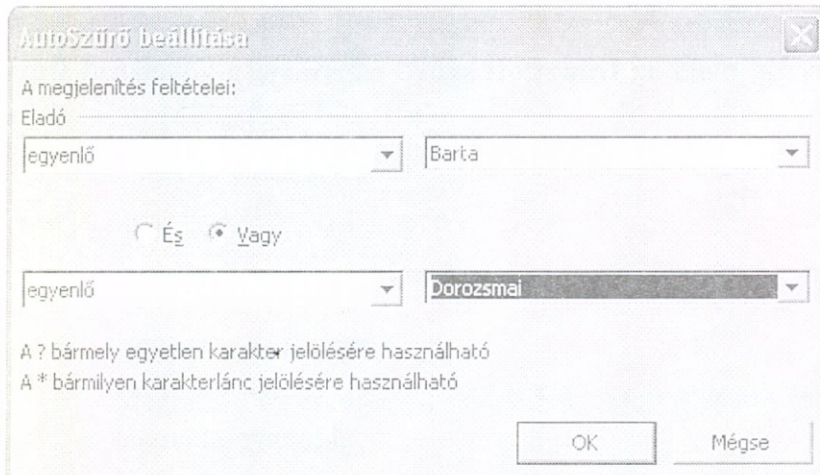
	A	B	C	D
1	Eladó	Árufajta	Egység	Forgalom
	Emelkedő sorrend	Készétel	5408	29 Ft
	Csökkenő sorrend	Éjtermék	6740	66 Ft
	(mind)	Éjtermék	3216	75 Ft
	(Helyezés...)	Készétel	9265	450 Ft
	(Egyéni...)	Éjtermék	5580	479 Ft
	Barta	Készétel	3448	668 Ft
	Dorozsmai	Készétel	3868	797 Ft
8	Barta	Készétel	3868	797 Ft

115. ábra Egy oszlopos szűrés

A szűrés menete (összetettebb feltételek esetén)

Összetettebb kritériumok: a szűréshez megadhatunk egy időben két szűrési feltételt is, melyeket egymással logikai kapcsolatba hozhatunk.

1. Kiadva az **Adatok** menü **Szűrő** parancsát, majd az **AutoSzűrő** parancsot, a legördülő parancslistából válasszuk az **Egyénit**.
2. Miután a lenyílókból kiválasztottuk a szűrési feltételeket (következő ábra) és eldöntöttük a két feltétel közötti logikai kapcsolatot, kattintsunk az OK gombra, és a szűrés megtörténik.



116. ábra Összetett szűrési feltétel

Helyettesítő karakterek: a helyettesítő karakterek hasonlóak, mint a Dos-ban használtak. A * karakter helyettesíti azt a karaktert, amelynek pozícióján áll és tőle jobbra az összes többi. A ? karakter, az aktuális karakterpozícióban helyettesít.

Egy speciális parancs, a Helyezés: előfordulhat, különösen a statisztikai feladatoknál, hogy adott mezőt vizsgálva ki szeretnénk szűrni, például a 10 legnagyobb értékű rekordot. Ezen esetekre megoldás lehet az AutoSzűrő parancs listájában a **Helyezés** parancs választása.

Kilépés az AutoSzűrőből: a parancs kikapcsolható az **Adatok** menü **Szűrő** parancsának, majd az **AutoSzűrő** parancsnak az újbóli kiadásával.

Irányított szűrő

Mikor használunk az AutoSzűrő helyett Irányított szűrőt? Ha a feltételek száma több mint kettő vagy ha a szűrést nem helyben szeretnénk végrehajtani, akkor Irányított szűrőt használunk.

Első lépés a szűrőfeltétel megadása: szűrőfeltételeinket magunknak kell a cellákban létrehozni. A szűrőfeltételek első sorában helyezkednek el a mező-nevek, míg második sorában (esetleg az azok alatti sorokban is) a feltételek.

Egymás alatti szűrőfeltételek: az alábbi képen látható szűrőfeltétellel kiszűrhetjük az adatbázisból azon rekordokat, ahol az **Eladó** mezőben **Dorozsmai**, míg a **Terület** mezőben **Észak**, **Nyugat** vagy **Dél** található. **Megjegyzendő:** az egymás alatti feltételek egymással **VAGY** kapcsolatban vannak.

Eladó	Terület
Dorozsmai	Észak
Dorozsmai	Nyugat
Dorozsmai	Dél

117. ábra Egymás alatti feltételek

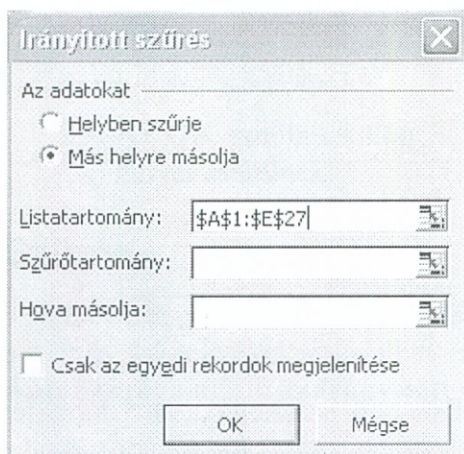
Egymás melletti szűrőfeltételek: az alábbi képen látható szűrőfeltétel kiszűri az adatbázisból azon rekordokat, amelyekben a **Forgalom** mező értéke **5000-nél** nagyobb, ugyanakkor **7000-nél** kisebb. **Megjegyzendő:** az egymás melletti feltételek egymással **ÉS** kapcsolatban vannak.

	J	K
5	Forgalom	Forgalom
6	>5000	<7000

118. ábra Egymás melletti feltételek

Az Irányított szűrés lépései

1. Készítsük el a feltétel (kritérium) tartományt
2. A parancs indításakor legyen aktív az adatbázis-táblázat (**adatbaz.xls**- Termékek nevű munkalap) egy tetszőleges mezejét tartalmazó cella.
3. Adjuk ki az **Adatok** menő **Szűrő** parancsát, majd az **Irányított szűrő** parancsot, a következő ábra a megjelenő párbeszédablakot mutatja.



119. ábra Az Irányított szűrés párbeszédablak

- ✦ A választókapcsoló párosból állítsuk be a **Más helyre másolja** választókapcsolót, ezzel elérjük, hogy az eredménytáblázatot az Excel a munkalap egy általunk adott helyére másolja majd, ezt a helyet a párbeszédablak **Hova másolja** mezejében adjuk meg.
 - ✦ A **Listatartomány** beviteli mezőben nincs teendőnk, mivel itt látható az adatbázis-táblázat tartománycíme, ezt az Excel detektálta.
 - ✦ A **Szűrőtartomány** beviteli mezőben adjuk meg az előzetesen elkészített feltételtartomány címét (kritérium).
 - ✦ A **Csak egyedi rekordok megjelenítése** jelölőnégyzetet akkor kell bekapcsolni, ha fennáll annak a veszélye, hogy egyes rekordok többszörösen rögzítettek. Ezen esetben (ha a jelölőnégyzet bekapcsolva van) a szűrés csak egyszer történik meg.
4. A szükséges beállításokat elvégezve, kattintsunk a párbeszédablak OK gombjára és a szűrés megtörténik. Ha akarjuk, a szűrt tartományon további adatbázis parancsokat hajthatunk végre.

Feladat: Irányított szűrő

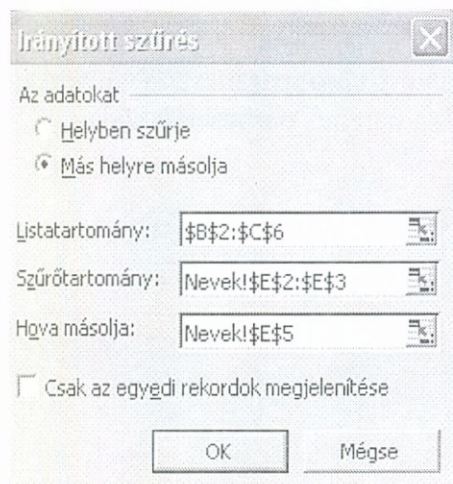
Az **adatbaz.xls** állomány **Nevek** nevű munkalapján található adatbázis-táblázat rekordjainak Irányított szűrővel történő szűrése a feladat. Szűrjük ki az adatbázisból azokat a rekordokat, amelyek **Kis** nevű kereskedőnk **Fizetési** adatai tartalmazzák.

1. Nyissuk meg az **adatbaz.xls** állományt, vigyük be a **Nevek** nevű munkalapon a kurzort az adatbázis-tábla egy hasznos mezejébe:

B2	B	C	D	E
	Vezetéknév	Fizetés		Vezetéknév
2	Kis	100000		Kis
3	Kiss	200000		
4	Nagy	300000		
5	Órsós	400000		

120. ábra Irányított szűrés

- A szűrési feltétel az E2:E3 cellatartományban adott.
- Adjuk ki az **Adatok** menü **Szűrő** parancsát, majd az **Irányított szűrő** parancsot. A párbeszédablakot állítsuk be az alábbi képernyőkivágás szerint:



121. ábra Szűrési beállítások

- Végezzük el a szűrést.
- Megnézve az eredményt, azt látjuk, hogy kiszűrte az Excel a Kiss-ek rekordjait is. A kritériumot kell megváltoztatnunk:

E3		fx " =Kis"		
	B	C	D	E
2	Vezetéknév	Fizetés		Vezetéknév
3	Kis	100000		=Kis
4	Kiss	200000		
5	Nagy	300000		
6	Orsós	400000		

122. ábra Szűrési feltétel

- A fenti kritériumra ("=Kis"), azért van szükség, mert a Kis kritérium önmagában csak azt jelenti, hogy a név Kis-sel kezdődik.

A gyakorlatban jól hasznosítható kritériumok

A következő két ábra közül az első bemutatunk egy minta adatbázis-táblát és azt követően, pedig néhány jól hasznosítható kritériumot.

	B	C
3	Név	Fiz
4	Péter János	500000
5	Kis Péter	600000
6	Kiss Pál	700000

123. ábra Minta adatbázis-tábla

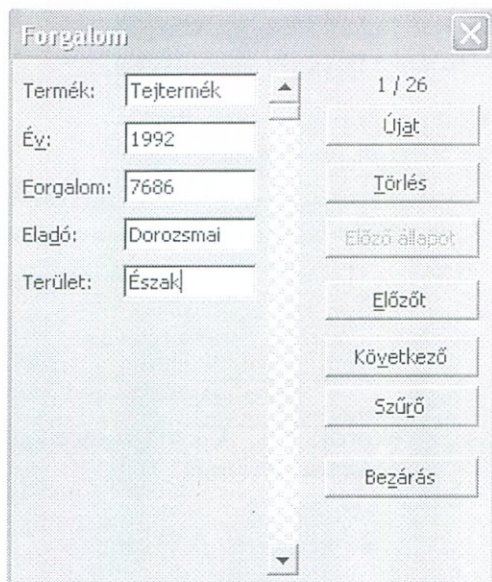
Név	Funkció
* Péter	Kiszűri a Péter keresztnévűeket
"=Kis *"	Kiszűri a Kis vezetéknévűeket
*Péter	Kiszűri azokat, akiknek a nevében Péter található
P*	Kiszűri azokat, akiknek a neve P -vel kezdődik
*P	Kiszűri azokat, akiknek a nevében P betű található

124. ábra Hasznos kritériumok

Úrlap parancs

Az **Úrlap parancs feladata**: a parancs segítséget nyújt adatbázisunk karbantartásában. Pillanatok alatt tudunk vele rekordokat törölni, újakat felvenni, meglévőket javítani. A parancs az egyszerű szűrést is támogatja, igaz (és ez nagy hátránya) az eredmény nem nyomtatható ki.

Az **Úrlap parancs párbeszédablaka**: miután kiadtuk az **Adatok** menü **Úrlap** parancsát (vigyázat: a fénykurzor a parancs kiadásakor az adatbázis egy mezőjén álljon), megjelenik a parancs párbeszédablaka.

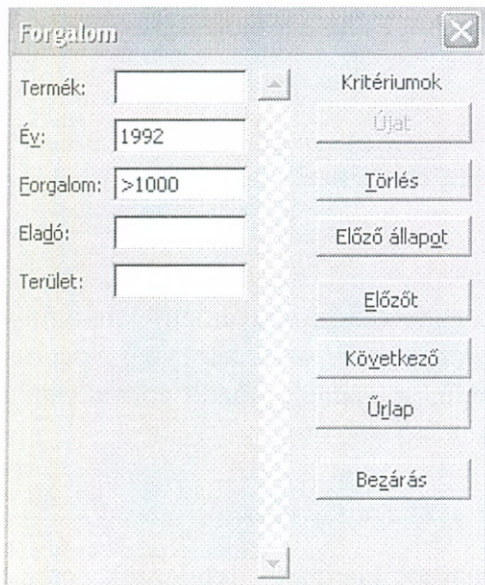


125. ábra Az Úrlap parancs párbeszédablaka

A párbeszédablak szövegdobozaiban megkapjuk az adatbázis első rekordjának mezőit. A jobb felső ablakrészben egy számpárt látunk, közöttük törtvonalat. A törtvonal előtti szám az aktuális rekord, míg az utána lévő az összes rekord számát mutatja az adatbázisban.

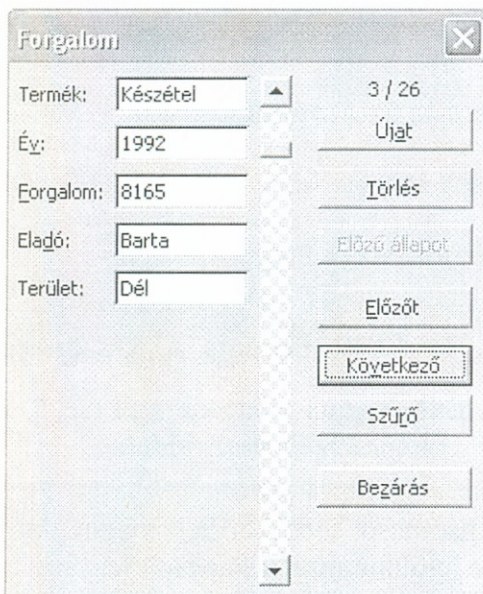
Szűrés az Úrlap paranccsal

1. Kattintva a **Szűrő** nyomógombra (125. ábra), a párbeszédablak minden mezője üres lesz, mivel ezekbe kell gépelni a szűrőfeltételeket.
 - ✦ A szűrőfeltételek csak önmagukban állhatnak, összetett nem lehet megadni. Ugyanakkor használhatók a feltételekben a régebben tanult logikai műveleti jelek is.
2. Az alábbi képernyőkivágáson (**adatbaz.xls**, **Forgalom** nevű munkalapja) látható szűrőfeltételt begépelve, kiszűrjük az adatbázisból azon rekordokat, amelyekben az **Év** mezőben **1992** és ugyanakkor a **Forgalom** mezőben nagyobb, mint **1000** érték található.



126. ábra Szűrőfeltételek

3. Az eredményrekordokat természetesen csak egyesével tekinthetjük meg, kereséssel. A kritériumoknak megfelelő rekordok keresését két nyomógombbal is indíthatjuk. Amennyiben lefelé irányban szándékozunk keresni úgy a **Következő**, ha felfelé, akkor az **Előzőt** nyomógombra kell kattintanunk.
4. Amikor az Excel talált egy a feltételeknek megfelelő rekordot, a keresés megáll és a rekord mezőit megjeleníti. Az alábbi képen látható, hogy a keresési feltételeknek megfelel az adatbázis harmadik rekordja.



127. ábra Egy megtalált rekord

5. Amennyiben tovább is szeretnénk keresni, úgy mindig kattintsunk a **Következő** nyomógombra. Mint látható, az eredményeket nem lehet kinyomtatni, ez a parancs nagy hátránya.

Rekordok szerkesztése az Űrlap parancssal

Miután megkerestük a szerkesztendő rekordot, annak mezői megjelennek a párbeszédablak szövegdobozzaiban. A mezőket az egyes szövegdobozokban szerkeszthetjük, a művelet végén az **ENTER** billentyű megnyomásával aktualizáljuk a rekordot.

Az Űrlap párbeszédablak nyomógombjainak szerepe

Újat: kattintva a nyomógombra, egy üres párbeszédablakot kapunk, amelynek szövegdobozzaiba gépelhetjük a mezők értékeit. Miután az új rekordot létrehoztuk, azt az **ENTER** billentyűvel érvényesítjük.

Törlés: a nyomógombra kattintva, az Excel engedélyt kér a rekord törlésére, és ha azt megadjuk, akkor úgy törli, hogy azt visszaállítani nem lehet többé.

Zárás: a parancsból való kilépésre szolgál.

Megjegyzés: a másik mód a kilépésre, ha kattintunk a párbeszédablak Bezárás (Close) ikonjára.

Részösszegek parancs

A **Részösszegek parancs feladata:** felhasználásával rekordcsoportjainkon (alapvetően) statisztikai műveleteket végezhetünk. Az eredményeket, az Excel **Részösszeg** függvényekkel hozza létre, egy speciális képernyőképen, az úgynevezett vázlatban (tagolás). A vázlat többszintű, a szintek között közlekedni a vázlatszint kapcsolókra történő kattintással lehet. A legmagasabb vázlatszint az 1-es.

A Részösszegek parancs végrehajtásának lépései

1. **Rendezzük sorba a rekordokat:** mielőtt a **Részösszegek** parancsot kiadnánk, rendezzük sorba a rekordokat azon mező szerint, amely alapján a rekordokat csoportosítani akarjuk.
2. Miután a rekordokat sorba rendeztük, adjuk ki az **Adatok** menü **Részösszegek** parancsát.
3. Válasszuk a megjelenő párbeszédablak **Csoportosítási alap** lenyílójában a csoportosítás alapját képező mező nevét.

A Részösszegek párbeszédablak kezelőszervei

- ✦ **Melyik függvénnyel:** a lenyílóból választhatjuk ki az Excel által beszúrandó függvény nevét.
- ✦ **Összegzendő oszlopok:** a listaablakban be kell kapcsolni azon mezők neve mellett a jelölőnégyzetet, amely mezőkbe a kiválasztott függvényt be akarjuk szűrni.
- ✦ **A jelölőnégyzetek szerepe**
 - ✦ Ha a **Részösszegek lecserélése** jelölőnégyzet bekapcsolt, akkor az Excel egy új függvény beszúrásakor törli az összes régit.
 - ✦ Ha az **Oldaltörlés a csoportok között** jelölőnégyzet bekapcsolt állapotában nyomtatunk, az Excel a csoportok közé oldaltörést szúr be.
 - ✦ Ha az **Összeg az adatok alatt** jelölőnégyzet bekapcsolt, akkor az Excel a függvényeket az adatscsoportok alá szúrja be.
- 4. Miután beállításainkat megtettük, kattintsunk az OK gombra, az Excel beszúrja a **Részösszeg** függvényeket.

A Részösszegek parancs ismételt végrehajtása

Ahhoz, hogy több **Részösszeg** függvényt szúrjunk be, a **Részösszegek** parancsot többször kell kiadni. Arra vigyázzunk, hogy a parancs ismételt kiadásakor a **Részösszegek lecserélése** jelölőnégyzet kikapcsolt legyen.

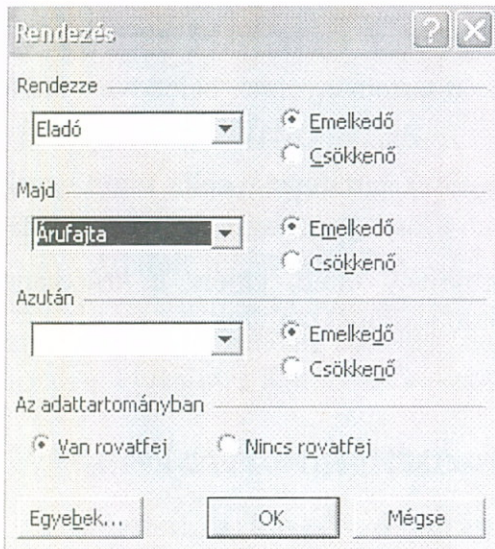
A Részösszegek törlése

Mint látható, a **Részösszegek** parancs végrehajtása átstrukturálja a munkalapot. Amennyiben szükséges a parancs előtti munkalap felépítés visszaállítása, akkor kattintsunk a **Részösszegek** parancs párbeszédablakában a **Mindent törli** gombra.

Gyakorlat: Részösszegek

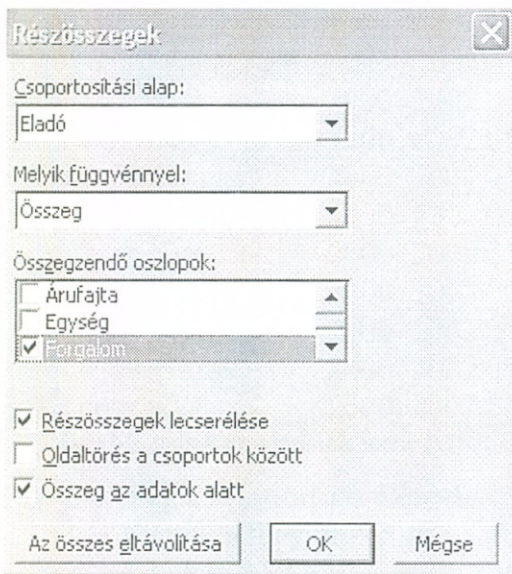
Az **adatbaz.xls** állomány **Termékek** nevű munkalapján eladónkénti bontás alapján ki szeretnénk számítani az összes forgalmat:

1. Tegyük aktívvá a **Termékek** munkalapot. Rendezzük sorba rekordjainkat az **Eladó** mező alapján.



128. ábra Sorba rendezés

- Adjuk ki az **Adatok** menü **Részösszegek** parancsát. A megjelenő párbeszédablakot az alábbiak szerint állítsuk be, majd kattintsunk az OK gombra.



129. ábra Részösszeget képzünk

- Egy speciális képernyőkép a **Tagolás (Vázlat)** jön létre, kattintsunk a 2. vázlat szint kapcsolóra. Az alábbi eredményt kapjuk:

		A	B	C	D
	1	Eladó	Árufajta	Egység	Forgalom
+	28	Barta Összesen			112 940 Ft
+	56	Dorozsmai Összesen			140 091 Ft
+	62	Orsós Összesen			33 539 Ft
-	63	Végösszeg			286 570 Ft

130. ábra A létrejött vázlat (tagolás)

Adatbázis függvények

Fontosabb adatbázis függvények:

AB.MIN

AB.MAX

AB.ÁTLAG

AB.DARAB

AB.DARAB2

AB.SZUM

Az AB.SZUM függvény szintaxisa: AB.SZUM(adatbázis;mező;kritérium)

Az adatbázis függvények Adatbázis paramétere: az adatbázis-táblázat tartománycíme, beleértve a mező-neveket. Ezt az Excel a függvények használatakor nem detektálja.

Az adatbázis függvények Mező paramétere: azon oszlop mező nevet tartalmazó cella címe, amely oszlopban a függvénnyel számolunk.

Az adatbázis függvények Kritérium paramétere: azon cellatartomány címe, amely a kritériumot tartalmazza. A kritériumokat hasonlóan kell létrehozni, mint azt az Irányított szűrőnél tanultuk.

Mintafeladat: Adatbázis függvények

Feladatunk annak kiszámítása, hogy Dorozsmai nevű eladónk 1992-ben (adatbaz.xls, Forgalom nevű munkalapja) összesen mennyi forgalmat ért el.

1. Állítsuk elő a kritériumot (131. ábra).

	G	H
10	Eladó	Év
11	Dorozsmai	1992

131. ábra A kritérium

2. Vigyük be a függvényt, Függvény varázslóval:

Függvényargumentumok

AB.SZUM

Adatbázis: A1:E27 = {"Termék";"Év";"For"

Mező: C1 = "Forgalom"

Kritérium: G10:H11 = G10:H11

= 32196

Az adatbázis adott feltételeknek eleget tevő rekordjaiból álló mezőben (oszlopban) összeadja az értékeket.

Kritérium a megadott feltételeket tartalmazó cellatartomány. A tartományban szerepel egy oszlopfelirat és alatta egy cellával a feltételt megadó címke.

Érték: 32196

Súgó a függvényről

Kész Mégse

132. ábra AB.SZUM függvény paramétere

3. Az eredmény a következő ábrán a G13-as cellában látható:

G13	=AB.SZUM(A1:E27;C1;G10:H11)						
	A	B	C	D	E	F	G
13	Készétel	1993	7 047 Ft	Dorozsmai	Nyugat		32196
14	Készétel	1992	7 191 Ft	Dorozsmai	Észak		
15	Tejtermék	1992	5 575 Ft	Dorozsmai	Kelet		
16	Tejtermék	1992	7 612 Ft	Barta	Dél		
17	Tejtermék	1992	4 873 Ft	Barta	Észak		
18	Tejtermék	1993	8 076 Ft	Dorozsmai	Dél		
19	Tejtermék	1992	3 338 Ft	Dorozsmai	Nyugat		
20	Tejtermék	1993	6 544 Ft	Dorozsmai	Nyugat		
21	Készétel	1993	6 955 Ft	Barta	Kelet		
22	Készétel	1993	4 138 Ft	Dorozsmai	Kelet		
23	Készétel	1992	8 447 Ft	Barta	Kelet		
24	Tejtermék	1993	8 516 Ft	Dorozsmai	Nyugat		
25	Készétel	1993	6 930 Ft	Barta	Nyugat		
26	Készétel	1992	2 686 Ft	Dorozsmai	Nyugat		
27	Készétel	1992	5 720 Ft	Dorozsmai	Kelet		

133. ábra Az AB.SZUM függvény eredménye

3. rész: Haladó Excel ismeretek

1. fejezet: Kimutatás

Az Excel **Kimutatás** parancsa segítségével adatbázis-táblából Excel táblázatot generálhatunk, ezzel azt jóval áttekinthetőbbé tesszük, egyúttal a mezőkön számításokat is végezhetünk.

Az **Adatok** menü **Kimutatás** parancsát egy Kimutatás varázsló valósítja meg, ebből máris kitűnik, hogy igen egyszerűen létrehozható egy kimutatás. Amin el kell gondolkoznunk, az a létrehozandó táblázat struktúrája. Jó azonban tudnunk, hogy a struktúra utólag, igen kényelmesen megváltoztatható. Egy feladat kapcsán kívánjuk a Kimutatás készítés alapjait szemléltetni.

Mintafeladat a Kimutatás készítés szemléltetésére

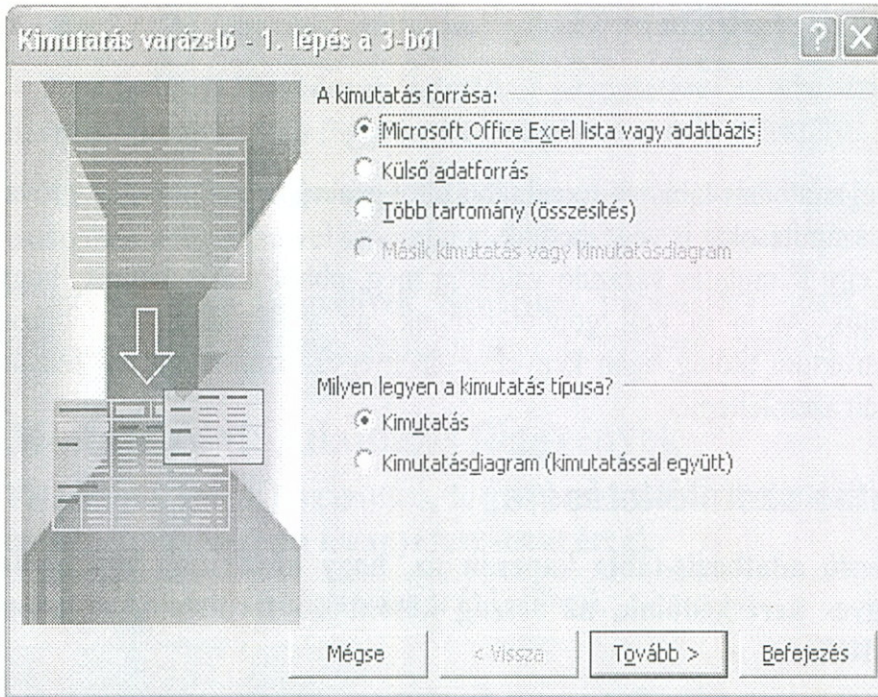
Feladatunk a következő ábrán található adatbázis-tábla kapcsán az, hogy készítsünk egy olyan kimutatást, amely megmutatja, hogy egyes kereskedőink, az ország különböző területein, az adott időszakban milyen volumenben értékesítettek.

	B	C	D	E
3	Név	Terület	Forgalom	Dátum
4	Kis	Dél	1000	2002.01.04
5	Nagy	Kelet	1200	2002.01.06
6	Orsós	Nyugat	1400	2002.01.08
7	Orosz	Észak	1600	2002.01.08
8	Veres	Dél	1800	2002.01.11
9	Kis	Kelet	2000	2002.01.10
10	Nagy	Nyugat	2200	2002.01.11
11	Orsós	Észak	2400	2002.01.12
12	Orosz	Dél	2600	2002.01.13

134. ábra Megadjuk az adatbázis forrását (KIMUTAT.XLS, Munka1)

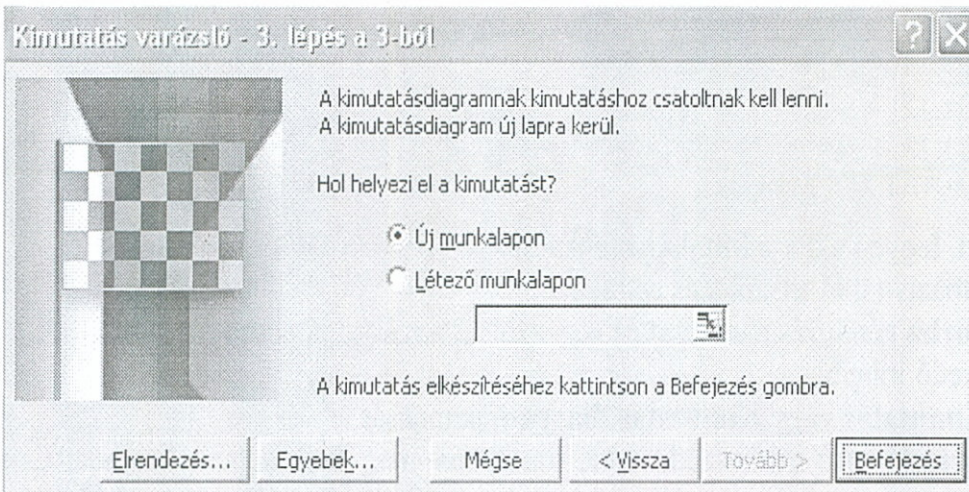
Nyissuk meg a **kimutat.xls** állományt, legyen aktív a Munka1 munkalap.

1. Vigyük a fénykurzort az adatbázis-tábla tetszőleges cellájára.
2. Adjuk ki az **Adatok** menü **Sorba rendezés** parancsát és végezzük el a rekordok sorba rendezését a **Terület** mező alapján, emelkedő irányban.
3. Adjuk ki az **Adatok** menü **Kimutatás vagy Kimutatás diagram** parancsát.
- ✦ Megjelenik a Kimutatás varázsló első párbeszédablaka, amelyben meg kell adnunk az adatbázis forrását a **Kimutatás forrása** keret választókapcsolóinak segítségével (állítsuk be a párbeszédablakot a 135. ábra szerint). A párbeszédablak **Milyen legyen a kimutatás típusa** keretében azért választottuk az alsó választókapcsolót, hogy a kimutatás egy diagrammal is kiegészüljön.



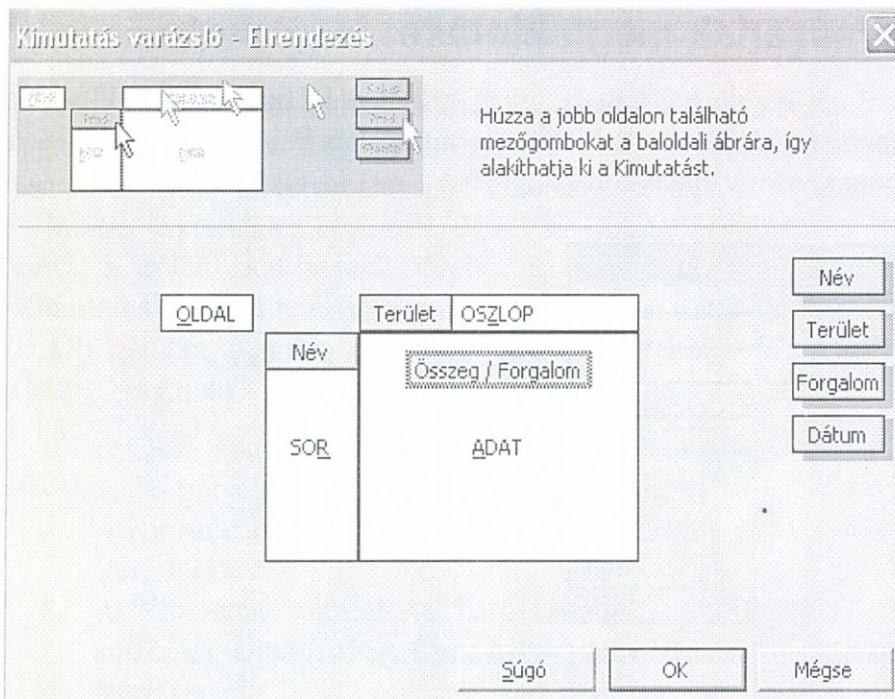
135. ábra az adatbázis forrását, egyúttal diagramot is készítünk.

4. Kattintsunk a **Tovább** gombra.
5. A Kimutatás varázsló második lépésének párbeszédablakában az Excel jelzi az általa detektált adatbázis-táblázat tartományt. Kattintsunk a **Tovább** gombra.
6. A Kimutatás varázsló harmadik lépésének párbeszédablakában (136. ábra) egyrészt megadhatjuk a létrehozandó kimutatás helyét (Hol helyezze el a kimutatást), másrészt különböző beállításokat tehetünk. Ha célunk ez utóbbi, akkor kattintsunk az **Egyebek** gombra.



136. ábra A Kimutatás varázsló 3. lépésnek párbeszédablaka

7. A Kimutatás varázsló harmadik lépése párbeszédablakának része egy **Elrendezés** (Layout) nevű gomb. Ide kattintva kialakíthatjuk a létrehozandó kimutatás táblázat struktúráját. Kattintsunk tehát erre a gombra.
- ✦ Létezik egy másik lehetőség is a kimutatás táblázat struktúrájának kialakítására, ehhez nem kell az **Elrendezés** gombra kattintani, hanem utólag a mezők húzásával lehet azt kialakítani.
8. A megjelenő **Kimutatás varázsló – Elrendezés** párbeszédablakában nincs is más feladatunk, mint a mezőket szimbolizáló gombokat egérrel behúzni a struktúrába. Az elkészült struktúrát mutatja a 137. ábra. Ha véletlenül egy mezőt feleslegesen húztunk be a struktúrát jelképező ablakba, akkor egérrel megfogva húzzuk ki. Nem kell a helyére visszahúzni azt.



137. ábra A Kimutatás varázsló Elrendezés párbeszédablaka

9. Kattintsunk az **OK** gombra, visszajutunk a Kimutatás varázsló harmadik lépésének párbeszédablakába. Kattintsunk annak **Befejezés** gombjára. Kimutatásunk létrejött (138. ábra).

	A	B	C	D	E	F
1	Ide húzhatja az Oldal mezőket					
2						
3	Összeg / Forgalom	Terület				
4	Név	Dél	Észak	Kelet	Nyugat	Végösszeg
5	Kis	1000		2000		3000
6	Nagy			1200	2200	3400
7	Orosz	2600	1600			4200
8	Orsós		2400		1400	3800
9	Veres	1800				1800
10	Végösszeg	5400	4000	3200	3600	16200

138. ábra Az elkészült kimutatás

Ezzel a feladatot megoldottuk. Az Excel beszúr egy munkalapot, erre helyezi a kimutatást, egyúttal beszúr egy diagram típusú lapot is, ezen készíti el a kimutatás alapján készült diagramot. **A mintafeladatot megoldottuk.** A következőkben röviden áttekintjük a kimutatás táblázat tovább formázásának lehetőségeit.

Mezők hozzáadása és törlése

A kimutatás táblázat elkészülte után a képernyőn megjelenik egy **Kimutatás mezőlista** nevű ablak, ebben félkövérek azon mezők nevei, amelyek a kimutatásban megtalálhatók. Megragadva ezen ablakban a **Dátum** mezőt és azt a kész kimutatás **Név** mezeje mellé húzva, ez a mező is részévé válik a kimutatásnak.

Mező hozzáadásánál lényegesen könnyebb egy felesleges mező törlése a kimutatásból. Ehhez nem kell mást tenni, mint megragadni egérrel a mező nevét szimbolizáló gombot és kihúzni a kimutatás területéről. Akkor engedjük csak fel csak az egér gombját, amikor a húzott gomb alatt egy áthúzás jel jelenik meg.

Az elrendezés megváltoztatása

Az elrendezés megváltoztatásához az adott mezőt szimbolizáló gombot be kell húzni a struktúra táblázat azon helyére, ahol azt látni szeretnénk.

