

**Devecz Ferenc – Juhász Tibor – Makány György – Végh András**

**Tanári kézikönyv az**  
**Informatika 10.**

**középiskolások számára**

tankönyvhöz

**Nemzeti Tankönyvkiadó, 2006**

Raktári szám: 16272

# Bevezetés

A tanári kézikönyv a Nemzeti Tankönyvkiadó 10. osztályos informatika tankönyvéhez készült (raktári szám: 16272). Felépítésében a tankönyv leckéit követi.

## Segédletek a tankönyvhöz

A 10. osztályos informatika tankönyvhöz munkafüzet (a továbbiakban Munkafüzet) és összefoglaló feladatsorokat tartalmazó füzet is készült (Feladatlapok). A tanári kézikönyvhöz mellékeljük a munkafüzet, illetve a feladatlapok feladatainak megoldását. A digitális formájú dokumentumokat a Nemzeti Tankönyvkiadó webhelyén találjuk meg ([www.ntk.hu](http://www.ntk.hu)).

A tanári kézikönyvben a munkafüzet feladataira az adott leckén belüli sorszámukkal hivatkozunk.

A Nemzeti Tankönyvkiadó webhelyéről letölthetők a tankönyv leckéihez, a munkafüzet és a feladatlap feladataihoz kapcsolódó példafájlok. A szövegben a továbbiakban nem jelöljük meg a példafájlok ezen forrását ([www.ntk.hu](http://www.ntk.hu)).

A tankönyv *Algoritmusok és adatok* fejezetéhez számos további kiegészítést, segédanyagot találunk a Nemzeti Tankönyvkiadó webhelyén ([www.ntk.hu](http://www.ntk.hu)).

Végezetül felhívjuk a figyelmet a Nemzeti Tankönyvkiadó webhelyén található, heti 1, illetve heti 2 órához illeszkedő tanmenetjavaslatokra.

A kollégáknak jó munkát, a diákoknak kellemes tanulást kívánnak

*a szerzők*

# Információs társadalom

## Az informatika fejlődéstörténete I.

### Információs korszakok

<b>Tanítási cél</b>	Annak megmutatása, hogy az egymást követő információs korszakok egyértelmű, de hullámzó fejlődési ívet írnak le az emberiség történetében. Milyen adatrögzítő, feldolgozó és továbbító (kommunikációs) technológiák és technikák jellemzik az egyes korszakokat?
<b>Fogalmak, definíciók</b>	Kommunikáció, jelrendszer, beszéd, írások, adathordozók, nyomtatás, távközlés (telefon, rádió, tévé).
<b>Tevékenységek</b>	Tanári bemutatás, beszélgetés, megvitatás.
<b>Megjegyzések</b>	Fontos az adatrögzítés és a kommunikáció általános összefüggéseinek bemutatása, megvitatása.
<b>Házi feladat</b>	Személyre szabott kutatómunka (dolgozat, kiselőadás, ill. prezentációkészítés) feladat adható az írás, a nyomtatás, a kommunikáció vagy a számítástechnika (lásd a következő anyagokat) gazdag történetéből.

### Kiegészítések

### Ellenőrző kérdések

1. Milyen előnyökkel járt az írás kifejlesztése?	Az ismeretek függetlenedtek az emlékezettől. Nagyon sok adatot pontosan lehetett rögzíteni, hosszú időre megőrizni. A távolsági kommunikáció fejlődött. A hallható és a látható kommunikáció kapcsolata fejlődött.
2. Milyen adatokat volt célszerű már az ókorban is írásban rögzíteni?	Célszerű volt írásban rögzíteni a gazdasági adatokat. A birtoklásra vonatkozó adatokat. A különféle tananyagokat, a tudományos és filozófiai ismereteket.
3. Miért tartjuk nagy találmánynak a nyomtatást?	A nyomtatás az „írás gépesítése”, ami tömeggyártást és alacsony árakat eredményezett. Nagy tömegek számára hozzáférhetővé tette az olvasást.
4. Miért állíthatjuk, hogy informatikai szempontból a távolságok megszűntek (a telefon és a rádió elterjedése óta)?	Természetesen nem a fizikai távolság szűnt meg. A rádióhullámok közel 300 000 km/s sebességgel terjednek, a Föld kerülete mindössze 40 000 km. A jel a másodperc tört része alatt eljuthat a Föld bármelyik részére. Ez az ember reakcióidejével azonos nagyságrendű, illetve kisebb idő. Ha valamit fél másodperc alatt elérhetünk, az emberi mércével „nagyon közel van”, hiszen a kommunikáció szempontjából az üzenetváltás ideje a lényeges tényező, nem a fizikai távolság. Ebben az értelemben mondhatjuk, hogy a távolságok megszűntek.
5. Kiscsoportban készítsenek előadást az ókori írásokról!	Keressünk forrásokat internetes keresőkkel és internetes katalógusokkal.
6. Készítsen kiselőadást a magyar rovásírásról!	Keressünk forrásokat internetes keresőkkel és internetes katalógusokkal.
7. Gyűjtsön forrásanyagot a telefon vagy a rádió feltalálásáról!	Keressünk forrásokat internetes keresőkkel és internetes katalógusokkal.

## Az informatika fejlődéstörténete II.

### A számítástechnika kezdetei

<b>Tanítási cél</b>	A számítástechnika előtörténetével – a mechanikus és elektromechanikus számológépek és számítógépek kifejlesztésével – prezentálhatjuk, hogy a mégoly forradalmian új informatikai találmány, mint a Neumann-féle elektronikus számítógép sem született előzmények nélkül. Az ember sokszori próbálkozással, részeredmények és kudarcok útján jut el korszakalkotó találmányokhoz, ha megérették tudományos, technikai, társadalmi és a gazdasági feltételek.
<b>Fogalmak, definíciók</b>	Számológép, lyukkártya, program, adat, bináris kód, operatív memória, háttértár, vezérlőegység, aritmetikai-logikai egység, bemeneti és kimeneti egységek, számítógép, Neumann-elv.
<b>Tevékenységek</b>	Tanári bemutató, kiselőadások.
<b>Megjegyzések</b>	Ha kevés az óraszám, az első két-három óra anyagát össze is vonhatjuk, de a kiselőadásokat célszerű megtartani.
<b>Házi feladat</b>	Személyre szabott kutatómunka (dolgozat, kiselőadás, ill. prezentációkészítés) feladat adható az írás, a nyomtatás, a kommunikáció vagy a számítástechnika (lásd a következő anyagokat) gazdag történetéből.
<b>Kiegészítések</b>	

### Ellenőrző kérdések

1. Mit talált fel Blaise Pascal?	Összeadó kivonó mechanikus számológépet.
2. Könyvtárban vagy az interneten keressen adatokat Blaise Pascalról!	
3. Könyvtárban vagy az interneten nézzen utána ki volt Gottfried Wilhelm Leibniz?	
4. Milyen találmányon dolgozott Charles Babbage?	Az első mechanikus számítógépen. Tk. 7. o.
5. Mikor építették az ENIAC-ot? Mit tud róla?	1943-1946. Tk. 7. o.
6. Ismertesse a Neumann-elvet!	Tk. 8. o.
7. Készítsen kiselőadást a számítástechnika történetének egy nevezetes eseményéről!	

## Az informatika fejlődéstörténete III.

### A számítógép-generációk és az Internet

<b>Tanítási cél</b>	Annak megláttatása, hogy a számítógép-generációk és az internet robbanás-szerű minőségi fejlődés eredményezett az adatkezelés, adatfeldolgozás, adattárolás és a kommunikáció területén.
<b>Fogalmak, definíciók</b>	Relé, elektroncső, ferritgyűrűs tároló, mágnesszalag, mágneslemez, gépi kód, magas szintű nyelv, tranzisztor, integrált áramkör, processzor, hálózat, internet.
<b>Tevékenységek</b>	Tanári prezentáció, eszközök esetleg fényképek bemutatása. Kiselőadások.
<b>Megjegyzések</b>	Az egymást követő számítógép-generációk a technológiai váltásokra alapozott gyors mennyiségi és minőségi fejlődést hoztak, miközben az alapvető elvek nem változtak.
<b>Házi feladat</b>	Személyre szabott kutatómunka (dolgozat, kiselőadás, ill. prezentációkészítés) feladat adható az írás, a nyomtatás, a kommunikáció vagy a számítástechnika (lásd a következő anyagokat) gazdag történetéből.
<b>Kiegészítések</b>	

### Ellenőrző kérdések

1. Melyek a számítógép-generációk legfőbb jellemzői?	Tk. 9-10. o.
2. Eredetileg milyen célkitűzésekkel építették az Internet elődjét, az ARPANET-et?	A számítógép-hálózat akkor is kommunikációképes legyen, amikor a hálózat egy része tönkremegy (támadás következtében). Tk. 10. o.
3. Milyen protokollt használ az internet?	TCP/IP
4. Mit jelent a http?	Tk. 11. o.

## Az informatika fejlődéstörténete IV.

### Az IKT forradalma

<b>Tanítási cél</b>	Az információs társadalmak kialakulását az IKT forradalma tette lehetővé, ezért célszerű a technológiát és a trendeket áttekinteni.
<b>Fogalmak, definíciók</b>	IKT, algoritmikus gondolkodás, internet, web 2.0, Moore-törvény,
<b>Tevékenységek</b>	Megbeszélés, bemutatás, kiselőadások.
<b>Megjegyzések</b>	Célszerű kiaknázni a tanulók egy részének jó motiválási lehetőségeit.
<b>Házi feladat</b>	Személyre szabott kutatómunka (dolgozat, kiselőadás, ill. prezentációkészítés) feladat adható az írás, a nyomtatás, a kommunikáció vagy a számítástechnika (lásd a következő anyagokat) gazdag történetéből.
<b>Kiegészítések</b>	A közeljövő információs technológiáinak szédületes lehetőségeinek bemutatása, megbeszélése.

### Ellenőrző kérdések

1. <i>Az informatikának milyen korszaka(i) jellemzik a XX. század második felét és a XXI. század elejét?</i>	A számítógépek forradalma és a világháló.
2. <i>A kommunikációs lehetőségek szempontjából hogyan jellemezné az internetet?</i>	Világméretű számítógépes hálózat. Sok adó és sok vevő modellt valósít meg. Többféle protokollt használhatunk: http, ftp, gopher,
3. <i>A Moore-törvény szerint milyen jellegű a számítástechnikai ipar fejlődése?</i>	A processzorok vonatkozásában gyors, exponenciális a fejlődés évtizedek óta. A memóriák, háttértárak is hasonló ütemben fejlődtek.
4. <i>Mit jelent a web 2.0?</i>	Tk. 13. o.
5. <i>Milyen (IKT) technológiai trendeket figyelhetünk meg a mai információs társadalomban?</i>	A miniaturizálás exponenciális üteme. A műveleti sebesség és a tárolókapacitás gyors növekedése. A személyi számítógépek elterjedése. A grafikus rendszerek (operációs rendszerek és alkalmazások) fejlődése, elterjedése. Az IKT térhódítása a társadalomban. Tk. 13 - 14. o.

## Az információs társadalomról

<b>Tanítási cél</b>	Az információs társadalom mibenlétének megértése.
<b>Fogalmak, definíciók</b>	Az információ (adat) mint árú, hálózati társadalom, hálózati gazdaság, globális gazdaság, elektronikus demokrácia.
<b>Tevékenységek</b>	Kiselőadások, vitafórum, csoportmunka.
<b>Megjegyzések</b>	Célszerű megvitatni a munkafüzetben vázolt, illetve ezekhez kapcsolódó problémákat.
<b>Házi feladat</b>	Robotokról szóló film, vagy site-ok megtekintése.
<b>Kiegészítések</b>	

### Ellenőrző kérdések

1. <i>Megváltozik-e a kapitalizmus termelési rendszere az információs társadalomban?</i>	Nem valószínű.
2. <i>Miért nem szimpatikus a társadalom Orwell-modellje?</i>	Mert informatikai társadalmi diktatúra.
3. <i>Mi kellene ahhoz, hogy egy társadalomban érvényesülhessen az elektronikus demokrácia?</i>	A tanulók fejtsék ki a véleményüket!
4. <i>A környezetvédelem szempontjából előnyös vagy hátrányos lesz-e a fejlett információs társadalom?</i>	Előnyös, de a helyzet ellentmondásos.
5. <i>Keressen forrásokat az interneten és készítsen kiselőadást az e-tanulásról (e-learning)!</i>	
6. <i>Keressen forrásokat az interneten és készítsen kiselőadást az e-közigazgatásról!</i>	
7. <i>Keressen forrásokat az interneten és készítsen kiselőadást az e-gazdaságról!</i>	
8. <i>Ismertesse Castells információs társadalomról alkotott modelljét!</i>	Tk. 15. o. Bővebben: az interneten.
9. <i>A háztartásoknak kb. hány százaléka rendelkezik internet-hozzáféréssel Magyarországon?</i>	2005-ben kb. 40 %-a. Jelenleg:
10. <i>Hány ember tekinthető internet-használónak Magyarországon?</i>	2005-ben kb. 3 millió fő. Jelenleg:

## Robotok

<b>Tanítási cél</b>	A személyi és más robotok lehetőségeinek feltárása, megbeszélése.
<b>Fogalmak, definíciók</b>	Robot, ipari robot, oktatórobot, harci robot, robot jármű, kiborg, android, reklám robot.
<b>Tevékenységek</b>	Bemutató, kiselőadás, beszélgetés. Film megtekintése.
<b>Megjegyzések</b>	Az informatika jövőbeni fejlődési trendjeinek egyike szinte biztosan a hétköznapiakban is használható robot lesz. A robot jó példa az intelligens informatikai (és kommunikációs) eszközre is.
<b>Házi feladat</b>	Robotokkal kapcsolatos játék vagy ismeretterjesztő film megtekintése.
<b>Kiegészítések</b>	

### Ellenőrző kérdések

1. Milyen robotokat ismer a filmekből?	R2D2, C3PO ...
2. Milyen feladatot látott el a Sojourner robot?	Tk. 17. o.
3. Mi az ipari robot? Mire alkalmas	Tk. 17-18. o.
4. Feltehetőleg milyenek lesznek, és mit tudnak majd a személyi robotok?	Tk. 18. o.
5. Mire használjuk az oktatórobotot?	A robotokkal kapcsolatos tanulásra. Később talán mindenféle tanulásra, ez a „robotpedagógus”. (Már ma is létezik, csak biológiai alapon.)



## Jog és etika az informatikában

<b>Tanítási cél</b>	Megismerni a szoftverekre és más szellemi termékekre vonatkozó legfontosabb felhasználói (és szerzői) jogokat. Megismertetni az IKT eszközök testi és lelki veszélyeit.
<b>Fogalmak, definíciók</b>	Szellemi termék, szerzői jog, felhasználói jog, szabad szoftver, freeware szoftver, shareware szoftver, félig szabad szoftver, kereskedelmi szoftver, licenc, ergonómia és munkavédelem, az IKT veszélyei, függőség, adatvédelem.
<b>Tevékenységek</b>	Licencek tanulmányozása, egy ergonomikus munkahely bemutatása. A számítógépes játékfüggőség káros hatásainak és elkerülésének megbeszélése.
<b>Megjegyzések</b>	A játékfüggőség megbeszélését, tiltását sok tanár elhanyagolja, pedig majdnem annyira káros, mintha a fiatal alkoholista lenne. Az agresszív játékok különösen veszélyesek, ha nincs erős környezeti (valóság) kontrol.

### Házi feladat

### Kiegészítések

### Ellenőrző kérdések

1. <i>Mi az adatvédelmi törvény lényege?</i>	Tk. 19. o.
2. <i>Milyen védelmet nyújt a szerzői jog?</i>	Tk. 19. o.
3. <i>A felhasználói jog szempontjából milyen szoftvereket ismer?</i>	Tk. 20. o.
4. <i>Mi az ergonómia?</i>	Tk. 20. o.
5. <i>Milyen veszélyeit tapasztalhatjuk a számítógépes munkahelynek?</i>	Tk. 20. o.
6. <i>Miért rossz a számítógépes függőség?</i>	Egyes számítógépes tevékenységek erős függőséget alakíthatnak ki egyes fiatalokban. Ez éppen úgy kerülendő, mint minden túlzó időrabló és káros szenvedély. Tk. 20. o.
7. <i>Kiscsoportban gyűjtsenek adatokat a számítógépes függőségről!</i>	

# Táblázatkezelés

## Bevezetés

A táblázatkezelés feldolgozásának szerkezete a tankönyv többi fejezetéhez hasonló: egy- egy témakört 1-2 tanítási órára terveztünk. Mivel a fejezet erősen gyakorlati jellegű, az ismeretszerzést konkrét, életszerű, egymáshoz kapcsolódó feladatok megoldásai köré szerveztük. A feldolgozás során fokozatosan, egyre jobban számítottunk a tanulók önálló munkájára, ezért az egyes témakörök kidolgozása a fejezet vége felé haladva nem annyira aprólékos, részletes, mint az elején.

A haladási ütem persze nagyon különböző lehet a tanulók előképzettségétől, képességeitől, a rendelkezésre álló órakerettől függően. A könyvhöz tartozó munkafüzetrel az egyéni, esetleg az otthoni tanuláshoz szeretnénk volna segítséget adni, különösen olyan esetekben, amikor az iskolai foglalkozásokkal a szükséges intenzitást, a gyakorlatszerzést nem lehet maradéktalanul biztosítani.

A fejezet tartalmát az érettségi kompetenciákhoz igazítottuk. Követelmény a táblázatok esztétikus formázása, az adattípusok ismerete, képletek alkalmazása a táblák celláiban, ezek másolása és mozgatása, munkalapfüggvények használata a kifejezésekben, a grafikonszerkesztés és függvényábrázolás. Az adatbázisok használatának előkészítéseként foglalkozunk az adatbázistáblák rendezésével, szűrésével, webes lekérdezésével és közzétételével. Az emelt és középszintű követelményrendszert a feldolgozott feladatok egyaránt lefedik.

A tömbfüggvényeket csak a gyakoriságok számítása miatt és az ehhez szükséges mértékben tárgyaltuk. Ennek hiányában a hisztogramok elkészítése aránytalanul bonyolult lett volna.

A szükséges matematikai háttérrel feltételeztük, hogy az legfeljebb a 10. évfolyam szintjén áll rendelkezésünkre. A tesztadatok generálását és a polárkoordinátákkal adott görbék ábrázolását ezért kiegészítő anyagnak szántuk, de ha a matematikai ismeretek lehetővé teszik, ezek is tárgyalhatók.