Az adatok egy adott részalmazának megtekintése

Kimutatásainkban rövid úton létrehozhatunk oldalmező(ke)t is. Ezen célból a **Dátum** mezőt (139. ábra) húzzuk be az Oldalmezőbe (ez a képernyőn kék kerettel látható munkalaprészt, „**Ide húzhatja az Oldalmezőket**” szöveggel). A 139. ábra mutatja az így átalakított kimutatást.

	A	B	C	D	E	F	
1	Dátum	Kimutatás mezőlista					
2		Húzza a tételeket a kimutatásra					
3	Összeg / Forgalom						
4	Név				at	Végösszeg	
5	Kis					3000	
6	Nagy				2200	3400	
7	Orosz					4200	
8	Orsós				1400	3800	
9	Veres					1800	
10	Végösszeg				8600	16200	
11							
12							
13							
14							
15							
16							
17							
18							
19							

139. ábra Oldalmezőt is tartalmazó Kimutatás

Vegyük észre, hogy a **Dátum** nevű Oldalmező lenyílóként viselkedik, kattintsunk annak listanyilára, ekkor láthatjuk, hogy táblázatunk a Dátum értékek alapján “lapozható”.

A kívánt részletek megjelenítése

Megjeleníthetünk vagy épp ellenkezőleg eltüntethetünk részleteket. Például ha nem szeretnénk a kimutatásban látni Nagy adatait, akkor kattintsunk a **Név** kimutatás mező lenyílójának listanyilára. Ekkor egy listában a jelölőnégyzetek mellett megjelennek a mező értékek, kapcsoljuk ki a **Nagy** mezőérték melletti jelölőnégyzetet.

	A	B
1	Dátum	(mind)
2		
3	Összeg / Forgalom	Terület
4	Név	Dél
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

140. ábra Nagy jelölőnégyzetét kikapcsoltuk

Kimutatásban végezhető számítások

Amikor a mintafeladatban létrehoztuk a kimutatás struktúráját, akkor ahhoz a Kimutatás varázsló harmadik lépésében, a **Kimutatás – Elrendezés** párbeszédablakban, a kimutatás ADAT mezejébe a **Forgalom** mezőt húztuk be. Ekkor az Excel alapértelmezetten összeadás műveletet végeztet el ezen mezőértékekkel.

Miután a kimutatás elkészült, és mi az összeadás műveletet le szeretnénk cserélni például átlagra, akkor ebből a célból kattintsunk duplán a már kész kimutatás **Összeg/Forgalom** mezejére. A megjelenő **Kimutatásmező** párbeszédablak **Mezőstatisztika** nevű listaablakában jelöljük ki az Átlagot, majd kattintsunk az **OK** gombra. Egy kis idő eltelte után az Excel lecseréli a műveletet átlagra. Ezt abból láthatjuk, hogy az **Összeg/Forgalom** mező új neve **Átlag/Forgalom** lesz.

Mi a teendő akkor, ha úgy akarjuk a mezőértékek átlagát kiszámíttatni, hogy egyúttal meg szeretnénk őrizni az összeadás művelet eredményeit is. Egy lehetséges megoldás:

1. A **Kimutatás** eszköztáron kattintsunk a **Kimutatás** lenyílóra és a megjelent parancslistából válasszuk a **Varázslót**.
2. A Kimutatás varázsló a harmadik lépéstől újra indul, kattintsunk a párbeszédablak **Elrendezés** gombjára, a már ismert **Elrendezés** párbeszédablakban a **Forgalom** mezőt húzzuk be a struktúra ADAT mezejébe.
 - ✦ A varázsló, a mező nevében az **Összeg/Forgalom** szöveggel jelzi, hogy a mező értékek összeadását tervezi (**Összeg**). Kattintsunk duplán az ADAT mezőben azon mezőt szimbolizáló gombra, amely mezőben a művelet cseréjét tervezzük. Az ekkor felkínált **Kimutatásmező** ablakban (ezt már ismerjük) cseréljük le a műveletet.
3. Már nincs is más teendőnk mint a Kimutatás varázslót a végéig leléptetni.

Mivel célunk csak a kimutatás alapjainak bemutatása volt, ezért ezen a ponton be is fejezhetnénk a Kimutatásról szóló rövid ismertetőnket, de további kedvcsinálás végett megmutatunk egy mind gyakrabban használt kimutatás készítési fogást. Igen sűrűn feladatunk, hogy olyan táblázatot állítsunk elő, ahol százalékos arányokat kell megjeleníteni.

1. Miközben aktív az elkészült kimutatás ADAT mezejének tetszőleges cellája, a **Kimutatás** eszköztáron kattintsunk a **Mezőbeállítások** ikonra.
2. A megjelent Kimutatásmező ablakban kattintsunk az **Egyebek** gombra. Az **Adatok megjelenítése** lenyíló listájában válasszuk az **Oszlop százaléka** elemet. Az ekkor létrejött kimutatás jól mutatja az egyes kereskedők területi eladásainak arányát.

	A	B	C	D	E	F
1	Dátum	(mind) ▼				
2						
3	Összeg / Forgalom	Terület ▼				
4	Név ▼	Dél	Észak	Kelet	Nyugat	Végösszeg
5	Kis	18,52%	0,00%	62,50%	0,00%	18,52%
6	Nagy	0,00%	0,00%	37,50%	61,11%	20,99%
7	Orosz	48,15%	40,00%	0,00%	0,00%	25,93%
8	Orsós	0,00%	60,00%	0,00%	38,89%	23,46%
9	Veres	33,33%	0,00%	0,00%	0,00%	11,11%
10	Végösszeg	100,00%	100,00%	100,00%	100,00%	100,00%

141. ábra Oszlop százaléka

2. fejezet: Külső adatok importja

Az Excel adatbázis-kezelésének egyik gyengéje a kezelhető rekordok és a mezők viszonylag kis száma. Ezért adatbázist kezelni nem Excellel kell. De, ha az adatbázis mérete kellőképpen kicsire szelődött, akkor igen hasznos lenne azt a továbbiakban az Excellel feldolgozni. Ez lehetséges, az Excel képes megnyitni (és azon munkát végezni) adatbáziskezelőkben készült állományokat. Ennek szemléltetésére az alábbiakban bemutatunk egy egyszerű külső lekérdezést.

Külső adatok importja a Query program segítségével

Az Excelnek része egy saját felhasználói felülettel rendelkező program a Query. A Query programmal megadhatjuk az ODBC adatforrásokat és visszakereshetjük az adatokat. Egyszerű lekérdezés készítéséhez a Lekérdezés varázslót, összetettebb lekérdezés készítéséhez pedig a Query magas szintű funkcióit alkalmazhatjuk. A Queryt közvetlenül az Excel programból is elérhetjük, de készíthetünk lekérdezést a Kimutatás vagy a Kimutatásdiagram varázslóval is.

A Query programban a DDE (Dynamic Data Exchange) - a Microsoft Windows alapú programok közti adatcserét lehetővé tevő szabványos protokoll - használatára is lehetőség van, amelyről bővebben a program súgójában olvashatunk.

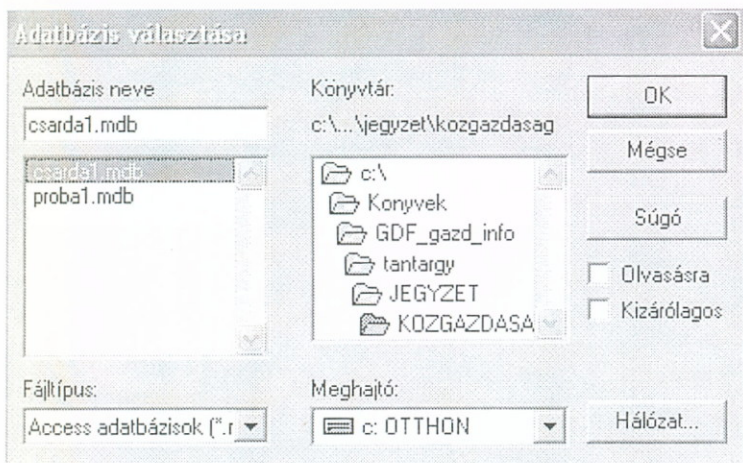
A Query programmal végrehajtott külső lekérdezéshez az alábbiak szükségesek:

- ✦ **A Query telepítése:** A Query az Excel választható programmodulja, amely a Lekérdezés varázslót is magában foglalja. Az **Adatok** menü **Külső adatok importálása** parancsának, **Új adatbázis-lekérdezés** alparancsát választva az Excel a legtöbb esetben felajánlja a Query telepítését.
- ✦ **ODBC illesztőprogramok telepítése:** Relációs adatbázisok, szövegfájlok vagy Query programot használó Excel adatok visszakereséséhez ODBC illesztőprogramra van szükségünk (ODBC: Open Database Connectivity). Minden egyes adatbázis-kezelő programnak (például Access vagy dBase) és adatbázis-kezelő rendszernek (például SQL Server) más és más illesztőprogramra van szüksége. A Query programmal együtt automatikusan ODBC illesztőprogramokat is telepítünk. Ha a megfelelő illesztőprogram nem került telepítésre a Query programmal együtt, akkor erről külön kell gondoskodnunk.

Mintafeladat a Külső adatok importjának szemléltetésére

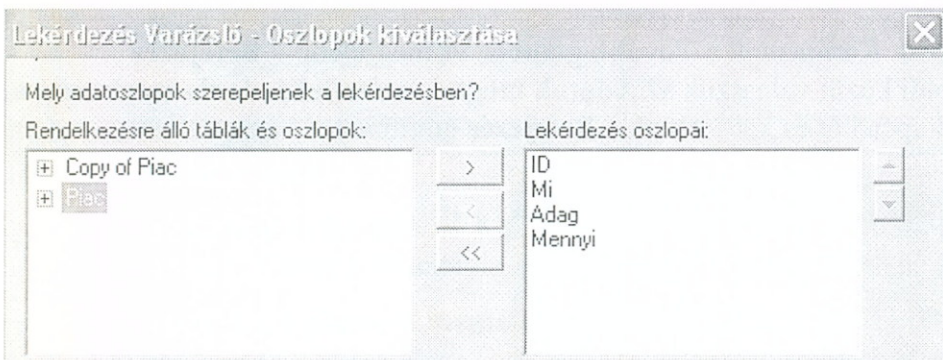
Mintafeladatunkhoz használjuk fel a következő adatbázis-táblát: CSARDA1.MDB

1. Adjuk ki az **Adatok** menü **Külső adatok importálása** parancsát.
2. A megjelenő parancsok közül válasszuk az **Új adatbázis-lekérdezést**.
3. Az elindult Query **Adatforrás kiválasztása** párbeszédablakában kattintsunk az **Adatbázisok** szegélyre. A listából válasszuk ki az MS Access adatbázis* elemet.
4. Kattintsunk az **OK** gombra.
5. Az **Adatbázis választása** párbeszédablak **Könyvtár** nevű listaablakában keressük meg azon könyvtárat, ahol a keresett Access állomány megtalálható. Kattintsunk annak nevére.



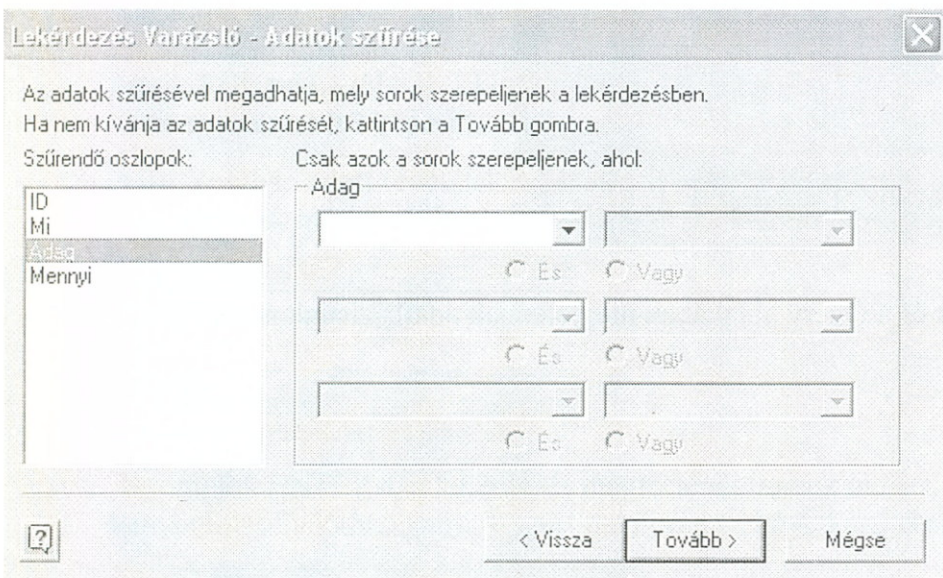
142. ábra Adatbázis állomány kiválasztása

6. Kattintsunk az **OK** gombra.
7. A **Lekérdezés varázsló - Oszlopok kiválasztása** nevű párbeszédablakának bal szélső listaablakában kattintsunk az adattábla nevére (Piac, 143. ábra), majd a nagyobb jelet tartalmazó ikonra. A jobboldali listaablakban megjelennek a lekérdezésben megjelenő mezők nevei.



143. ábra Az adattábla mezőinek kiválasztása

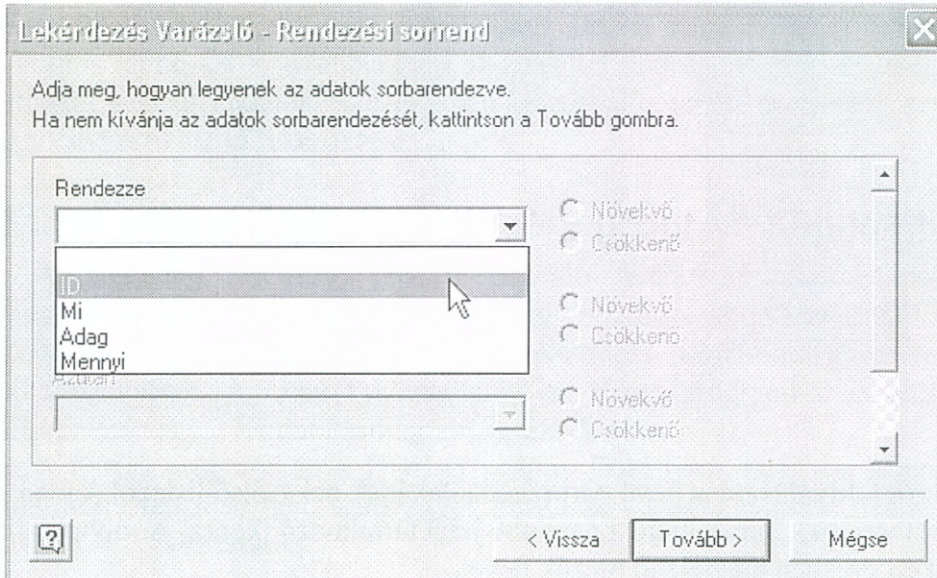
8. Kattintsunk a **Tovább** gombra. Megjelenik a **Lekérdezés Varázsló Adatok szűrése** nevű ablaka (144. ábra).



144. ábra Adatok szűrése

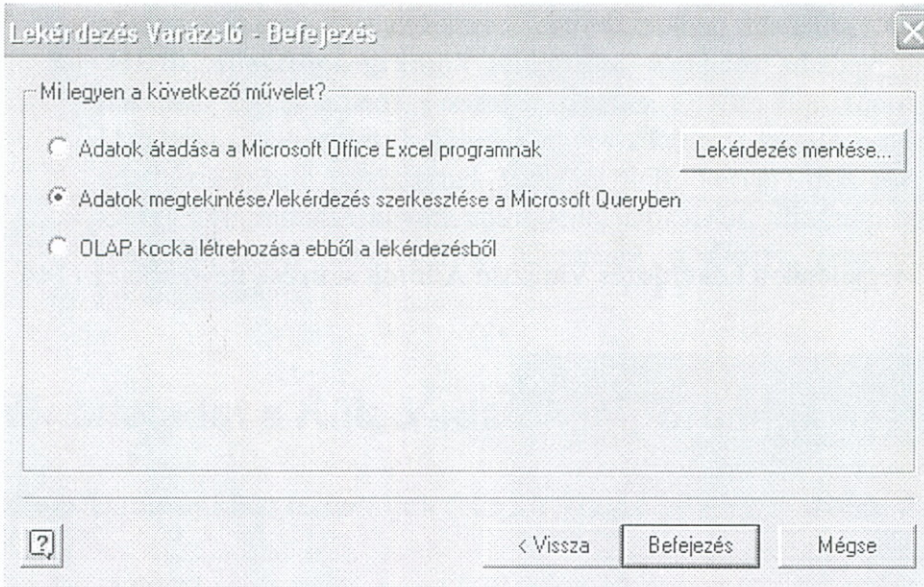
9. Ha rákattintunk a **Szűrendő oszlopok** nevű listaablakban bármelyik mező nevére (mi az **Adag** mezőre kattintottunk), akkor az Excel AutoSzűrőjéhez hasonló módon lehetőségünk nyílik szűrésre. Mi most nem tervezünk szűrést.
10. Kattintsunk a **Tovább** gombra. Megjelenik a **Rendezési sorrend** ablak.

11. Ha lenyitjuk a **Rendezze** lenyílt és abból mezőt választunk, lehetőség nyílik a rekordok sorba rendezésére.



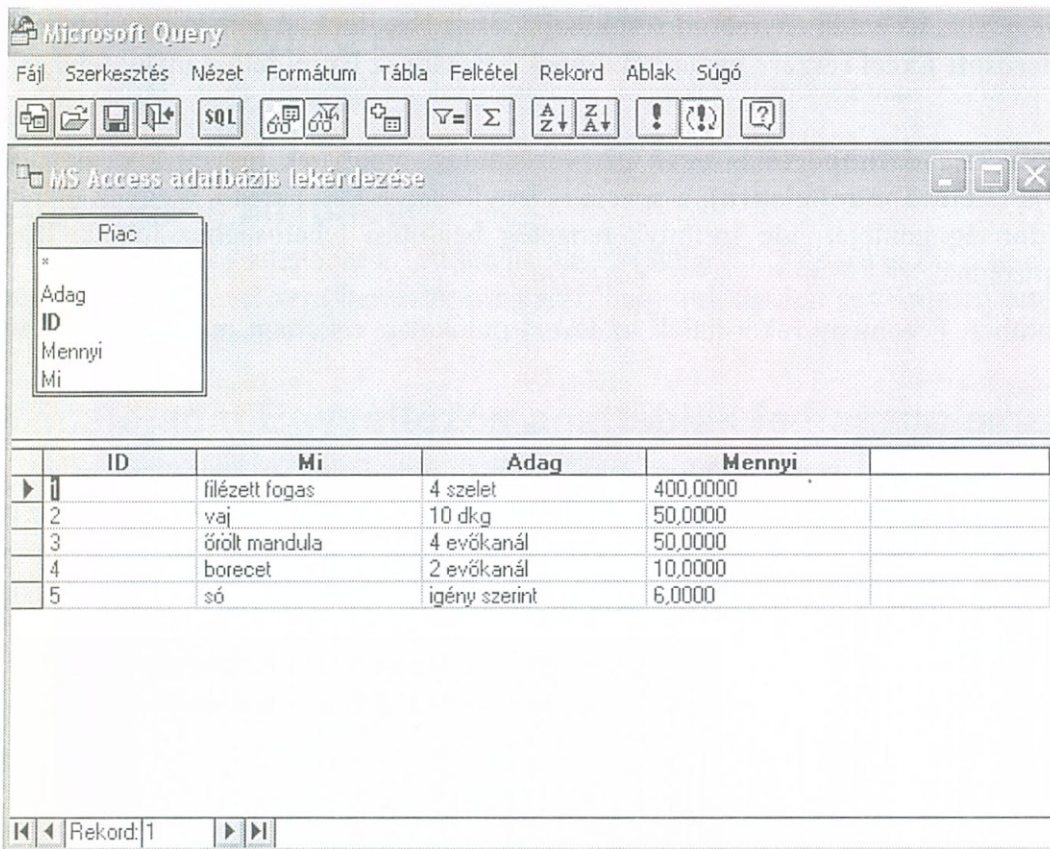
145. ábra Rendezési sorrend

12. Mi most nem tervezünk rendezést. Kattintsunk a **Tovább** gombra. A megjelenő – **Befejezés** párbeszédablak választókapcsolói közül válasszuk az **Adatok megtekintése/lekérdezés szerkesztése a Microsoft Queryben** választókapcsolót és kattintsunk a **Befejezés** gombra.



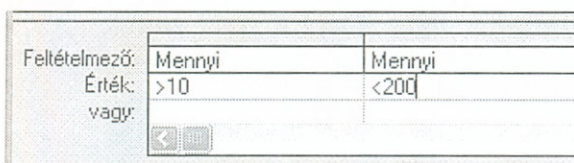
146. ábra Befejezés párbeszédablak

13. A Varázsló befejezi működését és a Query ablakában megjelennek adatbázisunk mezői (147. ábra).



147. ábra Adatbázisunk mezői

14. Most logikusan következik a szűrés művelete. Ennek több módja létezik. Szűrés végezhető ikonnal, paranccsal illetve feltételtábla segítségével. Mi most tekintsük át a feltételtáblával történő szűrést.
15. Kattintsuk a **Feltételek megjelenítése** ikonra. Ekkor egy újabb ablak jelenik meg az alkalmazás ablak közepén.
16. Feladatunk legyen azon rekordok kiszűrése, ahol a **Mennyi** mező értéke 10-nél nagyobb, de egyúttal kisebb mint 200.
17. Ragadjuk meg egérrel az alkalmazásablak bal felső sarkában látható, az adatbázis mezőit szimbolizáló kis ablakban a **Mennyi** mezőt, és húzzuk egérrel a feltételtábla **Feltételmezőjébe**, majd a mellette lévő másik **Feltételmezőbe** is. Ekkor, mivel a két feltétel egymás mellett található a Feltételablaktáblában, azok egymással **ÉS** kapcsolatban vannak.
18. A 16. pontban adott feltételeket gépeléssel vigyük be az érték mezőbe (nagyon hasonló, a feltétel megadásának módja, az Irányított szűrőnél alkalmazotthoz, lásd (148. ábra).



148. ábra A lekérdezési feltételeket előállítottuk

19. Nyomjunk ENTER-t, a lekérdezés megtörténik (ennek oka az, hogy az **Automatikus lekérdezés** ikon benyomott állapotban van). A szűrt eredményt a következő ábrán láthatjuk (149. ábra).

ID	Mi	Adag	Mennyi
2	vaj	10 dkg	50,0000
3	őrölt mandula	4 evőkanál	50,0000

149. ábra Eredmény rekordjaink

20. Ezen a ponton a rekordcsoportokon végezhetnénk számításokat az **Összegzés** ikonnal (nem csak összeadást, mert arra egymás után kattintgatva más és más műveleteket kapunk). Ehhez azonban kevés rekordunk van.

21. Mivel feladatunkat elvégeztük, az eredményrekordokat küldjük át az Excelnek. Adjuk ki a **Fájl** menü **Adatok átadása A Microsoft Excel részére** parancsát. Ennek hatására az Excel felhasználói felülete válik aktívvá.
22. Az Excelben megjelenik az **Adatimportálás** nevű párbeszédablak, amelynek legyen kiválasztva a felső választókapcsolója (**Létező munkalapra**), a speciális lenyílóban **\$A\$1** látható. Vegyük észre a párbeszédablak **Tulajdonság** gombját, ide kattintva rengeteg beállítási lehetőséghez jutunk. Ezek megismerése most nem célunk.
23. Kattintsunk az **OK** gombra. Eredmény rekordjaink az Excel munkalap celláiban megjelennek (150. ábra).

	A	B	C	D
1	ID	Mi	Adag	Mennyi
2	2	vaj	10 dkg	50
3	3	őrölt mandula	4 evőkanál	50

150. ábra Eredmény rekordjaink az Excel munkalap celláiban

Ezzel a mintafeladatot megoldottuk.

3. fejezet: Elemző eszközök

Jegyzetünk ezen fejezetében megismerjük az Excel főbb elemző eszközeit, az Egyváltozós adattáblától egészen az Esetvizsgálóig.

Egyváltozós adattábla

Az Egyváltozós adattábla a "Mi lenne, ha..." (What if...) típusú eszközöknek nevezett parancskészlet részét képezi. Segítségével arra keressük a választ, hogy miként hat egy változó értékének megváltozása egy vagy több függvényre (képletre).

Mintafeladat Egyváltozós adattáblák felhasználására

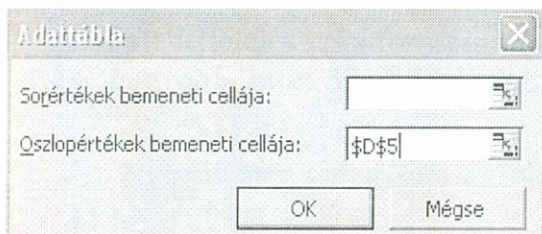
Feladatunk legyen annak kimutatása, hogy miként hat egy kölcsön törlesztő részleteire a kamatláb megváltozása. Magát a törlesztő részletet a **RÉSZLET** nevű függvénnyel számítsuk ki.

	A	B	C	D	E	F
2						
3						
4						
5			kamat			
6			honap	360		
7			penz	80000		
8						
9						
10						
11			\$222,22			
12		9,00%				
13		9,25%				
14		9,50%				
15		10,00%				
16		12,00%				
17		25,00%				
18		30,00%				

151. ábra Egyváltozós adattábla (EGYVALT.XLS)

- ✦ Vegyük észre, hogy a függvényben a változóra egy, az adattáblán kívüli üres cellával hivatkozunk. Ezért üres a D5-ös cella. A legfelső változót tartalmazó cella feletti cella üres kell, hogy maradjon (B11). A függvény vagy a képlet az adattábla első sorában helyezkedik el. A RÉSZLET függvény tehát a C11-es cellába került (a függvény a Szerkesztőlécen látható). A C12:C18 cellatartomány üres.

1. Jelöljük ki az adattáblát (B11:C18 cellatartomány).
2. Adjuk ki az **Adatok** menü **Adattábla** parancsát. A megjelenő párbeszédablakot állítsuk be a 152. ábra szerinti módon. Ezzel a párbeszédablak beállítással jelezzük az Excelnek, hogy a függvény első paraméterét ne a D5-ös cellából vegye, hanem az adattábla bal szélső oszlopából (mivel a párbeszédablak **Oszlopértékek bemeneti cellájaként** adtuk meg). Kattintsunk az **OK** gombra.



152. ábra A változók oszlopban található

A mintafeladatot megoldottuk (153. ábra), az adattáblából megtudjuk, hogy különböző kamatlábak mellett mennyi a havi törlesztés összege, ha 80000 Dollárt vettünk kölcsön 360 havi futamidőre.

	B	C
11		\$222,22
12	9,00%	643,6981
13	9,25%	658,1403
14	9,50%	672,6834
15	10,00%	702,0573
16	12,00%	822,8901
17	25,00%	1667,663
18	30,00%	2000,276

153. ábra Az elkészült Egyváltozós adattábla

Kétváltozós adattábla

A kétváltozós adattábla a "mi lenne, ha" (What if) típusú eszközöknek nevezett parancskészlet részét képezi. Arra keressük a választ, hogy miként hat két változó értékének megváltozása egy függvényre (képletre).

Mintafeladat Kétváltozós adattáblák felhasználására

Feladatunk legyen annak kimutatása, hogy miként hat egy kölcsön törlesztő részleteire a kamatláb és a futamidő-megváltozása. Magát a törlesztő részletet a RÉSZLET nevű függvénnyel számítsuk ki.

Tekintsük át, mennyiben kell másként dolgozni a Kétváltozós adattáblákkal, mint az Egyváltozóssal.

Az egyik változó értékei egy oszlopban, míg a másik változó értékei egy sorban vannak felsorolva.

Csak egy függvényt (képletet) használhatunk és az az adattábla bal felső cellájában szerepel.

1. Nyissuk meg a **ketvalt.xls** állományt (154. ábra) Az adattábla bal felső cellájában látható a kiértékelendő függvény (Szerkesztőlécen).

A9	fx =ABS(RÉSZLET(\$D\$5/12;\$D\$6;\$D\$7))					
	A	B	C	D	E	F
5			kamat	10%		
6			futamidő	360		
7			kölcsön	8000000		
8						
9	70205,72561	180	240	300	360	400
10	9,00%					
11	9,25%					
12	9,50%					

154. ábra Kétváltozós adattábla (KETVALT.XLS)

2. Jelöljük ki az adattáblát (A9:F12).
3. Adjuk ki az **Adatok** menü **Adattábla** parancsát, a megjelenő párbeszédablakot állítsuk be a 155. ábra szerint.

D5	=ABS(RÉSZLET(\$D\$5/12;\$D\$6;\$D\$7))					
	A	B	C	D	E	F
5			kamat	10%		
6			futamidő	360		
7			kölcsön	8000000		
8						
9	70205,73	180	240	300	360	400
10	9,00%					
11	9,25%					
12	9,50%					

Adattábla

Sorértékek bemeneti cellája:

Oszlopértékek bemeneti cellája:

OK Mégse

155. ábra Az Adattábla párbeszédablak beállítása

4. Kattintsunk az OK gombra. Az adattábla B10:F12 celláit kitölti az Excel.

	A	B	C	D	E	F
5			kamat	10%		
6			futamidő	360		
7			kölcsön	8000000		
8						
9	70205,73	180	240	300	360	400
10	9,00%	81141,33	71978,08	67135,71	64369,81	63181,01
11	9,25%	82335,38	73269,35	68510,55	65814,03	64663,91
12	9,50%	83537,97	74570,5	69895,73	67268,34	66156,4

156. ábra Az elkészült adattábla

- Fejtsük meg például a C10-es cella tartalmát. Ez a cella mutatja a havi törlesztés összegét akkor, ha 8 millió Forintot vettünk kölcsön 9%-os éves kamatláb mellett 240 havi futamidőre. A megoldás érdekessége, hogy az Excel a C10-es (no meg a többi) cellába nem a RÉSZLET függvényt írja be, hanem egy TÁBLA nevű függvényt.

Egyváltozós célérték keresés

A célérték keresés a "Mi lenne, ha..." (What if...) típusú parancskészletnek része. Segítségével meghatározhatjuk azt a bemeneti értéket, amely egy képlet (függvény) kívánt eredményét hozza. A célérték keresés során a Microsoft Excel egy adott cella értékét addig módosítja, amíg az azon alapuló képlet eredménye el nem éri a kívánt értéket (ha egyáltalán létezik megoldás).

Mintafeladat célérték keresésre

Mint azt az eddigiekből tudjuk egy kölcsön törlesztő részleteit a RÉSZLET nevű függvénnyel számíthatjuk ki. De mi a megoldás akkor, ha a kérdés úgy merül fel, hogy a leendő adósnak nincs annyi pénze törlesztésre, mint amennyit adott paraméterekkel kiszámoltunk. Ugyanakkor megelégedne kisebb kölcsönrel is. Magyarul, előírt a RÉSZLET függvény eredménye, a függvény egyik paraméterét kell úgy megváltoztatnunk, hogy ezt az eredményt kaphassuk.

- Nyissuk meg a **reszlet.xls** állományt. Vajon mi a kérdés, ha a havi törlesztést kiszámoltuk már? A kérdés az, hogy mekkora kölcsön jár annak, aki csak havi 30 000 Forintot tud törleszteni.

	A	B	C	D	E	F	G	H	I
1	Kölcsön		3 000 000 Forint						
2	Futamidő		120 hónap						
3	Éves kamat		4,50%						
4	Havi kamat		0,38%		Képlet: =C3/12				
5	Havi törlesztés		-31 092 Forint		Formátumkód: ###0" Forint";[Piros]-##0" Forint"				

157. ábra A havi törlesztés összege most (reszlet.xls)

- Legyen aktív a C5-ös cella (ez tartalmazza a RÉSZLET függvényt).
- Adjuk ki az **Eszközők** menü **Célértékkeresés** parancsát. A megjelenő párbeszédablakot állítsuk be a 158. ábra szerint.

158. ábra A RÉSZLET függvény eredménye legyen 30 000

- Kattintsunk az OK gombra. Mint azt a 159. ábra mutatja, az Excel talált megoldást.

159. ábra Az Excel talált megoldást

- Kattintsunk az OK gombra (159. ábra). A 160. ábra mutatja, hogy a havi 30 000 Forintos törlesztéshez mekkora kölcsön járul.

	A	B	C
1	Kölcsön		-2 894 680 Forint
2	Futamidő		120 hónap
3	Éves kamat		4,50%
4	Havi kamat		0,38%
5	Havi törlesztés		30 000 Forint

160. ábra A kölcsön összege a C1-es cellában látszik

Solver

A mindennapi életben gyakran merülnek fel optimalizálási feladatok. Például egy autógyár esetén, ahol három különböző típusú autót kell gyártani, amelyből kétféle nyereséges, de a harmadik (a dzsip) veszteséges. Ebből a hadsereg megrendelésére egy adott mennyiséget le kell gyártani, de mi legyen a fennmaradt gyártási kapacitással?

Az optimalizálási modell három összetevőből áll: döntési változók, célfüggvény és korlátozó feltételek.

A **döntési változókat** a döntést hozók befolyásolni tudják, például túlóra elrendelése stb...

A **célfüggvény** a döntési változók függvénye, ezt szeretnénk maximalizálni, minimalizálni vagy egy adott értékre állítani.

A **korlátozó feltételek** a változókra nézve adnak meg különböző korlátokat.

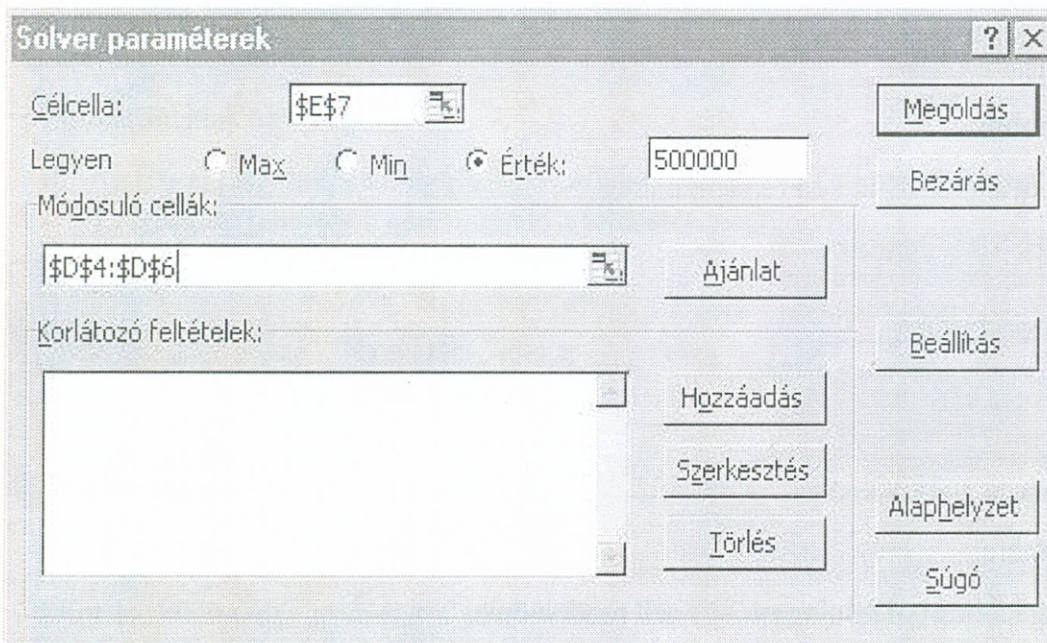
A Solver a döntési változók függvényében a célfüggvényt maximalizálja, minimalizálja, adott értékre állítja be. A Microsoft egy SOLVSAMP.XLS nevű Excel állományt szállít az Excellel, amelynek munkalapjain jellemző Solver feladatokat mutat be megoldva, kommentálva.

Mintafeladat a Solver bemutatására

	B	C	D	E	F	G	H	I	J
1									
2						Maximum 500 000 Dollár van. Minimum 4 alsó- Minimum 3 közép- Minimum 2 nagy kategóriás autó kell.			
3	Autó kategória	ár/autó	darabszám	Teljes ár					
4	alsó	14000	1	14000					
5	közép	20000	1	20000					
6	felső	40000	1	40000					
7			Összesen:	74000					
8					=C4*D4				
9									

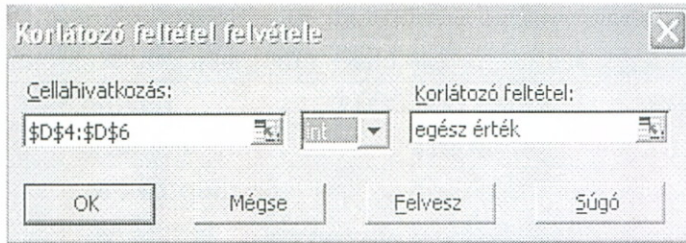
161. ábra Flotta.xls

1. Nyissuk meg a **flotta.xls** állományt.
 - ✦ Cégünk alsó-, közép- és felső kategóriás gépjárműveket kíván venni. Az **ár/autó** oszlopban az árak is adottak. A vásárlás céljára maximum 500 000 Dollárunk van. Minimum 4 alsó-, 3 közép- és 2 nagy kategóriájú (felső) autót szándékozunk venni. Kérdés, hogy ezen felül melyik kategóriából mennyit vegyünk. Hogy pénzünkből fusson.
2. Legyen aktív az E7-es cella.
3. Adjuk ki az **Eszközök** menü **Solver** parancsát. A párbeszédablakot a következő ábra szerinti módon állítsuk be:



162. ábra Solver paraméterek

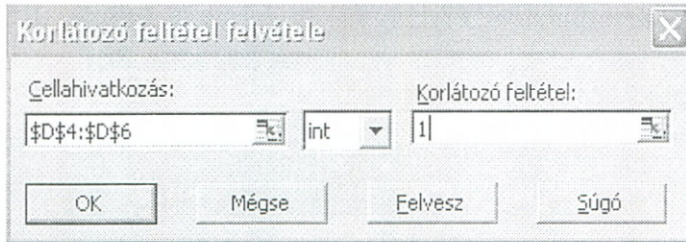
4. Mint a fenti ábrán látható az E7-es cellában lévő függvénynek az értékét 500000-re szeretnénk állítani. Ehhez az Excel módosíthatja a D4, D5 és D6-os cellák tartalmát (darabszámok).
5. Kattintsunk a **Hozzáadás** gombra, hogy peremfeltételeket adhassunk. A D4, D5, D6 cellákban csak egész számok lehetnek. Ezért kattintsunk a **Hozzáadás** gombra. A megjelenő párbeszédablakot állítsuk be a következő ábra szerint.