Ebben a segédletben megadtuk a leckékre bontott tanítási célokat, a tananyagban előforduló új fogalmakat és kifejezéseket, a szükséges tevékenységeket, valamint a témához kapcsolódó ellenőrző kérdések megoldásait.

A házi feladatok csak ajánlások. A rendelkezésre álló tanítási órakeret és a konkrét pedagógiai szituáció függvényében esetleg bővíthetők, módosíthatók, illetve el is hagyhatók. A témakör erősen gyakorlati jellege miatt azonban az önálló tanulói munkától természetesen nem lehet eltekinteni.

A feladatlapok a megoldások követelményeit tekintve a tankönyvet, illetve a munkafüzetet követik és két, akár külön-külön is feladható részből állnak. A négy sorozat mindegyikében az első hét feladat tartozik az első részhez, amelynek megoldásához nem szükséges, illetve nem is ajánlott gépet használni. A nyolcadik és a nagyobb sorszámú feladatok kidolgozása a táblázatkezelő használatát igényli. A feladatsorok megoldásához a teljes fejezet ismeretanyagára szükség lehet, ezért a dolgozatok megírását a téma zárását követően javasoljuk. A forrásadatokat a Központi Statisztikai Hivatal adatbázisából importáltuk, ezeket a dolgozatírás idejére természetesen aktualizálni lehet.

## A táblázatkezelők funkciói, munkakörnyezete, a táblázatok elemei

<b>Tanítási cél</b>	A táblázatkezelő használatának előkészítése, ismerkedés az alkalmazói környezettel. A tanulók előzetes ismereteinek felmérése.
<b>Fontos fogalmak, definíciók</b>	Munkafüzet, munkalap, sorok, oszlopok, cellák. Aktív cella, kijelölés, egyesítés és megszüntetése, tartományok. Cellák mozgatása, másolás, áthelyezés, beszúrás, beillesztés.
<b>Tevékenységek</b>	A táblázatkezelő alkalmazás specifikumainak megismerése, a menürendszerének összehasonlítása más alkalmazások környezetével. Az operációs rendszer paramétereinek (dátum-, idő-, pénznem, tizedesformátumok) hatása a táblázatkezelőre, az alkalmazás dokumentumainak betöltése és mentése. Mozgás és mozgatás a munkalapokon és a munkafüzetben.
<b>Megjegyzések</b>	A témakör tárgyalása során fel kell mérni a tanulók előzetes ismereteit. Mivel az ismétlődő kérdéssor nem feltételezi a táblázatkezelő ismeretét, az ellenőrző kérdéseket használhatjuk ebből a célból. A téma feldolgozásában használjuk ki a szövegszerkesztőkben már megismert fogalmakat és tevékenységeket, például a cellatartalom formázását.
<b>Házi feladat</b>	A tanulók előképzettségének, illetve a rendelkezésre álló órakeret függvénye. A táblázatkezelésben teljesen kezdő tanulók számára feltétlenül ajánljuk a munkafüzet feladatainak kidolgozását.

### Az ellenőrző kérdések és feladatok megoldásai

1. Melyek a táblázatkezelő programok fontosabb funkciói?	A cellák adataival műveletek végezhetők, az adatok grafikonokon ábrázolhatók, a táblázatok adatai szelektíven is visszakereshetők.
2. Hogyan nevezhető át a munkalap és a munkafüzet?	A munkafüzet egy adatfájl, az operációs rendszer szervizprogramjával nevezhető át. A munkalapnév a dokumentumablak alján, a munkalapfülről kért helyi menüvel változtatható meg.
3. Hogyan olvashatjuk ki az aktív cella címét?	Az aktív cella címe vagy a neve a szerkesztőléc bal oldalán, a névmezőben látható
4. Hogyan hivatkozhatunk a tartományokra?	Bal felső cella címe:jobb alsó cella címe
5. Miről ismerhető fel az egyesített cellatartomány?	A rácsvonalak törlődnek a belsejéből, a cellakurzor a tartomány határszakaszaira módosul, ha aktívvá válik.
6. Hogyan hivatkozhatunk az egyesített cellákra?	Bal felső cellájának a címével.
7. Mi lesz a Legszélesebb kijelölt parancs hatása?	A kijelölt cellában vagy tartományban az oszlopszélesség a leghosszabb cellaszöveg hosszára módosul
8. Miben különbözik a beszúrás a beillesztéstől?	Beszúráskor az új cella, tartomány, sor vagy oszlop számára helyet kell készíteni a régi adatok eltolásával, beillesztéskor az új adatok felülírják a régieket az érintett cellákban.
9. Soroljon fel néhányat a törlési lehetőségek közül?	A Del billentyűre cella tartalma törlődik, a formátuma nem. A Szerkesztés menü Tartalom törlése parancsával szelektív törlést kérhetünk, míg a helyi menüből is kérhető Törlés.. paranccsal a teljes cella törölhető a szomszédos cellák áthelyezésével.
10. Mi a munkafüzet alapértelmezett kiterjesztése?	Excelben xls, az Open Office.org-ban ods.

## Adattípusok, formátumok, adatbevitel

<b>Tanítási cél</b>	<p>Megismerjük a táblázatkezelő adattípusainak legfontosabb jellemzőit. Az automatikus típusfelismerés működése, az adattípusok explicit megadása, megváltoztatása.</p> <p>A táblázatkezelő eszközszerének néhány fontos jellemzője: automatikus kitöltés, munkalapfüggvények, munkalapvédelem, munkafüzet szintű nevek használata.</p>
<b>Fontos fogalmak, definíciók</b>	<p>Adattípusok és formátumok: szám, szöveg, logikai, hibajelzések. A pénznem, a százalék- és dátum-idő számformátum. Speciális adatformátumok, cellatartalom és megjelenítés.</p> <p>Munkalapfüggvény, automatikus kitöltés, zárolás, lapvédelem, munkafüzet szintű név.</p>
<b>Tevékenységek</b>	<p>Ezt a tanítási egységet elméleti előkészítésnek szántuk. Ezért nem lehet célunk a fogalmak készség szintű elsajátíttatása, ehhez további, elsősorban gyakorlati munkára lesz szükség. A foglalkozás során a fogalmak meghatározására és konkrét példákkal történő bemutatásra szorítkozunk.</p>
<b>Megjegyzések</b>	<p>A táblázatkezelő automatikus típusfelismerése nem mindig segíti a precíz fogalomalkotást. Feltétlenül mutassunk be a szöveges adat bevitelének aposztróffal való kényszerítését, a szám- és dátumformátumok közti kapcsolatot, a tizedesjegyek tárolása és kijelzése közti különbséget.</p> <p>A munkafüzet szintű nevek törlése a menürendszerből logikusan nem következik, hiszen a Nevek megadása ponttal kell indulni. Ezért fontosnak tartjuk a név törlésének bemutatását is.</p>
<b>Házi feladat</b>	<p>A munkalapfüggvények és a táblázatkezelő kifejezéseinek típuskezelése különbözik. Ajánljuk a SZUM függvény és az összeadás művelete közti különbség megértéséhez a munkafüzet 2. feladatát.</p>
<b>Kiegészítések</b>	<p>Ha a tanulók rendelkeznek valamilyen szintű programozási előismerettel, a programnyelv és a táblázatkezelő adattípusait és ezek kezelését feltétlenül ajánlott összehasonlítani. Azonban programozási ismeretek hiányában is érdemes kitérni röviden a formális nyelvek típuskezelési tulajdonságaira, összehasonlítva ezeket a táblázatkezelő típuskezelésével.</p>

### Az ellenőrző kérdések és feladatok megoldásai

1. Sorolja fel a táblázatkezelő alapadattípusait!	Szám, szöveg, logikai, hibajelzés.
2. Mi történik, ha Szöveg formátumú cellába számot írunk?	A táblázatkezelő balra igazítja a cellában, de számként értelmezi.
3. Hogyan vihetünk be számnak látszó adatot szöveggént az Általános/Standard formátumú cellákba?	Aposztróffal kell kezdeni a beírását.
4. Nevezze meg a táblázatkezelő logikai konstansait! Hogyan ismerhetjük fel őket a cellákban?	IGAZ és HAMIS. Nagybetűsre alakítva, középre igazítva jelennek meg a cellákban.
5. Nevezzen meg néhány speciális számformátumot!	Pénznem, százalék, könyvelői, tudományos, tört, egyéni.
6. Milyen tulajdonságai vannak a Százalék formátumnak?	A cellába írt értékek %-jellel lezárva jelennek meg a cellában. Valódi értékük a cellában megjelenő szám/100.
7. Hogyan adhatunk nevet a cellának vagy a tartományoknak?	A tartomány kijelölése után a szerkesztőléc bal oldalán lévő névmezben, vagy a Beszúrás→Név→Név megadása menüparancsokkal.
8. Hogyan változtathatjuk meg a neveket?	Beszúrás→Név→Név megadása→ Törlés után új nevet adhatunk.

9. Miért hívjuk a celláknak és tartományoknak adott neveket munkafüzet szintűnek?	Az egész munkafüzetben, annak minden munkalapján használhatók, a munkalap megadása nélkül.
10. Mi lesz a hatása, ha a cellák zárolt/védett jellemzőjét töröljük a Cellavédelem lapon?	Nem lehet védetté tenni őket a lapvédelem beállításával.

## Műveletek, kifejezések és képletek, formázás igazítással

<b>Tanítási cél</b>	Kifejezések és képletek megadásának elsajátítása. Táblaformátumok kialakítása.
<b>Fontos fogalmak, definíciók</b>	Tartományműveletek: kijelölés, egyesítés, közös rész képzése. Szövegműveletek: összefűzés, néhány fontosabb szövegfüggvény: BAL, JOBB, KÖZÉP. A táblázatkezelő műveleteinek elsőbbségi sorrendje. Típuskonverzió a műveletekben. A Formátum menü Igazítás lapjának lehetőségei.
<b>Tevékenységek</b>	A kifejezések megadásának gyakorlása. Egy adott táblázat formázása a Formátum menü segítségével.
<b>Megjegyzés</b>	A relációkat és logikai adatokat tartalmazó kifejezések és képletek a későbbi adatvizsgálatok, döntések nélkülözhetetlen eszközei. Ezért a szám- és szövegadatokkal végezhető műveletek mellett hagyjunk időt a logikai értéket adó kifejezések gyakorlására is. A kifejezések szerkezetének tárgyalását elméleti előkészítésnek szántuk. A hozzá kapcsolódó táblázat elkészítése azonban már gyakorlati anyag, amelynek feldolgozása során részben a tanulók önálló munkájára is számítunk.
<b>Kiegészítések</b>	A kifejezések megadása és a műveletek elsőbbségi sorrendje az adattípusokhoz hasonlóan szorosan kapcsolható a programozási ismeretek tanításához.

### Az ellenőrző kérdések és feladatok megoldásai

1. Milyen típusú adat lesz a hasonlító műveletek eredménye?	Logikai.
2. Mi lesz a Százalék formájú cellában, ha 10-et írunk be?	10%, vagy 0,1.
3. Van-e különbség az összeadás művelete és a SZUM munkalapfüggvény működése között?	Igen. A Szum munkalapfüggvény kihagyja az összeadásból a számként nem értelmezhető adatokat, míg az összeadás művelete az #ÉRTÉK hibaüzenetet adja ilyenkor.
4. Összeadható-e az IGAZ logikai konstans számokkal?	Igen. A táblázatkezelő az IGAZ értéket 1-nek, a HAMIS-t 0-nak tekinti.
5. Mi lesz az $=1<2<3$ képlet eredménye az Excel/Open Office elsőbbségi szabályai szerint?	Az Excelben HAMIS, a Calcban IGAZ. Az Excelben a balról jobbra elvégzett műveletsor előbb az $1<2=$ IGAZ értéket hozza létre, majd az IGAZ<3 eredménye HAMIS lesz. A Calc az $1<2$ eredményeként 1-et (IGAZ) ad és az $1<3$ is IGAZ lesz.
6. Mi a funkciója a BAL, a JOBB és a KÖZÉP függvényeknek?	Egy szöveg bal, jobb vagy középső részszovegét adják vissza adott hosszúságban: BAL(szöveg;részszoveg_hossz); JOBB(szöveg; részszoveg_hossz); KÖZÉP(szöveg;részszoveg_kezdete;részszoveg_hossza).
7. Milyen a művelettípusok elsőbbségi sorrendje?	Hivatkozási műveletek>Aritmetikai műveletek>Összefűzés>Hasonlító műveletek.
8. Milyen igazítási lehetőségeink vannak egy cellában?	Szövegigazítás vízszintesen, illetve függőlegesen, szöveghelyezés sortöréssel, lekicsinyítve, cellaegyesítéssel, a behúzás és az elforgatás megadása.

## Képletek másolása, automatikus kitöltés, irányított beillesztés

<b>Tanítási cél</b>	Táblakészítés. A táblázatkezelő másolási lehetőségeinek kihasználása a tábla létrehozásában. A konstansok, hivatkozások és képletek viselkedése a cellatartalom másolásakor. A hivatkozástípusok szerepe a táblatervezésben.
<b>Fontos fogalmak, definíciók</b>	Automatikus kitöltés, relatív, abszolút és vegyes hivatkozások.
<b>Tevékenységek</b>	Az irányított beillesztés. Táblatervezés. Az automatikus kitöltés és a képletek másolásának alkalmazása a táblakészítésben. Másolási lehetőségek. Sorok, oszlopok, cellák rögzítése másolásakor. Az irányított beillesztés változatainak kipróbálása.
<b>Megjegyzések</b>	Az automatikus kitöltés az újabb Excel változatokban szelektív lett, a művelet végén megjelenő listával pontosíthatjuk a másolást. A lista alapértelmezett értékei megfelelnek a korábbi verziók kitöltési lehetőségeivel.
<b>Házi feladat</b>	Munkafüzet 2., 3. és 4. feladata.

### Az ellenőrző kérdések és feladatok megoldásai

<i>1. Mit értünk azon, hogy a kerekítés abszolútértékes?</i>	A negatív számokat az abszolút értéküknek megfelelően kerekítjük, vagyis a -1,5-et a nála kisebb -2-re. Ez a kijelzésre és a KERÉKÍTÉS függvényre egyformán igaz.
<i>2. Hogyan folytatódik az autokitöltővel kihúzott 1,-1 számsorozat?</i>	1, -1, -3, -5, ...
<i>3. Hogyan tarthatjuk meg az automatikus kitöltéskor a cellák eredeti formátumát?</i>	Az intelligens címkét kibontva formátum nélküli kitöltést kérünk. További lehetőségek: Cellák másolása, kitöltés sorozattal, kitöltés csak formátummal.
<i>4. Mi lesz a C5 cellában lévő =D3*E2 képletből, ha az E7 cellába másoljuk?</i>	=F5*G4.
<i>5. Mi lesz a C5 cella =D\$3*\$E2 képletéből, ha az E7 cellába másoljuk?</i>	=F\$3*\$E4.
<i>6. Mikor kell abszolút módon hivatkozni egy cellára?</i>	Akkor, ha azt szeretnénk, hogy a cellára történő hivatkozások a képletek másolásakor változatlanok maradjanak.
<i>7. Hogyan változnak a relatív hivatkozások a képlet másolásakor?</i>	Legyen a forráscella címe Xn, a célcellaé Ym. Ekkor a relatív hivatkozás oszlopához (Y-X), sorához m-n adódik, ahol (Y-X) a cél és a forrásoszlop sorszámának különbsége..
<i>8. Hogyan változtathatjuk gyorsan a hivatkozás típusát?</i>	Az F4 billentyű ismételt leütésével.
<i>9. Soroljon fel néhány irányított beillesztési lehetőséget!</i>	Mindent, képletet, értéket, formátumot, értéket és számformátumot, szegélyt kivéve, művelettel, üresek átugrása, transzponálás.
<i>10. Hogyan lehet az autokitöltővel a beillesztést irányítani?</i>	A kitöltés végén, a megjelenő intelligens címkével.

## Képletek áthelyezése, munkalap-műveletek, nevek használata

<b>A lecke célja</b>	Táblakészítés. A táblázatkezelő cellamozgatási lehetőségeinek kihasználása a tábla létrehozásában. A konstansok, hivatkozások és képletek viselkedése a cellatartalom áthelyezésekor. Hivatkozás más munkalapok celláira és tartományaira. A másolás és az áthelyezés és a beszúrás műveletek működése munkalapokra. A munkafüzet szintű nevek használatának gyakorlása.
<b>Fontos fogalmak definíciók</b>	Cellamozgatás, másolt cellák beszúrása. Munkalapokra történő hivatkozások. Műveletek munkalapokkal.
<b>Tevékenységek</b>	Táblakészítés már létező táblázat átalakításával. Munkalapműveletek, lapok másolása és áthelyezése.
<b>Megjegyzések</b>	A háromdimenziós hivatkozásokat a tankönyv nem tárgyalja, de a munkafüzet 10. feladatában foglalkozik velük. Szorgalmi feladatként, esetleg konkrét alkalmazáshoz kapcsolva tárgyalhatók.
<b>Házi feladat</b>	A képletek áthelyezésére a munkafüzet 3., 4., a munkalapokkal végzendő műveletek gyakorlásához a 9. feladatot javasoljuk.