163. ábra D4:D6-ban csak egész számok lehetnek

6. Kattintsunk a **Felvesz** gombra.

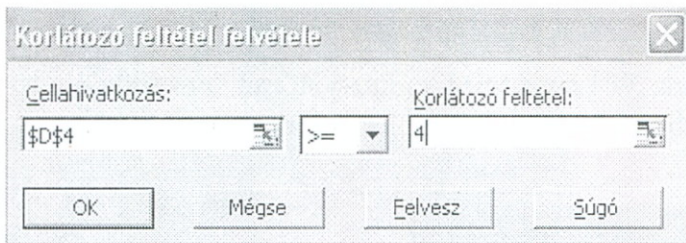
7. Hibajelzést kapunk. A párbeszédablakot állítsuk be a következő ábra alapján.



164. ábra A cellák értéke egész szám legyen

8. Kattintsunk a **Felvesz** gombra.

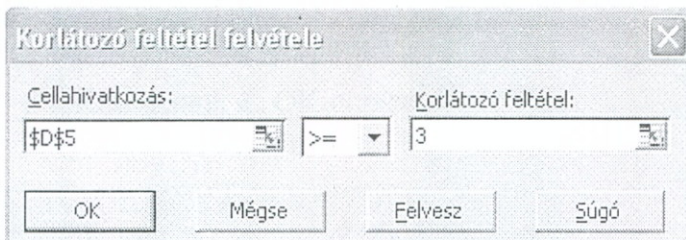
9. Alsó kategóriás személygépkocsiból minimum négyet kell rendelnünk:



165. ábra Alsó kategóriás gépkocsik minimális száma

10. Kattintsunk a **Felvesz** gombra.

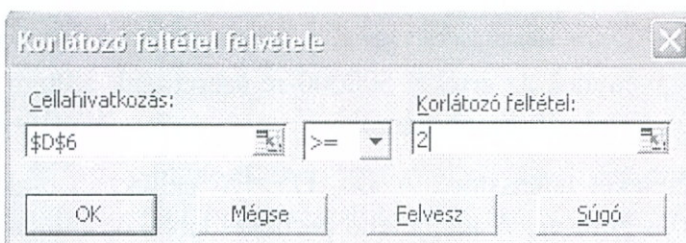
11. Középkategóriás autóból minimum 3-at kell rendelnünk:



166. ábra Középkategóriás autóból minimum 3-at kell rendelni

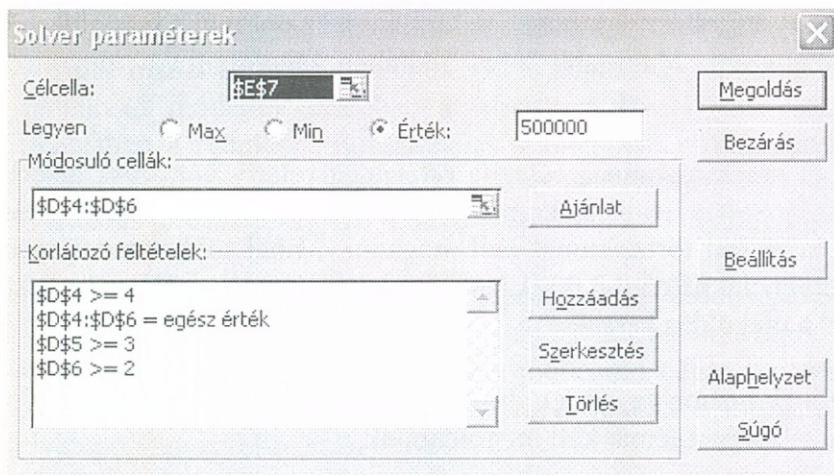
12. Kattintsunk a **Felvesz** gombra.

13. Felső kategóriás személygépkocsiból minimum 2-öt kell rendelnünk:



167. ábra Felső kategóriás autóból minimum 2-öt kell rendelnünk

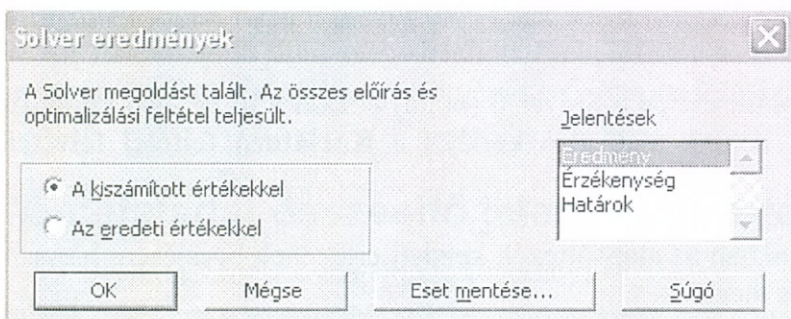
14. Kattintsunk az **OK** gombra. Megint megjelenik a **Solver paraméterek** párbeszédablak.



168. ábra Solver beállítások

15. Kattintsunk a **Megoldás** gombra.

16. Megjelenik a **Solver eredmények** nevű párbeszédablak. Kattintsunk a **Solver eredmények** nevű párbeszédablak **Jelentések** nevű listaablakában a az **Eredmény** bejegyzésre.



169. ábra Solver eredmények

17. Kattintsunk az OK gombra.

18. Az Excel kiszámítja a megoldást, a cellák tartalmát, ha kell megváltoztatja, beszúr egy munkalapot **Eredmény jelentés 1** néven. Ezzel a feladatot megoldottuk.

	B	C	D	E
2				
3	Autó kategória	ár/autó	darabszám	Teljes ár
4	alsó	14000	10	140000
5	közép	20000	6	120000
6	felső	40000	6	240000
7			Összesen:	500000

170. ábra Az eredmények (D4:D6 cellákban)

Mint az köztudott, az ördög a részletekben lakozik, ennek ellenére nem térünk ki a Solver mélységeire. Tervünk a Solver alapjainak bemutatása volt, ezért nem ismertetjük részletesen a Solver beállításait (**Beállítás** gomb). Az alábbiakban röviden ismertetjük (a Súgó alapján) a beállításokat.

Max. idő szerkesztődoboz. A megoldáskeresés idejét korlátozza. Bár legnagyobb lehetséges értéke 32 767, a 100 másodperc, mint alapérték, a legtöbb kis problémánál megfelel.

Lépésszám szerkesztődoboz. A megoldási eljárás idejét a köztes számolási lépések számának korlátozásával határolja be. Bár legnagyobb lehetséges értéke 32 767, a 100 lépés, mint alapérték, a legtöbb kisméretű problémánál megfelel.

Pontosság szerkesztődoboz. A megoldás pontosságát szabályozza; a beírt szám dönti el, hogy egy feltételcella értéke egyenlő-e a célértékkel, illetve, hogy kielégíti-e az alsó vagy felső korlátot. A pontosság egy 0 (nulla) és 1 közé eső törtszám.

Tűrés szerkesztődoboz. Azt szabja meg, hogy hány százaléknyi hibát engedünk meg az optimális megoldáshoz képest, ha a probléma korlátozó feltétele egész szám. Magasabb tűrés általában felgyorsítja a megoldási folyamatot.

Konvergencia szerkesztődoboz. Ha az utolsó öt iteráció során a célcella értékének relatív változása kisebb, mint a Konvergencia mezőben megadott érték, a Solver leáll. A konvergencia csak nemlineáris problémák esetén használható, értékét egy 0 (nulla) és 1 közé eső törtszámmal kell megadni. Minél több tizedesjegyet tartalmaz a begépett szám, annál kisebb a konvergencia; például 0,0001 kisebb relatív változást jelöl, mint 0,01. Minél kisebb a megadott érték, annál tovább tart a megoldás keresése.

Lineáris modell feltételezése jelölőnégyzet. Gyorsabb a megoldási folyamat, ha a modell minden összefüggése lineáris, tehát egy lineáris optimalizálási problémát kell megoldanunk.

Kijelzés lépésenként jelölőnégyzet. Megszakítja a Solver működését, hogy megmutassa minden egyes lépés eredményét.

Nagyságrendek felismerése jelölőnégyzet. Bekapcsolja az automatikus léptékváltást, ha a bemeneti és kimeneti értékek nagyságrendekkel különböznek egymáshoz képest - például amikor a nyereség (százalék) a beruházások összegétől (millió forint) függ.

Nemnegatív feltételezése jelölőnégyzet. Bekapcsolása esetén a Solver 0 (nulla) értéket állít be alsó korlátnak azoknál a módosuló celláknál, amelyekhez nem adtunk meg alsó korlátot a **Korlátozó feltétel felvétele** párbeszédpanel **Korlátozó feltétel** mezőjében.

Közelítés keret. Az egydimenziós keresési eljárásokban az alapváltozók kezdeti értékének becslésére használt módszert (négyzetes vagy érintőmódszer) határozza meg.

Differenciák keret. Azt határozza meg, hogy a cél- és a feltételfüggvények parciális deriváltjainak becslésére a Haladó vagy Centrális differenciálás módszerét használja-e a program.

Keresés keret. Azt adja meg, hogy a Newton vagy a Konjugált algoritmust használja-e, amikor a közelítő lépéseknél eldönti a keresési irányt.

Newton Egy kvázi Newton módszer, amely általában több memóriát, de kevesebb közelítő lépést használ, mint a konjugált módszer.

Konjugált Kevesebb memóriát használ, mint a Newton módszer, de általában több közelítő lépésre van szüksége az adott szintű pontosság eléréséhez. Próbáljuk ki ezt a módszert, ha a probléma nagyméretű és a memória korlátozott, vagy ha a közelítő lépések során az eredmények túl lassan változnak.

Modell betöltése gomb. Megjeleníti a **Modell betöltése** párbeszédpanelt, ahol megadhatjuk, melyik modellt szeretnénk betölteni.

Modell mentése gomb. Megjeleníti a **Modell mentése** párbeszédpanelt, ahol megadhatjuk, hova szeretnénk a modellt menteni. Csak akkor kattintsunk ide, ha a munkalappal együtt egynél több modellt szeretnénk menteni - az első modell mentése automatikus.

Néhány szó a jelentésekről

A Solver meghatározott típusú jelentéseket készít, és mindegyiket a munkafüzet külön lapján helyezi el. Az alábbiakban (a Súly alapján) röviden ismertetjük az egyes jelentés típusokat. Amennyiben az Olvasót a leírtaknál mélyebben érdeklik a jelentések, érdemes elolvasnia az **Excel közzgazdászoknak című** (ajánlott) könyv 22. fejezetét (is).

Eredmény jelentés. Felsorolja a célcellát és a módosuló cellákat eredeti és végső értékeikkel, korlátozó feltételeikkel és a feltételekre vonatkozó információkkal együtt.

Érzékenység jelentés Információt ad arról, mennyire érzékeny a megoldás a **Solver paraméterek** párbeszédpanel **Célcella** mezőjében lévő képlet vagy a korlátozó feltételek kis változásaira. Integer korlátokkal rendelkező modellek esetében a Microsoft Excel nem készít ilyen jelentést. A nemlineáris modellek a jelentésben a redukált gradiensek és a Lagrange multiplikátor értékét is feltüntetik. Lineáris modelleknél a jelentés tartalmazza a csökkentett költségeket, az árnyékárakat, az objektív koefficienseket (a megengedhető növekedés és csökkenés értékével), valamint a korlátozó feltétel jobb oldali tartományait.

Határok jelentés. Felsorolja a célcellát és a módosuló cellákat elfogadható értékeikkel, alsó és felső határaikkal és célértékeikkel együtt. Integer korlátokkal rendelkező modellek esetében a Microsoft Excel nem készít ilyen jelentést. Az alsó határ a módosuló cella által felvehető legkisebb érték, feltéve, hogy az összes több módosuló cella értéke rögzített, és a feltételek mind teljesülnek. A felső határ a módosuló cella által felvehető legnagyobb érték.

Az Esetvizsgáló

Az **Eszközök** menü **Esetvizsgáló** parancsával látványos összesítő jelentéseket készíthetünk. Segítségével például kiszámítható üzleti hasznunk, különböző változó feltételek függvényében. Ilyen változó lehet például a szolgáltatásokért fizetett összeg.

Az Esetvizsgáló által megvalósított funkciót szokásos a szakirodalomban Forgatókönyvnek (Scenario) is nevezni. Az Esetvizsgáló végeredménye egy szépen formázott táblázat, amely különböző eseteket jelenít meg.

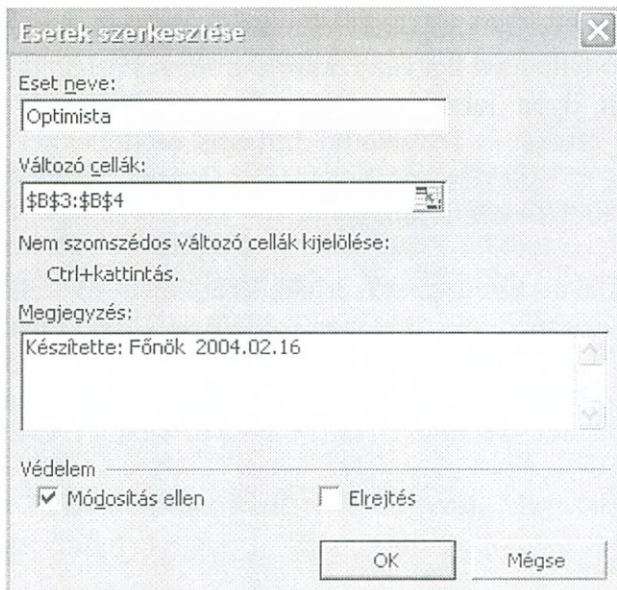
Mintafeladat összesítő jelentés készítésére

1. Nyissuk meg a **scenario.xls** állományt.

Bruttó_nyere...		=B3-B4					
	A	B	C	D	E	F	
1	Üzleti terv						
2		2005					
3	Bruttó bevétel	7 500 000 Ft		Elsődleges változó			
4	Eladott áruk beszerzési értéke	1 771 000 Ft		cellák			
5							
6	Bruttó nyereség	5 729 000 Ft					
7							
8		Cella	A képlet				
9		\$B\$6	=B3-B4				

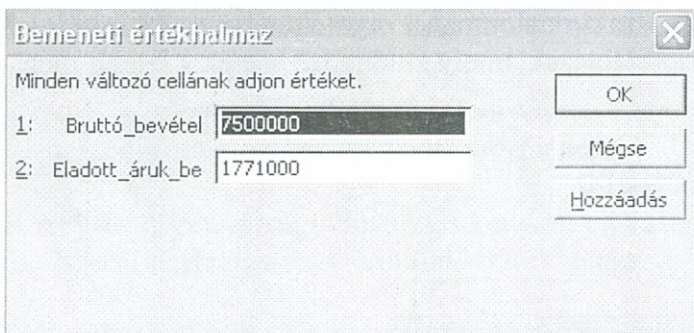
171. ábra Táblázat a mintafeladathoz (SCENARIO.XLS)

2. A B oszlop celláit nevezzük el a tőlük balra álló címkék segítségével.
3. Adjuk ki az **Eszközök** menü **Esetvizsgáló** parancsát. Kattintsunk a párbeszédablak **Felvesz** gombjára. A 172. ábra szerint állítsuk be az **Esetek szerkesztése** nevű párbeszédablakot. Majd kattintsunk az **OK** gombra.

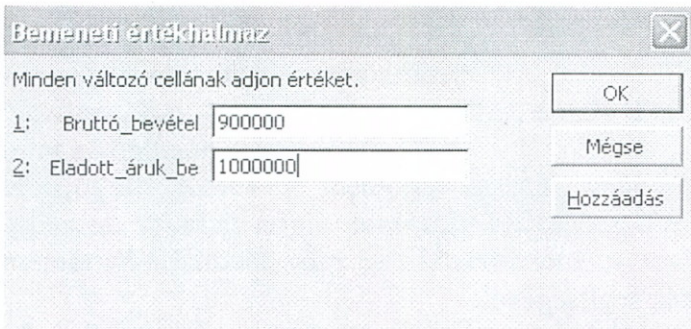


172. ábra Az első eset rögzítése

4. A **Bementi értékhalmoz** párbeszédablak jelzi a két bemeneti változó aktuális értékét (B3, B4 cellák).

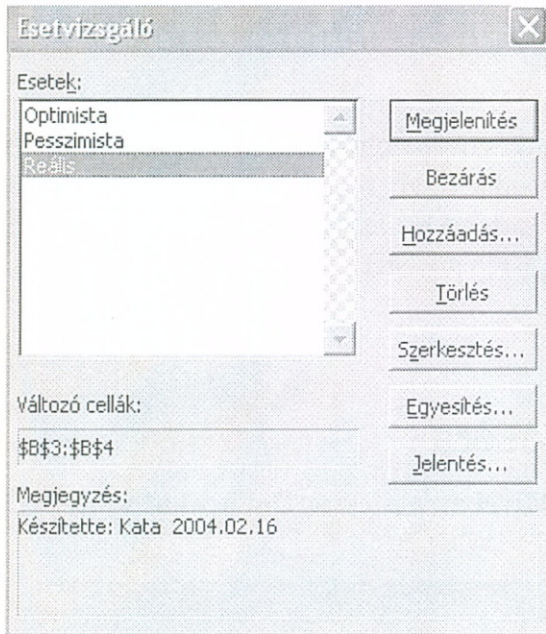


5. A felső szövegmezőbe gépeljük 9 000 000 –t, míg az alsóba 1 000 000-t.



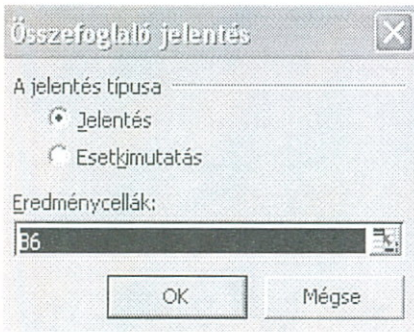
173. ábra Változó cellák értéke

6. Kattintsunk a **Hozzáadás** gombra, ekkor első esetünk rögzítésével (Optimista) végeztünk.
7. Az eddig ismertettek alapján, rögzítsünk még két esetet, egy pesszimistát és egy reálisnak tűnőt.
8. Miután mind a három esetet rögzítettük visszajutunk az **Esetvizsgáló** párbeszédablakába. Kattintsunk annak **Jelentés** nevű gombjára.



174. ábra Az eseteket rögzítettük

9. Az **Összefoglaló jelentés** párbeszédablakban az **Eredménycellák** nevű bevitelmezőben megjelenik a B6-os cella címe, az Excel tehát detektálta azon cella címét, amelyben a minket érdeklő függvény (képlet) van.



10. Kattintsunk az OK gombra.

11. A következő ábra mutatja az elkészült összehasonlító jelentést.

	1	2	A	B	C	D	E	F	G	H
	1									
	2									
	3									
	4									
	5									
	6									
	7									
	8									
	9									
	10									
	11									
	12									

Összehasonlító jelentés					
	Aktuális érték:	Optimista	Pessimista	Reális	
Módosuló cellák:					
Bruttó_bevétel	7 500 000 Ft	9 000 000 Ft	5 000 000 Ft	7 000 000 Ft	
Eladott áruk beszerzési értéke	1 771 000 Ft	1 000 000 Ft	1 771 000 Ft	2 000 000 Ft	
Eredménycellák:					
Bruttó nyereség	5 729 000 Ft	8 000 000 Ft	3 229 000 Ft	5 000 000 Ft	

Megj: Az Aktuális értékek oszlopban az Összefoglaló jelentés készítése idején a változó cellákban lévő értékek látszanak. Az egyes esetekre vonatkozó cellák szürke háttérrel lettek kiemelve.

175. ábra Az elkészült összehasonlító jelentés

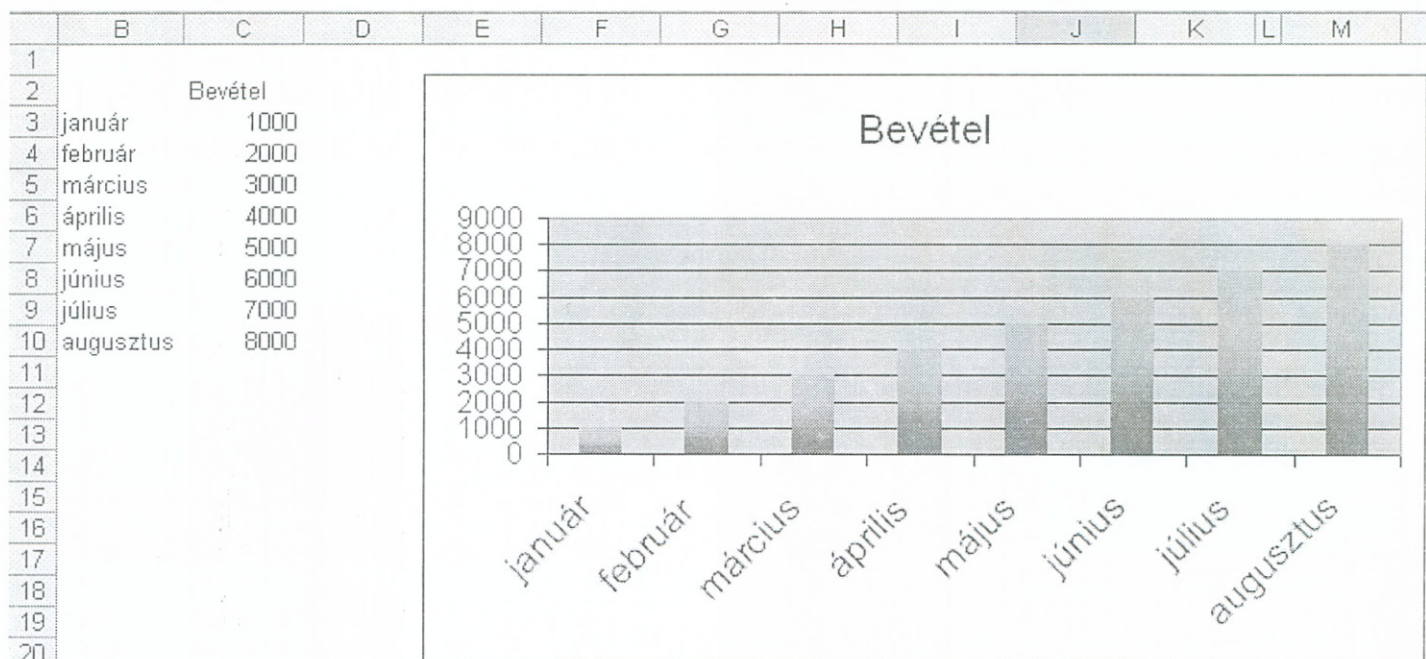
4. fejezet: Szemelvények a haladó diagramkészítési lehetőségek tárgyköréből

Jegyzetünk ezen fejezetében bemutatunk néhány haladó diagramkészítési lehetőséget. Fejezetünk elején bemutatjuk az értékjelzők grafikára történő cseréjének módját. A második részben egy igen hasznos technikát ismertetünk meg az Olvasóval, nevezetesen a Trendvonal diagramba történő beillesztésének módját. A Fejezet végén röviden ismertetjük a diagramok Mi van akkor, ha... (What if...) lehetőségét.

Értékjelzők grafikára cserélése

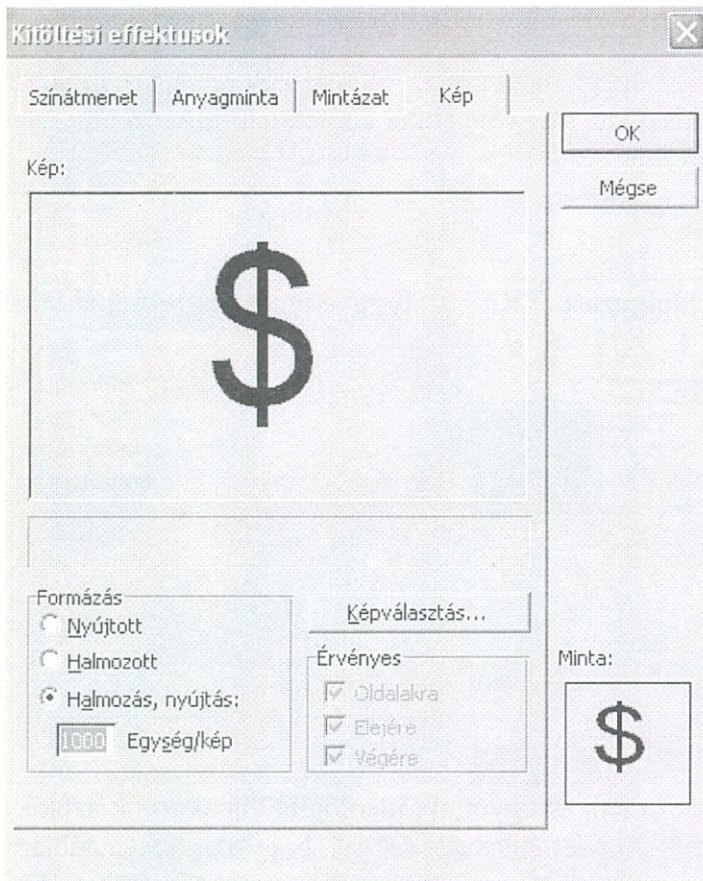
Mintafeladat értékjelzők cseréjének bemutatására

Tegyük fel elkészült diagramunk és unalmasnak találjuk azt. Mivel az oszlopok Dollárokat mutatnak, jobban néznének ki az oszlopok helyett, Dollár jelek.



176. ábra A szépítendő diagram

1. Nyissuk meg az **ertekjelzok.xls** állományt.
2. Gépeljük be egy dollár jelet egy üres oszlop tetszőleges cellájába. Ez a cella legyen aktív. Állítsuk be az oszlop szélességét ideálusra.
3. A **SHIFT** gombot nyomva kattintsunk a **Szerkesztés** menüre.
4. A legördülő parancsok közül válasszuk a **Kép másolása** parancsot.
5. A párbeszédablakban kattintsunk az **OK** gombra. A másolás megtörténik.
6. A diagramban kattintsunk bármelyik oszlopra, ezzel az összeset egyszerre kijelöltük.
7. Adjuk ki a beillesztésre szolgáló billentyűkombinációt (**CTRL+V**). Ezzel az értékjelzők cseréje megtörtént.
8. Ne fejezzük itt be. Kattintsunk bármelyik oszlopra jobb gombbal és a megjelenő rövidmenüből válasszuk az **Adatsorok formázása** parancsot. A megjelenő összetett párbeszédablakban kattintsunk a **Mintázat** szegélyre (ha nem ez lenne a kiválasztott).
9. Kattintsunk a **Kitöltési effektusok** gombra. Kattintsunk a **Kép** fülre (ha nem ez lenne aktív). A megjelenő párbeszédablakot állítsuk be a következő ábra szerinti módon:

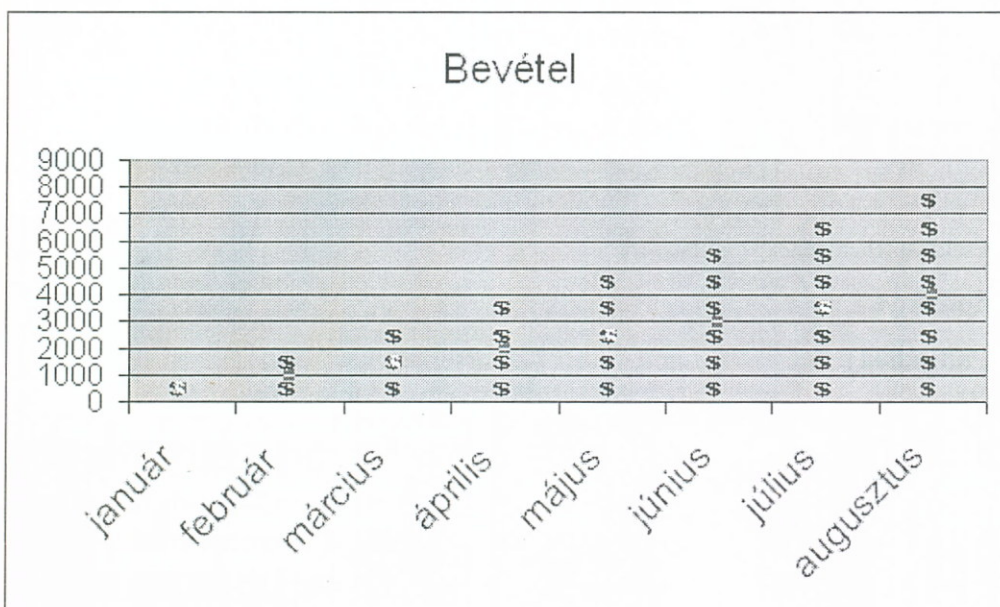


177. ábra A kép tulajdonságainak beállítása

10. Kattintsunk az OK gombra, majd a most megjelenő párbeszédablak OK gombjára. Az Excel máris lecserélte az értékjelzőket olyan dollárt szimbolizáló kis ábrákra, ahol egy-egy kis dollár jel ábra a függőleges tengelyen 1000 egységnyinek felel meg.

- ✦ Érdeemes tudni, hogy az értékjelző természetesen lecserélhető bármilyen más alkalmas grafikára is, ezen esetben talán érdekesebb az Excel **Beszúrás** menüjének **Kép** parancsát használni a **Másolás** parancs helyett.

A fentiek alapján szépített diagramot mutatja a következő ábra.



178. ábra A lecserélt értékjelzők

Trendvonal diagramba történő beszúrása

A trendszámítást az idősorokban lévő tartós irányzat meghatározására, előrejelzésére használjuk. Itt emlékeztetjük az Olvasót, ezen jegyzetre, annak is a Statisztikai függvényekkel foglalkozó alfejezetére, ahol bemutattuk a TREND nevű függvényt, amely lineáris trend függvény.

Mintafeladat trendvonal beszúrására

Feladatunk legyen a hivatkozott fejezet részben kidolgozott TREND függvényen alapuló példa. A következő ábra mutatja a táblázat jelenlegi állapotát.

F3		=TREND(\$B\$3:\$B\$9;\$C\$3:\$C\$9;\$D\$3;E3)					
	B	C	D	E	F	G	H
2	ismert_y	ismert_x	új_x	konstans	Trend		
3	10,1	1997	2004	HAMIS	13,11386		
4	11	1998		IGAZ	17,04286		
5	12,2	1999					
6	13,1	2000					
7	14,2	2001					
8	15	2002					
9	16	2003					
10							

179. ábra Lineáris trend (TREND.XLS)

Ha a fenti ábrán látható információkat szavakba akarjuk önteni és egyúttal „ideológiát” is akarunk azokhoz gyártani, akkor azt mondhatjuk, hogy kiszámoltuk lineáris trendet feltételezve azt, hogy cégünk a táblázat adatai alapján 2004-ben 17,04 millió Forint nyereséget ér majd el. A számításhoz figyelembe vettük cégünk nyereségi adatait 1997-től kezdve 2003-ig bezárva.

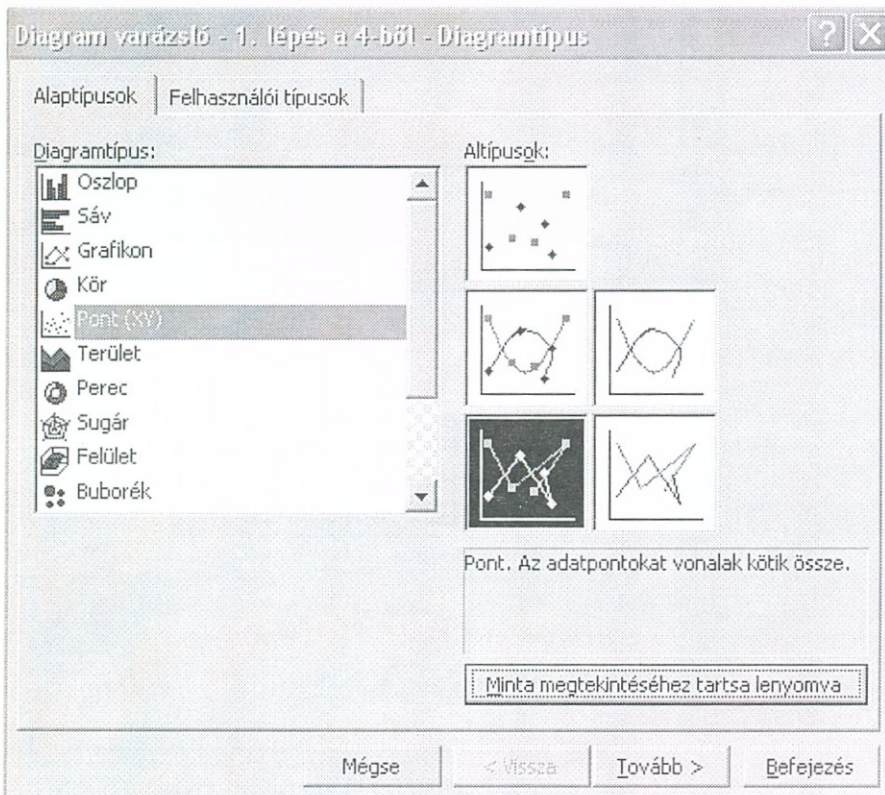
Most azt szeretnénk bizonyítani, hogy mindez a diagramok trendvonal beszúrási lehetőségével is kiszámítható. Ne feledjük, trendvonal térbeli diagramba nem szúrható be.

1. A **trend.xls** állományt nyissuk meg, a diagram alapját képező táblázat a C15:D22 cellatartományban található (180. ábra).

D16		=B3	
	C	D	E
15	ismert_x	ismert_y	
16	1997	10,1	
17	1998	11	
18	1999	12,2	
19	2000	13,1	
20	2001	14,2	
21	2002	15	
22	2003	16	

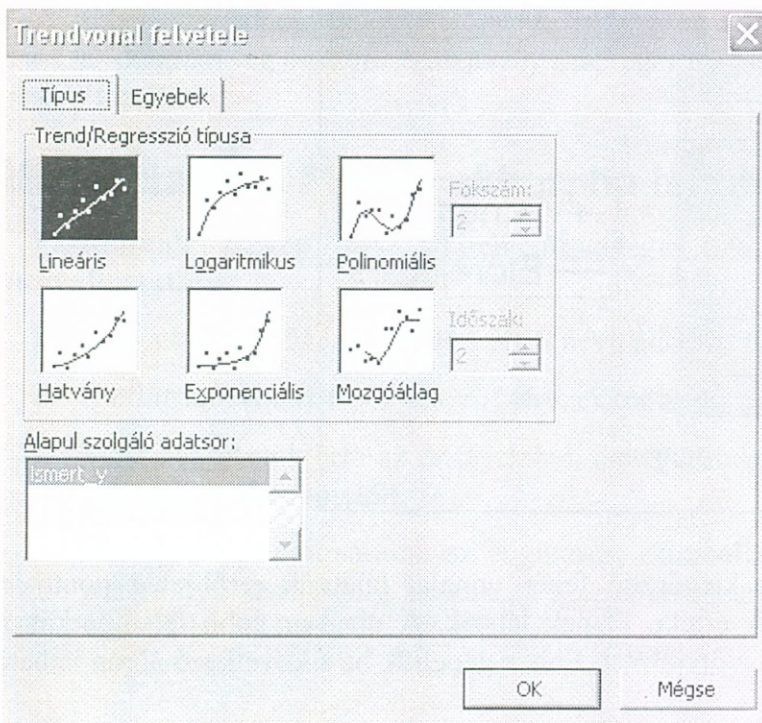
180. ábra Táblázat a diagramok trendvonal beszúrási lehetőségének bemutatására

2. Jelöljük ki a C15:D22 cellatartományt.
3. Készítsünk X-Y diagramot (3. altípusút).



181. ábra Az Altípus kiválasztva

- Kattintsunk valamelyik értékjelzőre jobb gombbal (az összes kijelölt lesz), erre az adatsorra illeszti majd az Excel a trendvonalat. A megjelenő rövid menüből válasszuk a **Trendvonal felvétele** parancsot.



182. ábra Válasszuk ki a trendvonal típusát

- Olyan trendvonal típust válasszunk, amely szerintünk a lehető legjobban megközelíti az adataink között lévő tendenciát. Ezért a fenti ábrán látható **Trendvonal felvétele** párbeszédablakban válasszuk a **Lineáris** ablakot.
- Mivel egyéves előrejelzést akarunk adataink alapján adni, kattintsunk a párbeszédablak **Egyebek** fülére. A most megjelenő párbeszédablakot állítsuk be a következő ábra alapján. Mint az látható (**Előre** léptetőjének **1** értékével), egyéves előrejelzést kérünk, valamint az **Egyenlet látszik a diagramon** és az **R^2 értéke látszik a diagramon** jelölőnégyzeteket is bekapcsoltuk.

Trendvonal felvétele

Típus Egyebek

Trendvonal neve

Automatikusan: Lineáris (ismert_y)

Megadott:

Előrejelzés

Előre: egység

Vissza: egység

Legyen a metszéspont =

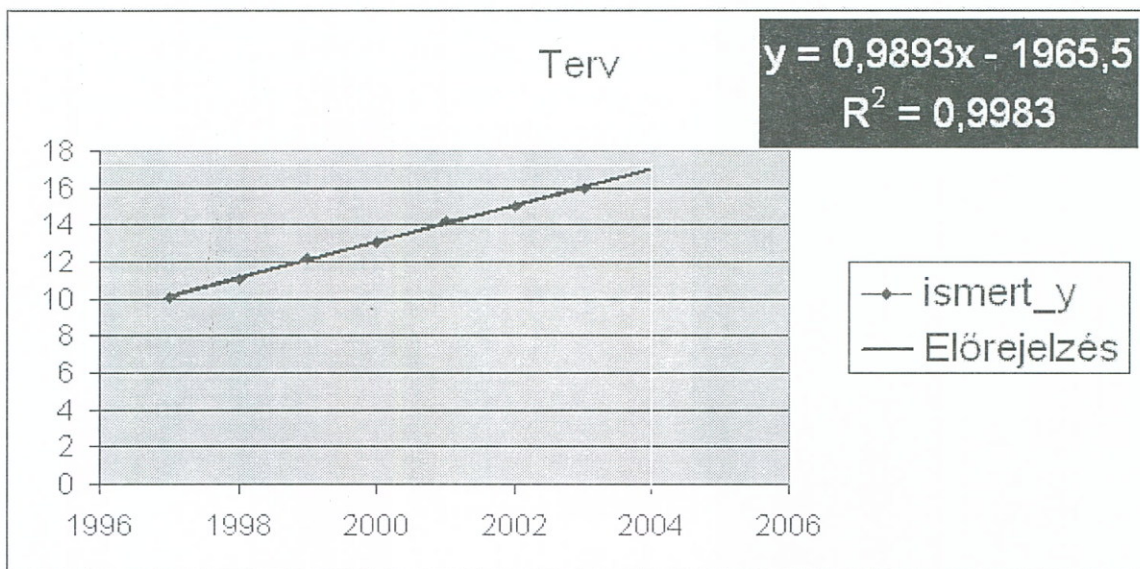
Egyenlet látszik a diagramon

R-négyzet értéke látszik a diagramon

OK Mégse

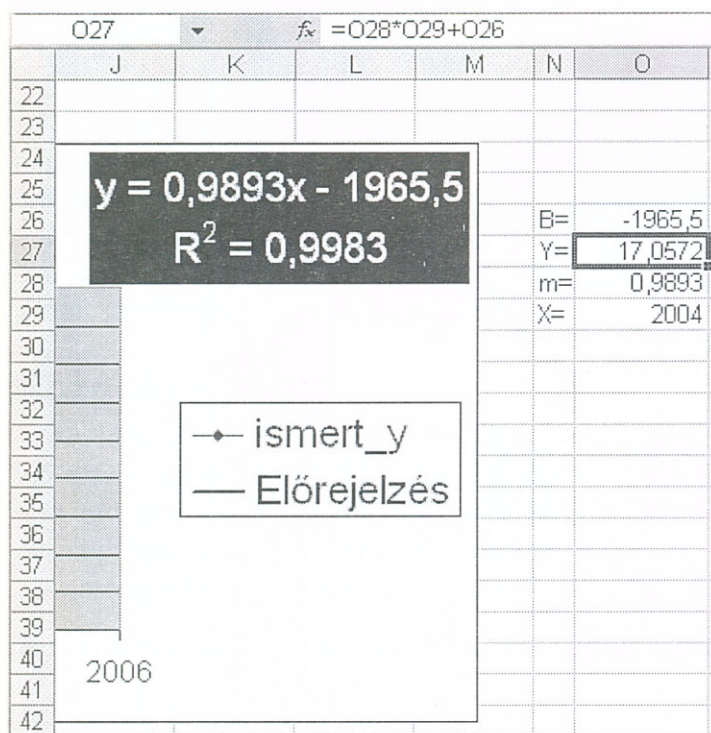
183. ábra Egyéves előrejelzést adunk

7. Kattintunk az **OK** gombra. Az elkészült diagram, a beszúrt trendvonal a következő ábrán látható.



184. ábra Diagram a beszúrt lineáris trendvonallal

8. Feladatunkat megoldottuk. A diagramba két kiegészítő fehér vonalat húztunk erről elég pontosan leolvasható a 2004. évi előrejelzés, amelynek értéke 17-nek látszik. A diagram jobb felső sarkában látható képlet alapján számítsuk ki a 2004. évi előrejelzést. Ehhez gépeljük be a következő ábrán látható adatokat és a képletet.



185. ábra Előrejelzés a diagramból nyert képlet alapján

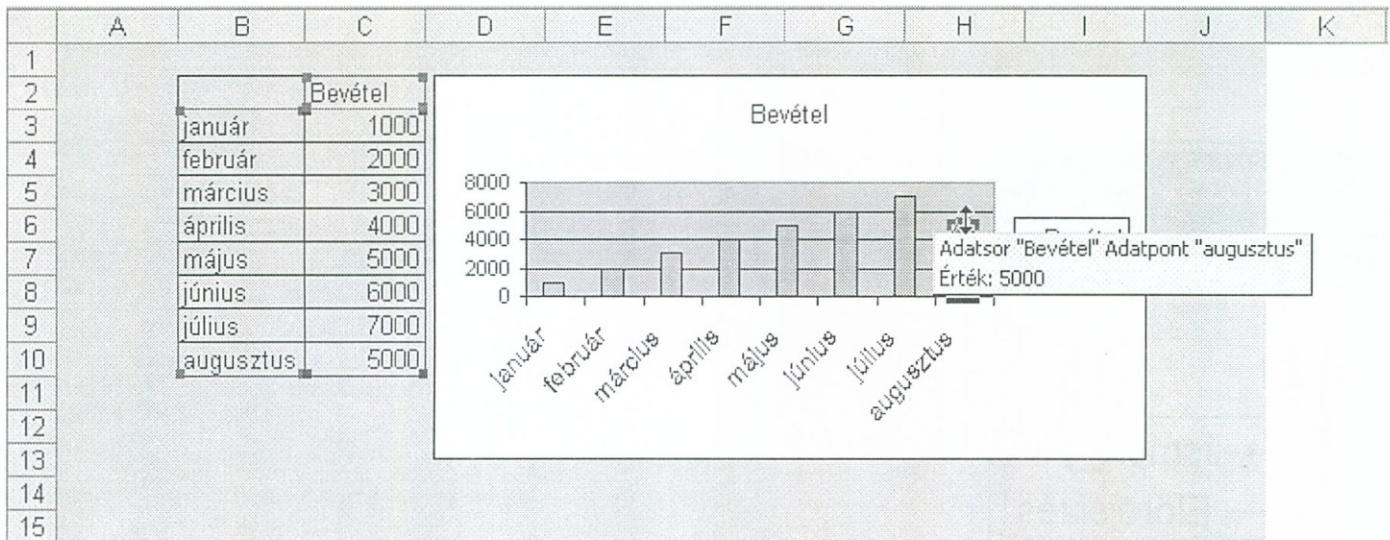
Mi van akkor, ha?

A síkbeli diagramok **Mi van akkor, ha** (What if) lehetősége egy hasznos tervezési eszköz. Ez nem más, mint a diagram egy adott értékjelzőjének egérrel történő megragadása, majd ezt követően értékének megváltoztatása

Mintafeladat a Mi van akkor, ha lehetőség szemléltetésére

Feladatunk legyen egy síkbeli függőleges oszlopdiagram egy adott értékjelzője értékének megváltoztatása.

1. Nyissuk meg a **MI_VAN_HA.XLS** nevű állományt.
2. Kattintsunk annak egy értékjelzőjére, ekkor az adatsorhoz tartozó összes értékjelző kijelölt lesz.
3. Most kattintsunk arra az értékjelzőre, amelyet meg akarunk változtatni, ez legyen az augusztus. Ekkor már csak ez lesz kijelölt.
4. Az egérrel pozícionáljunk az augusztust szimbolizáló oszlop tetejére, az egérkurzor alakja ekkor két irányú nyíl lesz. Addig húzzuk az egérrel az oszlopot, míg annak értéke az elvárt nem lesz. Az értéket állítsuk be 5000-re.



186. ábra Augusztust, 5000-re állítottuk

Ezzel feladatunkat elvégeztük.

- ✦ Vegyük észre, hogy miközben az értékjelzőt egérrel állítjuk, az ahhoz tartozó cellában annak aktuális értéke látható.

5. fejezet: Analitikai eszközkészlet

Jegyzetünk ezen fejezetében bemutatjuk az Excel adatelemző csomagját az **Analysis ToolPak**-et. Nem célunk a teljes csomagot ismertetni, hanem csak ízelítőként két módszert. Az egyik témakörünk a **Mozgóátlag**, míg a másik a **Hisztogram** lesz. A Sűgóban a **Statisztikai elemző eszközök** témakör tárgyalja az adatelemző módszereket.

A csomag által kínált módszerek az **Eszközök** menü **Adatelemzés** parancsán keresztül érhetők el. Amennyiben nem találjuk a parancsot az **Eszközök** menüben, adjuk ki az **Eszközök** menü **Bővítőmenykezelő** parancsát és a megjelenő **Bővítőmenykezelő** ablakban kapcsoljuk be az **Analysis ToolPak** jelölőnégyzetet.

Mozgóátlag

Az idősorok elemzésének egyik hasznos módszere a mozgóátlagolás. Mozgóátlagolással olyan részleteket tudhatunk meg a trendről, amelyek az adatok átlagolásával ki sem derülhetnének.

A mozgóátlagolás nem más mint az idősor egy adott elemének, adott környezetében található elemek átlaga. Az általában használt legkisebb elemszám a 3.

A mozgóátlag kiszámításra szolgáló összefüggés a következő ábrán látható:

$$F_{(t+1)} = \frac{1}{N} \sum_{j=1}^N A_{t-j+1}$$

ahol:

- N a mozgó átlag periódusainak száma
- A_j a tényleges érték a j időpontban
- F_j a becsült érték a j időpontban

187. ábra A Mozgóátlag számítás kiszámításának alapképlete

1. Mintafeladat mozgóátlagolásra

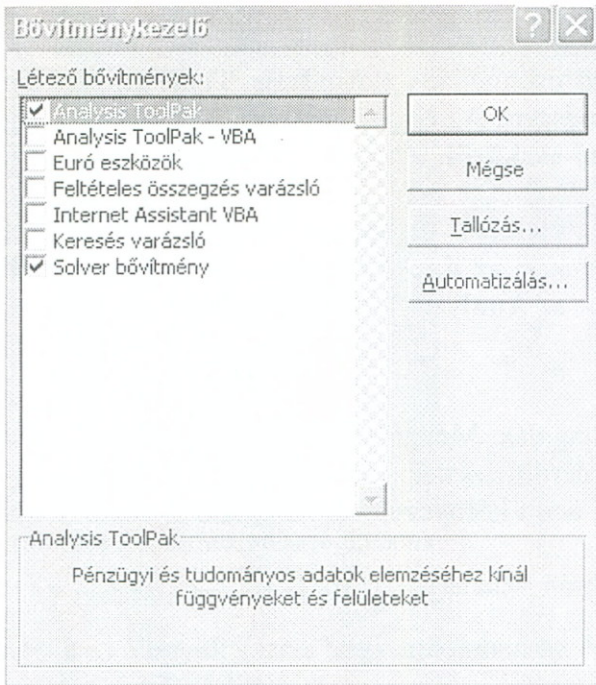
Feladatunk két év adatainak mozgóátlagolású ábrázolása diagramon.

1. Készítsük el az alábbi táblázatot tartalmazó Excel állományt (188. ábra).

	A	B	C	D	E
1			aktuális	becsült	szórás
2	2003	1. n. év	420		
3	2003	2. n. év	650		
4	2003	3. n. év	800		
5	2003	4. n. év	1420		
6	2004	1. n. év	1360		
7	2004	2. n. év	1600		
8	2004	3. n. év	2110		
9	2004	4. n. év	2400		

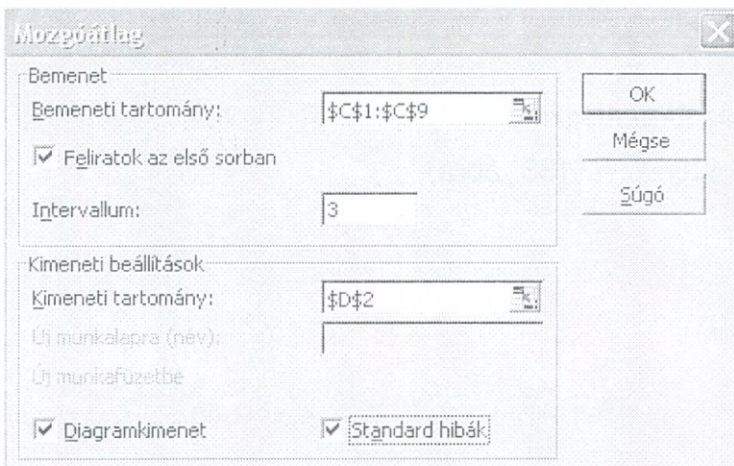
188. ábra Mintatáblázat a feladathoz (MOZGOATL.XLS)

2. Tegyük fel előállt azon eset, amikor, mivel még nem használtuk az adott gépen az Analysis ToolPak-et, nem találjuk az **Eszközök** menüben az **Adatelemzés** parancsot. Ezért adjuk ki az **Eszközök** menü **Bővítőmenykezelő** parancsát. Kapcsoljuk be az **Analysis ToolPak** jelölőnégyzetét (189. ábra).



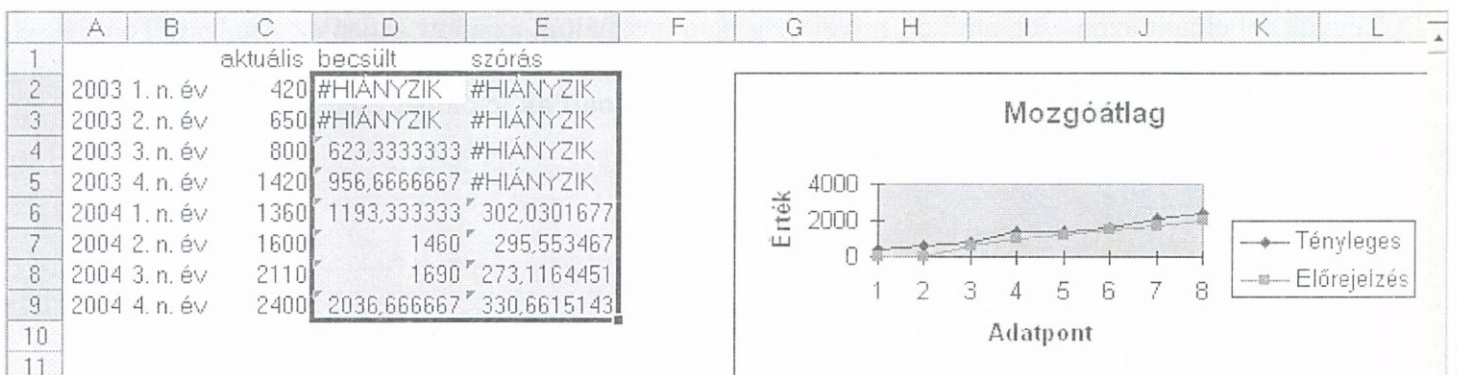
189. ábra A bővítmények listája

3. Kattintsunk az OK gombra. Egy kis idő múltán az Excel elindítja a megfelelő bővítményprogramot (az esetek többségében .XLA fájlokban található a bővítmények) és megjelenik az **Eszközők** menüben az **Adatelemzés** parancs.
4. Adjuk is ki ez utóbbit. Az **Adatelemzés** párbeszédablak listaablakában válasszuk ki a **Mozgóátlag**.
5. Kattintsunk az **OK** gombra.
6. A **Mozgóátlag** párbeszédablakot állítsuk be a 190. ábra alapján:



190. ábra A Mozgóátlagolás beállításai

7. Kattintsunk az OK gombra. Eredményül a következőket kapjuk (191. ábra):



191. ábra A Mozgóátlagolás eredménye

A tervezett feladatot megoldottuk. A témakör lezárásaként röviden ismertetjük a Mozgóátlag párbeszédpanel beállításait (a Súgó alapján).

Bemeneti tartomány. Itt adjuk meg az elemezni kívánt munkalaptartomány hivatkozását. A kiindulási adattartománynak minimum négy cellából álló oszlopot vagy sort kell tartalmaznia.

Feliratok. Amennyiben a bemeneti tartomány első oszlopában vagy első sorában feliratok vannak, kapcsoljuk be ezt a jelölőnégyzetet. Töröljük a jelölőnégyzetet, ha a bemeneti tartomány nem tartalmaz feliratokat. Ebben az esetben az Excel elkészíti az eredménytábla megfelelő feliratait.

Intervallum. Ide gépeljük be a mozgó átlagolásba bevont adatok számát. Alapértelmezett értéke: 3

Kimeneti tartomány. Itt adjuk meg az eredménytartomány bal felső sarkának hivatkozását. Amennyiben a **Standard hibák** jelölőnégyzet be van kapcsolva, az Excel kétszlopos eredménytáblázatot állít elő a Standard hiba értékeivel a jobb oldali oszlopban. Ha a bemenő adatok száma nem elegendő az előrejelzéshez vagy a Standard hiba kiszámítására, az Excel a #HIÁNYZIK hibaértéket adja eredményül.

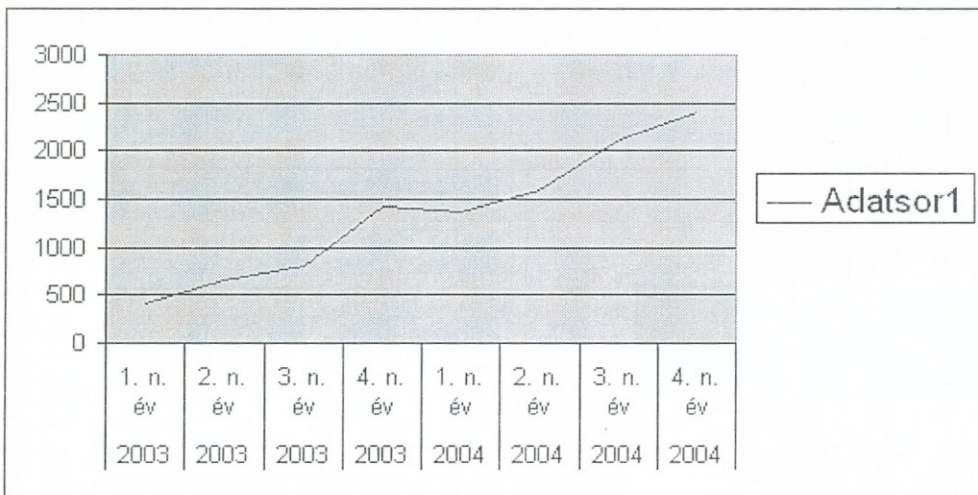
Megjegyzés: Az eredménytartománynak ugyanazon a munkalapon kell lennie, amelyen a bemeneti tartomány adatai vannak. Ezért az **Új munkalapra** és az **Új munkafüzetbe** funkciók nem érhetőek el.

Diagramkimenet. Ha bekapcsoljuk ezt a jelölőnégyzetet, akkor az eredménytáblával együtt egy beágyazott hisztogram-diagramot is kapunk.

Standard hibák. Akkor kapcsoljuk be ezt a jelölőnégyzetet, ha az eredménytábla egyik oszlopában meg szeretnénk jeleníteni a Standard hibákat. Töröljük a jelölőnégyzetet, ha egyoszlopos kimeneti táblázatot szeretnénk kapni, a standard hibák értékei nélkül.

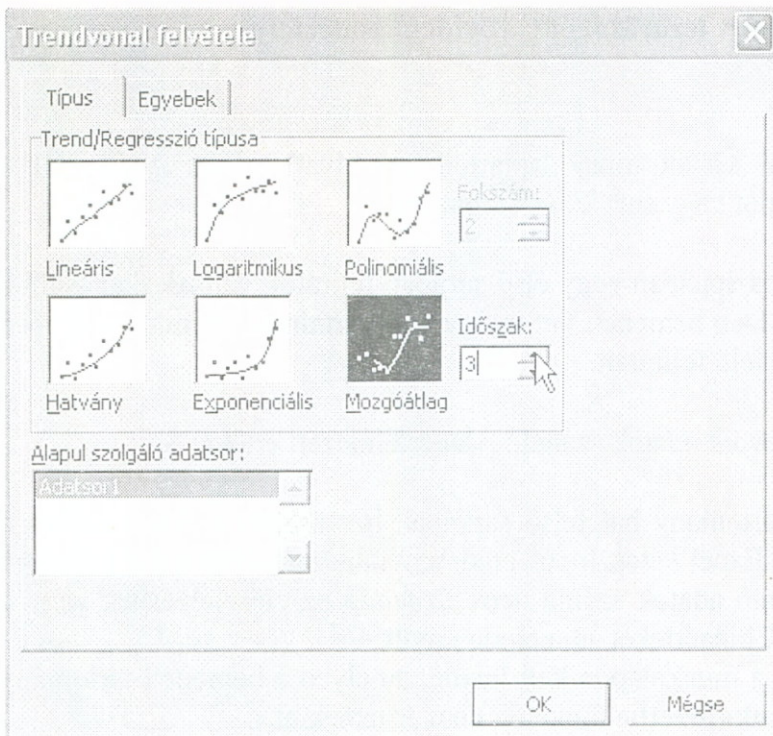
2. Mintafeladat diagrammal végzett mozgóátlagolásra

1. Elkészítettünk a mintatáblázat alapján (mozgoatl.xls) egy Grafikon típusú (vonaldiagram), 1-es altípusú diagramot a mintatáblázat A2:C9 cellatartományából (192. ábra).



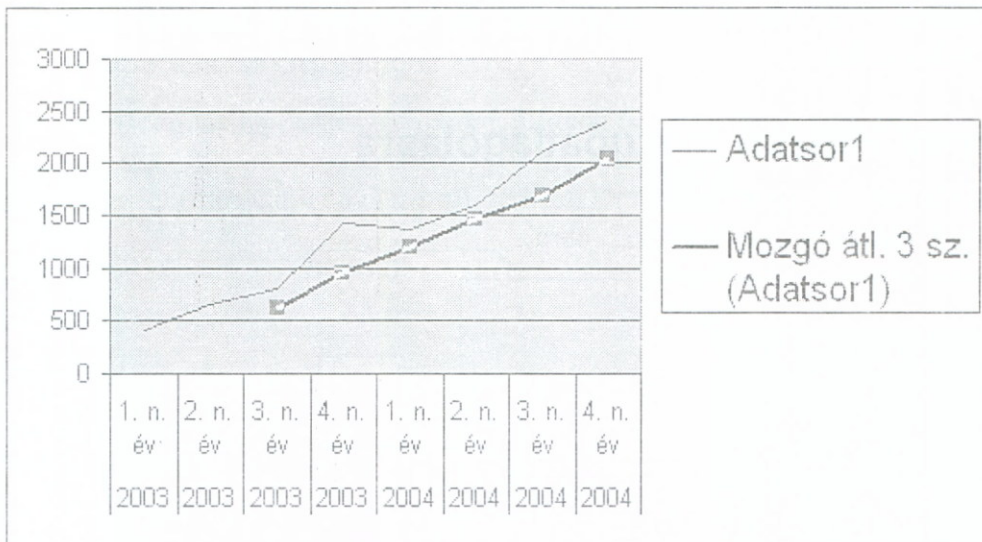
192. ábra Az elkészült vonaldiagram

2. Kattintsunk az egér jobb gombjával a adatsor tetszőleges pontjára. A megjelenő rövidmenüből válasszuk a **Trendvonal** felvétele parancsot. A párbeszédablakot állítsuk be az következő ábra szerint (3 időszakot állítunk be).



193. ábra Mozgóátlagláshoz készülünk

3. Kattintsunk a párbeszédablak OK gombjára.
4. Az elkészült diagram, a beszűrt mozgóátlagolású trendvonallal a következő ábrán látható. Ezzel feladatunkat megoldottuk.



194. ábra Az alsó vonal a beszűrt mozgóátlagolású trendvonal

Hisztogram

A Hisztogram módszerrel egy cellatartomány adatai és az adatkategóriák szerint egyenkénti és halmozott gyakoriságok számoltathatók ki. A módszerrel kiszámítható egy adathalmazban egy érték előfordulásainak száma is.

A hisztogram módszer kipróbálásához használjuk fel a jegyzet első fejezetében a GYAKORISÁG függvényben tanultakat, a gyakorló állomány is legyen ugyanaz (195. ábra).

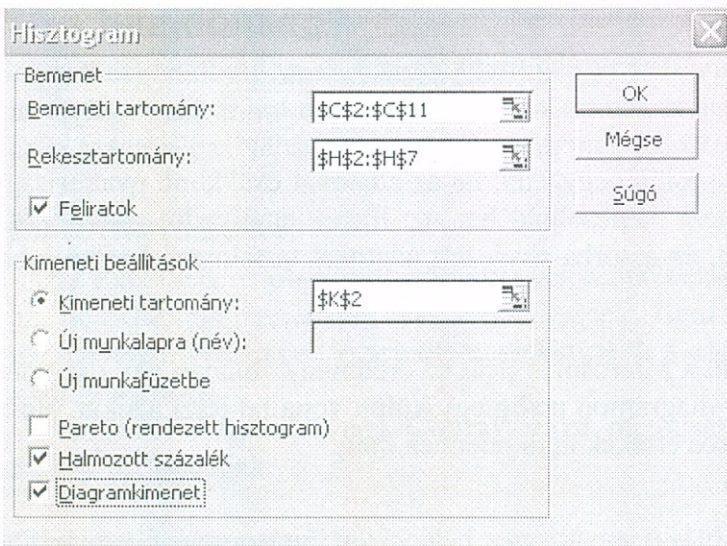
Mintafeladat Hisztogram készítésére

1. Nyissuk meg a **gyakorisag.xls** állományt. A táblázat bizonyára ismerős, ezt a táblázatot használtuk fel akkor is, amikor a **GYAKORISÁG** függvényt bemutattuk.

	A	B	C	D	E	F	G	H	I
1									
2	Sorszám	Név	Pontszám	Adattömb				Csoport_tömb	Gyakoriság
3	1.	Almási Pál	4					5	3
4	2.	Bendegúz Péter	5					6	1
5	3.	Csapodár Vilmos	6					7	1
6	4.	Keres Imre	7					9	2
7	5.	László Péter	8					10	2
8	6.	Másik Éva	9						0
9	7.	Piros Katalin	10						
10	8.	Rémes Zsófia	3						
11	9.	Veremi Gáspár	10						
12									

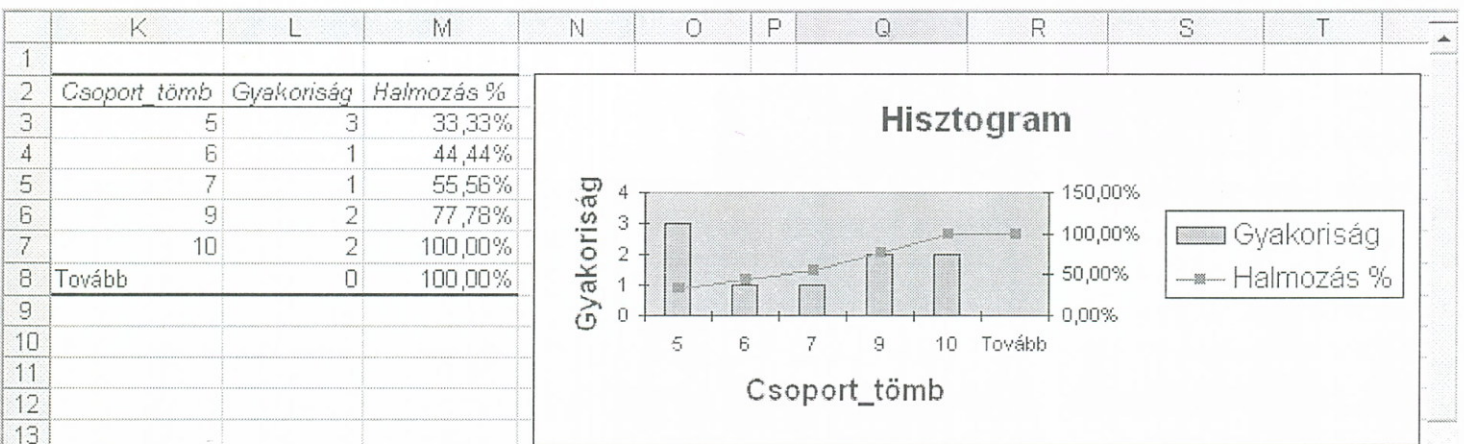
195. ábra Mintaállomány a Hisztogram készítéshez (GYAKORISAG.XLS)

- Adjuk ki az **Eszközők** menü **Adatelemzés** parancsát. A listaablakból válasszuk a **Hisztogramot**.
- A Hisztogram párbeszédablakot állítsuk be a 196. ábra szerint.



196. ábra Beállításaink

- Kattintsunk az **OK** gombra.
- Ténykedésünk eredménye látható a következő ábrán.



197. ábra Az eredmény

Fejezetünk végén röviden ismertetjük a **Hisztogram** párbeszédablak kezelőszerveinek funkcióit a Súgó alapján.

Bemeneti tartomány. Itt adjuk meg az elemezni kívánt munkalaptartomány hivatkozását.

Rekesztartomány (nem kötelező). Adjunk meg egy cellatartományt és tetszőleges határértékeket, amelyek az egyes kategóriákat határozzák meg. A határértékeknek növekvő sorrendben kell lenniük. Az Excel

megszámolja az aktuális és (ha van ilyen) a felülről csatlakozó kategória között lévő adatpontok számát. Az első kategóriaérték alatti és az utolsó kategóriaérték feletti értékeket mindig beleveszi számításába. Ha nem adunk meg kategóriákat, akkor az Excel létrehozza azokat úgy, hogy a legkisebb és a legnagyobb érték között egyenletesen felosztja az adattartományt.

Feliratok. Ha a bemeneti tartomány első sorában vagy első oszlopában vannak feliratok, kapcsoljuk be a jelölőnégyzetet. Kapcsoljuk ki, ha a bemeneti tartomány nem tartalmaz feliratokat. Az Excel el fogja készíteni az eredménytábla megfelelő feliratait.

Kimeneti tartomány. Adjuk meg az eredménytartomány bal felső sarkának hivatkozását. Az Excel automatikusan meghatározza az eredménytartomány méretét, és ha az eredménytartomány várhatóan felülírja a meglévő adatokat, figyelmeztetést ír ki.

Új munkalapra. Ha bekapcsoltuk, új munkalapot illeszt be abba a munkafüzetbe, amelyben az aktuális lap van, és az eredmények az új lap A1 cellájától kezdve kerülnek a lapra. Az új lapnak nevet is adhatunk a mezőben.

Új munkafüzetbe. Ha bekapcsoltuk, új munkafüzetet hoz létre egy új munkalappal, és az eredményeket annak a lapnak az A1 cellájától kezdve helyezi el.

Pareto (rendezett hisztogram). Kapcsoljuk be a Pareto jelölőnégyzetet, ha az adatokat csökkenő gyakorisági sorrendben szeretnénk látni. Ha a jelölőnégyzetet nem kapcsoljuk be, az Excel emelkedő gyakorisági sorrendben helyezi el az adatokat az eredménytáblában, és a sorba rendezett adatokat tartalmazó három jobb oldali oszlopot nem jeleníti meg.

Halmazott százalék. Ha a jelölőnégyzetet bekapcsoljuk, a Microsoft Excel az eredménytáblában egy oszlopot készít a halmazott százalékos értékeknek, a hisztogram-diagramon pedig egy külön vonallal jelzi azokat. Ha a jelölőnégyzetet nem kapcsoljuk be, a halmazott százalékos értékek nem jelennek meg.

Diagramkimenet. Ha bekapcsoljuk, az eredménytáblával együtt egy beágyazott hisztogram-diagram fog automatikusan megjelenni.

4. rész: Az Excel programozásának alapjai

1. fejezet: Alapok

Mi az objektum, objektumgyűjtemény (collection)?

Az Excel minden alkotóelemét egy-egy objektum alkotja. Objektum egy munkafüzet (Workbook) egy munkalap (Worksheet), de objektum maga az Excel is (Application).

Az Excel programozásakor ezen objektumokat kell kezelnünk. Mivel egyszerre akár több munkafüzet is nyitva lehet, ezért ezek a több példányban előforduló azonos típusú objektumok gyűjteményt (Collection) alkotnak.

Mind az objektumok, mind a gyűjtemények tulajdonságokkal (Properties) és metódusokkal (Method) egyaránt rendelkeznek.

Mi a tulajdonság?

A Workbook-nak van szerzője (**Author**). A Worksheet-nek van neve. A cellának van magassága, szélessége és értéke. Ezek mind tulajdonságok (Properties). Néhány tulajdonság megváltoztatása könnyű, ilyen például a cella magassága. Néhány tulajdonság nem megváltoztatható.

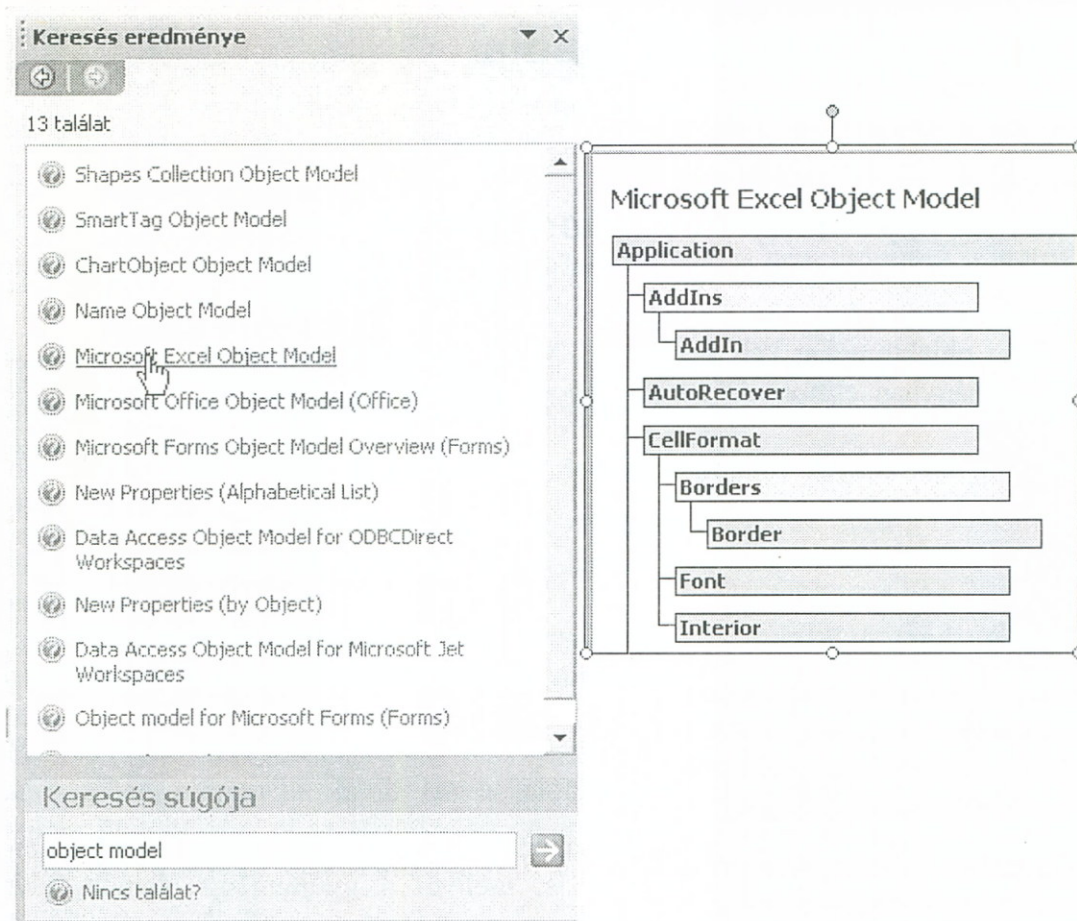
Mi a metódus?

Az Excelben, az objektumgyűjtemények rendelkeznek egy **Add** metódussal, mellyel új elemet lehet felvenni az objektumgyűjteménybe. De rendelkeznek **Item** metódussal is, mellyel a gyűjtemény elemeit lehet megnevezni. Az Excel **Worksheet** objektumok **Calculate** metódussal rendelkeznek, mellyel a cellák újból számíttathatóak.