### Az ellenőrző kérdések és feladatok megoldásai

1. A Fékút vagy a Megállási idő táblákban alkalmazott módszer hibája lehet nagyobb a $t_r=1,5s$ , $v_0=130$ km/h-hoz tartozó cellában?	A Fékút táblában az adatokat minden cellában újraszámoltuk, ezért itt a hiba cellánként változó, függ a számábrázolási korlátok pontosságától. A Megállási idő táblában a kezdőcella hibája jelenik meg minden más helyen is, feltéve, hogy a differenciák pontos adatok.
2. Változhat-e az áthelyezett cellában a csak abszolút hivatkozást tartalmazó képlet?	Igen ha egyidejűleg azt a cellát is mozgatjuk, amelyre abszolút módon hivatkoztunk.
3. Hogyan ugorhatunk gyorsan a munkalap legutolsó sorára és oszlopára?	A legutolsó sorra: Ctrl+↓, a legutolsó oszlopra Ctrl+→ (esetleg ismételt) alkalmazásával.
4. Hogyan jelölhetjük ki egy munkalap összes celláját?	Az oszlop és sornevek által meghatározott szürke gombbal a munkalap bal felső sarkában, vagy az A1 cellából indulva a Ctrl+Shift+↓ és a Ctrl+Shift+→ ismételt alkalmazásával.
5. Hogyan különböztethetők meg a kijelölt munkalapok a nem kijelöltektől?	A kijelölt lapok háttérszíne fehér, a nem kijelölteké szürke. Vagy a kijelölt lapok neve aláhúzott, a többié színes háttérű.
6. Milyen műveletek végezhetők a munkalapokon?	Új lapokat szúrhatunk be, törölhetjük és átnevezhetjük, másolhatjuk és áthelyezhetjük őket.
7. Hogyan hivatkozhatunk egy másik munkalapon lévő cellára?	Vagy munkafüzet szintű névvel, vagy Munkalapnév!Cellahivatkozás módon.
8. Hogyan viselkednek a munkalapnevek a képletekben?	A munkalapnevek abszolút hivatkozásként viselkednek.
9. Mi lesz a Munka1 lap A1 cellájában lévő =A2 képletből, ha az A2 cellát áthelyezzük a Munka3 lap B2 cellájára? Milyen képlet lesz a Munka3!B2-ben?	=Munka1.A2

## Statisztikai függvények, táblaformázás, grafikonkészítés

<b>Tanítási cél</b>	Adott formátumú táblázatok készítése. Külső objektumok használata a táblaformázásban. Néhány gyakrabban használt statisztikai munkalapfüggvény alkalmazása.
<b>Fontos fogalmak, definíciók</b>	A grafikonkészítés elemeinek bemutatása A SZUM, MAX, MIN és ÁTLAG függvények. A grafikonkészítés alapfogalmai: grafikontípusok, adatsorok, kategória és értéktengely.
<b>Tevékenységek</b>	Konkrét feladat adott formátumú tábláinak elkészítése. WordArt embléma készítése és beszúrása. Statisztikai munkalapfüggvények használata. Grafikonkészítés varázslóval, ismerkedés a grafikonok elemeivel az elemleírások segítségével.
<b>Megjegyzések</b>	Mutassuk be a munkalapfüggvények és a táblázatkezelő műveleteinek eltérő működését. Próbáljuk ki Word dokumentumok és képek beszúrását is.
<b>Házi feladat</b>	A munkafüzet feladatainak megoldásához további összesítő függvényekre is szükség lesz. Ezeket a mellékelt súgó segítségével nem nehéz értelmezni és használni. Mivel a korábban nem használt munkalapfüggvények alkalmazása a tárgy érettségi követelményeihez tartozik, a munkafüzet 2., 3., 4. és 5. feladatai közül néhányat feltétlenül javasolunk önálló megoldásra.

### Az ellenőrző kérdések és feladatok megoldásai

1. Mit kéne írni az =D13/\$K13*100 képlet helyett, ha a cella formátuma Százalék lenne?	=D13/\$K13
2. Mit értünk a diagramvarázsló kategória fogalma alatt?	A megjelenítendő adatok mennyiségét az értéktengelyen, az adattípusokat a kategóriatengelyeken szemléltetjük. Ha a kategóriák is számok, mint pl. a matematikai függvényeknél, akkor a kategóriatengely a koordináta-rendszer x tengelyével lehet azonos.
3. Lehetnek-e szóközök az összesítő függvények argumentumában megadott tartományban?	Igen, ezeket az összesítő függvények figyelmen kívül hagyják (Nem nullának tekintik!).
4. Miért ajánlott a tábla formázását a megoldás végére hagyni?	A másolási és az áthelyezési műveletek könnyen elronthatják a cellaformátumokat. Ezen az intelligens címkék következetes használatával is segíthetünk.
5. Mi a funkciója a szűkítőgombnak?	A viszonylag nagyméretű párbeszédablakok helyett egy olyan kisméretű ablakba vihetjük be a hivatkozást, ami nem takarja el a beíráshoz esetleg szükséges cellákat.
6. Hogyan módosíthatók a grafikonok elemei?	Az egérkurzort a grafikon fölé mozgatva helyi menüt kérünk a módosítandó elemről. Ehhez az Eszközök menü Beállítások parancsának Diagram lapján a Nevek kijelzését be kell kapcsolni!



## Dátum- és időértékek, adatok másolása és áthelyezése

<b>Tanítási cél</b>	Összetettebb táblázatok készítése adott minta alapján. A dátum- és időadatok használatának gyakorlása.
<b>Fontos fogalmak, definíciók</b>	Dátum- és időfüggvények: ÉV, HÓNAP, NAP, ÓRA, PERC, MPERC, MA, MOST.
<b>Tevékenységek</b>	Az INT függvény. Dátum és időértékek, logikai konstansok bevitelének gyakorlása. A táblázatok fejlécének formázása. Tábla összeállítása segédtáblákból. Műveletek dátum- és időadatokkal.
<b>Megjegyzések</b>	A táblaszerkesztés a HA függvény használatával elegánsabban is elvégezhető lenne. A tankönyvben választott megoldást az indokolta, hogy a HA bevezetése jelentősen megnehezítené az amúgy sem könnyű táblaszerkesztési műveleteket. Ezért választottuk a csúcsidős hívássorok megjelölésére az IGAZ logikai konstans. A HA függvény tárgyalásakor a problémára röviden vissza lehet térni.
<b>Házi feladat</b>	A munkafüzet 1-4. feladatai közül ajánljuk a választást. Az 5. és 6. feladat függvényeit ritkán használjuk, de fontos fogalmakat – a szöveges, szám és dátum típusú adatok jellemzőit – segítenek tisztázni. A 8., 9. és 10. feladatokat kiegészítő anyagnak szántuk.

### Az ellenőrző kérdések és feladatok megoldásai

1. <i>Hogyan ábrázolja a táblázatkezelő a dátumot és az időt?</i>	Az1900.01.01.nap az 1 egész számnak felel meg, ez naponként 1-gyel növekszik. A legnagyobb kezelhető dátum: 9999.12.31. Az időértékeket nulla, vagy 1-nél kisebb tizedes törtek ábrázolják. A 0:00:00-nak a 0, a 0,99999 a 23:59:59-nek felel meg.
2. <i>Hogyan írhatjuk be a cellába a 2006.03.01. adatot úgy, hogy a záró pont ne törlődjön?</i>	A megfelelő alak kiválasztható a cellaformátumokból.
3. <i>Hogyan lehet eltüntetni a cellákból a képletet úgy, hogy annak csak az értéke maradjon meg?</i>	A cella tartalmát vágólapra másoljuk, majd önmagára irányított beillesztést kérünk és csak az értéket másoljuk vissza.
4. <i>Mit jelent a 37:50:55 időformátum?</i>	A csak időt, vagyis dátumértéket nem tartalmazó cellákban az óra értékét ekkor nem periodikusan számolja, hanem 23-nál nagyobb órát is megenged. Ezt később napokat is tartalmazó dátum-idő formára alakíthatjuk.
5. <i>Milyen dátum- és időfüggvényeket ismer?</i>	Év, Hónap, Nap, Óra, Perc, Mperc, Ma, Most, Dátum, Idő, Időérték, Dátumérték.
6. <i>Hogyan változnak a hivatkozásneveket tartalmazó képletek a cella másolásakor?</i>	A képletekben a munkafüzet szintű nevek abszolút hivatkozásként viselkednek.

## Logikai műveletek és függvények, táblák rendezése

<b>Tanítási cél</b>	Feltételek kezelése logikai függvények segítségével. Számadatok kerekítésének szükségessége és szabályai a táblázatkezelőben. Adatok rendezése táblázatokban. Speciális adatformátumok használata.
<b>Fontos fogalmak, definíciók</b>	Új munkalapfüggvények: NEM, ÉS, VAGY, HA, KERÉKÍTÉS. Az adatrendezés szabályai.
<b>Tevékenységek</b>	A korábban elkészített tábláink kiegészítése feltételektől függő adatokkal. A HA, a VAGY és az ÉS függvények gyakorlása. Számadatok kerekítésének alkalmazása a KERÉKÍTÉS függvénnyel. A táblázatkezelő rendezési szabályainak bemutatása. Rendezés az eszköztár ikonjainak segítségével.
<b>Megjegyzések</b>	Egyéni számformátumok alkalmazása a mértékegységek jelzésének érdekében. Az IGAZ és HAMIS függvények részletes tárgyalását nem éreztük fontosnak alkalmazásuk gyakorisága miatt. Fontosnak tartjuk hangsúlyozni, hogy a KERÉKÍTÉS függvény abszolút értékesen működik.
<b>Házi feladat</b>	A munkafüzet feladatai mellett választhatjuk korábban elkészített tábláink kiegészítését is feltételektől függő adatokkal.

### Az ellenőrző kérdések és feladatok megoldásai

1. <i>Hogyan adhatunk egyéni számformátumot a celláknak?</i>	A Cellaformázás menüparancs Egyéni formátumát kell választani és meg kell adni egy mintát.
2. <i>IGAZ értéket ad-e az <math>=A2&lt;B2</math>, ha az <math>A2&gt;B2</math> HAMIS volt?</i>	Nem feltétlenül. Ha $A2=B2$ igaz volt, akkor az $A2<B2$ is hamis lesz.
3. <i>Mikor kapunk HAMIS logikai értéket az És függvény értékeként?</i>	Ha az és függvénynek van legalább egy hamis argumentuma.
4. <i>Mi lesz az <math>=Nem(És(IGAZ;HAMIS))</math> eredménye?</i>	IGAZ
5. <i>Mi lesz az eredménye az <math>=Ha(HAMIS; "Nem igaz!")</math> képletnek?</i>	HAMIS
6. <i>Miért nevezzük abszolútértékesnek a Kerekítés/Round függvény eredményét?</i>	A függvény nem a matematikai szabályoknak megfelelően kerekíti a negatív számokat. Például a -0,5 argumentum esetén -1-et kapunk, nem pedig nullát.
7. <i>Mi lesz a Kerekítés/(172,15;-1) eredménye?</i>	17
8. <i>Milyen speciális hasonlítási szabályokat használ a táblázatkezelő a rendezéskor?</i>	A kis- és nagybetűket nem különbözteti meg, A magyar kettős betűket általában felismeri, az üres cellák az adatok végére kerülnek, az aposztrófot és a kötőjelet általában nem veszi figyelembe.
9. <i>Mi lesz az 1., 2., 3., 4., 5., 6., 7., 8., 9., 10. sorozatból, ha növekvően rendezzük.</i>	1., 10., 2., 3., 4., 5., 6., 7., 8., 9.
10. <i>A név szerinti rendezéskor a sorszám oszlop is átrendeződik az Osztálystatisztika táblánkban. Állítsuk vissza az eredeti rendezettséget!</i>	Ha az egy karakteres sorszámokat balról egy-egy nullával kiegészítjük, akkor a pontra végződő szöveges sorszámok esetén is jó sorrendet kapunk.

## Nyomatás, részösszegek, tömbfüggvények

<b>Tanítási cél</b>	A táblázatkezelő nyomtatási jellegzetességei, táblázatok nyomtatásának gyakorlása. Részösszegek képzése. Összesítés rendezett és rendezetlen táblázatokban Előkészületek hisztogramok készítéséhez.
<b>Fontos fogalmak, definíciók</b>	Nyomatási terület, ismétlődő címsorok. Nyomatási nézetek. A RÉSZÖSSZEG, a DARABTELI és SZUMHA függvények, kritériumok. A GYAKORISÁG függvény. Tömbfüggvények.
<b>Tevékenységek</b>	A tábla nyomtatás előtti rendezése. Munkalapok adatainak nyomtatása: a nyomtatási területek kijelölése, ismétlődő címsorok, nyomtatási nézetek, az adatterület illesztése a papírmérethez. Összesítési szintek készítése, megjelenítése és elrejtése a korábbiakban elkészített táblázatokon. A SZUMHA és a DARABTELI függvények alkalmazása. Tömbfüggvények megadási módja, a GYAKORISÁG függvény használata, a szükséges intervallumok megadása, az eredmény értékelése.
<b>Megjegyzések</b>	Mutassunk rá a matematikai függvény és a munkalapfüggvény fogalmának különbözőségére a GYAKORISÁG függvénnyel kapcsolatban. Hagyjunk elegendő időt az adattömb, a csoporttömb és az eredménytömb jellemzőinek, az intervallumok megadásának gyakorlásához.
<b>Házi feladat</b>	A munkafüzet 8. feladatát ajánljuk, ami előkészíti a következő óra anyagát, a grafikonkészítést.

### Az ellenőrző kérdések és feladatok megoldásai

1. <i>Hogyan állíthatjuk be az oldalhatárokat a nyomtatáshoz?</i>	Az Oldaltörés megtekintése paranccsal, ami a Nézet menüből vagy a Nyomatási kép menüparancsból érhető el.
2. <i>Hogyan lehet törölni a részösszeg sorokat?</i>	A Részösszegek párbeszédablak Az összes eltávolítása nyomógombjával.
3. <i>Keresse meg a Súgóban, milyen számok azonosítják az összegző függvényeket a Részösszeg függvényben!</i>	1 ÁTLAG 4 MAX 7 SZÓRÁS 10 VAR 2 DARAB 5 MIN 8 SZÓRÁSP 11 VARP 3 DARAB2 6 SZORZAT 9 SZUM
4. <i>Miért tartozik a Részösszeg argumentumában ugyanahhoz az összegző függvényhez kétféle szám?</i>	Az 1..11 helyett a 101..111 számokat kell használni ugyanazon összegző függvényekhez, ha a Formátum menü Sorok elrejtése parancsával láthatatlanná tett sorokat nem akarjuk figyelembe venni a műveletben.
5. <i>Mi a szerepe a szintvonalaknak és vezérlőiknek?</i>	A szintjelző vonalak a részösszegek csoportjait mutatják. A számozott vezérlők segítségével egy adott csoportszint alatti összes csoport elrejthető vagy megjeleníthető, a + a csoport elemeinek megjelenítését, a – az elemek elrejtését eredményezi.
6. <i>Hogyan kell megadni a kritérium függvényargumentumot?</i>	A kritérium tetszőleges kifejezés lehet (szövegkonstansokra használhatjuk a joker-karaktereket is), amely a kritérium tartomány megfelelő elemével együtt egy logikai értéket szolgáltat.
7. <i>Milyen munkalapfüggvényeket nevezünk tömbfüggvényeknek?</i>	A kiértékeléskor több értéket visszaadó függvényeket tömbfüggvényeknek nevezünk.
8. <i>Hogyan lehet felismerni a tömbfüggvényeket?</i>	A tömbfüggvényt minden eredménycellájában ugyanaz a képlet adja meg és {} zárójelek fogják közre.

## Grafikonkészítés, tesztdatok generálása

<b>Tanítási cél</b>	Jártasság szerzése a grafikonok készítésében. Tesztdatok előállítás.
<b>Fontos fogalmak, definíciók</b>	Néhány további, fontosabb statisztikai függvény megismerése. A grafikonok elemei: címek és feliratok, diagramterület, rajzterület, jelmagyarázat, kategóriatengely, értéktengely, a tengelyek beosztása, rács- és segédrács-vonalak.
<b>Tevékenységek</b>	A VÉL függvény. Hisztogramok készítése. Grafikonkészítés már létező grafikon átalakításával. A diagramok formázása. Egyenletes eloszlás előállítása adott intervallumon. Az átlag, a MEDIÁN és a MÓDUSZ kiszámítása.
<b>Megjegyzések</b>	A tankönyvben szereplők mellett természetesen további statisztikai munkalap-függvényeket is megemlíthetünk, a tanulók matematikai ismereteitől függően, például a SZÓRÁS-t. Az inverz eloszlásfüggvények nem tartoznak a szorosan vett tantervi követelmények közé. Használatuk a VÉL függvényre, illetve az egyenletes eloszlásra támaszkodva azonban olyan kevés matematikai előkészítést követel, hogy jobb képességű csoportokban tárgyalhatók lehetnek.
<b>Házi feladat</b>	A munkafüzet 1-5. feladatai a grafikonkészítéshez, a 6., 7., 9., 10. sorszámúak a statisztikai függvények alkalmazásához készültek. A 8. feladat néhány jól felhasználható mátrixfüggvény (INDEX, OFSZET, INDIREKT) működésének megértését célozza. Házi feladatnak az 1-5. sorszámúakból való választást ajánljuk.

### Az ellenőrző kérdések és feladatok megoldásai

<i>1. Mit jelentenek a grafikontengelyek „kis léptékei”?</i>	A tengelyeken két beosztást jelölhetünk, a fő és a kis léptéket. A kis léptékekhez nem tartozik felirat, lépésközük a Tengely formázása ablak Skála, jelölésmódjuk a Mintázat lapján állítható be.
<i>2. Hogyan állíthatjuk a térhatású grafikonok látószögét?</i>	A rajzterületről, a diagramterületről vagy a Falakról kért helyi menü segítségével. A menüből a Térhatás párbeszédablakot kell kérni. Ennek eszközeivel a látószög változtatható.
<i>3. Hogyan kérhetünk véletlen racionális számokat az <math>[a;b]</math>, <math>a,b \in Q</math> intervallumra,</i>	Az $a+(b-a)*\text{vél}()$ kifejezéssel.
<i>4. Hogyan kérhetünk véletlen egész számokat egy adott <math>[a;b]</math>, <math>a,b \in Z</math> intervallumra?</i>	Az $\text{int}(a+(b-a+1)*\text{vél}())$ kifejezéssel.
<i>5. Hogyan lehetne az Int helyett a Kerekítés függvénnyel adott <math>[a;b]</math> intervallumra véletlen egészeket generálni?</i>	Ha az intervallum mindkét végpontja $\geq 0$ , akkor a $\text{Kerek.le}(a+(b-a+1)*\text{vél}())$ , ha a végpontok $\leq 0$ -k, akkor a $\text{Kerek.le}(b-(b-a+1)*\text{vél}())$ , ha a végpontok különböző előjelűek, akkor az intervallumot 0-nál meg kell osztani.
<i>6. Hogyan lehet szabályozni a képletfrissítéseket?</i>	Eszközők $\rightarrow$ Beállítások $\rightarrow$ Számolás lapon kikapcsolhatjuk az automatikus frissítést. Ezt ezután az F9- vagy a Shift + F9-cel lehet kérni.
<i>7. Jellemezzük az egyenletes eloszlást!</i>	Egyenlő hosszú részintervallumokra várhatóan ugyanannyi adat esik.
<i>8. Mit csinálnak az Inverz.Eloszlás függvények?</i>	A pont után megadott Eloszlású adatokat lehet előállítani velük a $\text{vél}()$ véletlenszám-generátor értékeiből.