Az Excel diagramok egy **ChartWizard** metódussal rendelkeznek, mellyel a diagram jellemzői gyorsan megváltoztathatóak.

Objektummodell

Az Excel objektumai hierarchiába szervezettek, ezt a hierarchiát Objektummodellnek nevezik (Object model), amely a Súgóban megjelenik és annak segítségével bejárható.



Az objektumok egymás alá rendelve, (bizonyos körülmények között) hasonló hatása van az alábbi parancsok mindegyikének:

Workbooks(1).Worksheets(1).Range("a1").Value=1

ActiveWorkbook.Worksheets(1).Range("a1").Value=1

Worksheets(1).Range("a1").Value=1

ActiveSheet.Range("a1").Value=1

Range("a1").Value=1

Range("a1")=1

Selection.Value=1

-ha az első munkafüzet aktív

-ha az első munkalap aktív

-ha az első munkalap aktív

-ha az első munkalap aktív

-ha az aktív cella az A1.

Direkt kódbevitel

Nagyszerű lehetőség a programozás alapjainak elsajátítására az **Immediate** párbeszédablakba történő kódbevitel. Miután elindítottuk az Excelt és **ALT+F11**-gyel átváltottunk a VBE ablakra, adjuk ki a **View** (Nézet) menü **Immediate** (Próbaablak) parancsát. A parancsokat begépelve a **Immediate** ablakba és **ENTER**-t nyomva azok végrehajtódnak (attól függően, hogy a kurzor melyik sorban áll, **ENTER**-re a kurzor pozícióban lévő parancs végrehajtódik). A következőkben bemutatunk egy a kódbevitel gyakorlására szolgáló feladatsort.

① Gépeljük be:

workbooks.Add

és nyomjunk **ENTER**-t.

A **Workbooks** a jelenleg használatban lévő munkafüzetek gyűjteménye, a parancs ezen gyűjteményhez hozzáad egy új üres munkafüzetet.

② Gépeljük be:

?workbooks.Count

és nyomjunk **ENTER**-t.

A **Count** tulajdonság segítségével egy adott gyűjtemény (most a **Workbook**-ok) elemeinek számát kérdezhetjük le.

③ Gépeljük be:

workbooks.Close és nyomjunk ENTER-t.

Ezzel az utasítással egyszerre zárjuk be a Workbooks gyűjtemény összes elemét (fájlt).

Lépjünk vissza kurzorral az első utasításra (workbooks.Add), segítségével nyissunk meg hat üres Workbook-ot.

④ Gépeljük be:

Activeworkbook.Close és nyomjunk ENTER-t.

A fenti utasítás bezárja az aktív munkafüzetet.

⑤ Gépeljük be:

?activeworkbook.Name és nyomjunk ENTER-t.

Az utasítás hatására a **Immediate** ablakban az utasítás alatti sorban megkapjuk az aktív Workbook nevét.

⑥ Gépeljük be:

workbooks("Munkafüzet5").Close és nyomjunk ENTER-t.

Az utasítás hatására a rendszer bezárja ezt a munkafüzetet.

⑦ Gépeljük be:

workbooks("Munkafüzet6").Activate és nyomjunk ENTER-t.

Az utasítás hatására a Munkafüzet6 lesz az aktív állomány (látható a képernyőn).

⑧ Gépeljük be:

?activeworkbook.Saved és nyomjunk ENTER-t.

Mivel a munkafüzetben még semmit nem változtattunk, ezért a True (Igaz) választ kapuk.

⑨ Gépeljük be:

Activeworkbook.Saved=false és nyomjunk ENTER-t.

Ezzel a **SAVED** tulajdonságot beállítottuk. Ha most próbáljuk bezárni a Workbook-ot például az **Activeworkbook.Close** utasítással, akkor az Excel figyelmeztet bennünket a file mentésére.

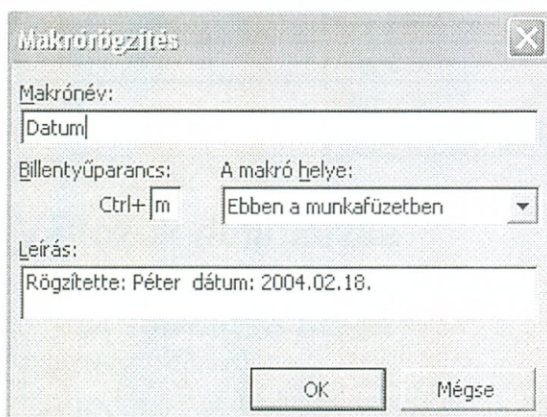
Makró rögzítés

Az Excel képes műveleteinkből VBA kódokat generálni egy makrórögzítési technikával. A következőkben egy mintafeladat kapcsán bemutatjuk a rögzítési technikát.

Mintafeladat a makró rögzítés szemléltetésére

Feladatunk egy olyan makró rögzítése, amely egy cellába beviszi a mai nap dátumát.

1. Adjuk ki az **Eszközök** menü **Makró**, majd **Új makró rögzítése** parancsát.
2. A megjelenő **Makrórögzítés** párbeszédablakot állítsuk be a következő ábra szerinti módon:



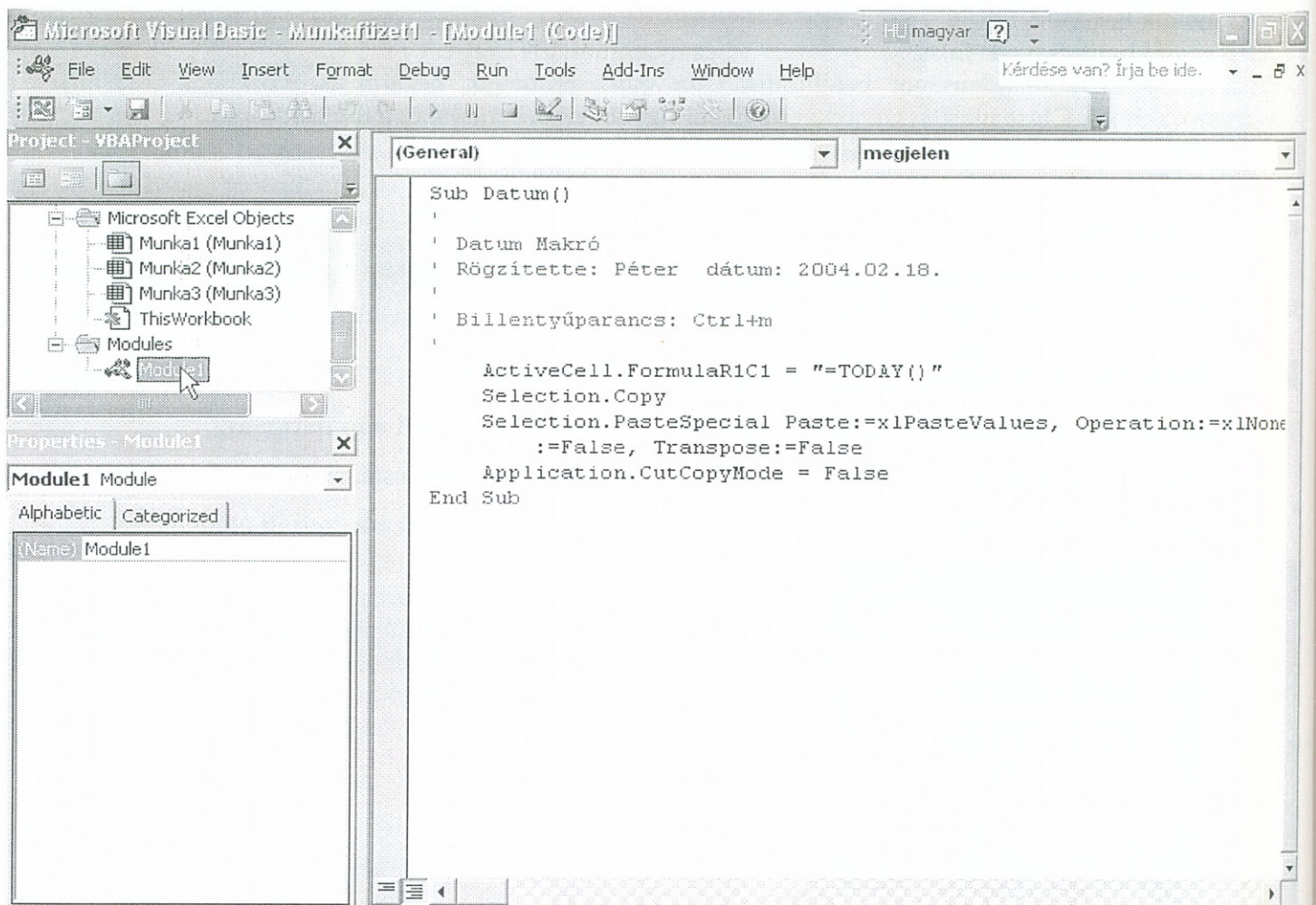
- ✦ A fenti párbeszédablak beállításával megadtuk a makró nevét (**Datum**), indító billentyűkombinációját (**CTRL+m**) és helyét (**Ebben a munkafüzetben**).

3. Kattintsunk az **OK** gombra. A rögzítés elindul, ettől az időponttól kezdve az Excel tevékenységeink alapján egy VBA makrót generál. Az Állapotsávon megjelenik a **Rögzítés** szó és a rögzítés folyamán végig ott marad.
4. Szúrjuk be Függvény varázslóval az aktuális cellába a **=MAO** függvényt.
5. Nyomjuk meg egyidejűleg a **CTRL** és a **C** billentyűket (másolás).
6. Adjuk ki a **Szerkesztés** menü **Irányított beillesztés** parancsát. A megjelenő párbeszédablakban kattintsunk az **Értéket** nevű választókapcsolóra.
7. Kattintsunk az **OK** gombra.
8. Nyomjuk le, majd engedjük fel az **ESC** billentyűt.
9. Fejezzük be a rögzítést, e célból kattintsunk egérrel a **Rögzítés** eszköztár, **Rögzítés vége** gombjára. Ekkor az Állapotsávról eltűnik a **Rögzítés** jelzés.

A rögzített kód

Nyilván az Olvasót érdekli a rögzített kód és egyúttal működtetni kívánja azt.

1. Adjuk ki az **Eszközök** menü **Makró**, majd **Visual Basic Editor** parancsát. Megjelenik a fejlesztésre szolgáló felület (Visual Basic Editor).



2. Kattintsunk duplán a **VBAProject** ablakban a **Modules** nevű könyvtárban a **Module1**-re. A fejlesztőfelület jobb oldalán a kód ablak megjelenik. A létrejött kódot ezzel megtaláltuk.
3. Indítsuk el a rögzített eljárást, ehhez váltsunk vissza az Excel képernyőre (kattintva a Taskbaron a Microsoft Office Excel nyomógombra).

- Adjuk ki az **Eszközök** menü **Makró**, majd **Makrók** parancsát. A megjelenő párbeszédablak listaablakában kattintással válasszuk ki makrónk nevét, majd kattintsunk az **Indítás** gombra. Makrónk lefut és az aktuális cellába beírja a mai nap dátumát.

Makró nyomógombbal történő indítása

Elkészült makróink számtalan módon indíthatóak. A következőkben röviden ismertetjük az egyik legnépszerűbb makróindítási módot, a nyomógombbal történő indítást.

- Jelenítsük meg az **Űrlapok** nevű eszköztárat.
- Kattintsunk a **Gomb** ikonra. Rajzoljuk meg a gombot. Amikor a gombot megrajzoltuk, megjelenik egy **Makró-hozzárendelés** nevű párbeszédablak. A listaablakban kattintással jelöljük ki a gombbal indítandó makró nevét, majd kattintsunk az OK gombra.
- A nyomógomb még mindig kijelölt, írjuk át feliratát.
- Kattintsunk egy cellára, ezzel befejeztük a gomb szerkesztését. A továbbiakban ezen gombra kattintva a hozzá rendelt eljárás mindig elindul majd.

Az Excel helyzetérzékeny Súgója (Help)

A VBA-ban történő programozást a **Help** (Súgó) menü is támogatja. Egy igen kényelmes megoldás egy parancs kijelölése a makróban az arra történő balgombos kattintással, majd az F1 megnyomása. Természetesen a Súgóból a mintapéldák, illetve azok részei programunkba átmásolhatók.

Adattípusok deklarálása

Változóinkat és konstansainkat különböző adattípusokban tárolhatjuk. Ha nem rendelünk adatunkhoz adattípust, az automatikusan ún. Variant típusú lesz. Az alábbiakban egy rövid táblázatban bemutatjuk adattípusainkat: Bővebb információ a Súgó: **Data Type** témakörében található.

Adattípus	Mit tárol	jele
Boolean (logikai)	True (igaz) vagy False (hamis)	
Integer	Egész számok	%
Long	Hosszú egész számok	&
Single	Tizedesszámok	!
Double	Nagy tizedesszámok	#
Currency	Pénznem	@
Date	Dátum	
Object	Objektum	
String	Karakterlánc	\$
Variant	Alapértelmezett	
Array	Tömbök	
User-defined	Többféle adattípus kombinációja	

Változók definiálása

Legegyszerűbb módja a változók Dim (Dimension) kulcsszóval (keyword) történő megadása.

Dim Változó1 As Integer

Ahol:

Dim Kulcsszó, mely jelzi, hogy változót deklarálunk

Változó1 A változó neve

As Kulcsszó, mely a változó neve és típusa között helyezkedik el (szeparálja őket)

Integer Kulcsszó, mely a változó adattípusát adja meg.

A változókat definiálhatjuk több sorban is:

Dim Súly As Double

Dim Neve As String

A változókat definiálhatjuk egy sorban:

Dim Súly As Double, Neve As String

Több azonos típusú változó egyszerre is megadható:

Dim a,b,c As String

Az adattípusokat az alábbi két függvény segítségével fejthetjük vissza:

↓ **VarType(Változó)**

↓ **TypeName()**

Típus deklaráció kezdőbetűk alapján a DefType paranccsal

Szintaxisa: **DefType Betűtartomány[,Betűtartomány]...**

Lehetséges szintaxisokat lásd az alábbi ábrán.

Parancs	Adattípus
DefBool	Boolean
DefInt	Integer
DefLng	Long
DefCur	Currency
DefSng	Single
DefDbl	Double
DefDate	Date
DefStr	String
DefObj	Object
DefVar	Variant

ALAPOK2.XLS-Változók:

DefInt A-K

' A változók nevei A-tól K-ig Integer-ek.

Sub Jó()

Dim felelet As String

felelet = "Helló!!!"

MsgBox (felelet)

End Sub

Sub Rossz()

CalcVar = 4

AnyVar = "Hello" ' Hiba:"Type mismatch" hiba.

MsgBox AnyVar

End Sub

A programon belül természetesen felüldefiniálható a változó típusa a **Dim** kulcsszóval.

ALAPOK2.XLS-Változók:

```
Sub Javított()
Dim AnyVar As String
AnyVar = "Hello"
MsgBox AnyVar
End Sub
```

Típusdeklaráló jelekkel történő adattípus megadás

A VB nyelvben szívesen használják, (Type declaration character) a VBA-ban nem, a típusdeklaráló jeleket.

Egy példa: (Változó%)

ALAPOK2.XLS-Változók:

```
Sub Típus()
Dim Változó%
Változó = 5
If VarType(Változó) = vbInteger Then
MsgBox "A Változó " & "Integer típusú"
End If
End Sub
```

Munka változókkal, Szöveg (String) típusú változók

Karakteres típusú értékeket tudunk bennük tárolni. Lehetőségünk van különböző adattípusok összefűzésére (Concatenation) a + vagy az & jellel. A szöveg hossza verziótól függően 2 milliárd (2^{31}) vagy 65 535 (az 5-ösben) karakter.

Ha változónkat a következő módon adjuk meg: **Dim Szöveg As String* 12**

akkor ezzel meghatároztuk a szöveg hosszát maximum 12 karakterekben, ha ennél hosszabbat adunk meg, akkor a szöveg vége elvész.

Ha változónkat a következő módon adjuk meg: **Dim Szöveg As String**

akkor a szöveg hossza a verziótól függő maximum.

A szövegben vezérlőkarakterek is lehetnek pl.: Chr(13) vagy VBCRLF.

Szöveg="Több soros a szöveg"+Chr(13)+"ez már új sorban lesz."

ALAPOK2.XLS-Változók:

```
Sub egyesít()
Változó1 = "34"
Változó2 = 6 ' Kezdőértékek.
eredmény = Változó1 + Változó2 ' Eredmény= 40.
MsgBox eredmény
Változó1 = "34": Változó2 = "6" ' Kezdőértékek.
eredmény = Változó1 + Változó2 ' Eredmény "346" (Konkatenáció).
MsgBox eredmény
End Sub
```

Dátum típusú változók

Megadásuk Hashmarkokkal (Andráskereszt) történik.

ALAPOK2.XLS-Változók:

```
Sub datum()
Dim MyDate As Date
MyDate = #5/2/97#           ' Dátum megadása
Date = MyDate
Cells(15,1)=Date
MsgBox Date
'Vigyázat átállítja a gép belső óráját 1997.05.02-re!!!!
End Sub
```

Aki dolgozik hibát is véthet, ahhoz, hogy a VBA jelezze számunkra a hibás (elgépelt) változó nevét, vagy a Modul első sorába írt **Option Explicit** parancsot kell használnunk, vagy be kell kapcsolnunk a **Tools** menü **Options** parancsának párbeszédablakában a **Require Variable Declaration** jelölőnégyzetet

A felhasználó által definiált adattípus (Type)

A következő példában tekintsünk meg egy eljárást a felhasználó által definiált adattípus használatára:
ALAPOK2.XLS-Tömbök:

```
Const ADÓKULCS = 0.25
Type Adataim
Név As String
Életkor As Integer
Beosztás As String
End Type

Sub Személyi_információk()
Dim Személy As Adataim
Személy.Név = "Kis János"
Személy.Életkor = 34
Személy.Beosztás = "tanár"
MsgBox Személy.Név & ", " & vbCrLf & Személy.Életkor _
& " éves," & Chr(10) & "beosztása:" & Személy.Beosztás
End Sub
```

Fenti példánkban a **Type** és az **End Type** parancsok között adtuk meg a felhasználó által definiált adattípusunkat, ezt mindjárt fel is használtunk a Személyi_információk című eljárásunkban.

A fenti példában az aláhúzás folytatásjelzésre szolgál, erre akkor van szükség, amikor a parancs nem fér ki egyetlen sorba. A folytatásjelzés létrehozásának módja: gépeljünk **SPACE**-t, majd aláhúzást karaktert, majd nyomjunk **ENTER**-t.

Megjegyzés: a változók kezdőértéke speciális is lehet.

Empty (üres) a változó kezdőértéke ha definiáláskor nem adtuk meg annak típusát. Ezért típusa ekkor **Variant** lesz.

Null a kezdőérték, ha nincs a változónak használható értéke.

Tömbökkel végzett munka

Tömbökkel végzett műveleteknél egy változónév több adatot képes tárolni. Elemeit az indexek révén érhetjük el.

A tömb megadására egy példa:

Dim Tömb (2,3) as Típus

Ahol a dimenziószám 2 és a maximális indexértékek 2 és 3.

Példákat találunk a tömbökre az **ALAPOK.XLS** munkafüzet **Alapok_D** nevű munkalapján.

Az indexelemek alapértéke 0 de az **Option Base {0 | 1}** paranccsal 1-be is állítható. Természetesen az alábbi forma sem tilos: **Dim Tömb1(5 to 25)**.

Dinamikus tömbök

Ha nem tudjuk vagy nem akarjuk előre megadni a tömb elemeinek számát, akkor dinamikus tömböt használunk és ekkor nem adunk a tömbnek indexet. Az alábbi példában az **Adat4()** a dinamikus tömb. Látható, hogy felhasználáskor meg kell adni az elemszámot (**ReDim Adat4(2)**).

Újabb **ReDim** paranccsal az elemszám megváltoztatható, de a tömb régi elemeinek értéke elveszik.

A **ReDim Preserve Adat4(20, 3)** parancs biztosítja a tömb régi elemeinek megtartását, egyúttal a tömb bővítését is.

ALAPOK2.XLS-Tömbök:

```
'Dinamikus tömb felhasználása (dynamic array)
'a ReDim parancs alkalmazásával.
Sub UseDynamicArray()
  Dim Adat4() As Variant
  ReDim Adat4(2)
  Adat4(1) = Int(90 * Rnd())
  Adat4(2) = Int(90 * Rnd())
  MsgBox "Lottó számok: " & Adat4(1) & ", " & Adat4(2)
  ReDim Adat4(10, 3)
  Adat4(1, 1) = "János"
  Adat4(1, 2) = 84
  Adat4(1, 3) = #3/21/12#
  MsgBox Adat4(1, 1) & ", kora: " & Adat4(1, 2) & _
    ", született: " & Adat4(1, 3)
End Sub
```

Egy adott tömb tartalmát egy munkalapra legegyszerűbben ciklus segítségével vihetjük át, pl. **For...Next** ciklussal. A **For...Next** ciklus lehetővé teszi a tömb elemeinek egyenkénti megcímzését és a célterületre való átvitelét.

Konstansok használata

A változókhoz hasonlóan a konstansok is elnevezett értékek (az érték nem változhat az eljárás végrehajtása közben). A konstansokat a **Const** paranccsal deklaráljuk. Nem kötelező, de szokásos a konstansok nevét nagy betűkkel írni. A beépített Excel és VBA konstansok "xl" és "vb" betűkombinációkkal kezdődnek.

A konstansokat rendszerint a modul tetején az eljárások fölött deklaráljuk, ekkor azokat a modul valamennyi eljárásában használhatjuk. A **Körterület** nevű függvénymakrónkban (a függvénymakrókról a későbbiekben még szó lesz) a PI értékét konstanssal adjuk meg, az alábbiak szerint:

ALAPOK2.XLS-Tömbök:

```
Function Körterület(Sugár As Double) As Double
  Const PI = 3.14
  Körterület = Sugár ^ 2 * PI
End Function
```

A fenti függvény próbája. programból:

```
Sub körter_proba()
  valasz = Körterület(10)
  MsgBox "A kör területe, ha a sugár 10 =" & valasz
End Sub
```

A függvényt a makró által létrehozott függvény begépelhető cellába is.

	MA	X	✓	fx	=Körterület(10)
	A	B	C	D	
1	=Körterület(10)				
2					

Műveletek

Programjainkban különböző típusú adatokkal dolgozhatunk, ehhez a műveletek széles skálája áll rendelkezésünkre.

Logikai műveletek

Logikai értékeket hasonlítanak össze (True, False). Az eredmény maga is egy logikai kifejezés.

Logikai_kifejezésA And Logikai_kifejezésB

Az And (logikai és) művelet eredménye akkor True (Igaz), ha mindkét kifejezés értéke True, ellenkező esetben False (Hamis).

```
Dim A, B, C, D, proba
A = 10: B = 8: C = 6: D = Null ' Kezdeti értékek.
proba = A > B And B > C ' A válasz: True.
```

Logikai_kifejezésA Or Logikai_kifejezésB

Az Or (logikai vagy) művelet eredménye akkor False (Hamis), ha mindkét kifejezés értéke False.

Logikai_kifejezésA Eqv Logikai_kifejezésB

Az Eqv (Ekvivalencia) művelet eredménye akkor True (Igaz), ha mindkét kifejezés értéke True vagy False.

Logikai_kifejezésA Xor Logikai_kifejezésB

Az Xor (logikai kizáró vagy) művelet eredménye False, ha mindkét kifejezés logikai értéke azonos.

Logikai_kifejezés1 Imp Logikai_kifejezés2

Az Imp (Logikai implikáció) művelet eredménye csak akkor False, amikor a Logikai_kifejezésA értéke True és a Logikai_kifejezésB értéke False.

Not Logikai_kifejezés

A Not (logikai negáció) művelet a Logikai_kifejezés értékét negálja (ellenkezőjére állítja).

Aritmetikai műveletek

Ide tartoznak az elemi matematikában megszokott műveleti jelek:

Összeadás (+)
 Kivonás (-)
 Lebegőpontos osztás (/)
 Egész osztás (\)
 Szorzás (*)
 Hatványozás (^)
 Osztás utáni maradék (Mod)

Példa aritmetikai műveletre:

```
Dim Szám, Var1, Var2
Szám = 2 + 2      ' Eredmény: 4.
Szám = 4257.04 + 98112 ' Eredmény: 102369.04.

Var1 = "34": Var2 = 6  ' Kiinduló értékek.
Szám = Var1 + Var2    ' Eredmény: 40.

Var1 = "34": Var2 = "6"      ' Kiinduló értékek.
Szám = Var1 + Var2    ' Eredmény: "346" (string concatenation).
```

Összehasonlító műveletek

Ezen műveletek segítségével két logikai kifejezést lehet összehasonlítani, a művelet eredménye is logikai kifejezés lesz:

Kisebb, mint (<)
 Nagyobb, mint (>)
 Kisebb vagy egyenlő (<=)
 Nagyobb vagy egyenlő (>=)
 Nem egyenlő (<>)
 Egyenlő (=)

Az összehasonlító műveletekhez tartozik a **Like** művelet is, amely két sztringet hasonlít össze – és az eredmény csak akkor True, ha a két sztring azonos karaktersorozatot tartalmaz.

Karakterláncok (sztringek) összefűzése

Műveletek segítségével két sztring egyesíthető, két műveleti jel szolgál a konkatenációra:

```
Eredmény= KifejezésA&KifejezésB
Eredmény= KifejezésA+KifejezésB
```

A fenti műveletek használatakor vigyáznunk kell arra, hogy a + jel az adattípustól függően az összeadás jele is, ezért inkább az & műveleti jel használata javasolt.

Kommentek (magyarázatok) használata a kódban

Eljárásainkba magyarázószövegeket gépelhetünk. A kommentek egysorosak is lehetnek. Azokat Aposztróf (') vagy a **Rem** (Remark) parancs, majd egy Space előzi meg, az alábbi módon:

```
Const PI = 3.14 ' A PI értéke.
' Ez egysoros magyarázat.
Rem Ez egysoros magyarázat
```

Végtelen ciklus és hibamegállás

Ha programunk nem akar “megállni” azaz végtelen ciklusba esik, onnan ESC gombbal, ha ez sem sikerül, akkor esetleg az **ALT+CTRL+DEL** gombbal tudjuk kihozni.

Bizonyos szituációkban szeretnénk, ha programunk a felhasználó által nem megszakítható lenne.

Amennyiben meg szeretnénk gátolni, hogy a felhasználó megszakítsa a programot, akkor használhatjuk az **Application** objektum **EnableCancelKey** tulajdonságát az alábbi értékek egyikével:

xlDisabled	A program nem megszakítható (ha kész a program, akkor használjuk),
xlInterrupt	Hibánál leáll, a felhasználó a programfutást megszakíthatja (pl.: ESC),
xlErrorHandler	A programban a saját hibakezelő eljárásunknak adja át a vezérlést, a hibakód ekkor 18-as értékű (trappable error code =18.)

ALAPOK2.XLS-Hibakez.:

```
Sub hiba()
On Error GoTo Cancel_Kezeles
Application.EnableCancelKey = xlErrorHandler
MsgBox "Ha unod a dolgot : Nyomj ESC -t a megálláshoz"
For x = 1 To 10000000
Next x

Cancel_Kezeles:
If Err = 18 Then
MsgBox "Kilőtted!!!!!"
End If
End Sub
```

Üzenetküldés

Adatok vagy üzenetek küldésének kényelmes módja az **MsgBox** függvény.

A parancs általános formája:

```
MsgBox(prompt[,buttons][,title][,helpfile,context])
```

Ahol:

Prompt: az üzenet szövege

Buttons: nyomógombok és szimbólumok (ezek jelennek meg a párbeszédablakban), kombináltan lehet őket használni, ekkor értéküket össze kell adni, lásd **ALAPOK2.XLS-MsgBox1** példát.

Title: a párbeszédablak felirata

helpfile, context: a Súlyó file és szövege.

A MsgBox függvény lehetséges **Buttons** paramétere:

Settings for the MsgBox's <i>buttons</i> argument			
	Constant	Value	Description
Group 1	vbOKOnly	0	Display OK button only
	vbOKCancel	1	Display OK and Cancel buttons
	vbAbortRetryIgnore	2	Display Abort, Retry, and Ignore buttons
	vbYesNoCancel	3	Display Yes, No, and Cancel buttons
	vbYesNo	4	Display Yes and No buttons
	vbRetryCancel	5	Display Retry and Cancel buttons
Group 2	vbCritical	16	Display Critical Message icon
	vbQuestion	32	Display Warning Query icon
	vbExclamation	48	Display Warning Message icon
	vbInformation	64	Display Information Message icon
Group 3	vbDefaultButton1	0	First button is default
	vbDefaultButton2	256	Second button is default
	vbDefaultButton3	512	Third button is default
	vbDefaultButton4	768	Fourth button is default

A MsgBox függvény visszaadott értéke:

Values returned by the MsgBox function		
Button	Constant	Numeric value
OK	vbOK	1
Cancel	vbCancel	2
Abort	vbAbort	3
Retry	vbRetry	4
Ignore	vbIgnore	5
Yes	vbYes	6
No	vbNo	7

ALAPOK2.XLS- MsgBox1:

```

Sub üzenet()
Dim üzenet As String
üzenet = "Folytassuk ?" ' Üzenet.
Stílus = vbYesNo + vbCritical + vbDefaultButton2 ' Nyomógombok.
Cím = "MsgBox DEMO" ' Cím.

Válasz = MsgBox(üzenet, Stílus, Cím)

If Válasz = vbYes Then ' A válasz Igen volt?

    Szöveg = "IGEN" ' Igen volt a válasz.

Else

    Szöveg = "NEM" ' Nem volt a válasz.

End If
MsgBox Válasz ' A 6 jelzi, hogy OK gombot nyomtunk
MsgBox Szöveg
End Sub

```

Információkérés (InputBox)

A függvény, illetve a metódus a felhasználótól kér információt egy megjelenő párbeszédablakkal.

Az InputBox lehet: InputBox függvény vagy InputBox metódus.

Példa InputBox függvényre:

ALAPOK2.XLS- InputBoxA:

```

Sub bekérés_fgv()
Dim Üzenet As String
Üzenet = "Adj be 1 és 3 közötti értéket."

Cím = "InputBox Próba" ' Cím.

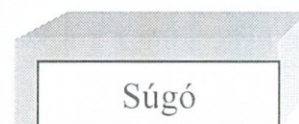
Alapérték = "1"

Eredmény = InputBox(Üzenet, Cím, Alapérték)

' helpfile és context használataa. A help gomb automatikusan megjelenik.
Eredmény = InputBox(Üzenet, Cím, , , "DEMO.HLP", 10)

' A párbeszédablakhoz pozíciója: 100,100
Eredmény = InputBox(Üzenet, Cím, Alapérték, 100, 100)
End Sub

```



Példa InputBox metódusra:
ALAPOK2.XLS- InputBox2:

```
Sub Bekérés_metódus()
Dim Üzenet As String
Üzenet = "Adj be 1 és 3 közötti értéket."

Cím = "InputBox Próba" ' Cím.

Alapérték = "1"

Eredmény = Application.InputBox(Üzenet, Cím, Alapérték, Type:=1)

End Sub
```

Az InputBox metódus szintaxisa:

object.InputBox(prompt, title, default, left, top, helpFile, helpContextID, type)

Ahol:

Prompt a felhasználónak szánt üzenet

title a párbeszédablak címe

default a párbeszédablak szövegdozójában megjelenő alapérték

left a párbeszédablak bal szélének távolsága a képernyő bal felső sarkától (pontokban) mérve

top a párbeszédablak tetejének távolsága a képernyő bal felső sarkától mérve

helpFile, helpContextID a Súlyó file és szövege

type a válasz típusa.

Igen hasznos a **type** paraméter, segítségével megszűrhetjük a beadott paraméter típusát. Íme egy táblázat a paraméterek értékéről:

Érték	Jelentés
0	Képlet
1	Szám
2	Szöveg
4	Logikai érték
8	cell referencia (Range object)
16	Egy hibaérték (pl.:#N/A)
64	Tömb

Vezérlési struktúrák

Feltételes elágazás If...Then...Else struktúra

Ezen az struktúrával bizonyos feltételek alapján megváltoztatható a parancsok sorrendje.

Szintaxis 1:

```
If feltétel Then parancsok [Else elseparancsok ]
```

Szintaxis 2:

```
If feltételek Then
    [parancsok]
[ElseIf feltétel-n Then
    [elseifparancsok]] ...
[Else
    [elseparancsok]]
End If
```

Ez a vezérlési szerkezet először megvizsgál egy kifejezést. Ha a vizsgált feltétel teljesül, a "Then"-t követő parancsokkal folytatódik a programvégrehajtás, egyébként az "Else"-t követőkkel.

ALAPOK2.XLS- IF:

```
Const ADÓKULCS1 = 0.12
Sub IF_szerkezet(Bruttó, Kedvezmény)

Dim Adóköteles
Dim adó
Adóköteles = Bruttó - Kedvezmény
If Adóköteles < 1500 Then
    adó = 0
Else
    adó = Adóköteles * ADÓKULCS1

End If
MsgBox ("Az adó értéke=" & adó)
End Sub

Sub hiv()
Call IF_szerkezet(10000, 200)
End Sub
```

Ha parancsunk egysoros, akkor elhagyható az End If, például:

```
If adó>0 Then MsgBox ("adó=" & adó)
```

Gyakran több feltételt kell vizsgálni, ekkor használjuk az ElseIf parancsot:

ALAPOK2.XLS- IF2:

```
Sub Minősítés()  
  Dim üzenet As String  
  Dim eredmény As Integer  
  üzenet = "Hány éves vagy?"  
  eredmény = CDbI(InputBox(üzenet, "Kor", 3))  
  If (eredmény >= 100) Then  
    MsgBox ("Ez igen szép kor!")  
  ElseIf eredmény < 100 And eredmény > 60 Then  
    MsgBox ("Az ember itt nyugdíjas!")  
  Else  
    MsgBox ("Még a nyugdíj odébb van!")  
  End If  
End Sub
```

Többirányú elágazás (Select Case)

Előfordulhat munkánk során, hogy egy adott értéket szeretnénk különböző értékekkel összehasonlítani, ekkor a **Select Case** vezérlési struktúrát használhatjuk az **If...Then...Else** helyett.

A **Select Case** szerkezet előbb kiértékeli a vizsgálat eredményét, majd a kapott értéket összehasonlítja minden egyes Case-zel a struktúrában. Ha azok egyeznek, végrehajtja a Case utáni parancsokat. A fenti, Minősítés nevű makrót átírtuk a Case szerkezetre:

ALAPOK2.XLS- IF3:

```

Sub Minősítés2()
  Dim üzenet As String
  Dim eredmény As Integer
  üzenet = " Hány éves vagy?"
  eredmény = Val(InputBox(üzenet, "Minősítés", 3))
  Select Case eredmény
  Case Is >= 100   'Ha az eredmény nagyobb egyenlő, mint 100.
    MsgBox ("Ez igen!")
  Case 61 To 99
    MsgBox ("Mit akarsz ez szép kor!")
  Case Else
    MsgBox ("Sokat kell még dolgozni!")
  End Select
End Sub

```

Ciklusképzés**A Do...Loop ciklusszervező utasítás**

A ciklusszervező utasítások egy utasítássorozat többszöri végrehajtására szolgálnak. A ciklus addig ismétlődik, míg egy feltétel igaz (While) vagy addig míg igazzá nem válik (Until).

A parancsnak többféle beírási módja van, vegyük sorra azokat.