## Keresőfüggvények, az adatok szűrése

<b>Tanítási cél</b>	Osztálystatisztika készítése. Nagyon összetett feltételek kezelése, a HA függvény kiváltása. Keresés táblázatokban. Táblázatok adatainak szűrése.
<b>Fontos fogalmak, definíciók</b>	Az FKERES és VKERES függvények.
<b>Tevékenységek</b>	Autósűrő, irányított sűrő, sűrőtartomány. A keresőfüggvények alkalmazása osztálystatisztika készítéséhez. A grafikonkészítés gyakorlása.
<b>Megjegyzések</b>	Adattáblák szűrésének gyakorlása, a szűrt adatok mentése. A szűrt adatok leválasztása a táblázatról irányított ugrással történhet. Ezt a műveletet az önálló tanulói feladatmegoldás során nem könnyű megtalálni a menürendszerben, ezért feltétlenül javasoljuk az előzetes bemutatását. Az FKERES és a VKERES függvények mellett jól használható a HOL.VAN függvény is. Ennek tárgyalására a tankönyv keretei miatt nem térhettünk ki, de a munkafüzet 5. feladatában hivatkozunk rá. A függvény működésének megértéséhez a Sűgő ad segítséget.
<b>Házi feladat</b>	Az irányított szűrés feltételeinek megadását a sűrőtartománnyal adhatjuk meg, amellyel a később tárgyalandó adatbázisfüggvények kritériumainak megadását készíthetjük elő. Házi feladatként ezért a munkafüzet 9. és 10. feladatát javasoljuk.

### Az ellenőrző kérdések és feladatok megoldásai

<i>1. Milyen argumentumai vannak az Fkeres függvénynek?</i>	Fkeres(Keresett adat;Tartomány;Helyettesítési oszlop;Keresési mód) A függvény a Keresett adatot a Tartomány első oszlopában keresi és a Keresési módtól függően a Helyettesítési oszlop megfelelő sorában lévő cella értékét adja vissza. A Helyettesítési mód logikai érték lehet.						
<i>2. Miben különbözik a Vkeres az Fkeres-től?</i>	A Vkeres függvény a Tartomány első sorában keres és a Helyettesítési oszlop helyett adott sorból veszi a megfelelő értéket.						
<i>3. Az oszlop- és tortagrafikonoknak ugyanaz az adatsora. Készítse el az oszloppgrafikonból a tortát helyi menük segítségével!</i>	A Diagramterületről kért helyi menüből kiválasztjuk a Minták parancsot, majd a megjelenő párbeszédablakban az oszlop helyett a kördiagram-csoportot kérjük. A diagram kijelölése után a menüsor Diagram csoportját is használhatjuk.						
<i>4. Hogyan adhatjuk meg a sűrőtartományt?</i>	A szűrendő tábla oszlopnevei alatt megadjuk a szűrés feltételeit. Egy oszlopnév többször is szerepelhet a tartományban: <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="border-bottom: 1px solid black; padding: 0 10px;">Elért pontszám</td> <td style="border-bottom: 1px solid black; padding: 0 10px;">Elért pontszám</td> <td style="border-bottom: 1px solid black; padding: 0 10px;">Értékelés</td> </tr> <tr> <td style="text-align: center;">&gt;40</td> <td style="text-align: center;">&lt;55</td> <td style="text-align: center;">1</td> </tr> </table>	Elért pontszám	Elért pontszám	Értékelés	>40	<55	1
Elért pontszám	Elért pontszám	Értékelés					
>40	<55	1					
<i>5. Hogyan kell megadni a sűrőtartományban az És műveletet?</i>	Egy sorban kell megadni a feltételeket. Például a fenti tartományban az Elért pontszám>45 ÉS Elért pontszám<55 feltételt adtuk meg.						
<i>6. Keresse meg a Sűgőben, hogy hogyan lehet a sűrőtartományban képleteket használni?</i>	Az alábbi példában az 50 évesnél idősebb nőket szűrtük ki egy táblázatból. A képletet olyan oszlopban kell elhelyezni, amilyen oszlopnév a táblában nincsen (50-nél több), a sűrőképlet pedig a tábla első adatsorára vonatkozik (B2). <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="border-bottom: 1px solid black; padding: 0 10px;">Név</td> <td style="border-bottom: 1px solid black; padding: 0 10px;">50-nél több</td> <td style="border-bottom: 1px solid black; padding: 0 10px;">Férfi/Nő</td> </tr> <tr> <td colspan="2" style="text-align: center;">=ÉV(MA())-ÉV(B2)&gt;50</td> <td style="text-align: center;">N</td> </tr> </table> (Jegyezzük meg, hogy az Év(Ma())-Év(B2) nem korrekt kifejezés az életkor pontos meghatározására.)	Név	50-nél több	Férfi/Nő	=ÉV(MA())-ÉV(B2)>50		N
Név	50-nél több	Férfi/Nő					
=ÉV(MA())-ÉV(B2)>50		N					
<i>7. Hogyan lehet a helyben szűrés eredményét kimenteni?</i>	A Szerkesztés menü Ugrás→Irányított→Csak látható cellák parancsával vágólapra lehet menteni.						
<i>8. Hogyan állíthatjuk vissza a helyben szűrt táblákat?</i>	Az Adatok→Szűrő→Minden látszik parancssal.						

## Függvények ábrázolása

<b>Tanítási cél</b>	Matematikai függvények grafikonjainak készítése.
<b>Fontos fogalmak, definíciók</b>	Matematikai képletek szerkesztésének gyakorlása az Equation Editorra. A PontXY típusú diagram jellemzői és változatai. Az ABS, GYÖK, SIN, COS, RADIÁN és FOK és PI munkalapfüggvények. Polárkoordináta-rendszer, polárkoordináták.
<b>Tevékenységek</b>	Képletszerkesztés Equation Editorral, az adatsorok felvétele. Egyenes és görbe vonalakkól, vonaldarabokból és pontsorokból álló grafikonok készítése a PontXY típusú diagramokkal. A több vonaldarabból álló függvény-grafikonok, szakadási helyek kezelése. Polárkoordinátákkal adott görbék ábrázolása. Az ábrázolás lépésközének megválasztása.
<b>Megjegyzések</b>	Az ábrázolás lépésközének megválasztása középiskolás matematikai eszközökkel nem tárgyalható. A megoldás a digitalizálás mintavételi problémáira vezethető vissza, de a helyzetet nehezíti, hogy a táblázatkezelő tartópontokat összekötő görbéinek szerkesztési módszerét nem ismerjük.
<b>Házi feladat</b>	A munkafüzet 2. és 10. feladatai.

### Az ellenőrző kérdések és feladatok megoldásai

<i>1. Hogyan ábrázolhatunk több függvényt egyetlen diagramon?</i>	A forrásadatok ablakban annyi adatsort kell felvenni, amennyi függvényt ábrázolunk.
<i>2. Mit értünk a diagram zárolása alatt?</i>	Ha az Eszközök menü Védelem... ablakában a munkalapvédelmet bekapcsoljuk, a zárolt diagramokon nem végezhető módosítás.
<i>3. Hogyan becsülhetjük a torzításmentes ábrázolás lépésközét a görbe vonalú grafikonoknál?</i>	Ha a lépésközök finomítása (kisebbre vétele) már nem ad látható változást a görbén, a függvény alakját torzításmentesnek tekinthetjük. A görbe leggyorsabban változó helyének megkeresése mintavételi problémák megoldására vezethet, ezért középiskolában nem tudunk vele foglalkozni.
<i>4. Hogyan ábrázoljuk azokat a függvényeket, amelyek a vizsgált intervallumon nem mindenütt vannak értelmezve?</i>	Az összefüggő értelmezési tartományokon külön-külön adatsorokat veszünk fel. Az adatsorok ne tartalmazzanak olyan intervallumpontokat, ahol a függvény nincs értelmezve.
<i>5. A grafikon adatpontjai mellett látni szeretnénk a koordinátákat! Hogyan lehet ezt beállítani?</i>	Az Adatsorok formázása→Felíratok→x érték, y érték kapcsolók beállításával. Az adatsorok formázását helyi menüből is kérhetjük.
<i>6. Mit értünk egy síkbeli pont polárkoordinátái alatt?</i>	Az $(r, \varphi)$ számpárt, ahol az $r$ a pont origótól való távolsága, $\varphi$ pedig az $r$ pontot az origóval összekötő szakasz és a Descartes-féle koordináta-rendszer $x$ tengelye által bezárt szög. A két koordináta-pár között az $x=r \cdot \cos(\varphi)$ és az $y=r \cdot \sin(\varphi)$ összefüggések állnak fenn.
<i>7. Milyen mértékegységben kell megadni a szögfüggvények argumentumait?</i>	Radiánban (ívmértékben)
<i>8. Periodikusak-e a Spirál és a Négylevelű rózsza függvények?</i>	A Spirál nem, a Négylevelű igen.
<i>9. Mire kell ügyelni, ha egy függvény grafikonját több adatsorból rakjuk össze?</i>	Az egyes adatsorok utolsó eleme a csatlakozó adatsor első elemével egyezzen meg, különben a grafikon szakadást mutathat ott is, ahol a függvény folytonos.

## Adatbázisok, külső kapcsolatok

<b>Tanítási cél</b>	Az adatbázistáblák és adatbázis függvények használatának megismerése. Külső adatforrások átvétele és exportja. A táblázatkezelő és a web kapcsolata.
<b>Fontos fogalmak, definíciók</b>	Adatbázistábla, adatúrlap, adatbázis függvények. Mező, kritériumtartomány. Sztatikus és interaktív állományok. Közzététel a weben. Szűrés nélküli weblekérdezések.
<b>Tevékenységek</b>	Az adatbázistáblák feltöltése úrlapok segítségével. A kritériumtartományok megadása, szerkesztése. Az AB függvények működésének kipróbálása. Táblák közzététele a weben. Egyszerű internetes weblekérdezések.
<b>Megjegyzések</b>	A webes lekérdezések az SQL ismerete nélkül is gyakorolhatók. A konkrét feladatokban elegendő a Weblekérdezés beállításait ismerni.
<b>Házi feladat</b>	Gyakorlásként az adatbázistáblák úrlapokkal való feltöltését és az internetes lekérdezéseket ajánljuk.

### Az ellenőrző kérdések és feladatok megoldásai:

<i>1. Mit nevezünk az adatbázis tábla rekordjának, illetve mezőjének?</i>	A tábla egy-egy sorát rekordnak, oszlopait pedig mezőknek nevezzük.
<i>2. Milyen műveleteket végezhetünk az Adatúrlapokkal?</i>	Az Adatúrlapok Adatbázistáblák feltöltésére és karbantartására valók. A tábla számított cellái nem változtathatók meg, a szűrés csak az úrlapon megjelenő rekordokra értendő, nem pedig a tábla rekordjaira.
<i>3. Milyen az adatbázis függvények közös szintaxisa?</i>	AB.függvény(Adatbázistábla;Mező;Kritériumtartomány), ahol a Mező annak az oszlopnak a neve a táblában, amelyen a függvény „dolgozni fog”, a Kritériumtartomány pedig a szűrőtartományokhoz hasonlóan megadott feltételrendszer. A függvény a tábla csak azon sorait veszi figyelembe, amelyekre a kritériumtartomány IGAZ értéket ad.
<i>4. Milyen tulajdonságai vannak a sztatikus, illetve az interaktív weblapoknak?</i>	Az interaktív weblapok a böngészőkkel olvashatók és módosíthatók, de az Excelben közvetlenül nem nyithatók meg. A sztatikus lapok a böngészőkkel csak olvashatók, de a táblázatkezelőben közvetlenül is megnyithatók.
<i>5. Milyen feltételei vannak az Excel munkafüzet-elemek közzétételének?</i>	Az IExplorer legalább 4.01. verziója és az MSOffice Web Components program szükséges.
<i>6. Hogyan kérhetünk weblekérdezést?</i>	Adatok→Külső adatok importálása→Új weblekérdezés. Ezután kiválaszthatjuk a lekérdezendő objektumot,
<i>7. Hogyan állíthatjuk be a külső adattartományok jellemzőit?</i>	A weblekérdezés menüjéből a Beállítások párbeszédablak kapcsolóival megadhatjuk a külső tartományokból átveendő formátumokat, a dátumkezelést, stb.
<i>8. Milyen formátumok vesznek el a weblapkén való mentéskor?</i>	A cellákon belüli sortörés, a szöveg függőleges igazítása, az egyéni formátumok.

## Rendszerezés

<b>Tanítási cél</b>	A táblázatkezelés témakörének összefoglalása, az ismeretek rendszerezése, a témakört záró dolgozat elméleti előkészítése.
<b>Fontos fogalmak, definíciók</b>	Új fogalmakat erre a foglalkozásra nem terveztünk. A munkalapfüggvények magyar és angol elnevezéseit azonban ajánlott áttekinteni, ha ez a korábbiakban még nem szerepelt.
<b>Tevékenységek</b>	A rendszerezésben az egyes foglalkozások témaköreit foglaltuk össze néhány mondatban, kiemelve a legfontosabbakat. A foglalkozás lényegi részeit a kapcsolódó ellenőrző kérdések feldolgozása jelentheti.
<b>Házi feladat</b>	A kidolgozott feladatok áttekintése.

### Az ellenőrző kérdések és feladatok megoldásai

1. Sorolja fel az eszköztár speciális, csak a táblázatkezelőkre értelmezhető ikonjait.	AutoSzum, Rendezés-növekvő, Rendezés-csökkenő, Diagram varázsló, Cellaegyesítés, Pénznem, Százalék, Ezres csoport.
2. Mi a szerepe a szerkesztőlécnek?	Beviteli mezőként használható, másrészt mutatja a cella teljes tartalmát a kijelzéstől függően. Nem látjuk a cella tartalmát a szerkesztőlécen sem, ha a cella rejtett és a lapvédelem be van kapcsolva.
3. Milyen adattípusnak értelmezi a 3.2., illetve a 3. formátumú adatot az Excel bevitelkor?	A 3.2. dátum típusú lesz, a 3. szöveges.
4. A Formátum menü Cellák... ablakának Szám lapján milyen formátumokkal végezhetünk matematikai műveleteket?	A Szöveg kivételével bármelyikkel, kivéve még az Általános formátum szöveges adatait.
5. Milyen aritmetikai műveleti jeleket (operátorokat) használhatunk?	Előjelek, +, -, *, /, %, ^
6. A különböző típusú adatokat összehasonlítva melyek az IGAZ értékű relációk?	Igaz eredményt adnak a szám<szöveg, a szöveg<HAMIS és a HAMIS<IGAZ relációk, illetve az ezekből következő tranzitív műveletek.
7. Mi történik a képletek hivatkozásaival másolásakor?	A relatív hivatkozások sor- és oszlopindexei az elmozdulás sor- és oszlopszámának előjeles értékével módosulnak, az abszolút hivatkozások indexei nem változnak.
8. Hogyan viselkednek a munkafüzet szintű nevek a cellák másolásaakor, illetve áthelyezésekor?	Abszolút hivatkozásként viselkednek.
9. Mire lesz jó, ha a munkalapfüleknek különböző színeket adunk?	Könnyebben megkülönböztethető az aktív lap a többtől, áttekinthetőbb lesz a munkafüzet.
10. Mit értünk a diagram kategóriatengelyén?	A diagramok adatsorainak oszlopnevei, ha az adatok sorokban, vagy az oszlopok sorainak nevei, ha oszlopokban vannak.
11. Hogyan módosíthatjuk a diagramot annak létrehozása után?	Az Eszközök menü Beállítások... ablakának Diagram lapján a Nevek kijelzését kapcsoljuk be. Ekkor az egeret a diagram fölé mozgatva a diagram objektumairól elemleírásokat kapunk és helyi menüt kérve az adott elem módosítható. A diagram egy elemét kijelölve a menüsorban megjelenő Diagram segítségével is elvégezhetők a módosítások.
12. Milyen formátumot kell adni a cellának, hogy az a dátum és az időadatot is tartalmazza?	A formátumjelölőnek legyen dátum és időrésze is, vagy például a 37:30:55, vagy a 30:55,2 jelölőkkel.
13. Milyen dátum- és időfüggvényeket ismer?	Év, Hónap, Nap, Óra, Perc, Mperc, Ma, Most, Dátum, Idő, Dátumérték, Időérték, Nap360, Hét.Napja.



14. Hogyan kell megadni a Ha függvényt, hogyan működik?	Ha(Logikai-vizsgálat;Érték_ha_igaz;Érték_ha_hamis) A logikai vizsgálat kivételével az argumentumok hiányozhatnak. A függvény a logikai vizsgálat során a számértékeket is logikai konstansokká konvertálja. Ha a vizsgálat eredménye IGAZ, akkor az Érték_ha_igaz, ha HAMIS, akkor az Érték_ha_hamis lesz a visszaadott adat. Az argumentumok további Ha függvényeket tartalmazhatnak.
15. Hogyan rendezi a táblázatkezelő az adatokat?	A fontosabb szabályok: - Az ugyanazon kis- és nagybetűk között nem tesz különbséget. - A magyar kettős betűket általában felismeri, - Az üres cellák mindig a rendezett adatok végére kerülnek. - Az aposztrófot és a kötőjelet általában nem veszi figyelembe, csak akkor, ha a hasonlításban egyedül ezekben különböznek az adatok.
16. Hogyan kell megadni a Gyakoriság függvényt? Mi a funkciója?	Gyakoriság(Adattömb;Csoporttömb) A hisztogramokhoz készíthetjük el vele az Adattömb elemeinek a Csoporttömb intervallumainak megfelelő eloszlását. Tömbfüggvényként kell megadni (Ctrl+Shift+Enter).
17. Mit csinál a Darabtelési és a Szumma függvény?	Darabtelési: megszámolja azokat a nem üres cellákat egy tartományban, amelyek egy kritériummal adott feltételeknek eleget tesznek. Szumma: összegzi egy tartomány azon elemeit, amelyekre egy kritériumtartomány megfelelő elemei adott feltételnek megfelelnek.
18. Hogyan kell megadni a kritériumot a Darabtelési és a Szumma függvényekhez?	Darabtelési: tetszőleges relációs kifejezés jobb oldala az operátorjellel, aminek baloldali operandusaként egy adott cella értékét feltételezzük. Ha a relációs operátor az =, akkor ez elhagyható. Szöveg típusú adatra használhatjuk a * és a ? joker-karaktereket. Szumma: Ugyanaz igaz rá, mint a Darabtelési kritériumára.
19. Mit nevezünk hisztogramnak?	A hisztogram olyan diagram, amely egy adatsorozat értékeinek eloszlását ábrázolja adott, egymáshoz csatlakozó intervallumrendszeren.
20. Hogyan állíthatunk elő véletlen egészeket az [a;b] intervallumra?	Int(a+(b-a+1)*vél())
21. Mi a különbség az auto- és az irányított szűrő között?	Az irányított szűrővel összetettebb szűrőfeltételeket adhatunk meg, és a szűrt adatok elhelyezését is rugalmasabban adhatjuk meg.
22. Hogyan ábrázolhatunk egyenes szakaszokból álló függvénygörbét?	A PontXY grafikontípus egyenes szakaszos változatát használva elegendő a szakaszok végpontjaiban megadni az összetartozó értékeket.
23. Hogyan kell megadni a lépésközt görbék ábrázolásakor?	Ha a lépésköz csökkentésével már nem változik a görbe érzékelhetően, nem érdemes a tartópontokat sűríteni. A szükséges lépésköz a görbe különböző részein más-más méretű lehet.
24. Milyen az adatbázistábla függvények formája?	AB.függvény(Adatbázis_tábla;Mező;Kritériumtartomány)
25. Mit jelent a munkafüzet elemek sztatikus és az interaktív Webes közzététele?	Sztatikus: a táblázatkezelővel közvetlenül megnyithatók és módosíthatók. A böngészőkkel nem módosíthatók. Interaktív: a böngészőkkel megnyithatók és módosíthatók, a táblázatkezelővel közvetlenül nem.

# Adatbázis-kezelés

## Bevezetés

Az adatbázis-kezelés igen eltérő bonyolultsági szinten közelíthető része az informatikai ismereteknek. Sok szakértő véleménye az, hogy nem szabad tanítani sem általános- sem középiskolában, mert fogalmi rendszere magas absztrakciós szintet és élettapasztalatot igényel, és a 18 év alatti korosztály számára ez nem adatik meg. Ugyanakkor mások – a többség - véleménye szerint van olyan szelete e tárgykörnek, amely tanítható, sőt tanítandó is. Ennek eredménye az, hogy az adatbázisok témaköre megjelent a NAT és az érettségi követelményeiben is. Annak meghatározása azonban, hogy mely ismeretek, milyen fogalmak és összefüggések kapjanak szerepet és különböző hangsúlyokat a tananyagban, valószínűleg örök vita tárgya marad. Borítékolható, hogy a szoftverek fejlődése, a gazdaság által támasztott igények legalább olyan gyorsan változnak, mint a középiskolás korosztály hozzáállása.

A tanár szerepe tehát évről évre az, hogy ismételten átgondolja ennek a témakörnek az aktualitásait, jelentőségét az IT kultúrában, és ennek megfelelő tudatossággal tanítsa diákjait. Érdemes nem feledni, hogy az információrobbanás korában kulcsfontosságú az adatok feldolgozása, tárolása, erre a tevékenységre épül sok ember, vállalkozás, szervezet léte és működése.

A tankönyv ezen fejezetének ismeretfeldolgozási vezérlő elve a spirális tananyag-elrendezés. Képzelnünk el egy szabálytalan alapú, csúcán álló tölcserőt! Ez a tudástérnek az adatbázis-kezeléssel kapcsolatos ismereteit tartalmazza. Minél távolabb állunk benne a csúcstól, annál bonyolultabb, magasabb absztrakciós szintű tudáselemet találunk ott. A palást sem merev, inkább kívülről befelé fokozatos fogalmi átmenetek más ismeretekből származtatva. Vagy a nézőpontot megfordítva az adatbázis-kezelés fogalmaival leírható a valóság jelentős része, és ezáltal hatással van a világgal való kapcsolatunkra. Ha tölcserőt jól megválasztott távolságokra a csúcstól elmetsszük, akkor megkaphatjuk a NAT, az érettségi, az egyetemi vagy a professzionális követelményszinteket. Ez az ideális modell. A különböző követelmények „síkjai” ma még igencsak eltérő bonyolultságú ismeret-elemeket tartalmaznak.

Ez a modell nem engedi meg a szakszerűtlen megfogalmazásokat a kevésbé magas szinteken sem, hiszen a téves, félrevezető ismeretekre nem építhető ugyanannak a fogalomnak mélyebb elsajátítása, szélesebb összefüggéseinek feltárása. Ezért a középiskolában is precíz fogalmakat kell átadni, inkább kevesebbet, de érvényeset. Nehéz feladat, de az adatbázis-kezelésnek ki kell alakítani egy látszólag koherens, működő modelljét, amely a középiskolai korosztály számára megfelelő szinten érthető és használható.

A tankönyv adatbázis-kezelés fejezete az alapfogalmak tárgyalásán és két szoftvercsomag használatán keresztül kívánja elérni a közép- és emelt szintű érettségi követelményeiben leírt célokat.

Ne essünk abba a hibába, hogy a programokat tanítsuk meg a diákoknak, hiszen azok csak eszközök, átmenet az előző és a következő programverziók között. Készítsük fel tanulóinkat a folyton változó szoftver-környezetre, bátran váltsunk, ha megjelenik az újabb verzió. A programfejlesztők valószínűleg jobbat alkotnak a régebbinél, már ezért is érdemes áttérni. Persze annyi késlekedést azért hagyjunk, amennyi nekünk, tanároknak szükséges az újdonságok megismeréséhez. Az sem baj, ha a tankönyv képei nem pontosan az alkalmazott szoftver képét mutatják. Keressük inkább a megfelelő funkciókat, mint az adott szoftver menü-struktúrájában elfoglalt helyeit.

A fejezet első felében (1...8.lecke) egyetlen adattáblás „adatbázisok” szerepelnek. Itt azok a fogalmak és technikák kerülnek sorra, amelyek inkább elméleti jelentőségűek, vagy a szoftverek használatában újdonságok. A második rész zömmel visszatérés az első rész fogalmaira és technikáira, de több adattáblával, kibővítve az ismereteket a kapcsolatok kezelésével. Íme a spirális tananyag-elrendezés!

A tankönyv, a feladatlapok és a munkafüzet példáinak megoldása megtalálható a Tankönyvkiadó weboldalán mindkét programra.

És most lássuk, mire kell figyelniünk az egyes fejezetekhez kapcsolódó órákon!

## Az adatbázis-kezelés fogalmai, adatbázis modellek

<b>Tanítási cél</b>	Az adatbázisok kialakulásának és elterjedtségének bemutatása, alapfogalmak tanítása
<b>Fontos fogalmak, definíciók</b>	Adat, digitális és analóg adat, adatbázis, redundancia, adatbázis-kezelő szoftverek. Adatmodellek: hierarchikus, hálós, relációs, objektum-alapú; egyed, reláció, a reláció tulajdonságai
<b>Tevékenységek</b>	Beszélgetés, a diákok bevonása a példák keresésébe
<b>Megjegyzés</b>	Motivációs óra. Roppant fontos a további órákhoz való attitűd kialakításában.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. A tényleges adatok mellett tartalmazza az adatok szerkezetének leírását, az adatok közötti összefüggéseket, a megjelenésükhöz szükséges leíró definíciókat is.</li> <li>2. Hierarchikus modellel, hiszen az adatok (személyek) között amúgy is hierarchikus kapcsolat áll fenn.</li> <li>3. A – A, C – D, B – E</li> </ol>

## Programok, táblák, adattípusok, kulcsok

<b>Tanítási cél</b>	Bevezetés az adatbázis-kezelő programok használatába. A szoftver kiválasztásának szempontjai, környezetfüggőség.																																																						
<b>Fontos fogalmak, definíciók</b>	Adatbázis-kezelő programok, programtípusok. Varázsló és Tündér. Adatbázis, tábla, mező, adattípus. Kulcs, elsődleges kulcs. Az óra legfontosabb fogalma a kulcs!																																																						
<b>Tevékenységek</b>	Tanári irányítású munka a számítógépeknél. A demonstráció céljára alkalmazható például projektor.																																																						
<b>Megjegyzés</b>	Az adattípusok tárgyalásakor inkább szemléltető példákat kell alkalmazni, a könyvbeli táblázatokhoz referenciaként célszerű a későbbiekben visszatérni.																																																						
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. <math>0,95\text{MiB} = 996147.2</math> bájt, <math>973\text{KB} = 973000</math> bájt, tehát <math>973\text{KB} &lt; 0,95\text{MiB}</math>.</li> <li>2. A tábla mezői: <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="padding-right: 20px;">Azonosító</td> <td style="padding-right: 20px;">sorszám</td> <td>(kulcsmező)</td> </tr> <tr> <td>Név</td> <td>karakter</td> <td>40</td> </tr> <tr> <td>Születési_idő</td> <td>dátum</td> <td></td> </tr> <tr> <td>Születési_hely</td> <td>karakter</td> <td>20</td> </tr> <tr> <td>Állampolgárság</td> <td>karakter</td> <td>15</td> </tr> <tr> <td>Lakcím</td> <td>karakter</td> <td>40</td> </tr> <tr> <td>Ált_isk_név</td> <td>karakter</td> <td>50</td> </tr> <tr> <td>Ált_isk_cím</td> <td>karakter</td> <td>50</td> </tr> <tr> <td>Középisk_név</td> <td>karakter</td> <td>50</td> </tr> <tr> <td>Középisk_cím</td> <td>karakter</td> <td>50</td> </tr> </table> </li> <li>3. A tábla mezői: <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="padding-right: 20px;">Rendszám</td> <td style="padding-right: 20px;">karakter</td> <td>7</td> </tr> <tr> <td>Alvázsám</td> <td>karakter</td> <td>30</td> </tr> <tr> <td>Motorszám</td> <td>karakter</td> <td>30</td> </tr> <tr> <td>Gyártó</td> <td>karakter</td> <td>20</td> </tr> <tr> <td>Típus</td> <td>karakter</td> <td>20</td> </tr> <tr> <td>Évjárat</td> <td>numerikus</td> <td>4</td> </tr> <tr> <td>Szín</td> <td>karakter</td> <td>10</td> </tr> <tr> <td>Ár</td> <td>pénznem ... és további technikai adatok, kívánság szerint (üzemanyag, tulajdonosok száma, stb.)</td> <td></td> </tr> </table> </li> <li>4. A kottákat egyedileg azonosító adatok, például raktári szám, a (szerző, cím, kiadó, évszám) mezőcsoport.</li> </ol>	Azonosító	sorszám	(kulcsmező)	Név	karakter	40	Születési_idő	dátum		Születési_hely	karakter	20	Állampolgárság	karakter	15	Lakcím	karakter	40	Ált_isk_név	karakter	50	Ált_isk_cím	karakter	50	Középisk_név	karakter	50	Középisk_cím	karakter	50	Rendszám	karakter	7	Alvázsám	karakter	30	Motorszám	karakter	30	Gyártó	karakter	20	Típus	karakter	20	Évjárat	numerikus	4	Szín	karakter	10	Ár	pénznem ... és további technikai adatok, kívánság szerint (üzemanyag, tulajdonosok száma, stb.)	
Azonosító	sorszám	(kulcsmező)																																																					
Név	karakter	40																																																					
Születési_idő	dátum																																																						
Születési_hely	karakter	20																																																					
Állampolgárság	karakter	15																																																					
Lakcím	karakter	40																																																					
Ált_isk_név	karakter	50																																																					
Ált_isk_cím	karakter	50																																																					
Középisk_név	karakter	50																																																					
Középisk_cím	karakter	50																																																					
Rendszám	karakter	7																																																					
Alvázsám	karakter	30																																																					
Motorszám	karakter	30																																																					
Gyártó	karakter	20																																																					
Típus	karakter	20																																																					
Évjárat	numerikus	4																																																					
Szín	karakter	10																																																					
Ár	pénznem ... és további technikai adatok, kívánság szerint (üzemanyag, tulajdonosok száma, stb.)																																																						

## Rendezés, indexelés, az adatok érvényessége

<b>Tanítási cél</b>	Betekintés az adatbázisok belső szerkezetébe. Az adathelyesség szükségessége (annyira értékes egy adatbázis, amennyire jók az adatai).
<b>Fontos fogalmak, definíciók</b>	Logikai és fizikai sorrend, rekordmutató, index, indextábla, indexelés. Beviteli maszk, érvényességi szabály és szöveg.
<b>Tevékenységek</b>	Tanári közlés után irányított géphasználat a sorrendek és a beviteli ellenőrzés bemutatásakor, azután önálló próbálgatás a bevitel biztonságának ellenőrzésekor.
<b>Megjegyzés</b>	
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. A fizikai rendezés az eredeti táblának megfelelő új területet igényel, ezért a legalább három mezős, már néhány rekordot tartalmazó táblánál is több helyet foglal. Extrém esetek: egy-két igen kisméretű mezőből álló táblát nem érdemes indexelni csupán a helytakarékosság miatt.</li> <li>2. Előbb fizikailag, mert előfordulhat, hogy az eredeti sorrendből gyorsabban előáll a fizikai másolat, mint az indextábla léte miatt már más szempont szerinti sorrendben álló táblából. Egyébként a sorrend nem számít, ha mindkettőt az eredeti (rendezetlen) táblából származtatjuk.</li> <li>3. Az alkalmazható érvényesítési szabály például: &gt;-50 AND &lt;50</li> </ol>

## Űrlapok használata

<b>Tanítási cél</b>	Az adatbázis kezelésének egyszerűbbé és esztétikusabbá tétele. Kóstoló a szoftver-ergonómia alapjaiból.
<b>Fontos fogalmak, definíciók</b>	Űrlap, vezérlőelemek, azok objektum-jellegű tulajdonságlapjai.
<b>Tevékenységek</b>	Példa után önálló űrlapok készítése. Címkek, navigációs eszköztár. A kombinált lista alkalmazása
<b>Megjegyzés</b>	
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Logikus, az adatforráshoz hasonló formátumú, jól olvasható, stb.</li> <li>2. Az előző fejezet kérdései között megoldott feladathoz illeszkedő űrlapot a segédprogrammal egyszerűen el kell készíteni.</li> </ol>

## Interaktív adatkezelés

<b>Tanítási cél</b>	Az adatok módosítása, törlése, másolása, kivágása, beszúrása, szűrése kezelői beavatkozással. Az adatmódosítás felelősségének tisztázása (jogosultság!). A rekordok szűrése
<b>Fontos fogalmak, definíciók</b>	Logikai műveletek, szűrés úrlappal, kijelöléssel, kizárással. Szűrés.
<b>Tevékenységek</b>	Az adatok módosítása, törlése, másolása, kivágása, beszúrása, szűrése saját gépen, előre adott adatbázis táblájában.
<b>Megjegyzés</b>	
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. A veszélytelen műveletek: <ul style="list-style-type: none"> <li>– rendezés</li> <li>– rekord törlése</li> <li>– adatmódosítás</li> <li>– rekordok másolása</li> <li>– mező törlése a rekordszerkezetből</li> <li>– szűrés kijelöléssel</li> <li>– mező hozzáadása a rekordszerkezethez.</li> </ul> </li> <li>2. A VAGY műveletet, hiszen a pécsi vagy szegedi lakosok felelnek meg a feltételnek (egyszerre a kettőnek valószínűleg egyik sem)</li> <li>3. Az ÉS műveletet, hiszen egyszerre szőke és kékszemű lányokat keresünk.</li> </ol>

## Egyszerű lekérdezések

<b>Tanítási cél</b>	Az adatbázisok adatainak kívánság szerinti visszanyerése.
<b>Fontos fogalmak, definíciók</b>	Egyszerű (választó) lekérdezés.
<b>Tevékenységek</b>	Frontális bemutató után egyéni munkában sok lekérdezés készítése varázsló vagy tündér segítségével, gyakoroltatás.
<b>Megjegyzés</b>	Alapvetően munkáltató óra. Sok feladatot kell előkészíteni!
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Független a rekordok számától és a lekérdezés eredményétől.</li> <li>2. Nem kötelező, de az egyértelműség kedvéért ajánlott.</li> </ol>

## Lekérdezések szerkesztése

<b>Tanítási cél</b>	Lekérdezések készítése más eszközökkel (QBE, nézet).
<b>Fontos fogalmak, definíciók</b>	Minta alapú lekérdezés, QBE, irányított szűrés. Nézet.
<b>Tevékenységek</b>	Cél, hogy a lekérdezések készítése önállóan segítség nélkül sikerüljön.
<b>Megjegyzés</b>	A dátumok megadása szokatlan, programfüggő, külön figyelmet igényel!
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Azért, hogy a csak feltételben vagy sorrendképzésben szereplő mezők megjelenítését kikapcsolhassuk.</li> <li>2. A mezőnév, vagy a képletre utaló kifejezés.</li> </ol>

## Jelentés készítése, nyomtatása

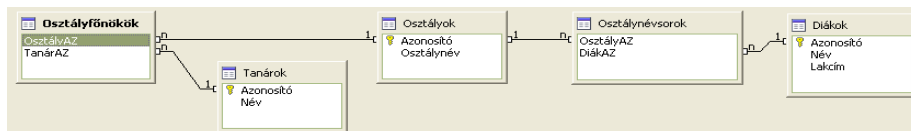
<b>Tanítási cél</b>	Jelentés készítése tetszőleges egytáblás adatbázisból.
<b>Fontos fogalmak, definíciók</b>	Jelentés, sablon, stílus, tájolás, jelentésfej, oldalfej, csoportfej, törzs, csoportláb, oldalláb, jelentésláb
<b>Tevékenységek</b>	Jelentések készítése varázsló vagy tündér segédprogrammal, sok változatban.
<b>Megjegyzés</b>	Abban az esetben, ha van elég idő, érdemes a kézi szerkesztést megmutatni, előbb-utóbb szükség is lesz rá.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Például úgy, hogy nem nyomtatóra, hanem állományba nyomtatunk, majd ezt küldjük el a hálózaton (célszerű tömöríteni!).</li> <li>2. Szerepel benne „-ként” raggal bővített, mezőnévre utaló határozószó.</li> </ol>

## Összetett adatbázis tervezése

<b>Tanítási cél</b>	Többláblás adatbázis tervezése, a kapcsolódó új fogalmak megértése. A tervezés fontosságának megértése.
<b>Fontos fogalmak, definíciók</b>	Normalizálás, első, második és harmadik normálforma, redundancia, teljes függőség, tranzitív függőség.
<b>Tevékenységek</b>	Tisztán elméleti óra. Aktív figyelem, közös munka szükséges.
<b>Megjegyzés</b>	Azt az adatmodellt, amelyről nem tudjuk, hogy teljesíti-e valamelyik normálforma feltételeit, nulladik normálformájának nevezzük. Ez az állapot a normalizálás kiinduló helyzete. Szám szerint öt normálformát ismerünk, illetve a szakirodalom ismer még egy hatodikat is, amelyet megalkotóiról Boyce-Codd normálformának neveznek. A normalizálás során az egymás után következő normálformák feltételeit ellenőrizzük, és ha szükséges, az adatmodellt módosítjuk. Ha az adatmodell az ötödik normálformát is teljesíti, akkor az adatmodellt, illetve az adatmodell alapján felépített adatbázist teljesen normalizáltnak nevezzük. A normalizált tervezés támogatja az adatintegritást a redundancia és az inkonzisztencia minimalizálásával, de lassíthatja az adatok lekérdezését. Ha a hangsúly az adatok gyors visszakeresésén van, előfordulhat, hogy egy alacsonyabb normálformával kell megelégednünk.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Csupán az a feltétel, hogy az adatok táblázatos formában álljanak és az oszlopok első sorában álljon az oszlop neve mint mezőnév.</li> <li>2. A normalizálás célja az adatintegritás (az adatok sértetlen egysége) a redundancia csökkentésével és az összefüggések feltárásával.</li> <li>3. Mivel redundancia nincs, csak akkor normalizáljunk, ha adatok közötti összefüggésekért azt tenni érdemes. Valószínűleg azonban nem lesz rá szükség.</li> </ol>