Do While...Loop

A vizsgált feltételt a ciklus végrehajtása előtt kiértékeli. Ez azt is jelenti, hogyha a vizsgált feltétel nem teljesül (False), akkor a ciklusmagban lévő parancsok egyszer sem hajtódnak végre:

ALAPOK2.XLS- WHILE:

```

Sub CiklusA()
  Ciklusszámláló = 0      'A ciklusszámláló kezdőértéke.
  Do While számláló < 100
    számláló = számláló + 1
    Ciklusszámláló = Ciklusszámláló + 1 'A ciklusszámláló növelése.
    Worksheets(3).Cells(Ciklusszámláló, 1) = Ciklusszámláló
  Loop
  Szöveg1 = "Számoló értéke a ciklus végén="
  Szöveg2 = Str(számláló) & "-zal."
  MsgBox (Szöveg1 & Szöveg2)
End Sub

```

Do...Loop While

A vizsgálat a ciklus végén történik, ezért a ciklus legalább egyszer lefut:

ALAPOK2.XLS- WHILE2:

```

Sub CiklusB()
  Dim Szöveg1, Szöveg2 As String
  Ciklusszámláló = 0 'A ciklusszámláló kezdőértéke.
  Do
    számláló = számláló + 1

```

```
Ciklusszámláló = Ciklusszámláló + 1 'A ciklusszámláló növelése.
Worksheets(3).Cells(Ciklusszámláló, 2) = Ciklusszámláló
```

```
Loop While számláló < 100      'Kilépés a ciklusból amikor
      'a számláló értéke 100
      'vagy több.
```

```
Szöveg1 = "Számoló értéke a ciklus végén="
Szöveg2 = Str(számláló) & "-zal."
MsgBox (Szöveg1 & Szöveg2)
End Sub
```

Do Until...Loop

A Do While...Loop vezérlőstruktúrához hasonlóan működik, a ciklus azonban addig fut, amíg a vizsgált feltétel nem teljesül (False):

ALAPOK2.XLS- Do UNTIL:

```
Sub CiklusC()
  Ciklusszámláló = 0      'A ciklusszámláló kezdőértéke.
  Do Until számláló = 100

    számláló = számláló + 1
    Ciklusszámláló = Ciklusszámláló + 1 'A ciklusszámláló növelése.
    Worksheets(3).Cells(Ciklusszámláló, 3) = Ciklusszámláló

  Loop

  Szöveg1 = "Számoló értéke a ciklus végén="
  Szöveg2 = Str(számláló) & "-zal."
  MsgBox (Szöveg1 & Szöveg2)
End Sub
```

Do...Loop Until

A Do...Loop While vezérlőstruktúrához hasonlóan működik. A ciklusnak azonban akkor van vége, amikor a vizsgált feltétel teljesül (True):

ALAPOK2.XLS- Do LOOP:

```
Sub CiklusD()
  Ciklusszámláló = 0      'A ciklusszámláló kezdőértéke.
  Do
    számláló = számláló + 1
    Ciklusszámláló = Ciklusszámláló + 1 'A ciklusszámláló növelése.
    Worksheets(3).Cells(Ciklusszámláló, 4) = Ciklusszámláló
  Loop Until Not (számláló < 100)

  Szöveg1 = "Számoló értéke a ciklus végén="
  Szöveg2 = Str(számláló) & "-zal."
  MsgBox (Szöveg1 & Szöveg2)
End Sub
```

For...Next

A **For...Next** vezérlőstruktúra a **Do...Loop** struktúrához hasonlóan működik, de ekkor tudjuk, hogy a parancssorozatot hányszor kell megismételni.

A **For...Next** ciklusban automatikusan nő vagy csökken a számláló értéke, jelezve, hogy a ciklus hányszor futott le. Az alábbi példa az első munkalapon az A oszlop első 10 cellájába bevisz egy-egy olyan értéket, amely egyenlő a sor számával:

ALAPOK2.XLS- FOR:

```
Sub ciklus_for()
Worksheets(1).Select
For sorszám = 1 To 10
ActiveSheet.Cells(sorszám, 1).Value = sorszám
Next sorszám
End Sub
```

A ciklusszámláló értéke nemcsak növelhető, hanem csökkenthető is (Step-1, a ciklusszámláló értéke eggyel csökken):

ALAPOK2.XLS- FOR2:

```
Sub faktoriális()
Dim üzenet
Dim Válasz As Long
Dim egyik As Long
üzenet = "Hányadik faktoriális számoljuk ki?"
Válasz = Val(InputBox(üzenet, "Faktoriális"))
együk = 1
For i = Válasz To 1 Step -1
együk = egyik * i
Next i
MsgBox (Válasz & " faktoriálisa=" & egyik)
End Sub
```

For Each...Next

A **For Each...Next** ciklusszerkezet egy objektumgyűjtemény minden egyes elemén egy parancsszériát hajt végre.

A következő példánkban az összes nyitott munkafüzetet (Workbooks gyűjtemény) bezárjuk, míg az ezt követően az aktív munkafüzetet nem zárjuk be:

ALAPOK2.XLS- FOR EACH:

```
Sub Bezár()
For Each MunkaFüzet In Application.Workbooks
MunkaFüzet.Close
Next MunkaFüzet
End Sub
```

```

Sub bezar2()
For Each Munkafuzet In Application.Workbooks
If Munkafuzet.Name <> ThisWorkbook.Name Then
Munkafuzet.Close savechanges:=True
End If
Next Munkafuzet
End Sub

```

Nem zárja be az aktív munkafüzetet

Exit

Az **Exit Do** és az **Exit For** parancsok segítségével kiléphetünk a ciklusból, annak szabályos befejezése előtt.

Nézzünk mindkét parancsra egy-egy példát:

ALAPOK2.XLS- EXIT DO:

```

Sub kilép_Do()
Worksheets(1).Select
Worksheets(1).Cells(5, 1).Value = 77
sorszamlalo = 1
Do
If Cells(sorszamlalo, 1).Value = 77 Then
MsgBox ("Megtaláltuk a 77-et tartalmazó cellát!")
Exit Do
End If
sorszamlalo = sorszamlalo + 1
Folytatás = MsgBox("Folytassuk a keresést?", vbYesNo)
Loop While Folytatás = vbYes
End Sub

```

ALAPOK2.XLS- EXIT FOR:

```

Sub Keres()
' A program megadja egy szövegben az r betű pozícióját.
Üzenet = "Adj be egy szöveget, megadom az r betű pozícióját!"
szöveg = InputBox(Üzenet, "R betű keresés")
For karakter = 1 To Len(szöveg)
If Mid(szöveg, karakter, 1) = "r" Then
MsgBox ("A(z) " & karakter & "." & "pozícióban" _
& " r" & " betű van!")
Exit For
End If
Next karakter
End Sub

```

2. fejezet: Események

A programozóknak gyakran feladata, hogy egy adott esemény makrót indítson el valamilyen feladat elvégzéséhez. A Microsoft Excel rendelkezik ezzel a képességgel is. Az események mindig objektumokhoz vannak hozzárendelve. Erre példa lehet egy munkalap vagy egy munkafüzet.

Az alábbi mintapéldákban csak néhányat tárgyalunk a rendelkezésre álló események közül, és általában a MsgBox üzenetablakokat használjuk annak jelzésére, hová ugrik a tényleges kód.

Ha többet szeretnénk megtudni ezen és más eseményekről, kattintsunk az Office Segédre a VBE környezetben és keresőszóként adjuk meg az objektumnevet és az "events" szót (pl. "worksheet events" a munkalap eseményekhez).

BeforeDoubleClick

Az események közül nagyon jól felhasználható a **BeforeDoubleClick** esemény a munkalapokon. Nagyon egyszerű példa lehet erre az alábbi:

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Excel.Range, Cancel As Boolean)
MsgBox "Duplán kattintott a következő cellára: " & Target.Address
Cancel = True
End Sub
```

Ez az esemény akkor következik be, amikor a felhasználó duplán rákattint a munkalap valamelyik cellájára. A **Target** paraméter átadódik a makrónak, így a programozó tudni fogja, melyik cellára kattintottunk duplán. A **Cancel** argumentum alapértelmezett értéke False (hamis), de megváltoztatható True-ra (igaz) a kódon belül. A **Cancel** argumentum True-ra állításával törölhetjük az eseményhez tartozó alapértelmezett műveletet. Ebben az esetben a cellára való dupla kattintáshoz tartozó alapértelmezett művelet a szerkesztő módba való átlépés. Mivel a **Cancel** argumentumot True-ra állítottuk, ez nem fog megtörténni. Ha azt szeretnénk, hogy megtörténjen az alapértelmezett művelet, töröljük a **Cancel=True** sort a kódból.

Change

Egy másik nagyon hasznos esemény a munkalap **Change** eseménye. Ha a felhasználó új értéket ír bármelyik cellába, ez az esemény következik be.

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
MsgBox "Ezen a lapon megváltozott a következő cella: " & Target.Address
End Sub
```

Mint látható, eseményekkel is vezérelhetjük programjainkat. Például megadhatjuk azt, hogy milyen utasítást akarunk végrehajtani, ha a munkalapon egy tartományt kijelölünk stb...

Azt, hogy milyen eseményeket tudunk figyelni, az objektumok határozzák meg.

Munkalap eseményt (Worksheet) kezelő eljárás készítése

1. Nyissuk meg a VBE-t (ALT+F11).
2. Kattintsunk duplán a Munka1 (Sheet1) munkalap moduljára. Ekkor megnyílik egy új modul.
3. A modul bal oldali lenyíló listáját nyissuk le (válasszuk a **Worksheet**-et).
 - ✦ Ekkor az ablak jobb oldali lenyíló listájából kiválaszthatjuk azon eseményt (events), amelyhez programot szeretnénk írni. Ez természetesen munkalaponként más lehet. Az eljárás neve mindig az eseményre jellemző, például a **SelectionChange** esemény, akkor működik, ha a munkalapon tartományt adunk meg. Az eljárások paraméterekkel is rendelkezh(et)nek. Például a **SelectionChange** esetén a **Target** paraméter megadja az újonnan kijelölt tartományt.

Mintafeladat: egy munkalapeseemény kezelő eljárás

Mintafeladat: készítsünk olyan eljárást, amely ahányszor a Munka1 munkalapon a kijelölést megváltoztatjuk, a cellákat zöldre színezi.

```
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
Target.Interior.ColorIndex = 4
End Sub
```

Próba: váltsunk át a Munka1 munkalapra (ennek írtuk meg a **Worksheet_SelectionChange** eseményét). Jelöljük ki egy tartományt, ennek celláit az Excel színezi.

Ha a Munka2-ön tennénk ezt, nem történne semmi, mert ennek még nem írtuk meg a **Worksheet_SelectionChange** eseményét.

Mintafeladat: példa egy munkafüzeteseemény (Workbook) kezelő eljárásra

Ha azt szeretnénk, hogy minden egyes munkalapunkon történjen meg a cellák színezése, ha tartományt változtatunk, akkor a munkafüzet **SheetSelectionChange** eseményét kell megírunk. Előbb töröljük a Munka1-hez rendelt **Worksheet_SelectionChange** eljárást (jelöljük ki majd DEL).

1. Kattintsunk duplán a VBA project ablakban a munkafüzet moduljára (ThisWorkbook). A jobb oldali ablakban kattintsunk a baloldali lenyílóra, válasszuk a **Workbook** objektumot. Majd a jobb oldali lenyílóra s innen válasszuk a munkafüzet **SheetSelectionChange** eseményt. **Megj:** alapesemény az **Open** (megnyitás).
2. Írjuk meg a programot. Látható, hogy az eljárásnak más paramétere is van a **Targeten** kívül. Ez az **Sh** (azaz: Sheet, a munkalap amelyen az esemény történt).

Íme az eseménykezelő:

```
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Excel.Range)
Target.Interior.ColorIndex = 4
MsgBox Sh.Name & "munkalapon a tartomány címe: " & Target.Address
End Sub
```

Próba: bármelyik munkalapon is jelölünk ki egy tartományt azt az Excel beszínezi. Csak már azt sajnáljuk, hogy ha most egy másik Excel állományt megnyitunk abban a program nem működik Ebben az esetben olyan eljárásra van szükségünk, amely Alkalmazás (Application) szinten működik.

Az Alkalmazás (Application) eseményei

Amennyiben olyan eseményvezérelt eljárást akarunk írni, amely az összes Excel munkafüzetében egyaránt működik, akkor az Application objektum eseményeit kell felhasználnunk. Nem célunk egy ilyen eljárás készítése, azon olvasók, akiket érdekel egy ilyen eljárás működése, tanulmányozzák az **APPEVENT.XLS** állomány eljárásait.

Egyéb események

ON...

Ezen események az Excel 95 VBA nyelvéből öröklődtek. Az **ONENTRY** helyett például a **Worksheet_Change** eseményt kínálja fel az Excel.

ONTIME

Ezen esemény segítségével egy adott időpontban egy eljárást elindíthatunk.

Néhány példa ONTIME-ra:

Ez a parancs a **my_Procedure** eljárást 15 másodperc múlva elindítja:

```
Application.OnTime Now + TimeValue("00:00:15"), "my_Procedure"
```

Ez a parancs a **my_Procedure** eljárást 17:00:00-kor elindítja:

```
Application.OnTime TimeValue("17:00:00"), "my_Procedure"
```

Ez a parancs kilövi az előző példában beállított eseményt (Schedule:=False):

```
Application.OnTime EarliestTime:=TimeValue("17:00:00"), _  
Procedure:="my_Procedure", Schedule:=False
```

Példák:

Munkalap eseményekre:

```
CHAP05_2.XLS ---SHEET1 munkalap
```

Munkafüzet eseményekre:

```
CHAP05_2.XLS ---ThisWorkBook
```

Application eseményekre:

```
APPEVENT.XLS
```

3. fejezet: Az Application objektum (kiegészítő interface elemek)

Application tulajdonságok

Calculation: megadja az Excel átszámítási módjait: pl.: **xlCalculationAutomatic**.

Caption: Az alkalmazás címsorán kijelzendő szöveg. Ha értékét **Empty**-be állítjuk, visszaáll a "Microsoft Excel" szöveg.

Cursor: négy Cursor alak közül választhatunk:

XINormal,	a Windows Vezérlőpult alapján áll be,
XIWait	"Homokóra",
XINorthWestArrow	"nyilas",
XIBeam	I alakú kurzor.

DisplayAlert: ha értéke **False**, akkor az Excel szolgálati üzenetei nem jelennek meg. Alapértelmezett értéke a **True**.

DisplayFormulaBar: ha értéke **False**, akkor nem látható a Szerkesztőléc.

DisplayStatusBar: ha értéke **False**, akkor nem lehet látni az Állapotsort.

Height: az alkalmazás ablak magassága, pontokban mérve.

Left: Az alkalmazás ablak bal szélének távolsága a képernyő bal szélétől (pontokban mérve).

OperatingSystem: operációs rendszer.

ScreenUpdating: ha értéke **False**, akkor az Excel nem frissíti a képernyőt (mintegy "fagyasztja" azt).

StatusBar: az Állapotsorra írandó szöveg, ha értékét **False**-ra állítjuk, visszaáll az Excel eredeti Állapotsora.

Version: Az Excel verziószáma.

Width: Az Excel ablak szélessége.

WindowState: Az Excel ablak állapota:

XINormal: az ablak méretét a Height és Width tulajdonságok határozzák meg.

XIMaximized: az ablak kitölti a teljes képernyőt.

XIMinimized: az ablak ikon méretű.

CHAP05-1m.XLS

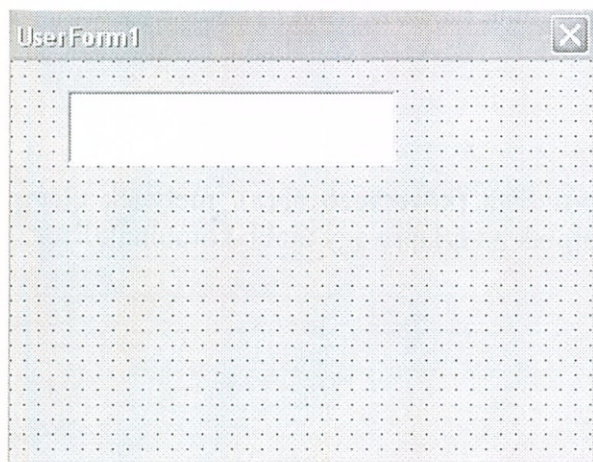
```
Sub Chap05dProc36_SetApplicationProperties()
Worksheets(1).Activate
With Application
  If UCase(Left(.OperatingSystem, 3)) <> "MAC" Then
    .Caption = "Saját felhasználás"
    MsgBox "Application.Caption = ""Saját felhasználás""""
  End If
  .DisplayFormulaBar = False
  MsgBox "A Szerkesztőléc kikapcsolt."
  .DisplayStatusBar = False
  MsgBox "Az Állapotsor kikapcsolt"
  .DisplayStatusBar = True
  .StatusBar = "Saját állapotsor üzenet"
  MsgBox "Az Állapotsor üzenet: Saját állapotsor üzenet"

  .ScreenUpdating = False
  If UCase(Left(.OperatingSystem, 3)) <> "MAC" Then
    .WindowState = xlNormal
    .Height = 200
    .Width = 200
    .Top = 100
    .Left = 100
    .ScreenUpdating = True
    MsgBox "Application.ScreenUpdating = False" & Chr(13) & _
      "Application.WindowState = xlNormal" & Chr(13) & _
      "Application.Height = 200" & Chr(13) & _
      "Application.Width = 200" & Chr(13) & _
      "Application.Top = 100" & Chr(13) & _
      "Application.Left = 100" & Chr(13) & _
      "Application.ScreenUpdating = True"
    .WindowState = xlMaximized
    MsgBox "Application.WindowState = xlMaximized"
    .Caption = Empty
  End If
  .DisplayFormulaBar = True
  .StatusBar = False
  '.DisplayAlerts = False
  '.Quit
End With
End Sub
```

4. fejezet: Űrlapok

Mintafeladat: Beviteli mező (TextBox)

1. Nyissunk meg egy új, üres file-t, abba szúrjunk be egy UserForm-ot, arra rajzoljunk Beviteli mezőt (TextBox) a ToolBox eszköztár megfelelő gombjára (TextBox) kattintva. Annak **ControlSource** tulajdonságát állítsuk **A5**-re.



2. Ezzel megadtuk azt, hogy az a szöveg, amelyet a beviteli mezőbe írunk a **Munka1!A5**-ös cellájába kerül majd.
3. **Próba:** Jelöljük ki az űrlapot és adjuk ki a **Run** (Indítás) menü **Run Sub/Userform** (Eljárás/UserForm futtatása) parancsát. Zárjuk be az űrlapot a párbeszédpanel jobb felső sarkában lévő keresztre való kattintással, majd visszatérve a munkalapra láthatjuk, hogy a beviteli mező tartalma a Munka1 munkalap A5-ös cellájába bekerült.

A fenti feladványt programozással is oldjuk meg.

1. Ehhez azt az eseményt használjuk fel, amely akkor történik ha a beviteli mező tartalma megváltozik. Neve: **TextBox1_Change**.
2. Írjuk az eljárásba: **Worksheets("Munka1").Range("A5")=TextBox1**
3. Majd indítsuk el az űrlap megjelenítést, ezért nyomjunk F5-öt, majd gépeljünk a Beviteli mezőbe valamit, a gépelt tartalom a beviteli mezőben azonnal megjelenik.
4. Az elkészült állományt mentjük el **URLAP2.xls** néven.

Kombinált lista (URLAP3.XLS Munka1) /ComboBox/

A kombinált lista **RowSource** tulajdonsága szolgál a lista helyének megadására, míg a **ControlSource** tulajdonság adja meg, hogy a lista melyik elemét választottuk ki.

Feladat: készítsünk egy űrlapot, amelyre majd a kombinált listát rajzoljuk. Egy munkalapon (például a Munka1-en, az első tanfolyam neve az A2-ben legyen) készítsük el képzeletbeli tanfolyamok nevének listáját.

1. Készítsünk a Munka1 munkalapra egy táblázatot:

	A
2	Excel I.
3	Excel II.
4	Excel III.
5	Word I.
6	Word II.

2. Egy űrlapra rajzoljunk ComboBox-ot.
3. Váltunk át az űrlapra és a kombinált listára jobb gombbal kattintva, a megjelenő parancsválogatásból válasszuk a **Properties** (Tulajdonságok) parancsot.
4. A **RowSource** tulajdonságot állítsuk be: **Munka1!A2:A6**-ra, míg a **ControlSource** tulajdonságot: **Munka1!A1**-re.

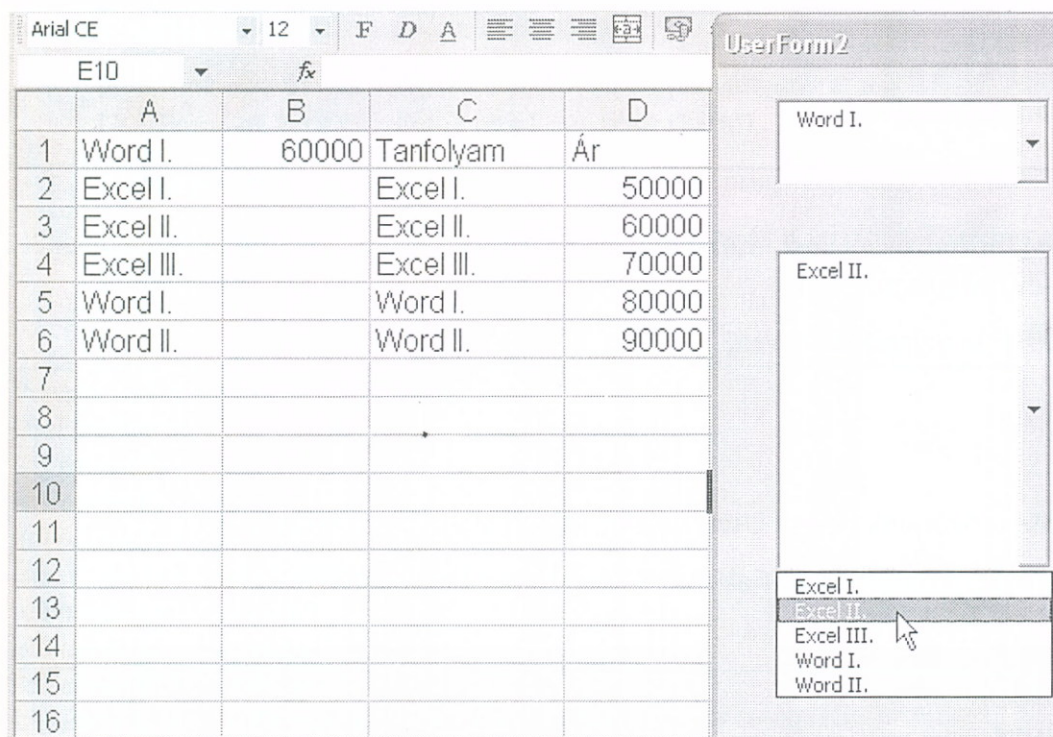
Próba: legyen aktív az űrlap, indítsuk el az űrlap megjelenítést, nyomjunk F5-öt. A listát nyissuk le, válasszunk elemet, zárjuk a párbeszédablakot a jobb felső sarkában lévő "keresztlet". Az A1-es cellában a kiválasztott elem megjelenik.

Kétoszlopos kombinált lista

1. A lista legyen kétoszlopos, miután elkészült a kétoszlopos lista (következő ábra), melynek elemeit begépeztük a Munka1 munkalap C2:D6 tartományába, rajzoljunk az űrlapra egy újabb kombinált listát.

	C	D
1	Tanfolyam	Ár
2	Excel I.	50000
3	Excel II.	60000
4	Excel III.	70000
5	Word I.	80000
6	Word II.	90000

2. A **RowSource** tulajdonságot állítsuk be: **Munka1!C2:D6**-ra, míg a **ControlSource** tulajdonságot: **Munka1!B1**-re. Az Excellel a **ColumnCount** tulajdonság 2-be történő beállításával közöljük, hogy a kombinált lista kétoszlopos.
3. Ugyanakkor azt is meg kell adnunk, hogy a lista melyik oszlopát szeretnénk használni. Ezt a **BoundColumn** tulajdonsággal adjuk meg, ide írjunk be 2-öt (ár).
4. Végül azt is megadhatjuk (ha akarjuk), hogy milyen szélesek legyenek a sorforrás egyes oszlopai (**ColumnWidths**), az oszlopszélesség értékeket egymástól pontosvesszővel válasszuk el, legyen ez most: 1 cm;0 cm. **Megjegyzés:** ezen értékeket az Excel átírja pontba.
5. **Próba:** legyen aktív az űrlap, indítsuk el az űrlap megjelenítést (F5), a listát nyissuk le, válasszunk elemet, zárjuk a párbeszédablakot a jobb felső sarkában lévő "keresztlet". A B1-es cellában a kiválasztott elem megjelenik.



A listaelemek meghatározása programból

1. Rajzoljunk egy újabb ComboBox-ot (ComboBox3).
2. Ez már a 3. lesz ugyanazon az űrlapon.
3. A lista elemeit tömbben deklaráljuk. A sorforrás tulajdonságot az űrlap Initialize eseményében írjuk meg (azaz ez a programrész akkor fog működni, amikor az űrlapot megnyitjuk). Duplán kattintva az űrlapra a **UserForm_Initialize** esemény megírható:

```
Private Sub UserForm_Initialize()
Dim Tanfolyamok(5) As String
Tanfolyamok(0) = "Excel I."
Tanfolyamok(1) = " Excel II. "
Tanfolyamok(2) = " Excel III. "
Tanfolyamok(3) = "Word I."
Tanfolyamok(4) = "Word II."
Tanfolyamok(5) = "Word III."
ComboBox3.List = Tanfolyamok
End Sub
```

4. Mivel rajzoltunk egy újabb ComboBox-ot, melynek az Excel a ComboBox3 nevet adta, a vezérlést is ehhez a névhez rendeltük, lásd a fenti kódot: **ComboBox3.List = Tanfolyamok**
5. Ha az eredményt is programból szeretnénk feldolgozni, akkor ehhez a ComboBox **List** tulajdonságát használhatjuk fel. Például a **ComboBox3.List(0)** hivatkozással a ComboBox listájának első elemére mutatunk rá.

Vajon hogyan jeleníthetjük meg egy cellában a ComboBox-ban kiválasztott elemet. Ezt a ComboBox **ListIndex** tulajdonsága adja meg. A következő programrészlet a B1-es cellába írja a kiválasztott listaelem értékét.

1. Miután egérrel duplán kattintottunk a ComboBox-ra megkaptuk a **ComboBox3_Change** eseményének kódját, az alábbiakat írjuk bele:

```
Private Sub ComboBox3_Change()
Worksheets("munka1").Range("b1") = _
ComboBox3.List(ComboBox3.ListIndex)
End Sub
```

2. Indítsuk el a `UserForm_initialize` eljárást, válasszuk ki a lista egy elemét, az azonnal megjelenik a Munka1 munkalap B1-es cellájában.
3. Az elkészült file-t mentjük le `urlap3.xls` néven.

Kétoszlopos lista

1. Nyissunk meg egy új file-t. Szűrjünk be egy UserForm-ot, rajzoljunk rá egy ComboBox-ot.
2. Annak `UserForm_Initialize` eseményében írjuk meg a kétoszlopos lista adatait tömbben:

```
Private Sub UserForm_Initialize()
Dim Tanfolyamok(4, 1) As String
Tanfolyamok (0, 0) = "Excel I."
Tanfolyamok (0, 1) = "10000"
Tanfolyamok (1, 0) = "Excel II."
Tanfolyamok (1, 1) = "15000"
Tanfolyamok (2, 0) = "Excel III."
Tanfolyamok (2, 1) = "20000"

Tanfolyamok (3, 0) = "Word I."
Tanfolyamok (3, 1) = "10000"
Tanfolyamok (4, 0) = "Word II."
Tanfolyamok (4, 1) = "14000"
ComboBox1.List = Tanfolyamok
ComboBox1.ColumnCount = 2
ComboBox1.ColumnWidths = "1 cm;0 cm"

End Sub
```

A `ComboBox1_Change` eseménybe az alábbiakat írjuk:

```
Private Sub ComboBox1_Change()
Worksheets("munka1").Range("a1") = _
ComboBox1.List(ComboBox1.ListIndex, 0) 'Tanfolyam neve
Worksheets("munka1").Range("b1") = _
ComboBox1.List(ComboBox1.ListIndex, 1) ' Ára
End Sub
```

3. Mentjük le az állományt `Urlap4.xls` néven. Próbáljuk ki programunkat a `UserForm_Initialize` elindításával. A Munka1 munkalap A1-es cellájába a tanfolyam neve, míg a B1-be, annak ára kerül.

Listablak

Hasonló a control a ComboBox-hoz. Ezt is a `RowSource`, `ControlSource` és a `List` tulajdonságokkal (Properties) vezérelhetjük.

Feladatunk több listaelem kiválasztása.

1. Nyissunk egy új munkafüzetet, ALT+F11-gyel váltsunk át a VBE-re. Fűzzünk be a **Insert** menüvel egy UserForm-ot, arra rajzoljunk egy ListBox-ot (Lista).
2. A lista elemekkel való feltöltésére használjuk az űrlap **Initialize** eljárását.
3. Ehhez bal gombbal duplán kattintsunk az űrlapra és a felkínált **UserForm_Click** eseményt cseréljük le Initialize-ra.

```
Private Sub UserForm_Initialize()
Dim lstForrás(6) As String
lstForrás(0) = "hétfő"
lstForrás(1) = "kedd"
lstForrás(2) = "szerda"
lstForrás(3) = "csütörtök"
lstForrás(4) = "péntek"
lstForrás(5) = "szombat"
lstForrás(6) = "vasárnap"
ListBox1.List = lstForrás 'a listadoboz feltöltése a tömb elemeivel.
End Sub
```

4. Úgy szeretnénk beállítani a listát, hogy minden eleme előtt megjelenjen egy jelölőnégyzet (CheckBox), ehhez annak **ListStyle** tulajdonságát (a Properties ablakban) állítsuk be: **fmListStyleOption**-ra.
5. Mivel egyszerre több elemet akarunk kiválasztani a listából, ezért állítsuk a ListBox **MultiSelect** tulajdonságát: **fmMultiSelectMulti**-ra.
6. **Tegyünk próbát:** kattintsunk az űrlapra, nyomjuk meg az F5-öt. A megjelenő listában jelöljük ki több elemet.
7. Már csak az a feladat, hogy megtudjuk, mely elemeket választottuk ki a listában. A kiértékelésre felhasználhatjuk az űrlap **Terminate** vagy a lista **Change** eseményét egyaránt. Az előbbi akkor következik be, ha bezárjuk az űrlapot, az utóbbi akkor, ha megváltoztatjuk a listaelemek kijelölését.
8. Használjuk az űrlap **UserForm_Terminate** eseményét:

```
Private Sub UserForm_Terminate()
Dim i As Byte
For i = 1 To ListBox1.ListCount ' Hány soros a lista
Cells(i, 1) = ListBox1.List(i - 1) 'lista eleme
Cells(i, 2) = ListBox1.Selected(i - 1) 'Bekapcsolt-e ez az elem?
Next i
End Sub
```

9. A fenti program a munkalap bal szélső két oszlopába egymás mellé listázza a lista elemét és egy logikai értéket, mely jelzi, hogy az elem bekapcsolt-e. A program szép futásához az űrlap Initialize eljárásba be kell írni egy olyan parancsot, amely törli a munkalap első két oszlopát.

Írjuk át a programot, úgy, hogy az a listában kiválasztott elemeket írja be a munkalap egymás alatti celláiba:

```
Private Sub UserForm_Terminate()
Dim i As Byte
Dim k As Byte
k = 1
For i = 1 To ListBox1.ListCount ' Hány soros a lista
If ListBox1.Selected(i - 1) Then 'ha az elem kiválasztott
Cells(k, 1) = ListBox1.List(i - 1) 'akkor az elem kerüljön be
'ebbe a cellába.
k = k + 1 'a következő sor címe, csak
'akkor nő eggyel, ha az elem kiválasztott.
End If
Next i
End Sub
```

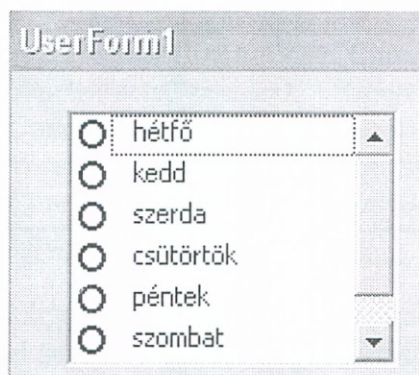
10. Ha a **MultiSelect** tulajdonságot **fmMultiSelectExtended** értékre állítjuk be, akkor a kijelölésnél használható a **CTRL** és a **SHIFT** billentyű több listaelem egyszerre történő kijelölésére.
11. Az állományt mentjük el **URLAP5.XLS** néven.

Lista (ListBox), mint választócsoport /USERFORM2, URLAP5.XLS/

Természetesen ilyenkor a listának csak egy eleme választható ki.

1. Szűrjünk be ebbe a projectbe egy másik űrlapot. Ebbe is rajzoljunk egy **ListBox**-ot (Lista).
2. Az űrlap **Initialize** eljárását felhasználva állítsuk be vezérlést, ugyanúgy, mint a Több listaelem kiválasztása c. részben volt. A ListBox **ListStyle** tulajdonságát a **Properties** ablakban állítsuk be **fmListStyleOption**-ra. Miután kattintással aktívvá tettük az űrlapot, nyomjunk F5-öt a próbához.
3. Mivel egyszerre csak egy elemet akarunk kiválasztani a listából, ezért állítsuk a ListBox **MultiSelect** tulajdonságát: **fmMultiSelectSingle**-re (ennek hatására választókapcsolók jelennek meg az elemek előtt).

Párbeszédablakunk az alábbi:



A kiválasztott elem kiértékelése:

1. A **ControlSource** tulajdonságba írjuk be egy cella címét, oda kerül a választás eredménye. Például: **Munka1!a11**.

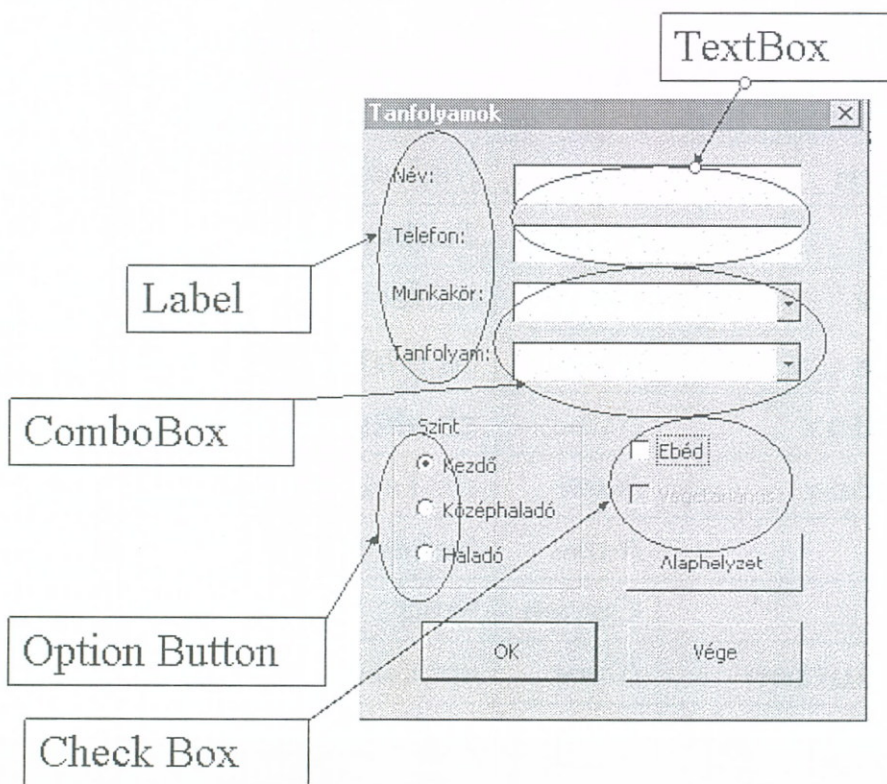
- Amennyiben programmal kívánjuk kezelni a Listát, érdemes annak kezelését az űrlap **Terminate** (űrlap bezárása) eseményébe megírni:

```
Private Sub UserForm_Terminate()
Range("b5") = ListBox1.ListIndex 'listabeli sorszám
Range("b6") = ListBox1.List(ListBox1.ListIndex) 'kiválasztott elem
End Sub
```

Próba: tegyük aktívvá kattintással az űrlapot, az F5-öt megnyomva az űrlap megjelenik. Válasszuk ki egy elemet, kattintva az űrlap jobb felső sarkában lévő keresztre a **Terminate** esemény elindul.

Összefoglaló feladat

Feladat: Készítsünk egy olyan űrlapot, mint amilyen a következő két képen látható! A beállítandó tulajdonságokat a két ábrát követő táblázatban megadjuk.



Az alábbi táblázat az egyes controlok nevét, típusát, beállítandó tulajdonságait tartalmazza:

<i>Control</i>	<i>Type</i>	<i>Property</i>	<i>Setting</i>
UserForm	UserForm	<i>Name</i>	frmKonyveles
		<i>Caption</i>	Tanfolyamok
Név	Text Box	<i>Name</i>	txtNev
Telefon	Text Box	<i>Name</i>	txtTelefon
Munkakör	Combo Box	<i>Name</i>	cboMunkakor
Tanfolyam	Combo Box	<i>Name</i>	cboTanfolyam
Szint	Frame	<i>Name</i>	fraLevel
		<i>Caption</i>	Szint
Kezdő	Option Button	<i>Name</i>	optKezdo
Középhaladó	Option Button	<i>Name</i>	optKozephalado
Haladó	Option Button	<i>Name</i>	optHalado
Ebéd	Check Box	<i>Name</i>	chkEbed
Vegetáriánus	Check Box	<i>Name</i>	chkVeget
		<i>Enabled</i>	False
OK	Command Button	<i>Name</i>	cmdOk
		<i>Caption</i>	OK
		<i>Default</i>	True
Vége	Command Button	<i>Name</i>	cmdVege
		<i>Caption</i>	Cancel

<i>Control</i>	<i>Type</i>	<i>Property</i>	<i>Setting</i>
		<i>Cancel</i>	True
Alaphelyzet	Command Button	<i>Name</i>	cmdAlaphelyzet

A következő ábra a táblázat címsorát tartalmazza (nem kell kitölteni a második sort, az csak tájékoztató jelleggel látható). Készítsük el a címsort.

	A	B	C	D	E	F	G	H
1	<i>Név</i>	<i>Telefon</i>	<i>Munkakör</i>	<i>Tanfolyam</i>	<i>Szint</i>	<i>Ebéd</i>	<i>Vegetáriánus</i>	
2	Kis Imre	12345678	adminisztrátor	PowerPoint	Középhaladó	Igen	Igen	
3								Könyvelés
4								
5								

- Duplán kattintva az űrlapra írjuk meg az inicializációs eljárást, amellyel feltöltjük a **ComboBox**-okat egyúttal üresre állítjuk azok értékét. Ugyanakkor itt állítjuk be a **CheckBox**-ok alapállapotát. A **Vegetáriánus CheckBox**-ot tulajdonság megadásával bénítsuk. Csak akkor engedélyezzük azt, ha az **Ebéd CheckBox** bekapcsolt. Végül az utolsó paranccsal a **Focus** a **txtNevre** visszük. Ne feledjük ez az eljárás az űrlap megjelenésekor hajtódik végre.

```

Private Sub UserForm_Initialize()
    txtNev.Value = ""
    txtTelefon.Value = ""
    With cboMunkakor
        .AddItem "vezető"
        .AddItem "tanár"
        .AddItem "kereskedő"
        .AddItem "közgazdász"
        .AddItem "fejlesztő"
        .AddItem "diszpécser"
        .AddItem "adminisztrátor"
    End With
    cboMunkakor.Value = ""
    With cbotanfolyam
        .AddItem "Access"
        .AddItem "Excel"
        .AddItem "PowerPoint"
        .AddItem "Word"
        .AddItem "FrontPage"
    End With
    cbotanfolyam.Value = ""
    optKezdo = True
    chkEbed = False
    chkVeget = False
    txtNev.SetFocus
End Sub

```

2. Írjuk meg a **Vége** gomb kattintás eseményét lekezelő eljárást:

```
Private Sub cmdVege_Click()  
    Unload Me  
End Sub
```

3. Írjuk meg az **Alaphelyzet** gomb kattintás eseményét lekezelő eljárást:

```
Private Sub cmdAlaphelyzet_Click()  
    Call UserForm_Initialize  
End Sub
```

4. Írjuk meg az **OK** gomb kattintás eseményét lekezelő eljárást:

```
Private Sub cmdOK_Click()  
    ActiveWorkbook.Sheets(1).Activate  
    Range("A1").Select  
    Do  
        If IsEmpty(ActiveCell) = False Then  
            ActiveCell.Offset(1, 0).Select  
        End If  
        Loop Until IsEmpty(ActiveCell) = True  
        ActiveCell.Value = txtNev.Value  
        ActiveCell.Offset(0, 1) = txtTelefon.Value  
        ActiveCell.Offset(0, 2) = cboMunkakor.Value  
        ActiveCell.Offset(0, 3) = cbotanfolyam.Value  
        If optKezdo = True Then  
            ActiveCell.Offset(0, 4).Value = "Kezdő"  
        ElseIf optKozephalado = True Then  
            ActiveCell.Offset(0, 4).Value = "Középhaladó"  
        Else  
            ActiveCell.Offset(0, 4).Value = "Haladó"  
        End If  
        If chkEbed = True Then  
            ActiveCell.Offset(0, 5).Value = "Igen"  
        Else  
            ActiveCell.Offset(0, 5).Value = "Nem"  
        End If  
        If chkVeget = True Then  
            ActiveCell.Offset(0, 6).Value = "Igen"  
        Else  
            If chkEbed = False Then  
                ActiveCell.Offset(0, 6).Value = ""  
            Else  
                ActiveCell.Offset(0, 6).Value = "Nem"  
            End If  
        End If  
        Range("A1").Select  
    End Sub
```

5. A **Vegetáriánus CheckBox** tiltása/engedélyzése céljából kattintsunk duplán a **chkEbed CheckBox**-ra és írjuk meg az alábbi eljárást:

```
Private Sub chkEbed_Change()
  If chkEbed = True Then
    chkVeget.Enabled = True
  Else
    chkVeget.Enabled = False
    chkVeget = False
  End If
End Sub
```

6. Írjuk meg a Module1-re az űrlapot megjelenítő eljárást, amelyet egy a Munka1 munkalapra az Űrlap eszköztár segítségével rajzolt nyomógombbal tegyünk indíthatóvá:

```
Sub Konyveles()
  frmKonyveles.Show
End Sub
```

7. Mentsük le az Excel állományt, majd próbáljuk ki elkészült alkalmazásunkat.

UserForm tulajdonságok, metódusok, események

Az alábbiakban röviden összefoglaljuk a fontosabb űrlap tulajdonságokat, metódusokat, eseményeket.

UserForm tulajdonságok

Name: a programban erre a névre kell hivatkozni.

BackColor: az űrlap színe.

Caption: az űrlap címsorában ezt a szöveget látjuk.

Left, Top, Height, Width: az űrlap helye és mérete (pontokban).

Picture: az űrlap háttéréül szolgáló kép. A **LoadPicture** függvény eredményét átadhatjuk a **Picture** tulajdonságnak. A kép vezérlésére felhasználhatjuk a **PictureAlignment**, **PictureSizeMode**, **PictureTiling** tulajdonságokat is.

StartPosition: az űrlap középére igazítására szolgál.

Értékei: 1-**CenterOwner:** az Excel ablak közepén

2-**CenterScreen:** a képernyő közepén

3-**None:** nincs középére igazítás (alapértelmezett: a képernyő bal felső sarka).

UserForm metódusok

Show: kijelzi az űrlapot (modális).

Hide: elrejt az űrlapot.

PrintForm: űrlap nyomtatása

UserForm parancsok:

Load: betölti az űrlapot, de nem jeleníti meg.

Unload: eltünteti a képernyőről az űrlapot, egyúttal a memóriából is törli.

UserForm események

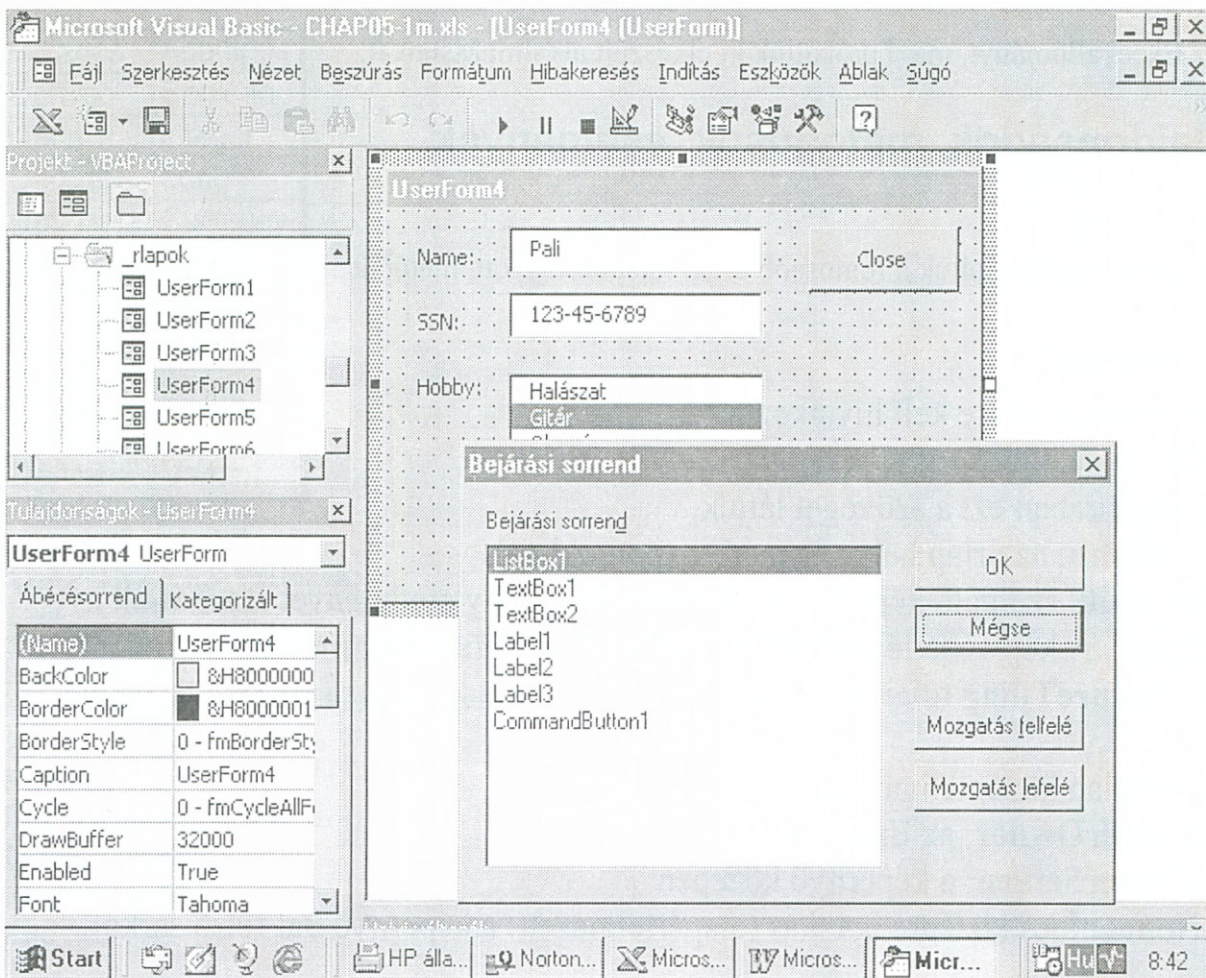
QueryClose: az űrlap bezárása előtt alakul ki. A **Cancel** argumentumot True-ba állítva megelőzhető az űrlap bezárása.

Initialize: az űrlap kijelzése előtt alakul ki.

Terminate: kialakul, miután az űrlapot bezártuk.

A Tab sorrend beállítása (Tab Order)

A Tab sorrend az űrlap elemeinek **TAB** billentyűvel történő bejárásai sorrendje. Mint az köztudott, az űrlapon a Controlok a TAB billentyű nyomogatásával egyesével aktívvá tehetők. A bejárás sorrendje megváltoztatható.



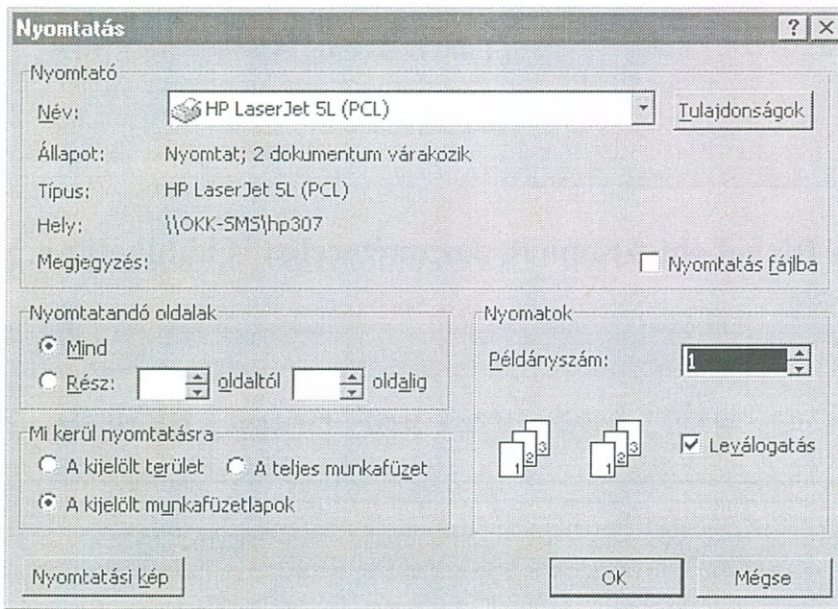
Miközben a **Chap05-1m.xls** állomány megnyitott és ALT+F11-gyel átváltottunk annak VBE képernyőjére (fenti kép), válasszuk ki a VBAProject-ek közül a UserForm4-et. Adjuk ki a **View** (Nézet) menü **Tab Order** (Bejárás sorrend) parancsát. A megjelenő párbeszédablakban láthatjuk a TAB sorrendet, annak kezelőszerveivel azt meg is változtathatjuk.

Beépített Excel párbeszédablakok kijelzése

A **Dialogs** gyűjtemény (Collection) szolgál a beépített párbeszédablakok kijelzésére. A **Dialogs** paramétere egy konstans, amely a kijelzendő ablak nevére utal. Természetesen a kijelzésre szükség van a **Show** metódusra is.

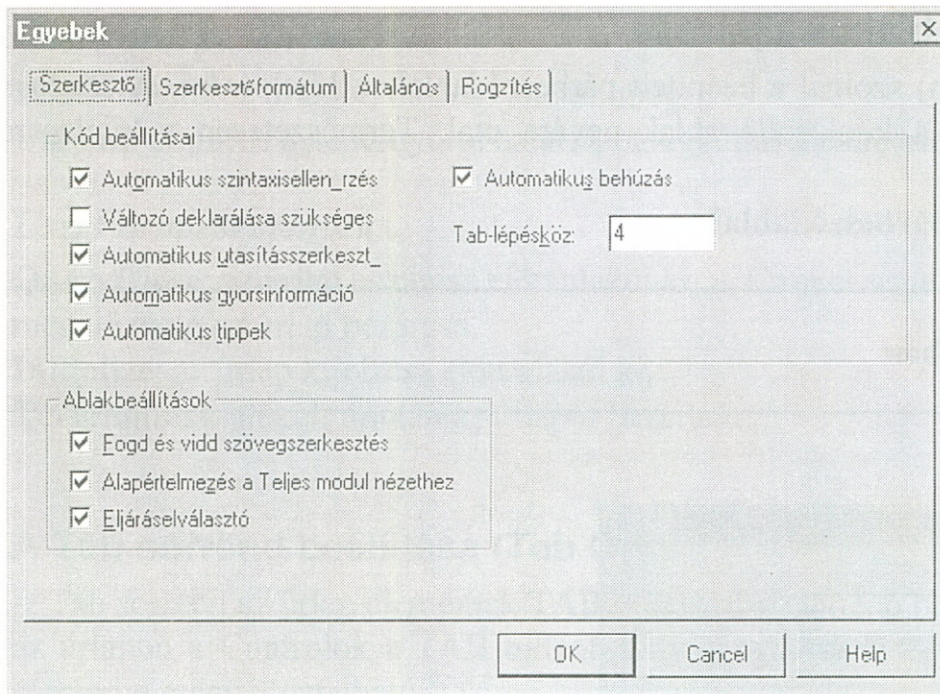
Az alábbi eljárás kijelzi a **Print** párbeszédablakot.

```
Sub Chap05bProc28_DisplayPrint()
    Application.Dialogs(xlDialogPrint).Show
End Sub
```



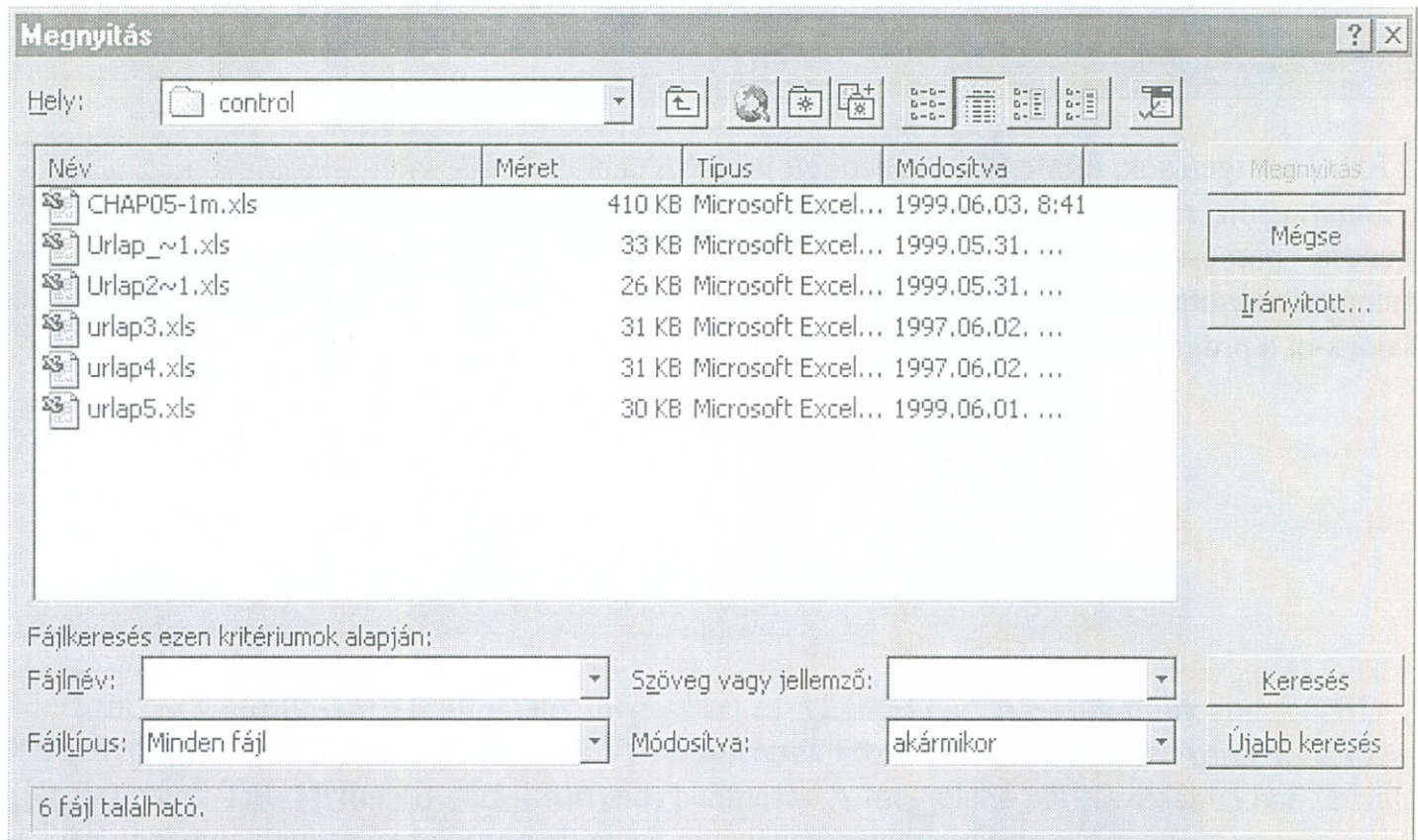
Az Excelnek igen sok, mintegy 200 beépített párbeszédablaka van.

A konstansok megkeresése: kapcsoljuk be a **Tools** menü (Eszközök) **Options** (Egyebek) parancsa párbeszédablakának **Edit** (Szerkesztő) nevű lapján az **Auto List Members** (Automatikus utasításszerkesztő) **CheckBox**-át. Ennek hatására, amikor begépeljük majd a **Dialogs**-ot a programba, a lista automatikusan megjelenik.

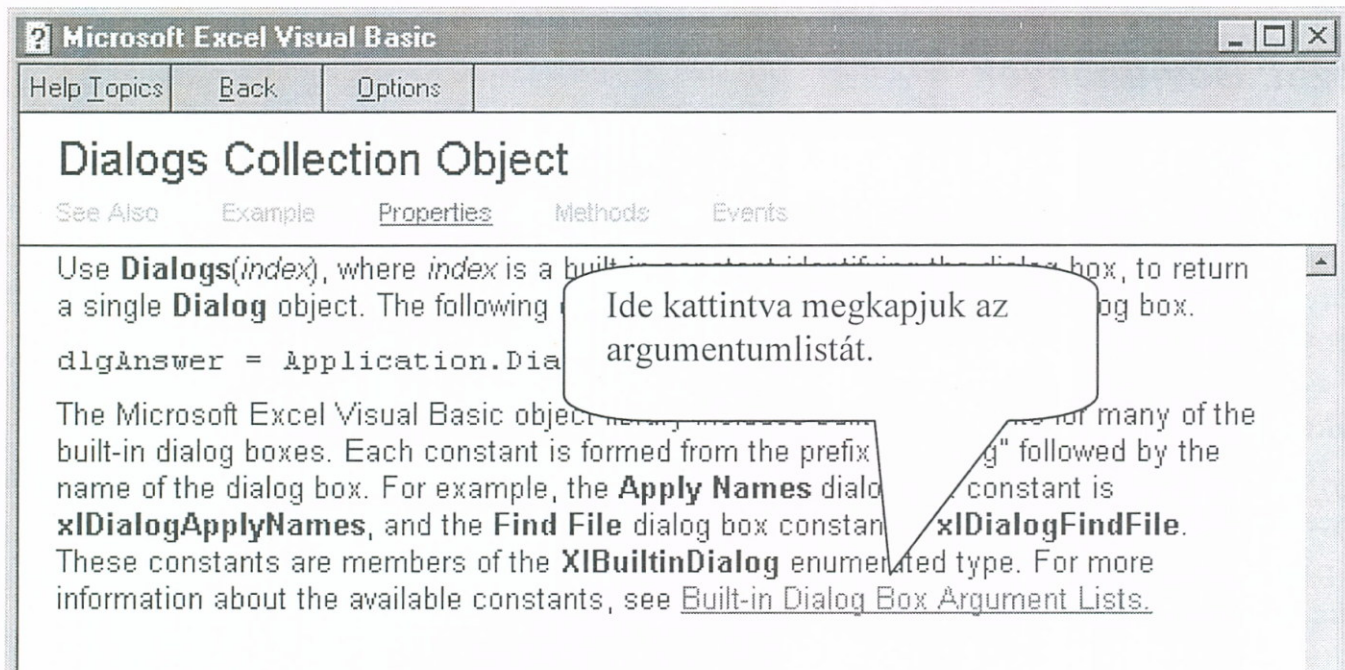


Amikor a Show metódust használjuk a Dialog objektumnál, paramétereket is küldhetünk át az ablakba. Erre jó példa a következő.

```
Sub Chap05bProc29_DisplayOpen()
    Application.Dialogs(xlDialogOpen).Show "*"
End Sub
```



Súgó:



Egy részlet az argumentumlistából:

Dialog box constant	Argument list(s)
<code>xlDialogActivate</code>	<code>window_text, pane_num</code>
<code>xlDialogActiveCellFont</code>	<code>font, font_style, size, strikethrough, superscript, subscript, outline, shadow, underline, color, normal, background, start_char, char_count</code>

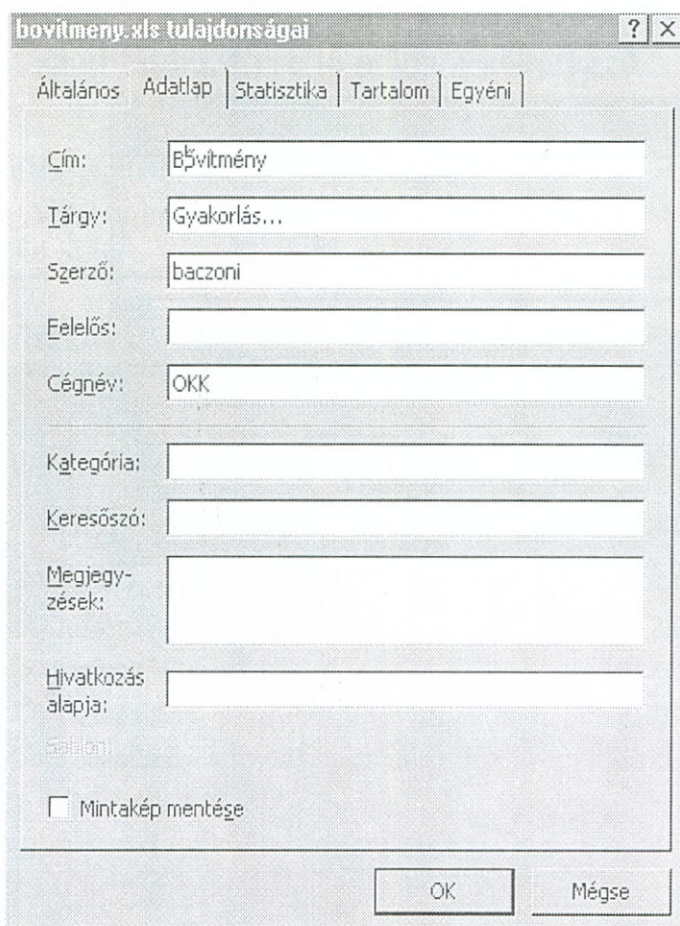
5. fejezet: Bővítmények

Bővítmények

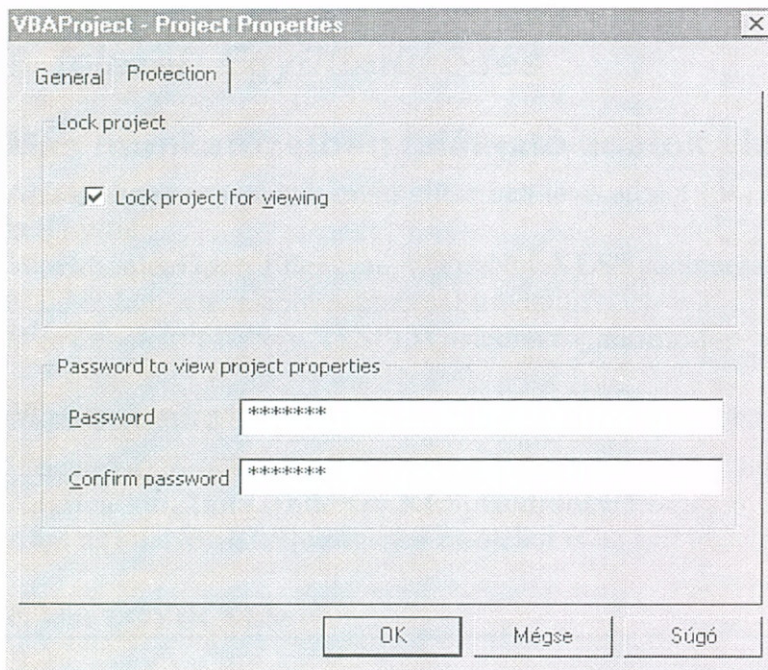
A bővítmények használata talán a legjobb módszer programjaink terjesztésére. A bővítmények vagy .XLA (Excellel készültek) vagy .XLL kiterjesztésűek, ez utóbbiak valamilyen fejlesztőrendszerrel (a Legtöbbször C nyelven) készültek. Az alábbiakban röviden ismertetjük a bővítménykészítés módját.

A projektet jelszóval védhetjük.

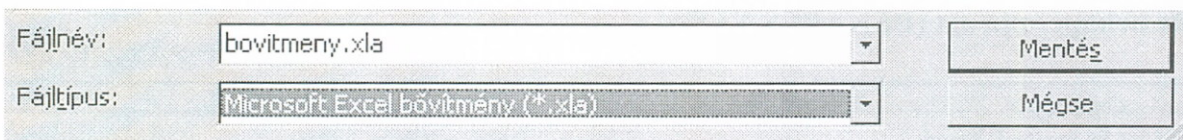
1. Nyissuk meg a .XLS állományt. Az állomány legyen rejtett (**Ablak** menü **Elrejtés** parancsa). Amikor az Excelből kilépünk az rejtettként menti az .XLS állományt.
2. A **Fájl** menü az **Adatlap** (Properties) parancsának segítségével állítsuk be az adatlapot. A legfontosabb a **Cím**, ez lesz a bővítmény címe.



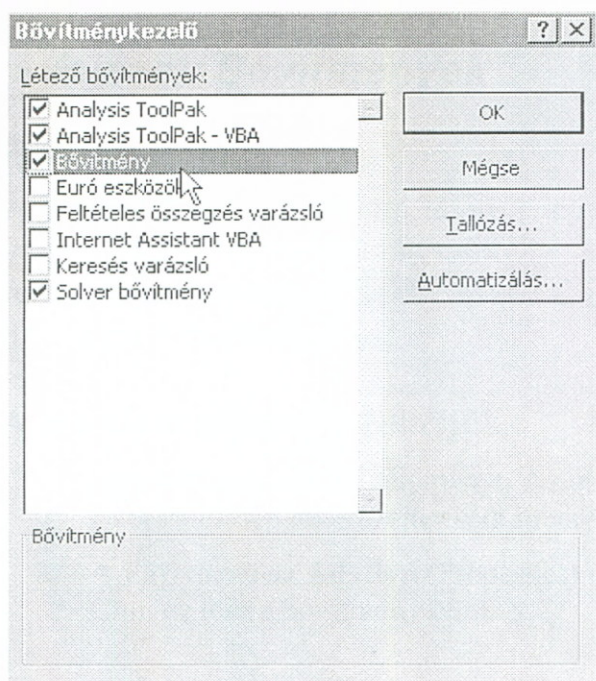
3. Kapcsoljunk át a VBE-re: ALT+F11. Adjuk ki a **Tools** menü **Compile VBAproject** parancsát. Nem kapunk visszajelzést a kompilálás végéről.
4. Adjuk ki a **Tools** menü **VBAProject Properties** parancsát (Eszközök / Egyedi Bővítmények tulajdonságai). A Védelem (Protection) lapon levédhetjük jelszóval projektünket.



5. Visszatérve az Excel ablakába, adjuk ki a **Fájl** menü **Mentés másként** parancsát (ehhez fel kell fedni az .XLS állomány ablakát, de ne feledjük elrejteni és így kilépni az Excelből), a megjelenő párbeszédablakban a **Fájltípus** lenyílókat nyissuk le. Az Add-In-t (**Microsoft Excel bővítmény**) válasszuk.



6. Ezzel bővítményünk előállt. Az elkészült állomány kiterjesztése .XLA lett, és helyileg a **Bővítmények** nevű könyvtárba került (Excel 2003).
7. Az Excelből lépünk ki, majd indítsuk azt újra el, építsük be makróinkat. Ehhez adjuk ki az **Eszközök** menü **Bővítménykezelő** parancsát. A megjelenő párbeszédablakban kapcsoljuk be bővítményünk jelölőnégyzetét. Majd nyomjunk **OK**-t. Az Excel betölti a bővítményt (ezután az Excel, minden elindulása után betölti ezt a bővítményt).



Végül, ne feledjük a bővítmény makrót úgy kell megírunk, hogy abban gondoskodni kell a bővítmény makróinak indításáról, ennek egyik legszebb módja, ha saját parancso(ka)t írunk A saját parancsok felhelyezéséhez írjunk **Workbook_Open** míg azok törléséhez a **Workbook_Close** eseménykezelőket. Jegyzetünk Függvényekről szóló fejezetében bemutatunk majd egy példát beépülő létrehozására.

Egy példa annak ellenőrzésére, hogy a bővítmény makró (Beállítások) **checkBox-a be van e kapcsolva**, ha nem akkor az eljárás bekapcsolja azt (BEKAPCS.XLS):

```
Private Sub Workbook_Open()
If AddIns("Beállítások").Installed Then
'Rendben telepítve a beépülő
Else
AddIns("Beállítások").Installed = True
End If
End Sub
```

Ha a bővítményt eltávolítjuk, gondoskodnunk kell a saját menük törléséről (**BeforeClose** eseménykezelő), a függvényeknél nincs ilyen gond azok automatikusan eltűnnek a Függvény Varázslóból.

6. fejezet: Együtműködés

Más munkafüzetben található adatok elérése

Ahhoz, hogy egy másik munkafüzetben lévő adatot (pl.: egy cella tartalma) elérhessünk, a munkafüzetet meg kell nyitni:

Workbooks.Open Filename:="FileNév.XLS" parancssal.

Majd érdemes az alábbi paranccsal folytatni:

Workbooks("FileNév.XLS").Worksheets(„munkalap_neve”).Range(“a2”)

Más munkafüzetben található programok elindítása

Érdemes az **Application.Run** parancsot használni erre a célra az alábbi formában:

Application.Run („[FileNév.XLS]Modulnév!Program_Név” [;argumentum1 [;arg2...]])

Ahol a [FileNév.XLS] szögletes zárójeleit is be kell gépelni.

Példa: EGYIK.XLS:

```
' meghív Makró
' Rögzítette: SZAMALK RT., dátum: 1998.05.25.
```

```
Sub meghív()
    Application.Run Macro:="masik.xls!zöldít"
End Sub
```

Természetesen a
Az EGYIK.XLS-ben
Fest zöldre!

MASIK.XLS:

```
' zöldít Makró
' Rögzítette: SZAMALK RT., dátum: 1998.05.25.
```

```
Sub zöldít()
    With Selection.Interior
        .ColorIndex = 4
        .Pattern = xlSolid
    End With
End Sub
```

7. fejezet: Hibakezelés

Az előre becsülhető hibákat futás közben kell kezelnünk.

A hibák feldolgozására az ON ERROR GOTO... utasítások szolgálnak.

Alapvető ismeretek a hibafeldolgozásról (hibafeldolgozó függvények és parancsok)

A következő mnemonicokkal sűrűn találkozhatunk a különböző hibakezelő rutinokban:

Err Objektum, alapértelmezett tulajdonága a Number (hibakód).

Erl: A tulajdonság a hibás sor számát adja vissza.

Error: Függvény, amely a hibaüzenet adja.

Error: Parancs, amely hibát szimulál, a hibakód az **Err**-ben jelenik meg.

A legtöbb eljárásba (parancs- vagy függvénymakró) illik elhelyezni egy hibakezelő rutint, amelynek egyik legfontosabb eleme a program elejére írandó **ON ERROR GOTO Címke** parancs. A Címkére (kettőspontot kell tartalmaznia a név után), akkor adódik a vezérlés, ha a program futása közben hiba lép fel. Nagyon fontos, hogy a program "normál" végére egy kilépő utasítást (pl.: Exit Sub) helyezünk, így akadályozva meg azt, hogy helyes működéskor a hibakezelő rutin működésbe lépjen.

A hibafeldolgozó rutinban a **RESUME 0** parancs visszaadja a vezérlést a hibás utasításra, a **RESUME NEXT** parancs viszont a hibás utasítást követő utasításra adja a vezérlést. Az **ON ERROR GoTo 0** parancs visszaadja a hibafeldolgozást a rendszernek.

Az **Err.Raise Number:=szám** parancs segítségével hibakódot tudunk generálni (a szám a hibakód).

A következő eljárást megoldott állapotban a **HIBAKER.XLS Modul3**-ban letároltuk.

```
Sub Osztas()
    Dim Long1 As Long
    Dim Long2 As Double
    On Error GoTo errorcode
    Long1 = Val(InputBox("Adj egy poz. Egész számot", _
        "A 10-et osztjuk"))
    Long2 = 10 / Long1
    MsgBox "10 osztva " & Long1 & " = " & Long2
    Exit Sub
errorcode:
    If Err = 11 Then
        MsgBox "Ne ossz nullával!!!"
    Else
        MsgBox "Valami más gond van."
    End If
End Sub
```

A Sűgóban fel vannak sorolva, a fontosabb elfogható hibák kódjai ("Trappable Errors")

Részlet a hibakódokból: Error Code	Error Message
3	Return without GoSub
5	Invalid procedure call or argument
6	Overflow
7	Out of memory
9	Subscript out of range

Az Err objektum fontosabb tulajdonságai:

Property	Érték
Number	A hiba száma (error number).
Source	Az aktuális Visual Basic project neve.
Description	A Hibaüzenet

Egy példa, amely az Err objektumok használatát mutatja: HIBAKER.XLS Sugobol nevű modulban található.

```

Sub Sugobol()
Dim Üzenet
On Error Resume Next ' Defer error handling.
Err.Clear
Err.Raise 6 ' Egy Túlsordulás hibát okozunk, "Overflow" error.
' Check for error, then show message.
If Err.Number <> 0 Then
    Üzenet = "A hibát, kódja # " & Str(Err.Number) & " okozta a: " _
        & Err.Source & Chr(13) & Err.Description
    MsgBox Üzenet, "Error"
End If
End Sub

```

8. fejezet: Függvények

Munkalapfüggvények

Az Excel munkalapfüggvények egy része a VBA-ban is felhasználható. Kivételt képeznek azon függvények, amelyeknek a VBA-ban is megfelelőjük. Például az **SQRT** (GYÖK) munkalapfüggvény nem használható a szabványos módon, mert a VBA-ban van azonos működésű függvény, az **SQR**.

Megjegyzés: ha mégis használni szeretnénk ezen függvényeket, akkor alkalmazzuk az **Evaluate** metódust:

Például: `MsgBox Evaluate("=sqrt(100)")`

A következő mintafeladat példát mutat a **SUM** munkalapfüggvény parancsmakróban történő felhasználására:

```
'WorksheetFunction
Sub osszead2()
eredmeny = Application.WorksheetFunction.Sum(Range("a1:a10"))
MsgBox eredmeny & " WorksheetFunction"
End Sub
```

Mint a fenti példaprogramban látható a függvény nevét a **WorksheetFunction** tulajdonság után adtuk meg. Meg kell jegyezni, hogy igazán régi verziójú Exceleekben a **WorksheetFunction** tulajdonság nem létezik, azokban az **Application**-t egyből a függvénynévnek kell követnie.

A fontosabb VBA függvények (Function)

Név	Funkció
Abs	Abs érték
Date	Rendszer dátum
Fix	A szám egész része, poz. Számnál le- neg. számnál felkerekít
Int	A szám egész része, mindig lekerekít
Is..	Információs függvények (pl: IsError)
Len	Sztring hossza byte-okban
Now	Akt. Dátum és idő
Shell	Program futtatás
Sqr	Négyzetgyök

Példák függvényekre: FGV1.XLS (Modul1 és Modul2)

Felhasználó által definiált függvények

Függvényt felhasználó is létrehozhat a **Function** kulcsszó segítségével. A függvények a **Felhasználói** kategóriában jönnek létre, de a **MacroOptions** metódussal más kategóriákba is áthelyezhetők.