## Kapcsolatok, hivatkozások

<b>Tanítási cél</b>	A táblák közötti kapcsolatok rögzítése, a hivatkozási integritás védelmének megértése
<b>Fontos fogalmak, definíciók</b>	Kapcsolatok (relációk): 1:1, 1:N, N:M, hivatkozási integritás, kaszkádolt frissítés, kaszkádolt törlés. Törlési anomália, adatfelvételi anomália, adatmódosítási anomália.
<b>Tevékenységek</b>	Részben elméleti, részben gépen végzett munka.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1.–3. Hozzuk létre az adatbázist a tankönyv útmutatása alapján.</li> <li>4. Elvileg lehet, de ez az adatbázis tervezési hibájára utal.</li> <li>5. Mivel mindegyik egyedi érték, csak 1:1 kapcsolat lehetséges.</li> <li>6. A táblák közötti felállított adatkapcsolatok érvényben maradását minden törléskor, módosításkor, új rekord bevitelkor ellenőrzi az adatbázis-kezelő rendszer.</li> </ol>



## Többléptáblás lekérdezések

<b>Tanítási cél</b>	Lekérdezés létrehozása több táblás adatbázisban.
<b>Fontos fogalmak, definíciók</b>	Táblanévvel minősített mezőnevek és kapcsolatok
<b>Tevékenységek</b>	Ismétlő jellegű óra, önálló, gépen végzett munka.
<b>Megjegyzés</b>	A spirális tananyag-elrendezés újabb példája, szinte semmi újdonság, csak összetettebb környezetben néhány ismert tevékenység végzése.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Válassz egyet az egyes és kettes borítékokból! (Persze itt a VAGY értelemben)</li> <li>2. Lehet, mert a lekérdezés megfogalmazásakor is összekapcsolhatjuk a táblákat, feltételek beállításával.</li> </ol>

## Összetett adatbázisok űrlapjai

<b>Tanítási cél</b>	Komplex űrlapok szerkesztése többléptáblás adatbázisban.
<b>Fontos fogalmak, definíciók</b>	Segédűrlap, kulcs - idegen kulcs kapcsolat.
<b>Tevékenységek</b>	Gépen végzendő, először irányított, majd önálló órai munka.
<b>Megjegyzés</b>	Nagyon fontos a táblák közötti kapcsolat lényegének ismerete, átlátása.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Akkor célszerű a segédűrlap, ha olyan tábla eléréséhez készítünk űrlapot, amelyik egy másikkal áll kapcsolatban, és látni szeretnénk a másik tábla hivatkozott rekordjait is a szám formátumú kulcs-érték helyett.</li> <li>2. Igen, mert csak az adatok közötti kapcsolatra van szükség, nem a kapcsolat definiáltságára.</li> </ol>



## Az adatbázis-kezelés közös nyelve, az SQL

<b>Tanítási cél</b>	Az SQL nyelv elemeinek bemutatása
<b>Fontos fogalmak, definíciók</b>	Lekérdező nyelv, az SQL alapszavai
<b>Tevékenységek</b>	Gép mellett végzett, programozáshoz hasonló munka. Az eddig elkészült lekérdezések vizsgálata SQL nézetben, új lekérdezések készítése közvetlenül SQL-ben.
<b>Megjegyzés</b>	Az SQL jó összefoglalása az eddigieknek, hiszen ugyanazokról a fogalmakról van szó, de más eszköz segítségével. Különösen fontos az emelt szintű érettségit tenni szándékozóknek.
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<ol style="list-style-type: none"> <li>1. Előnye az univerzalitása, hiszen az adatbázis-kezelés közös nyelvről van szó. Mindazon adatkezelő műveletek, amelyeket az adatbázisokon el kell végeznünk, SQL parancsokkal is megfogalmazhatóak.</li> <li>2. A nyelv feladatait és így utasításait három csoportba foglalhatjuk: DDL = adatstruktúrát definiáló utasítások, DML = adatokon műveletet végző utasítások, DCL = Kommunikációs, adatbiztonsági és egyéb utasítások.</li> <li>3. A SELECT utasításhoz kapcsolódó kötelező kulcsszó: FROM opcionális kulcsszavak: DISTINCTROW, WHERE, JOIN, HAVING, GROUP BY, ORDER BY, ASC, DESC (SQL implementációtól függően a lista bővíthet)</li> </ol>

### Különleges lekérdezések

<b>Tanítási cél</b>	További lehetőségek használata az adatbázisok kezelésében.														
<b>Fontos fogalmak, definíciók</b>	Adatfrissítő, módosító, táblakészítő, törlő, azonosakat és nem azonosakat kereső, keresztáblás lekérdezések														
<b>Tevékenységek</b>	Gyakorlati munka, amelyet a tanár elméleti magyarázata és demonstrációja egészít ki.														
<b>Megjegyzés</b>	Az adatokon végzett visszavonhatatlan manipulációk miatt érdemes erre a célra külön példa-adatbázissal készülni.														
<b>Válaszok a tankönyv ellenőrző kérdéseire</b>	<p>1. feladat:</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;">Frissítő</td> <td>SELECT, <u>UPDATE</u>, INSERT, DELETE, TRANSFORM</td> </tr> <tr> <td>hozzáfűző</td> <td>SELECT, UPDATE, <u>INSERT</u>, DELETE, TRANSFORM</td> </tr> <tr> <td>táblakészítő</td> <td><u>SELECT</u>, UPDATE, INSERT, DELETE, TRANSFORM</td> </tr> <tr> <td>törlő</td> <td>SELECT, UPDATE, INSERT, <u>DELETE</u>, TRANSFORM</td> </tr> <tr> <td>azonosakat kereső</td> <td><u>SELECT</u>, UPDATE, INSERT, DELETE, TRANSFORM</td> </tr> <tr> <td>nem egyezőket kereső</td> <td><u>SELECT</u>, UPDATE, INSERT, DELETE, TRANSFORM</td> </tr> <tr> <td>keresztáblás</td> <td>SELECT, UPDATE, INSERT, DELETE, <u>TRANSFORM</u></td> </tr> </table> <p>2. feladat: Jogosultsági, integritás-védelmi, zárolási (többfelhasználós környezetben) okból.</p> <p>3. feladat: A SELECT INTO paraméterében hálózati hivatkozással másik gépre másoljuk az adattáblákat.</p>	Frissítő	SELECT, <u>UPDATE</u> , INSERT, DELETE, TRANSFORM	hozzáfűző	SELECT, UPDATE, <u>INSERT</u> , DELETE, TRANSFORM	táblakészítő	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM	törlő	SELECT, UPDATE, INSERT, <u>DELETE</u> , TRANSFORM	azonosakat kereső	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM	nem egyezőket kereső	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM	keresztáblás	SELECT, UPDATE, INSERT, DELETE, <u>TRANSFORM</u>
Frissítő	SELECT, <u>UPDATE</u> , INSERT, DELETE, TRANSFORM														
hozzáfűző	SELECT, UPDATE, <u>INSERT</u> , DELETE, TRANSFORM														
táblakészítő	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM														
törlő	SELECT, UPDATE, INSERT, <u>DELETE</u> , TRANSFORM														
azonosakat kereső	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM														
nem egyezőket kereső	<u>SELECT</u> , UPDATE, INSERT, DELETE, TRANSFORM														
keresztáblás	SELECT, UPDATE, INSERT, DELETE, <u>TRANSFORM</u>														

# Algoritmusok és adatok

## Bevezetés

### Az Algoritmusok fejezet a tantervben

A **Kerettanterv** az általános iskolai informatikaoktatás céljával tűzi ki:

- algoritmusok készítését, leírását valamilyen algoritmus-leíró nyelven;
- algoritmusok kódolását egyszerű programozási nyelven;
- a szekvenciális és feltételes vezérlést, ciklust tartalmazó programok értelmezését, kódolását, kipróbálását;
- grafikával, szöveggel kapcsolatos programok készítését;
- különböző számtípusú adatok használatát;
- elemi és összetett adatok ismeretét, adatok sorozatának alkalmazását.

A középiskolai kerettanterv a következő feladatokat írja elő a fejezet számára:

- algoritmusok megfogalmazása, tervezése és készítése;
- algoritmusok leírása algoritmus-leíró nyelven;
- az algoritmikus gondolkodásmód fejlesztése;
- elemi és összetett adatok kezelése (karakterek, egész és valós számok, logikai értékek, tömbök);
- a feladatmegoldáshoz szükséges adattípus kiválasztása;
- a programkészítés lépéseinek ismerete és végrehajtása (feladat-meghatározás, tervezés, kódolás, kipróbálás);
- algoritmusok kódolása egyszerű programozási nyelven;
- adatgyűjtési és feldolgozási algoritmusok értelmezése és alkalmazása.

A belépő tevékenységformák között az alapvető adatgyűjtési–feldolgozási algoritmusok közé sorolja<sup>1</sup>

- az összegzést,
- megszámlálást,
- keresést,
- és a rendezést.

A **Nemzeti alaptanterv** a fejlesztési feladatok között általánosságban is megfogalmazza az algoritmizálás, adatmodellezés szerepét. Elvárja a tanulóktól, hogy ismerjék fel a hétköznapi életben és az iskolában előforduló tevékenységek algoritmizálható részleteit.

Az alaptanterv jelenlegi formájában csak általánosan és röviden tárgyalja az egyes fejlesztési feladatokat. Lényegében a 7.-8. évfolyamon írja elő a (nem „egyszerű”) fejlesztő rendszerrel történő feladatmegoldást, a megoldáshoz szükséges algoritmuselemek ismeretét, az algoritmusok tervezését, megvalósítását és végrehajtását. Ugyancsak ennél a korosztálynál említi a lépésenkénti finomítás elvét, továbbá az elemi és összetett adatok kezelését. A középiskolában ehhez látszólag nem sokat fűz hozzá, a típusalgoritmusokkal bővíti a fejlesztési feladatokat.

Az algoritmikus gondolkodásmód elemeit természetesen a többi fejezetben (és nem csak informatikaórán) is felhasználjuk. Kialakítása sem kötődik csak ehhez a fejezethez és csak az informatikaórákhoz.

Az Algoritmusok és adatok fejezet anyaga csupán az emelt szintű érettségi követelményei között szerepel, a középszintű érettségien közvetlenül nem kéri számon.

Bár a tantervek közvetlenül nem említik a programozás tanítását, az algoritmusok programozási nyelven történő kódolása természetesen feltételezi egy programozási nyelv elemeinek és alkalmazásának ismeretét.

---

<sup>1</sup> A Kerettanterv 2000-es változatában szereplő eldöntést és kiválasztást törölték a módosított változatból.

## Az Algoritmusok és adatok fejezet tanítása

A Kerettanterv az iskola feladataként szabja meg, hogy készítse fel a tanulókat a megfelelő információ-szerzési, feldolgozási, tárolási és átadási technikákra, valamint ismertesse meg velük az információkezelés jogi és etikai szabályait. „*Ennek leghatékonyabb módját a több évig tanult informatika tantárgy, és az iskolai élet egészét átható informatikai nevelés biztosíthatja.*” – olvashatjuk benne.

A szűkre szabott órakeret miatt még a tanterv jelentősen csökkentett, módosított változata is túlzó célokat és feladatokat tűz ki. Heti 2 órával számolva a 9. évfolyamra előírt 8 témakör mindegyikére átlagosan 9 óra jutna, az összefoglalást és számonkéréstől eltekintve mindössze 7 óra. Ez még akkor is kevés, ha nem egyformán osztjuk el az óraszámokat a fejezetek között.

A szűkös óraszámnak főleg az Algoritmusok és adatok fejezet látja kárát. Mivel közvetlenül nem szerepel a középszintű érettségiben, a legtöbb esetben teljesen kimarad az informatikaórán tárgyalt anyagból.

A körülményekhez igazodva a fejezet tanítását három szinten tartjuk lehetségesnek, nagyjából a rendelkezésre álló óraszám függvényében.

1. Külön fejezetként egyáltalán nem tárgyaljuk az algoritmusokat. Az algoritmikus gondolkodásmód kialakítására nagy súlyt fektetünk a tanterv többi fejezeténél. Az adattípusok ismertetésére főleg a táblázatkezelés és adatbázis-kezelés során nyílik mód.
2. Általános szinten tárgyaljuk az algoritmusokat, konkrét programozási nyelv nélkül. Ekkor természetesen nem nyílik mód az algoritmusok kódolására valamilyen programozási nyelven. Egy programozás nyelv használata további időt igényel a fejlesztőrendszer megismertetéséhez is. (Ezt az időigényt nagymértékben csökkenthetjük a Visual Basic Script nyelv alkalmazásával, lásd később.)
3. Megfelelő óraszám esetén természetesen van módunk a tantervi előírások maradéktalan teljesítésére, programok tervezésére, kódolására, kipróbálására, egy programozási nyelv alapelemeinek elsajátítására, az előírt algoritmusok megismertetésére.

## A tankönyv Algoritmusok és adatok fejezete

A tankönyv Algoritmusok és adatok fejezete úgy épül fel, hogy elősegítse a 2. és 3. szintnek megfelelő óratervezést. A konkrét programozási nyelvekkel kapcsolatos ismeretek jól elkülönülnek a szövegben, a teljes fejezet tárgyalható a programozási nyelvekre vonatkozó részek nélkül is! Néhány kivételtől eltekintve csak mondatszerű leírással szerepelnek az algoritmusok, a forráskódokat a tankönyv webes kiegészítésében találjuk ([www.ntk.hu](http://www.ntk.hu)).

Az NTK Digitális Kultúra sorozatának tankönyvei (9. és 10. osztály) két évfolyamra osztják el a középiskolai informatika anyagát. A tankönyvek tervezésénél heti 2 órával számoltunk, ehhez illesztettük az egyes fejezetek terjedelmét. Az algoritmusok és adatok fejezet 15 órát kapott, amelyből le kell vonni az összefoglalásra és számonkérésre vonatkozó 2 órát (lásd a javasolt tanmenetet: [www.ntk.hu](http://www.ntk.hu)). A tantervi követelmények elsajátítására tehát 13 óra marad. Ennyi idő alatt kell eljutni az alapoktól az előírt algoritmusok (köztük egy rendezési algoritmus) elkészítéséig és alkalmazásáig, beleértve a számítógépes megvalósítást.

Ez az óraszám nem elegendő az algoritmusok ÉS egy programozási nyelv alapjainak az elsajátításához, egy programozási nyelv fejlesztői környezetének megismeréséhez. Maga a **tanterv sem említi szó szerint a programozási nyelvek tanítását**. Hangsúlyozzuk tehát, hogy a körülmények kényszere miatt **a fejezet nem programozást tanít**, csupán megpróbálja bemutatni az előírt algoritmusok kódolását egy programozási nyelven. Kifejezetten programozást valószínűleg más felépítésben, más módszerrel (és több órában!) tanítanánk.

A fejezet leckéi olyan sorrendben és olyan tartalommal követik egymást, hogy a rendelkezésre álló 13 óra alatt el lehessen jutni az előírt algoritmusok megismertetéséig. Ez a kényszer természetesen végigkíséri az egész fejezetet. Az algoritmusok definícióját, a feltételes elágazások, ciklusok fogalmát ismétlésnek tekintjük, mert az általános iskolai tanterv előírja az ismeretüket. A programozási nyelvek fejlesztői környezetéből csupán annyit tárgyalunk, amennyi a forráskód begépeléséhez, illetve a program futtatásához szükséges. Nem tudunk kitérni a hibakereső, hibajavító eszközökre. Az egyes nyelvek elemi adattípusait csupán felsoroljuk, mivel a Pascalban a deklaráció, Visual Basic Scriptben pedig a numerikus értékek beolvasása igényli az ismeretüket. Nem tudunk elmerülni a numerikus kifejezések és a kifejezésekbe beírható függvények részleteiben sem.

A tárgyalásra kerülő algoritmusok mindegyike igényelné a tömbök alkalmazását. Az elemi adattípusok és értékadó utasítások viszonylag száraz ismerethalmaza után azonban beiktattuk – egyelőre tömbök nélkül – az összegezési algoritmus tárgyalását. Ez már módot ad érdekesebb, valóban számítógép alkalmazását igénylő

algoritmusok kódolására. A tömbök tárgyalása után következik a többi előírt algoritmus bemutatása, lecként egy-egy algoritmussal és egy-egy példával.

Még egyszer hangsúlyozzuk, hogy a rendelkezésre álló óraszám és a tanterv követelményeinek ellentmondása meglehetősen szűkre szabta a didaktikai–módszertani eszközök körét, nagyon feszes ritmust diktál, tömör, célratoró megfogalmazást igényel. Csak egyetlen esetben bővítettük a kötelező tananyagot. Feltétlenül szükségesnek tartottuk a Visual Basic Script grafikus felhasználói felület-kezelésnek és eseménykezelésének az ismertetését. Többek között ezek az eszközei teszik különösen alkalmassá a Visual Basic Scriptet az algoritmusok számítógépes megvalósításának a bemutatására.

## A tankönyvi fejezet és az emelt szintű érettségi

Az Algoritmusok és adatok fejezet speciális helyzetet foglal el a többi fejezethez viszonyítva. Közvetlenül ugyanis nem szerepel a középszintű érettségi követelményei között.

A fejezet összeállításánál a minden diákra érvényes tantervi követelményeket tartottuk szem előtt. Anyaga tehát csak részben fedi le az emelt szintű érettségi követelményeit. A tankönyvi fejezet nem tárgyalja:

- az eldöntés, kiválasztás, maximumkeresés, kiválogatás algoritmusát;
- a beillesztéses rendezés kivételével a rendezési algoritmusokat;
- a rekurziót;
- függvények és eljárások definiálásának módját, az eljárások paramétereinek típusait;
- a halmaz és rekord adatszerkezetet (csak utal rájuk);
- a fájlkezelést;
- a tesztelés és hibakeresés módszereit, eszközeit;
- a hatékonyság növelésének módszereit.