Mit érdemes tudni a felhasználó által készített függvényekről

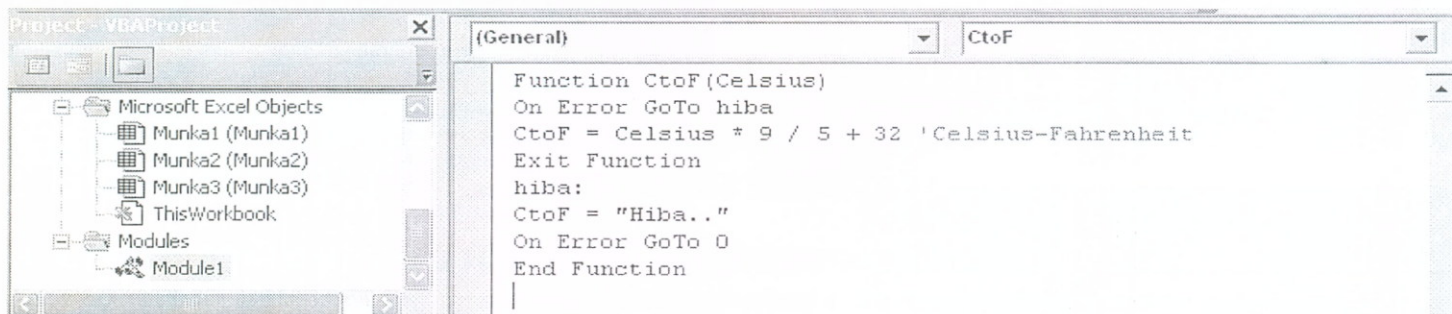
A függvénymakró az eredményt a nevének megfelelő változóban adja.

Ha a függvénymakró nevét a PRIVATE kulcsszó előzi meg, akkor a függvény nem jelenik meg a Függvényvarázsló párbeszédablakában.

A függvénymakrókat nem kell elindítani, mert attól az időponttól kezdve, hogy a függvénymakrókat tartalmazó Excel állományt megnyitjuk az működik, mindaddig, míg az állományt be nem zárjuk. A függvénymakrókat érdemes Bővítőmennybe (.XLA) helyezni.

Példa egy függvénymakróra

A következőkben bemutatjuk egy **CtoF** nevű függvénymakró létrehozásának lépéseit. A függvénymakró Celsius fokban beadott értéket Fahrenheit-be konvertál. Az állományt .XLA-ként mentettük el.

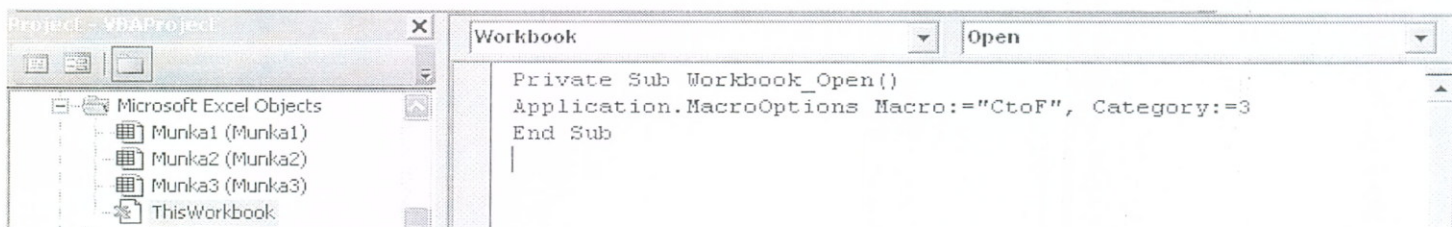


The screenshot shows the VBA Editor with the Project Explorer on the left and the VBA Editor window on the right. The Project Explorer shows a project named 'Project - VBAProject' with a folder 'Microsoft Excel Objects' containing 'Munka1 (Munka1)', 'Munka2 (Munka2)', 'Munka3 (Munka3)', and 'ThisWorkbook', and a folder 'Modules' containing 'Module1'. The VBA Editor window shows the code for the 'CtoF' function in the 'General' view.

```

Function CtoF(Celsius)
On Error GoTo hiba
CtoF = Celsius * 9 / 5 + 32 'Celsius-Fahrenheit
Exit Function
hiba:
CtoF = "Hiba.."
On Error GoTo 0
End Function

```



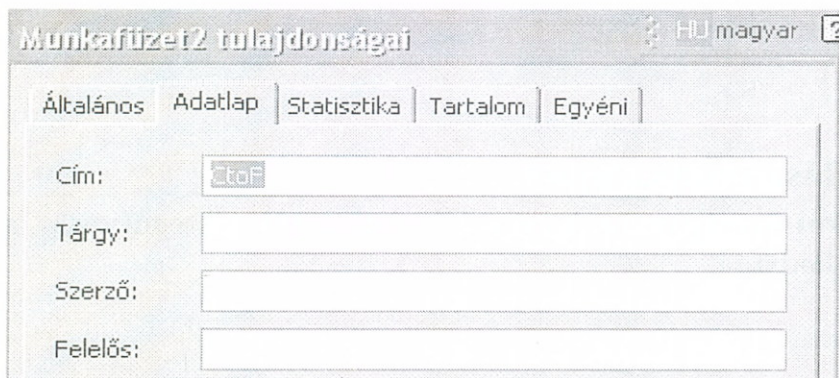
The screenshot shows the VBA Editor with the Project Explorer on the left and the VBA Editor window on the right. The Project Explorer shows a project named 'Project - VBAProject' with a folder 'Microsoft Excel Objects' containing 'Munka1 (Munka1)', 'Munka2 (Munka2)', 'Munka3 (Munka3)', and 'ThisWorkbook'. The VBA Editor window shows the code for the 'Workbook_Open' macro in the 'Workbook' view.

```

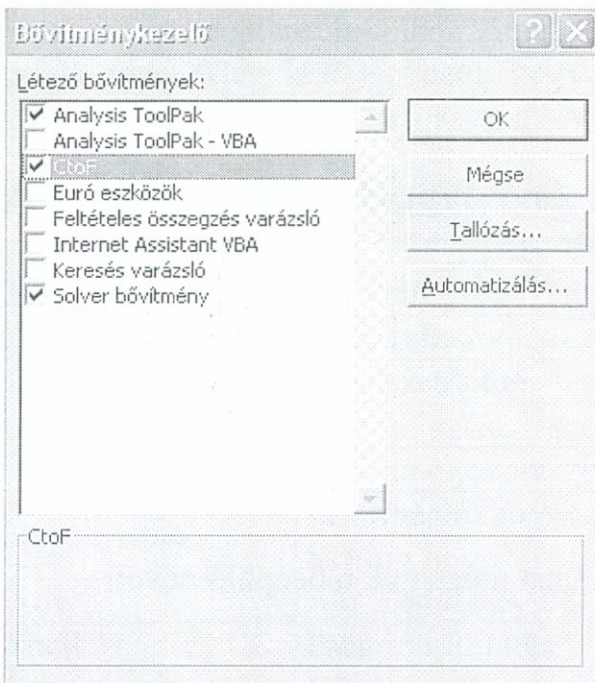
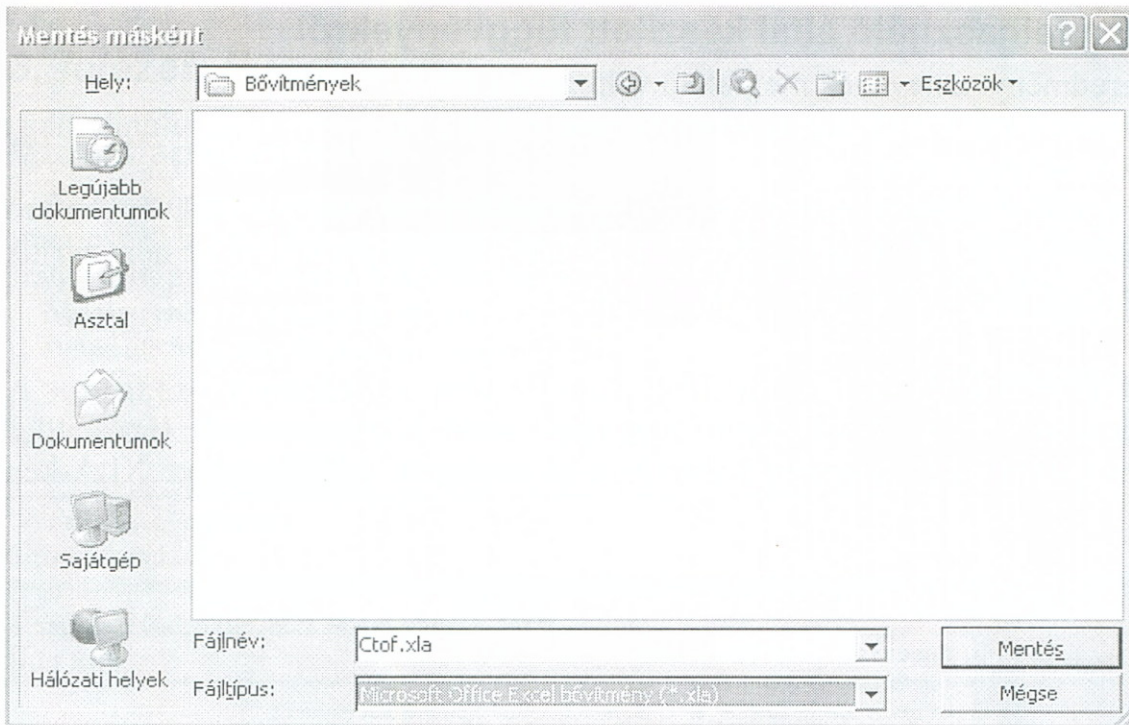
Private Sub Workbook_Open()
Application.MacroOptions Macro:="CtoF", Category:=3
End Sub

```

A **Fájl** menü **Tulajdonságok** parancsának párbeszédablakában megadjuk a beépülő nevét:

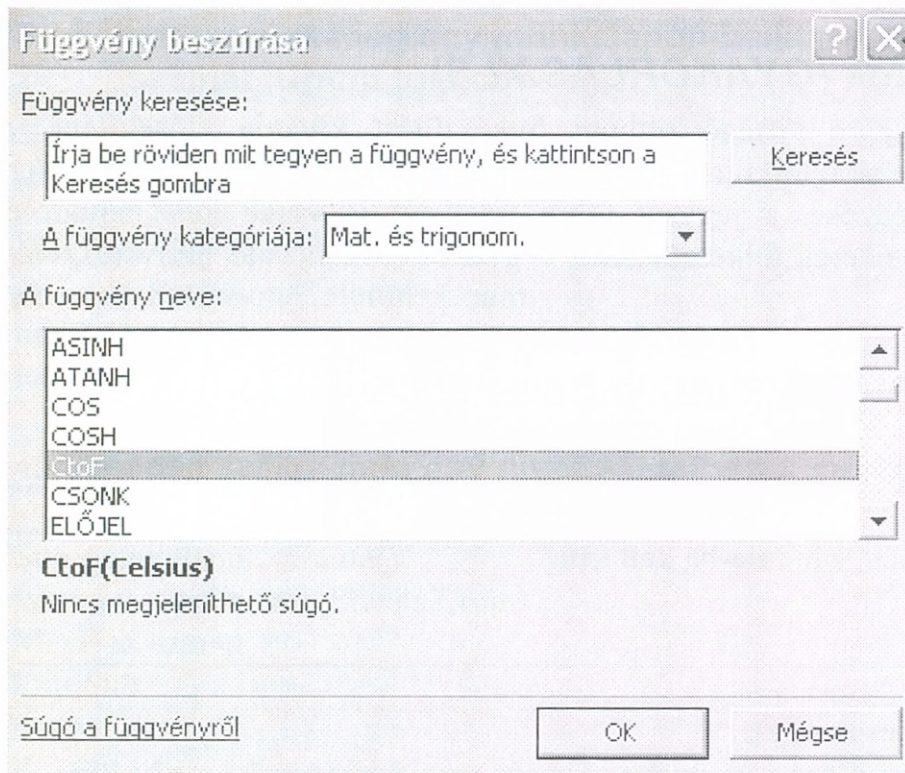


The screenshot shows the 'Munkafüzet2 tulajdonságai' dialog box with the 'Általános' tab selected. The dialog box has a title bar with 'HU magyar' and a help icon. The tabs are 'Általános', 'Adatlap', 'Statistika', 'Tartalom', and 'Egyéni'. The 'Általános' tab contains four text boxes: 'Cím:' with the value 'CtoF', 'Tárgy:', 'Szerző:', and 'Felelős:'.

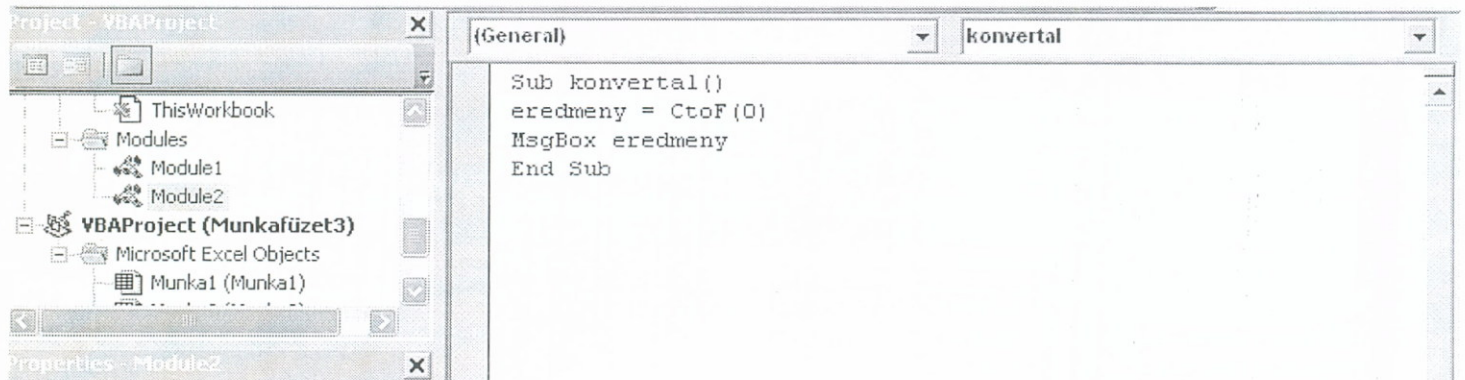


A CtoF függvénymakró felhasználása munkalapon

Mint az látható, a függvényt a **Matematikai és trigonom.** kategóriába helyeztük (category:=3), a Függvény varázslóval tehát bevihető a cellákba.



A CtoF függénymakró felhasználása parancsmakrókban



9. fejezet: Gyakorló feladatok (GYAKORLAS.XLS)

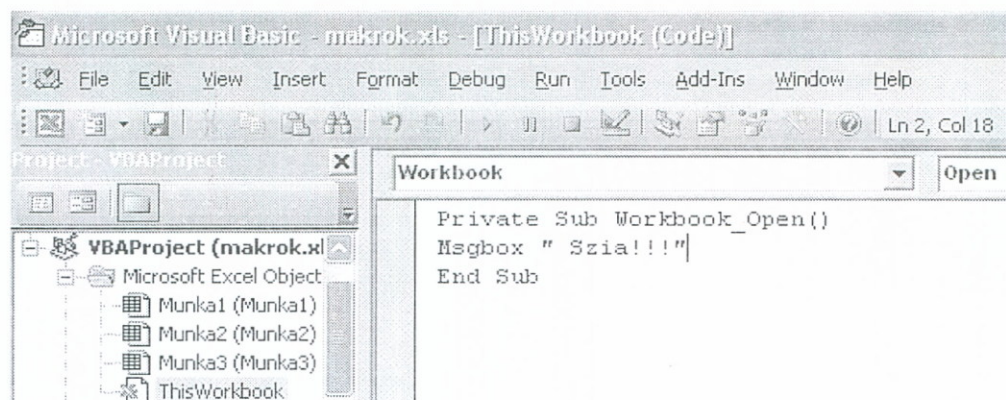
Jegyzetünk utolsó fejezete a vizsgára történő otthoni felkészülést kívánja elősegíteni. A **GYAKORLAS.XLS** nevű állomány nem tartalmazza ezen fejezet összes eljárását. Vigyázat nem elegendő csak ezen feladatok elvégzése, a jegyzet többi feladatát is végre kell hajtani és természetesen nem nélkülözhető a sikeres felkészüléshez a gyakorlatokon való aktív részvétel sem.

1. Feladat: készítsünk egy olyan makrót, amely a munkafüzet megnyitásakor automatikusan elindul:

```
Sub Auto_Open()
'Az Auto_Open egy kiemelt makrónév, Module-ba kell írni.
Msgbox "Szia!!!"
End Sub
```

“Korszerűbb” megoldás (Workbook esemény):

```
Private Sub Workbook_Open()
Msgbox " Szia!!!"
End Sub
```



2. Feladat: készítsünk egy olyan makrót, amely a fájl bezárásakor működik

```
Sub auto_close()
'Az Auto_Close egy kiemelt makrónév, Module-ba kell írni.
MsgBox "Vége!!!"
End Sub
```

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
MsgBox " Szia!!!"
End Sub
```

3. Feladat: adjuk meg egy munkalapon belül a kiválasztott sorok és oszlopok számát (Count tulajdonság):

```
Sub Kivalasztott_sorok_es_oszlopok_szama()  
szama = Selection.Rows.Count  
MsgBox "Kiválasztott sorok száma: "& szama  
szama = Selection.Columns.Count  
MsgBox "Kiválasztott oszlopok száma: "& szama  
End Sub
```

4. Feladat: Adjuk meg egy munkafüzetben a munkalapok számát (Sheets.Count):

```
Sub munkalapok_szama()  
szama = Application.Sheets.Count  
MsgBox szama  
End Sub
```

5. Feladat: képezzünk egy MsgBox-ban új sort (vbCrLf):

```
Sub ketsoros()  
MsgBox "Első sorban" & vbCrLf & "Második sorban"  
End Sub
```

6. Feladat: írjunk egy olyan eljárást, amely a fájl mentése előtt az aktív munkalap A1-es című cellájába beviszi az aktuális dátumot (Date függvény):

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)  
Range("A1") = date  
End Sub
```

7. Feladat: vizsgáljuk meg az aktív cella tartalmát, és minősítsük azt:

```
Sub ellenorzes()  
If Application.IsText(ActiveCell) = True Then  
MsgBox "szöveg"  
ElseIf ActiveCell = "" Then  
MsgBox "üres cella "  
ElseIf ActiveCell.HasFormula Then  
MsgBox "képlet"  
ElseIf IsDate(ActiveCell.Value) = True Then  
MsgBox "dátum"  
ElseIf Application.IsLogical(ActiveCell) Then  
MsgBox "logikai"  
ElseIf IsNumeric(ActiveCell) = True Then  
MsgBox "szám"  
Else  
End If  
End Sub
```

8. Feladat: adjuk meg az aktív cella pozícióját:

```
Sub poz()  
sor = ActiveCell.Row  
oszlop = ActiveCell.Column  
Msgbox "sor: " & sor & " oszlop: " & oszlop  
End Sub
```

9. Feladat: töröljük a kijelölt tartomány üres sorait, a vizsgálat a tartomány balszélső oszlopában történjék:

```
Sub uresek_torlese()  
Rng = Selection.Rows.Count  
ActiveCell.Offset(0, 0).Select 'Bal felső cella  
Application.ScreenUpdating = False  
For i = 1 To Rng  
If ActiveCell.Value = "" Then 'üres  
Selection.EntireRow.Delete 'sor törlése  
Else  
ActiveCell.Offset(1, 0).Select 'következő sor  
End If  
Next i  
Application.ScreenUpdating = True  
End Sub
```

10. Feladat: töröljük a tartományneveket (Names gyűjtemény) a fájlból:

```
Sub Nevek_torlese()  
Dim nev As Name  
For Each nev In Names  
ActiveWorkbook.Names(nev.Name).Delete  
Next nev  
End Sub
```

11. Feladat: egy kijelölt oszlopban vizsgáljuk meg az adatokat. A duplikált adatok színe legyen piros, a vizsgálat oszlopírányban történjék, a program futtatása előtt az oszloptartományt jelöljük ki:

```
Sub Dupla_piros()  
Application.ScreenUpdating = False  
Rng = Selection.Rows.Count 'kiválasztott sorok száma  
For i = Rng To 1 Step -1  
myCheck = ActiveCell  
ActiveCell.Offset(1, 0).Select 'következő sor  
For j = 1 To i  
If ActiveCell = myCheck Then  
Selection.Font.Bold = True  
Selection.Font.ColorIndex = 3 'piros  
End If  
ActiveCell.Offset(1, 0).Select 'következő sor  
Next j  
ActiveCell.Offset(-i, 0).Select 'eggyel feljebb  
Next i  
Application.ScreenUpdating = True  
End Sub
```

12. Feladat: az A oszlop minden második celláját vizsgáljuk meg, hogy üres-e. Amikor üres cellához érünk a vizsgálatot hagyjuk abba. Amennyiben egy vizsgált cella nem üres, akkor a cella sorának celláit színezzük be szürkére:

```
Sub Minden_masodik_sor1()  
Range("A2").EntireRow.Select  
Do While ActiveCell.Value <> ""  
Selection.Interior.ColorIndex = 15  
ActiveCell.Offset(2, 0).EntireRow.Select  
Loop  
End Sub
```

```
Sub Minden_masodik_sor2()  
  Dim i As Integer  
  i = 2  
  Do Until IsEmpty(Cells(i, 1))  
    Cells(i, 1).EntireRow.Interior.ColorIndex = 15  
    i = i + 2  
  Loop  
End Sub
```

```
Sub Minden_masodik_sor3()  
  i = 0  
  Do  
    i = i + 2  
    If IsEmpty(Cells(i, 1)) Then Exit Do  
    Rows(i).Interior.ColorIndex = 15  
  Loop  
End Sub
```

- 13. Feladat:** rejtjük el az aktív munkafüzet első munkalapját (ne feledjük felfedni a 3. sorban található kóddal), a VeryHidden-nel elrejtett munkalapot Excel paranccsal nem lehet felfedni:

```
Sub elrejtes()  
  Sheets(1).Visible = xlSheetVeryHidden  
  'Sheets(1).Visible = True  
End Sub
```

- 14. Feladat:** szűrjünk be annyi sort, amennyit a felhasználó kér:

```
Sub beszuras()  
  Dim Rng  
  Rng = InputBox("Hány sort szűrjünk be?")  
  Range(ActiveCell.Offset(0, 0), ActiveCell.Offset(Rng - 1, 0)).Select  
  Selection.EntireRow.Insert  
End Sub
```

- 15. Feladat: a kiválasztott tartomány celláiban lévő szövegek összes karakterét alakítsuk át kisbetűsre (LCase-kisbetűs, UCase-nagybetűs VBA függvény):**

```
Sub kisbetus()  
Dim cell As Range  
For Each cell In Selection.Cells  
If cell.HasFormula = False Then  
cell = LCase(cell)  
End If  
Next  
End Sub
```

- 16. Feladat: a kiválasztott tartomány celláiban lévő szövegek összes szavát alakítsuk át nagy kezdőbetűsre (Proper munkalapfüggvény):**

```
Sub Nagy_KezdoBetuk()  
Dim cell As Range  
For Each cell In Selection.Cells  
If cell.HasFormula = False Then  
cell = Application.Proper(cell)  
End If  
Next  
End Sub
```

- 17. Feladat: hozzunk létre egy lottószámot 1 és 49 között:**

```
Sub lotto()  
Randomize  
szam = Int((49 - 1 + 1) * Rnd + 1)  
MsgBox ("A Lottószám: ") & (szam)  
End Sub
```

- 18. Készítsen egy olyan makrót, amely 10sec. elteltével üzenetet küld:**

```
Sub timer()  
Application.Wait Now + TimeValue("00:00:10")  
MsgBox ("10 sec eltelt")  
End Sub
```

- 19. Feladat: feladatunk munkalap logikai végének meghatározása.**

Ez kézzel végezve történhet a CTRL+END billentyűkombinációval, vagy az **Irányított kijelölés** (F5) parancs párbeszédablaka **Utolsó cella** nevű választókapcsolójának segítségével. A funkciónak van egy kis "hibája", ha például beletörlünk a tartomány végébe, nem mutatja jól az

Excel a munkalap logikai végét. Feladatunk az, hogy vigyük a kurzort a tényleges utolsó kitöltött cellára.

A megoldáshoz felhasználjuk a **DARAB2** (angol neve: COUNTA) nevű függvényt, amely kiszámolja egy tartományban (ez most egy oszlop vagy egy sor) a kitöltött cellák számát, a vizsgálat akkor zárul, amikor a **COUNTA** függvény eredménye nem nulla. Ekkor olyan sorhoz vagy oszlophoz értünk, amely nem üres. Végül az eljárást léptessük le:

```
Public Sub LastCellsWithData()
' Irányított kijelölés: Utolsó cella
Set Utolso_cella = ActiveSheet.Cells.SpecialCells(xlLastCell)

sor = Utolso_cella.Row
Do While Application.CountA(ActiveSheet.Rows(sor)) = 0
    sor = sor - 1
Loop
utolso_sor = sor 'utolsó adatot tartalmazó sor

oszlop = Utolso_cella.Column
Do While Application.CountA(ActiveSheet.Columns(oszlop)) = 0
    oszlop = oszlop - 1
Loop
utolso_oszlop = oszlop 'utolsó adatot tartalmazó oszlop
ActiveSheet.Cells(utolso_sor, utolso_oszlop).Activate
End Sub
```

Az Eljárás leléptetése

1. Hozzuk létre a következő táblázatot, azon munkalap legyen aktív, amelyen ez a táblázat található:

	A	B	C	D	E	F	G
1							
2		1	1	1			
3		1	1	1			
4		1	1	1			
5		1	1	1			
6		1	1	1			
7		1	1	1			
8		1	1	1			
9		1	1	1			
10		1	1			2	2
11						2	2
12						2	2
13						2	2

2. Adjuk ki a **CTRL+END** billentyűkombinációt (munkalap logikai vége) a G13-as cella lesz aktív.

3. Töröljük ki az F10:F13 cellatartományt. Kiadva a **CTRL+END**-et, azt találjuk, hogy továbbra is a G13 a logikailag utolsó cella.
4. **ALT+F11** billentyűkombinációval váltsunk át a fejlesztői felületre (VBE).
5. Legyen a fénykurzor a **Sub LastCellsWithData** nevű eljárás teszőleges sorában.
6. Az eljárás utasításonkénti lelépetetésére az **F8**-as funkciógomb (**Debug/Step Into**) szolgál. Azon sor kiemelő színe, amely végrehajtásra kerül sárga. Hajtsuk végre az eljárást a While-ig. Álljunk egérrel a **sor** nevű változóra, ekkor az Excel elárulja annak értékét (10, azaz az utolsó olyan sor, ahol van adat, a 10. sor).

```
'19.feladat
Public Sub LastCellsWithData()
' Irányított kijelölés: Utolsó cella
Set Utolso_cella = ActiveSheet.Cells.SpecialCells(xlLastCell)

sor = Utolso_cella.Row
⇒ sor = 10 While Application.CountA(ActiveSheet.Rows(sor)) = 0
sor = sor - 1
```

7. A következő **F8** nyomásra belépünk a **Do...Loop** ciklusszerkezetbe, a ciklusokat általában nem léptetjük le (mivel túl sokáig tart azok végrehajtása), ezért nyomjunk **SHIFT+F8**-at (Debug/Step Over). A kiemelés a **Loop**-ot követő parancssorra kerül. Most valóban **F8**-at nyomjunk. Ha a már megismert módszerrel megnézzük az **utolso_sor** nevű változót, azt találjuk, hogy annak értéke 10.
8. A következő Do...Loop-ot szintén a SHIFT+F8-cal lépjük át. Az F8 következő megnyomásra az **utolso_oszlop** nevű változó értéket kap, amely most 4 (ez igaz, mivel az utolsó kitöltött oszlop a 4. azaz a D).
9. F8-cal végrehajtva az eljárás utolsó előtti utasítását, az aktív cella a D10-es lesz.

```
Loop
utolso_oszlop = oszlop 'utolsó adatot tartalmazó oszlop
⇒ ActiveSheet.Cells(utolso_sor, utolso_oszlop).Activate
End Sub
```

20. Feladat: üres sorok automatikus törlése

```
Private Sub Worksheet_Change(ByVal Target As Range)
'Üres sorok törlése.

'Végtelen ciklus megelőzése
Application.EnableEvents = False
'Több mint egy cella van kijelölve?
If Target.Cells.Count > 1 Then GoTo SelectionCode 'Igen
  If WorksheetFunction.CountA(Target.EntireRow) = 0 Then
    Target.EntireRow.Delete
  End If
Application.EnableEvents = True
Exit Sub

SelectionCode:
  If WorksheetFunction.CountA(Selection.EntireRow) = 0 Then
    Selection.EntireRow.Delete
  End If
  Application.EnableEvents = True
End Sub
```

21. Feladat: az A oszlop- illetve az első sor utolsó kitöltött cellájának megkeresése (ez kézzel a CTRL+FELNYÍL és CTRL+BALNYÍL kombinációkkal is elvégezhető, miután az A65536, illetve IV1-es cellára vittük a fénykurzort):

```
Sub LastCellInColumn()
  Range("A65536").End(xlup).Select
End Sub
```

Az első sor utolsó kitöltött cellájának megkeresése

```
Sub LastCellInRow()
  Range("IV1").End(xlToLeft).Select
End Sub
```

22. Feladat: előzzük meg, hogy a felhasználó a fájlt más néven mentse, azaz ne jelenjen meg a Mentés párbeszédablak (Cancel=true).

```
Private Sub Workbook_BeforeSave_
  (ByVal SaveAsUI As Boolean, Cancel As Boolean)
  If SaveAsUI = True Then Cancel = True
End Sub
```

23. Feladat: az aktív munkalapon az aktív cella sorában a cellák színe legyen szürke (Feltételes formázás):

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
Dim strRow As String
Cells.FormatConditions.Delete 'Feltételes formázás törlése

With Target.EntireRow 'kiválasztott sor
strRow = .Address
.FormatConditions.Delete
.FormatConditions.Add Type:=xlExpression, _
Formula1:="=DARAB2(" & strRow & ")>0"

.FormatConditions(1).Font.Bold = True
.FormatConditions(1).Interior.ColorIndex = 15
End With
End Sub
```

24. Feladat: oldjuk azt meg, hogy egy adott munkalapon a B1:F20 és a H1:L20 cellatartomány celláit ne lehessen kijelölni:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
Dim rngTilos As Range

Set rngTilos = Union(Range("B1:F20"), Range("H1:L20"))
If Intersect(Target, rngTilos) Is Nothing Then Exit Sub
Range("A1").Select
MsgBox "Ezen cellák nem jelölhetők ki: " _
& rngTilos.Address, vbCritical
End Sub
```

25. Feladat: készítsünk üdvözlő képernyőt (UserForm1 nevű úrlapon):

```
Private Sub Workbook_Open()
Application.OnTime Now + TimeValue("00:00:15"), "HideSplash"
UserForm1.Show
End Sub

Sub HideSplash()
Unload UserForm1
End Sub
```

26. Oldjuk meg azt, hogyha egy adott munkalapon, jobbgombbal kattintunk egy cellára, ne a rövidmenüt kapjuk, hanem ehelyett induljon el a Függvény varázsló

```
Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean)
On Error GoTo hiba
Cancel = True
Application.Dialogs(xlDialogFunctionWizard).Show
Exit Sub
hiba:
MsgBox "Hiba történt"
End Sub
```

27. Feladat: egy adott munkalapon vizsgáljuk meg az összes kitöltött cellát, ha a cella értéke kisebb mínusz száznál, akkor a szám színe legyen piros, egyébként pedig kék.

```
Sub Fest()
Dim cella As Range
Dim t As Double
t = Timer
For Each cella In ActiveSheet.UsedRange
If cella.Value < -100 Then
cella.Font.ColorIndex = 3 'piros
Else
cella.Font.ColorIndex = 5 'kék
End If
Next cella
MsgBox Round(Timer - t, 3) & " másodperc telt el"
End Sub
```

28. Feladat: feladatunk egy cellatartomány öt különböző színnel, felváltva történő színezése.

```
Sub ColorChange()
Dim dTime As Date
Static iCount As Integer
'Written by OzGrid.com
dTime = Now
Application.OnTime dTime + TimeValue("00:00:02"), "ColorChange"

Range("C3:G13").Interior.ColorIndex = Choose(iCount, 3, 36, 50, 7, 34)

If iCount = 5 Then
    iCount = 0
    Application.OnTime dTime + TimeValue("00:00:02"), "ColorChange", , False
End If
iCount = iCount + 1
End Sub
```

29. Feladat: egy cellatartomány számtartalmú celláinak értékét állításuk 0-ba

Két eljárást írunk. A második eljárásban, ahol a munkalap összes kitöltött celláját vizsgáljuk (UsedRange), azt is meg kell oldani, hogy ha nincs a munkalapon egyetlen számtartalmú cella sem, az eljárás ne álljon le hibával (On Error GoTo).

```
Sub ResetTest2()
For Each n In Range("A16:G28") ' Tartomány
    If IsNumeric(n) Then
        n.Value = 0
    End If
Next n
End Sub
```

```
Sub ResetValues()  
  On Error GoTo ErrorHandler  
  For Each n In ActiveSheet.UsedRange  
    If n.Value <> 0 Then  
      n.Value = 0  
    End If  
  TypeMismatch:  
  Next n  
ErrorHandler:  
  If Err = 13 Then      'Type Mismatch  
    Resume TypeMismatch  
  End If  
End Sub
```

30. Feladat: ha bármelyik munkalapon egy cellába 500-at vagy ennél nagyobb értékű számot gépelünk be, akkor annak a betűtípusát állítsuk be félkövérré (OnEntry), az OnEntry-t a Sűgő már nem tartalmazza:

```
Sub Auto_Open()  
  ActiveSheet.OnEntry = "Action"  
End Sub  
  
Sub Action()  
  If IsNumeric(ActiveCell) Then  
    ActiveCell.Font.Bold = ActiveCell.Value >= 500  
  End If  
End Sub  
  
Sub Auto_Close()  
  ActiveSheet.OnEntry = "" 'Az OnEntry leállítása  
End Sub
```

31. Feladat: a kijelölt tartományt nevezzük el:

```
Sub neve()  
  Selection.Name = "Maci"  
End Sub
```

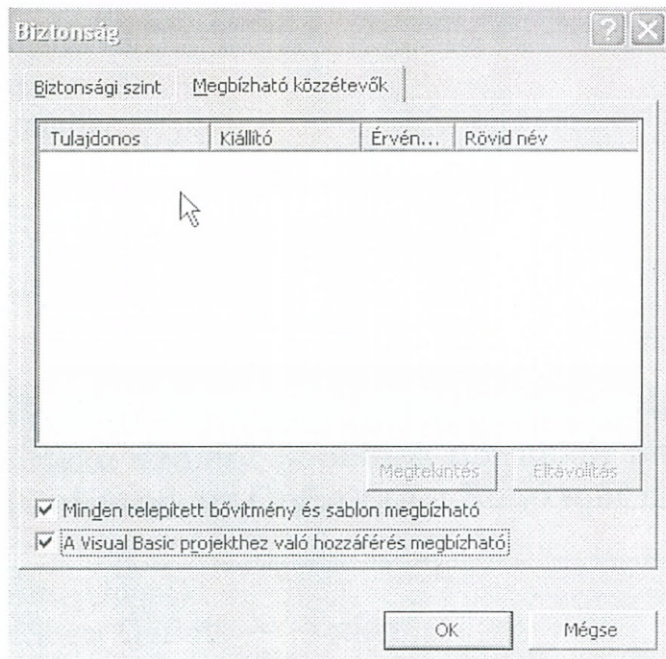
32. Feladat: hozzunk létre egy üzenetet az Állapotsoron (Statusbar)

```
Sub StatusBarExample()  
    Application.ScreenUpdating = False  
    ' Kikapcsolja a képernyő frissítést  
    Application.DisplayStatusBar = True  
    Application.StatusBar = "Kis türelmet kérek 1..."  
    ' Itt egy hasznos kódrészlet lehetne  
    Application.Wait Now + TimeValue("00:00:10")  
    Application.StatusBar = " Kis türelmet kérek 2..."  
    ' Itt egy hasznos kódrészlet lehetne  
    Application.Wait Now + TimeValue("00:00:10")  
    Application.StatusBar = False  
End Sub
```

33. Feladat: lássuk az élőlábban középen lent, a fájl nevét és elérési útját (az összes munkalapon):

```
Sub FilePathInFooter()  
    Dim i As Integer, FilePath As String  
  
    FilePath = ActiveWorkbook.FullName  
    For i = 1 To Worksheets.Count Step 1  
        Worksheets(i).PageSetup.CenterFooter = FilePath  
    Next i  
End Sub
```


34. Feladat: Hozzunk létre egy 10 lefutást biztosító ciklust a Do...Loop szerkezettel, a ciklusváltozó értékét írassuk ki az Immediate ablakba, ahová az eljárások váltunk át (a következő képkivágás mutatja a szükséges biztonsági beállítást, amelyet azután változtassunk vissza az eredetire):



```
Sub do_while()  
Dim start As Integer  
start = 10  
Do While start > 0  
start = start - 1  
Debug.Print start  
Loop  
Application.VBE.Windows.Item("Immediate").SetFocus  
End Sub
```