## A kiválasztott programozási nyelvek

Az algoritmusok számítógépes megvalósításához egy programozási nyelv ismeretére van szükség. Az általános iskolában a diákok általában megismerkednek a Logo alapelemeivel, ez a programozási nyelv azonban nem alkalmas a középiskolai tanterv követelményeinek teljesítéséhez (különböző adatszerkezetek bemutatása, az előírt algoritmusok egyszerű megvalósítása stb.).

Az algoritmusok kódolásához a tankönyvben a Pascal és a Visual Basic Script nyelvet választottuk. A Pascal valószínűleg a legelterjedtebb programozási nyelv a középiskolákban, a Visual Basic Script pedig véleményünk szerint kiemelkedően alkalmas a programozás alapelemeinek a tanítására. A két nyelv párhuzamos tárgyalása megfelel a többi gyakorlati fejezet felépítésének (kétféle alkalmazás bemutatása), lehetővé teszi a tanárok számára a választást, ideális körülmények között pedig módot ad a két programozási nyelv összehasonlítására, a jellemzők általánosítására is.

## A Free Pascal programozási nyelv

A középiskolai oktatásban általában a Turbo Pascal fejlesztői környezetét használják a programozás tanítására. A Pascal választása esetén azonban javasoljuk az áttérést az ingyenes **Free Pascal**-ra, melynek integrált fejlesztői környezete nagymértékben hasonlít a Turbo Pascaléra. A Free Pascal szintaxisa kompatibilis a Turbo Pascallal. A Free Pascal letölthető a [www.freepascal.org](http://www.freepascal.org) webhelyről.

A tankönyvi fejezet szempontjából a Free Pascal a következő előnyökkel rendelkezik a Turbo Pascalhoz képest:

- Kezeli a virtuális memóriát, így még nagy tömböknél sem jön létre túlcsordulás.
- 32 bites fordítóprogramja a Windows alatt futó programokat generál. A programok karakteres felületű ablakokban futnak, a Turbo Pascalban előforduló problémák nélkül. Megjegyezzük, hogy a Free Pascal grafikus üzemmódja külön ablakot nyit meg, amely a szokásos módon a vágólapra menthető (*Alt+PrintScreen*).
- Nincs szükség a Windows alatt a *newdelay* unit beillesztésére.
- *Strutils* unitja tartalmazza az alapvető sztringfüggvényeket (*leftstr*, *rightstr*, *midstr* stb.).
- Az *Edit* menüből elérhető a Windows vágólapja (másolás és beillesztés is).
- Automatikus kódkiegészítő (*Code Completion*) és kódsablon (*Code Template*) funkciója nagymértékben megkönnyíti a forráskód begépelését.

- Windows és Linux disztribúciókkal is rendelkezik (továbbá még néhány operációs rendszeren használható).
- Jelenleg is folyik a fejlesztése, támogatása.

A tankönyv kéziratának elkészítésekor (2006 elején) a 2.0.2 változat volt elérhető a Free Pascal webhelyén. A tankönyvben ennek logóját helyzetük el a vonatkozó megjegyzések mellé. Az újabb, 2.0.4-es változatban megváltoztatták a logót.



A Free Pascal telepítése nem igényel különösebb beavatkozást.

Megjegyezzük, hogy a Free Pascal teljes dokumentációja (felhasználói kézikönyv, programozói kézikönyv, referencia stb.) elérhető a [www.freepascal.org](http://www.freepascal.org) webhelyén. 2006-ban megjelent az első magyar nyelvű könyv is a Free Pascalról (lásd: [Free Pascal](#)).

## A Visual Basic Script programozási nyelv

A tanterv által előírt követelmények teljesítése (az algoritmusok kódolása valamilyen programozási nyelven) és a rendelkezésre álló szűkös órakeret ellentmondása olyan programozási nyelv használatát várja el, melynek segítségével a lehető legegyszerűbb és leggyorsabb módon mutathatjuk be az algoritmusok számítógépes megvalósítását. Erre a célra a Visual Basic Scriptet tartjuk a legalkalmasabbnak. Hangsúlyozzuk, hogy **a Visual Basic Script programok futtatásához nincs szükség az Internet Explorerre (vagy más böngészőre), és nincs szükség a HTML-kód ismeretére sem!** Windows operációs rendszer használata esetén minden további előkészítés, telepítés nélkül tudunk írni és futtatni<sup>2</sup> Visual Basic Script programokat. **A forráskód egy egyszerű szövegszerkesztővel, például a Jegyzetömbbel elkészíthető, és *hta* kiterjesztéssel mentve az Intézőből dupla kattintással indítható!**

A programozási nyelvek „egyszerűségének” egyik mutatója, hogy milyen hosszú a „Helló világ!” program<sup>3</sup> forráskódja, továbbá mennyi idő alatt magyarázható el, érthető meg a működése. A forráskód Visual Basic Scriptben:

```
<script language="vbs">
  document.write "Helló világ!"
</script>
```

Gépeljük be a szöveget a Jegyzetömbbe, mentsük el a fájlt *Helló.hta* néven, majd dupla kattintással indítjuk el. Megnyílik egy ablak a „Helló világ!” szöveggel.

A program magyarázata két sorban összefoglalható. Az utasításokat a forráskódban a `<script>` és `</script>` sorok<sup>4</sup> közé írjuk. Az első sorban megadjuk a programozási nyelvet (rövidítve: vbs)<sup>5</sup>. A `document.write` utasítás<sup>6</sup> kiírja az utána álló, idézőjelek közé tett szöveget az ablakba.

A fenti VBScript programmal „egyenértékű” Pascal program például 7 sorból áll, soronként külön-külön (a `uses` utasításnál például nem is olyan egyszerű) magyarázattal:

```
program hello;
uses crt;
begin
  clrscr;
  writeln('Helló világ!');
  readln;
end.
```

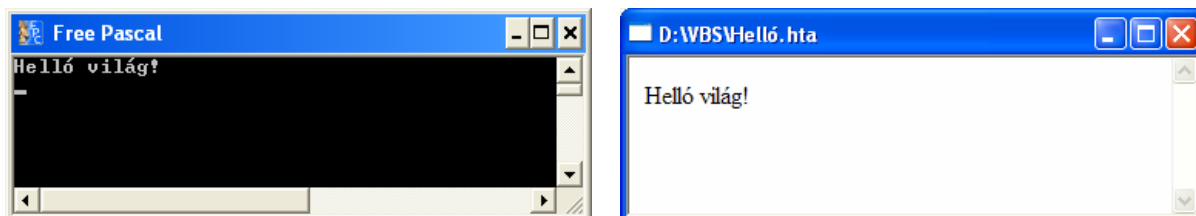
<sup>2</sup> A programokat a Windows VBScript interpretere futtatja (`\Windows\system32\vbscript.dll`).

<sup>3</sup> A program kiírja a képernyőre a „Helló világ!” szöveget. A Wikipedia több mint 100 programozási nyelven mutatja be a forráskódot ([hu.wikipedia.org](http://hu.wikipedia.org)).

<sup>4</sup> Valójában HTML-elemek, de ez nem lényeges a program szempontjából.

<sup>5</sup> Mivel más nyelv is használható. Alapértelmezett a JavaScript nyelv.

<sup>6</sup> Valójában a `document` objektum `write` metódusa, de ez nem lényeges a program szempontjából. A minősített eljárás-hívás azonban előrevetíti az objektumok használatát.



A Helló világ! program ablaka Pascalban és VBScriptben

### A Visual Basic Script előnyei a tankönyvi fejezet tanításánál:

- Grafikus felhasználói felületen fut, a programok a Windowsban szokásos ablakokat nyitnak meg.
- Nincs szükség semmilyen előkészítésre és telepítésre, a Windows rendelkezik a forráskód megírásához és a programok futtatásához szükséges eszközökkel (Jegyzetömb, beépített interpreter).
- A programok írásához nincsen szükség a fejlesztői felület ismertetésére, ismeretére!
- **Szintaktikus vagy futási hibák esetén magyar nyelvű üzenetablak jelenik meg a hiba helyével és megnevezésével!**
- Egyszerű, áttekinthető, következetes szintaxissal rendelkezik. A strukturális elemeknek mindig van zárótagja (például *end if*). Nincs szükség például a pontosvesszők keresgélésére az egymásba ágyazott elágazásoknál.
- Egymásba ágyazott *if ... else* utasítások helyett valódi többágú szelekciót alkalmazhatunk (*if ... elseif*).
- Ismeri a ciklusok mind a négyféle változatát (elől/hátultesztelő, kilépési/ismétlési feltétel).<sup>7</sup>
- A sztringeket a magyar ábécének megfelelő módon is összehasonlíthatjuk!
- Lehetővé teszi a grafikus felhasználói felület input/output elemeinek nagyon egyszerű kezelését (például szövegdoboz, parancsgomb).
- Erősen motiválja a diákokat, hogy a VBScript a weblapok egyik programozási nyelve.



Az interpreter hibaiüzenetei<sup>8</sup>

### A Visual Basic Script további előnyei

- Megalapozza az objektumorientált programozási nyelvek tanulását.
- Egyszerűen bemutatathatjuk a segítségével az eseményvezérelt programozási technikát.
- Felhasználhatjuk dinamikus weblapok, kliens- és szerveroldali szkriptek készítésére.
- A grafikus felhasználói felületen futó programok fejlesztése jóval könnyebb, mint más fejlesztői környezetek (például Lazarus, Delphi) esetén. A forráskód is jóval egyszerűbb.
- Felhasználható az MS Office alkalmazások programozásához. A makrók Visual Basic for Applications (VBA) programozási nyelvének szintaxisa és utasításkészlete szinte teljes egészében megegyezik a VBScripttel.
- A VBScript ismeretében könnyen áttérhetünk a Visual Basic-re, melynek 2005-ös változata tiszta objektumorientált, széles körben használható, modern programozási nyelv. A Visual Basic 2005 szerepel az emelt szintű informatika érettségien engedélyezett szoftverek listáján (2007). A Visual Basic 6.0 fejlesztői környezete szintén szerepel az informatika érettségien és az informatika OKTV-n engedélyezett szoftverek listáján.

<sup>7</sup> Lásd például a 10. osztályos informatika-tankönyv 124. oldalán.

<sup>8</sup> A hibakeresővel kapcsolatban lásd: Visual Basic Script összefoglaló ([www.ntk.hu](http://www.ntk.hu)).

- Nagyon egyszerű a modulokból álló programok összeillesztése, így **lehetővé teszi a csoportmunkát**. Nincs szükség unitok készítésére és fordítására. Az egyes csoportok által megírt eljárások minden további fordítás, linkelés nélkül csatolhatók a forráskódhoz.

## Segédanyagok a tankönyvhöz

A tankönyvhöz munkafüzet és a számonkérést segítő feladatlapok készültek (raktári szám: 16272/M, illetve 16272/F).

A munkafüzet feladatsorai a tankönyv leckéinek beosztását követik. A feladatokban szereplő fájlokat a Nemzeti Tankönyvkiadó webhelyéről tölthetjük le ([www.ntk.hu](http://www.ntk.hu)).

Az egyes feladatlapok különböző stílusban és nehézségi fokozatban mutatják be a tananyag számonkérését. Nem egyformán súlyozzák a témakör egyes részeit. Az A) és B) feladatsor megoldása nem igényli számítógép alkalmazását, programozási nyelv ismeretét. A C) feladatsor elméleti kérdéseket és egyszerű gyakorlati-programozási feladatokat is tartalmaz. A D) feladatsorban egy több részből álló program megírása szerepel. A feladatsor szövegében említett forráskód a Nemzeti Tankönyvkiadó webhelyén található ([www.ntk.hu](http://www.ntk.hu)).

A munkafüzetben és a feladatlapokban szereplő feladatok megoldásait a kiadó webhelyéről tölthetjük le ([www.ntk.hu](http://www.ntk.hu)).

A Nemzeti Tankönyvkiadónál hamarosan megjelenik a 9. és 10. osztályos tankönyvekhez kapcsolódó informatika feladatgyűjtemény.

## Segédanyagok az Interneten

A nyomtatott segédanyagok mellett a Nemzeti Tankönyvkiadó webhelyén ([www.ntk.hu](http://www.ntk.hu)) megtaláljuk:

- a tankönyvben szereplő példaprogramok forráskódját Free Pascal és Visual Basic Script nyelven (*hta* és *htm* formátumban is);
- a munkafüzetben szereplő feladatok forrásfájljait és megoldásait (Pascal, illetve VBScript nyelven);
- a feladatlapok forrásfájljait és megoldásait (Pascal, illetve VBScript nyelven);
- a programozási tételeket bemutató forrásfájlokat;
- lebegőpontos konvertálást végző programokat (*dectohex.hta*, *hextodec.hta*);
- a Pascal és a Visual Basic Script tanult elemeinek összefoglaló táblázatát;
- a Pascal és a Visual Basic Script szövegformázási eszközeinek összefoglaló táblázatát;
- Első lépések – VBScript bemutatót;
- egy részletes Visual Basic Script programozási útmutatót;
- VBScript gyorsreferencia-kártyát;
- teljes Microsoft VBScript referenciát;
- Microsoft VBScript Scripting Help-et.

Felhívjuk a figyelmet a Microsoft webhelyén található [VBScript User's Guide](#)-ra.



# AI-Hvárizmi öröksége

## Az óra célja

Az általános iskolában tanult definíciók ismétlése, pontosítása, szükség esetén bővítése.

## Fontos fogalmak, definíciók

Algoritmus, elemi lépés, utasítás, szekvencia, feltételes elágazás, szelekció, ciklus (számlálós/feltételes, előltesztelő/hátultesztelő), ciklusváltozó.

## Tevékenységek

- algoritmusok megfogalmazása, elemzése;
- algoritmusok elemi lépésekre bontása;
- a strukturált programozás építőelemeinek felismerése.

## Az óra menete

A mobiltelefon használatakor előforduló tevékenységek elemzésével idézzük fel az algoritmus, feltételes elágazás, ciklus fogalmát. Pontosítsuk a definíciókat, rögzítsük a feltételes elágazások és ciklusok típusait. Fordítsunk figyelmet az előltesztelő-hátultesztelő ciklusok, illetve az ismétlési és kilépési feltételek megkülönböztetésére. Vezessük be a ciklusváltozó fogalmát. Keressünk további példát a mobiltelefonhoz kapcsolódó algoritmusokra.

Beszélgük meg a Munkafüzet 2. feladatát.

## Megjegyzések

Az előforduló fogalmak szerepelnek az általános iskolai tantervben, így a tankönyv leckéje ismétlésnek tekinthető. Számítsunk azonban arra, hogy jó néhány diák eddigi tanulmányai során nem ismerkedett meg ezekkel a fogalmakkal.

A tankönyv leckéjében központi szerepet játszanak a mobiltelefon használatánál előforduló algoritmusok. Ezen tevékenységek hozzátartoznak a diákok mindennapi életéhez. A további leckékben is gyakran idézzük a mobiltelefonnal kapcsolatos algoritmusokat.

## Házi feladat

Munkafüzet 3. és 4. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Lásd a tankönyv kiemelt definícióit.
2. Nézzünk utána a mobiltelefon dokumentációjában.
3. Például ha hazaérünk és sötét van, akkor kapcsoljuk fel a villanyt.
4. Számlálós ciklus: meghatározott számú kört futni testnevelés órán. Feltételes ciklus: verstanulás, amíg hibátlanul nem megy.
5. a) Feltételes előltesztelő ciklussal:  
nyomd le a 7-es billentyűt!  
ISMÉTLÉS AMÍG nem S-betű  
nyomd le a 7-es billentyűt!  
CIKLUS VÉGE  
b) Feltételes, hátultesztelő ciklussal:  
ISMÉTLÉS  
nyomd le a 7-es billentyűt!  
AMÍG nem S-betű  
CIKLUS VÉGE
6. Kítáb al-dzsabr valmukábala (magyarul: A helyreállítás és az egyszerűsítés könyve)



# Algoritmisleíró eszközök

## Az óra célja

Algoritmusok szemléltetése folyamatábrával. A mondatszerű leírás bevezetése és alkalmazása.

## Fontos fogalmak, definíciók

Folyamatábra (blokkdiagram), utasítás, elágazás, ciklus, input/output ábrázolása a folyamatábrán, mondatszerű leírás.

## Tevékenységek

- Algoritmus folyamatábrájának elkészítése.
- Kész folyamatábra megértése, elemzése.
- Algoritmus mondatszerű leírásának elkészítése.
- Mondatszerű leírás értelmezése.

## Az óra menete

Idézzük fel az általános iskolában tanult módszereket az algoritmusok szemléltetésére. Szükség esetén ismer-tessük a folyamatábra elemeit.

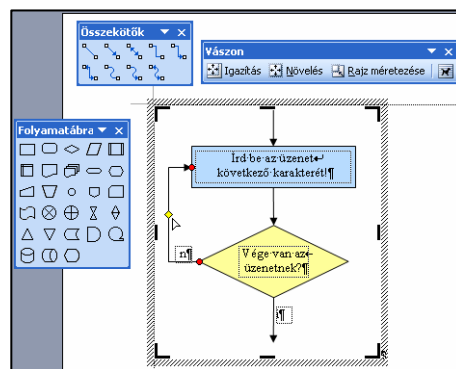
A tankönyvi példák alapján vezessük be az algoritmusok mondatszerű leírását.

Oldjuk meg a tankönyv 103. oldalának 3. feladatát, majd a Munkafüzet 3. feladatát.

## Megjegyzések

Az algoritmusok szemléltetése szintén része az általános iskolai tantervnek.

A grafikus szemléltetéshez a folyamatábrát és a struktúra-diagramot szokták használni. Mindkét módszernek inkább csak hátrányai vannak, de a folyamatábra legalább szemléletes. Megjegyezzük, hogy a modern szövegszerkesztők rendelkeznek folyamatábra-rajzoló eszközökkel. Az MS Word-ben használjuk például a *Rajzolás* eszköztár *Alakzatok/Folyamatábra* menüjét. A menü a felső sor mozgató-sával leválasztható az eszköztárról, így mindig a képernyőn marad. A folyamatábra elemeit *Összekötők* alkalmazásával kössük össze! Az összekötők együtt mozognak az elemekkel. Ehhez hasonló segítségeszközök a struktogramok rajzolásánál nem állnak a rendelkezésünkre.



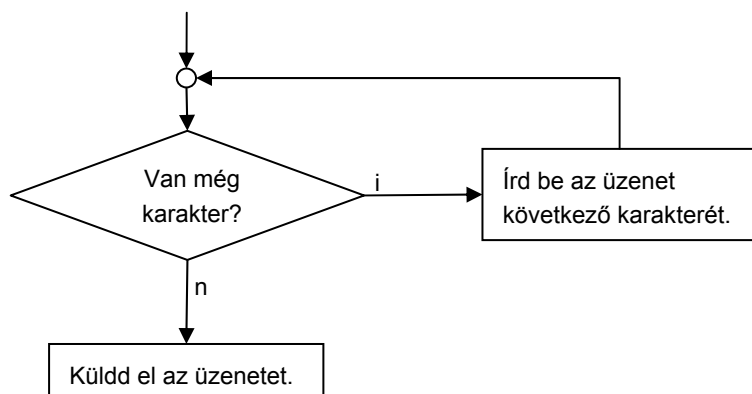
Folyamatábra rajzolása a Worddel

## Házi feladat

Munkafüzet 1. és 4. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Feltételes, hátultesztelő ciklus.
2. Elöltesztelő ciklussal, például:



3. Az SMS-küldés folyamatábrájához hasonlóan készíthetjük el.

4. Az algoritmus mondatszerű leírása ( $a$  div a maradékos osztás hányadosát, a  $mod$  pedig a maradékot jelenti):

*Átváltás kettes számrendszerbe*

Be: szám

bináris = ""

CIKLUS AMÍG szám  $\neq$  0

bináris = (szám mod 2) hozzáfűz bináris

szám = szám div 2

CIKLUS VÉGE

Ki: bináris

*Átváltás vége*

5. Az SMS-küldés folyamatábrájához hasonlóan készíthetjük el.

6. számláló = kezdőérték

CIKLUS AMÍG számláló  $\leq$  végérték

*ciklusmag utasításai*

számláló = számláló + 1

CIKLUS VÉGE

## Algoritmusok és a számítógép

### Az óra célja

Ismerkedés a programkészítés folyamatával, a programozási nyelvekkel, a programkészítés lépéseivel. A felhasználói és fejlesztői dokumentáció szerepe, részei.

### Fontos fogalmak, definíciók

Számítógépes program, programozási nyelv, kulcsszó, szintaxis, generációk, gépi kód, assembly, magas szintű programozási nyelvek, forráskód, fordítóprogram (*compiler*), értelmezőprogram (*interpreter*), szkript, integrált fejlesztői környezet, specifikáció, felülről lefelé való tervezés, kódolás, tesztelés, felhasználói és fejlesztői dokumentáció.

### Tevékenységek

- Felhasználói dokumentáció elemzése.

### Az óra menete

Vezessük be a számítógépes program és a programozási nyelv fogalmát. Ismertessük a szintaxis és a kulcsszavak szerepét.

Tekintsük át a programozási nyelvek generációit. A tankönyvi példák alapján mutassuk be az egyes generációk jellegzetességeit.

Oldjuk meg a Munkafüzet 2. feladatát.

Ismertessük a fordítóprogram és az értelmezőprogram célját, eltérő működését. Emeljük ki az integrált fejlesztőkörnyezet szerepét a programkészítés folyamatában.

Példákon keresztül mutassuk be a programkészítés lépéseit, a specifikáció fontosságát.

Soroljuk fel a felhasználói és fejlesztői dokumentáció összetevőit, emeljük ki jelentőségüket.

Beszéljük meg a Munkafüzet 3. a) feladatát.

### Megjegyzések

A lecke viszonylag sok fogalmat tartalmaz. Ezek megértése azonban nem jelent különösebb nehézséget, egy részük már ismert is a diákok számára.

A különböző generációk jellegzetességeit a tankönyvben szereplő példák (gépi kód, assembly, Pascal, Prolog), illetve a Munkafüzet 2. feladata alapján mutathatjuk be. A gépi kódú és assembly példák az Intel-mikroprocesszorok utasításkészletéből származnak.

Ha a diákok tanultak Logót, annak jellegzetességeit is felhasználhatjuk a fogalmak magyarázatánál.

### Házi feladat

Munkafüzet 3. b)–g) feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Lásd a tankönyv kiemelt definícióit.
2. A szkript definícióját lásd a tankönyv 105. oldalán. Szkriptnyelv például a Visual Basic Script vagy a JavaScript.
3. Lásd a 105. oldalon lévő definíciót.
4. A program forráskódját a 104. oldalon láthatjuk (Imagine Logo).
5. Az ablak címéből és a szövegdobozban olvasható feliratról arra következtethetünk, hogy a program a nevén szólítva üdvözölni fogja a felhasználót.
6. A szempontokat a tankönyv 106. oldalán találjuk.

## A programozási nyelvek elemei

### Az óra célja

Ismerkedés a tanult programozási nyelv fejlesztői környezetével, egyszerű program forráskódjának begépelése, mentése, a program futtatása.

### Fontos fogalmak, definíciók

Pascal: utasítás-zárójel, megjegyzés. Kulcsszavak: *begin, clrscr, end, program, readln, unit, uses, writeln*.

VBScript: *script*-elem, *language* tulajdonság, *br*-elem, megjegyzés. Utasítás: *document.write*.

### Tevékenységek

- Egyszerű program forráskódjának begépelés a tanult programozási nyelven.
- A begépelte program mentése, futtatása.
- A hibaüzenetek értelmezése, szintaktikus hibák javítása.
- Pascalban átváltás a felhasználói ablakba, *exe*-fájl készítése.
- VBScriptben mentés és megnyitás *hta*, illetve *htm* kiterjesztéssel.

### Az óra menete

Ismertessük a tankönyvben szereplő egyszerű példaprogram szerkezetét, utasításait.

Nyissuk meg a forrásfájl elkészítéséhez használt fejlesztői környezetet vagy szövegszerkesztőt, majd gépeljük be a forráskódot. Mutassuk be a forráskód mentését, a program futtatását. Az integrált fejlesztői rendszer használatából csak arra térjünk ki, amire szükség van ezeknek a tevékenységeknek a végrehajtásához.

Mutassuk be a már elkészült fájlok ismételt megnyitását is.

A program futtatásakor bizonyára előfordulnak szintaktikus hibák. Értelmezzük a hibaüzeneteket, térjünk ki a hibák javításának majd az ismételt futtatásnak a módjára.

Adjunk lehetőséget a diákoknak arra, hogy szabadon kísérletezhessenek a forráskód módosításával.

Pascal használata esetén röviden térjünk ki a telepítés lépéseire.

VBScript használata esetén ismertessük a nyelv különböző változatait (lásd a tankönyv 108. oldalán). Emeljük ki a program módosításának menetét (lásd a tankönyv 109. oldalán).

Készítsük el a Munkafüzet 4. feladatának programját.

### Megjegyzések

A tanult programozási nyelv kiválasztásával kapcsolatban lásd a tanári kézikönyv [Bevezetését](#).

A Free Pascal 2.0.4-es változatában módosították a fejlesztői környezet logóját.

VBScript használata esetén javasoljuk a *hta* kiterjesztés használatát.

VBScript programokat lehetőség szerint a FrontPage 2003 segítségével készítsünk.<sup>9</sup> Automatikus kódkiegészítő (IntelliSense) funkciója és a megjelenő szintaxis-minták nagymértékben megkönnyítik a forráskód begépelését (lásd a tankönyv 109. oldalán lévő ábrát).

Ha nem áll rendelkezésünkre a FrontPage 2003, akkor a forráskód elkészítéséhez az ingyenes *Notepad2* szövegszerkesztőt javasoljuk, melynek magyar nyelvű változata letölthető a [flo's freeware](#) webhelyről. A *Notepad2* telepítés nélkül futtatható. A program megjeleníti a sorok sorszámát, módosítható színekkel kiemeli a forráskód különböző szintaktikai elemeit, automatikus behúzásokat alakít ki és számos további eszközzel rendelkezik.

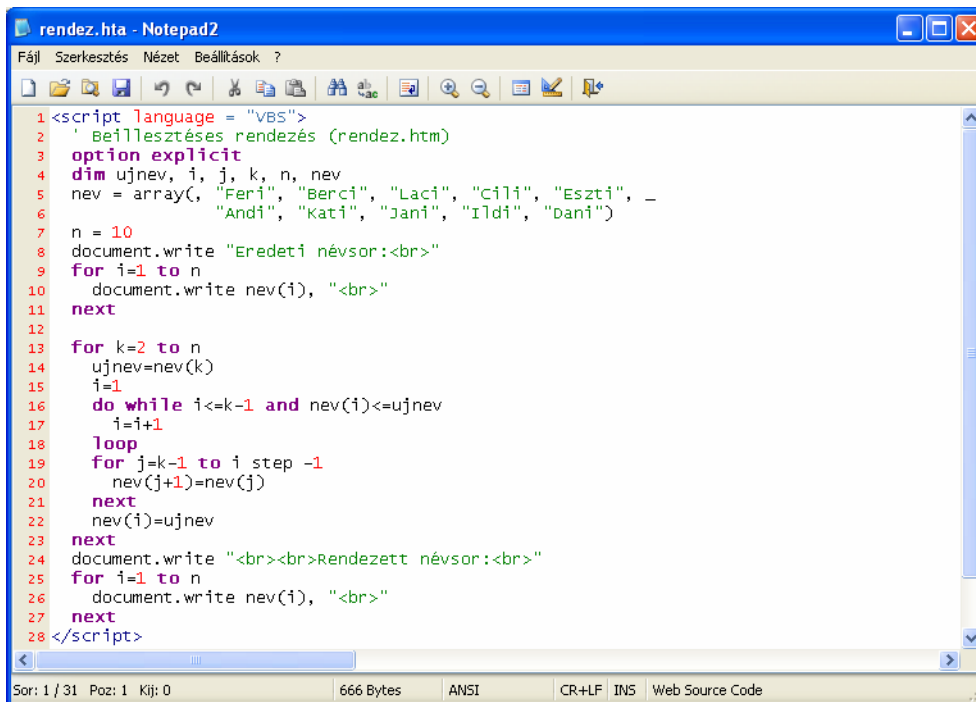
---

<sup>9</sup> A FrontPage nyomkövető és hibakereső eszközökkel is rendelkezik.

A *Notepad2* magyar nyelvű változatának szüksége van az *msvcr70.dll* fájlra, amit a környezeti változók beállítására helyett egyszerűen átmásolhatunk a *Notepad2*-t tartalmazó mappába. A *dll* fájlt megtaláljuk például a `C:\Program Files\Microsoft Office\OFFICE10\VS Runtime` illetve a `C:\Program Files\Microsoft Office\OFFICE11\VS Runtime` mappában, vagy letölthetjük a [DLL-files.com](http://DLL-files.com) webhelyéről.

A *Notepad2*-t Pascal-programok begépeléséhez is használhatjuk, mert a Pascal editoránál kényelmesebb eszközöket nyújt a forráskód elkészítéséhez.

A *Notepad2* *Nézet/Séma* kiválasztása menüjével választhatjuk ki a programozási nyelvhez tartozó sémát, amit hozzá is rendelhetünk az adott kiterjesztéshez.



```
rendez.hta - Notepad2
Fájl Szerkesztés Nézet Beállítások ?
1 <script language = "VBS">
2 ' Beillesztés rendezés (rendez.htm)
3 option explicit
4 dim ujnev, i, j, k, n, nev
5 nev = array( "Feri", "Berci", "Laci", "Cili", "Eszti", _
6           "Andi", "Kati", "Jani", "Ildi", "Dani")
7 n = 10
8 document.write "Eredeti névsor:<br>"
9 for i=1 to n
10 document.write nev(i), "<br>"
11 next
12
13 for k=2 to n
14 ujnev=nev(k)
15 i=1
16 do while i<=k-1 and nev(i)<=ujnev
17 i=i+1
18 loop
19 for j=k-1 to i step -1
20 nev(j+1)=nev(j)
21 next
22 nev(i)=ujnev
23 next
24 document.write "<br><br>Rendezett névsor:<br>"
25 for i=1 to n
26 document.write nev(i), "<br>"
27 next
28 </script>
```

*Forráskódkészítés a Notepad2 programmal*

### Házi feladat

A Munkafüzet 3. és 5. vagy 6. feladata.

### Az ellenőrző kérdések és feladatok megoldása

1. A Free Pascal letölthető a [www.freepascal.org](http://www.freepascal.org) webhelyről. A VBScript programok futtatásához nincs szükség további letöltésekre.
2. Nyissuk meg a forráskód begépeléséhez használt editort, majd írjunk be valamilyen egyszerű programot.
3. A folyamatábra csupán az egyes utasításokat tartalmazó téglalapokból áll.
4. Az output utasításban szereplő karaktersorozat helyett írjuk be például, hogy „Üdvözöllek!”
5. Lásd például az [internetes segédanyagok](#) között található szövegformázási összefoglalót.
6. A *script*-elemet a *body*-elemben helyezhetjük el:

```
<html>
  <head>
  </head>
  <body>
    <script>
      a script utasításai
    </script>
  </body>
</html>
```

7. Lásd például az [internetes segédanyagok](#) között található VBScript programozási összefoglalót.

# Elemi adattípusok

## Az óra célja

A változók fogalmának, legfontosabb típusainak, a deklaráció módjának a megismerése. Konstansok bevezetése.

## Fontos fogalmak, definíciók

Változó, elemi adattípus, numerikus, szöveges (karakterlánc, sztring), logikai típus. Egész és valós (lebegőpontos) típus. Deklaráció.

Pascal kulcsszavak: *const*, *var*, a változótípusok kulcsszavai.

VBScript kulcsszavak és utasítások: *const*, *dim*, *option explicit*.

## Tevékenységek

- Az adatnak megfelelő elemi változótípus kiválasztása.
- Változók és konstansok deklarációja a tanult programozási nyelven.

## Az óra menete

Vezessük be a változó definícióját, ismertetjük szerepét a számítógépes programokban.

Tekintsük át az elemi adattípusokat. Emeljük ki a típus helyes megválasztásának szempontjait (értékkészlet, memória-felhasználás).

Elemezzünk különböző adatokat, milyen típusú változót választanánk a tárolásukhoz? Használjuk fel a Munkafüzet 4. feladatát.

Ismertessük a változók deklarációjának célját és módját.

Módosítsuk az előző óra egyszerű programját úgy, hogy a képernyőn megjelenő szöveget először változóban tároljuk, majd a változó értékét írjuk ki.

Vezessük be a konstans fogalmát, emeljük ki szerepét a program utólagos módosításánál.

Ha van rá idő és mód, akkor mutassunk példákat a lebegőpontos számábrázolásra (lásd a tankönyv 112. oldalán).

## Megjegyzések

A tankönyvi lecke meglehetősen lexikális-jellegű ismereteket foglal össze. Színesebbé tehetjük a tárgyalás-módot, ha az egyes típusokat egy programban deklaráljuk, majd kiírjuk a változók értékét. Kísérletezzünk különböző számtartományokkal, több-kevesebb tizedesjegyet tartalmazó valós értékekkel. Szükség esetén már ebben a leckében bevezethetjük az értékadó utasítás fogalmát.

Felidézhetjük a 9. osztályos anyagból a decimális-bináris, illetve kettes komplement kódba történő konverziót (Munkafüzet 2. feladat). Gyakoroltathatjuk a lebegőpontos átváltásokat is (Munkafüzet 3. feladat). Kidolgozott példákat a munkafüzet forrásfájljai között találunk (lásd: [www.ntk.hu](http://www.ntk.hu)). A konverziót például a tankönyv forrásfájljai között található programokkal ellenőrizhetjük (*dectohex.hta*, *hexodec.hta*).

Egy további lehetőség, ha a lecke anyagát nem külön órán tárgyaljuk, hanem belefoglaljuk a további leckékbe úgy, hogy mindig csak az éppen aktuális és szükséges ismeretekre térünk ki.

## Házi feladat

Tankönyv 112. oldal 4. és 5. feladat, Munkafüzet 1. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Lásd a tanult programozási nyelv dokumentációját.
2. Lásd a tankönyv 111. oldalán lévő táblázatot és a Munkafüzet 4. feladatát.
3. Lásd a tankönyv 111. oldalán lévő táblázatot.
4. Körülbelül 4,2 milliárd, azaz  $2^{32}$  érték tárolható. A tároláshoz tehát  $32/8 = 4$  bájtot használnak fel.
5. Valós.
6. Lásd a tanult programozási nyelv dokumentációját.

# Értékadás a változóknak

## Az óra célja

Az értékadó utasítások megismerése, adatok beolvasása a billentyűzetről és kiírása a képernyőre, formázott megjelenítés. Kifejezések alapműveletekkel, egyszerű függvények használata. Típuskonverzió a beolvasásnál (VBScript).

## Fontos fogalmak, definíciók

Értékadó utasítás, kifejezés, egyszerűbb matematikai és sztringkezelő függvények.

Pascal utasítások: *readln*, *write* (beolvasás).

VBScript függvények: *formatnumber*, *inputbox*, típuskonverziós függvények.

## Tevékenységek

- Aritmetikai kifejezések használata, sztringek összefűzése.
- Értékadó utasítások kódolása.
- A változók értékének megjelenítése a képernyőn, formázott kiírás.
- Adatok beolvasása a billentyűzetről.
- VBScript: az *inputbox* használata, típuskonverzió a beolvasásnál.
- Egyszerűbb függvények használata a kifejezésekben és az értékadó utasításokban.

## Az óra menete

Definiáljuk a kifejezés fogalmát, mutassunk példát aritmetikai kifejezésekre a tanult programozási nyelvben. Emlékeztessük a diákokat a műveletek precedenciájára.

Vezessük be az értékadó utasítást, mutassuk meg a szintaxisát. Írjunk fel példaként numerikus és sztringkifejezéseket tartalmazó értékadó utasításokat.

Térjünk ki az  $a = a + 1$  jellegű értékadások jelentésére, szerepére.

Ismertessük a változók értékének megjelenítésére vonatkozó módszereket. Használjuk fel a formázott megjelenítés eszközeit.

Nyissuk meg és elemezzük a forrásfájlok között található *bank* programot. Végezzünk kisebb módosításokat a programon. További példákat válogathatunk a Munkafüzet 2., 3., 6. és 7. feladatából.

Mutassuk be az adatbeolvasás módját. VBScriptben emeljük ki a típuskonverzió fontosságát a beolvasásnál.

Nyissuk meg és elemezzük a forrásfájlok között található *szia* programot.

Használjunk fel néhány egyszerűbb függvényt a kifejezésekben (lásd a tankönyv 115. oldalán). A véletlenszám-generátor segítségével már most is érdekes programokat készíthetünk (például lottószámok választása 5 értékadó utasítással). A függvények használatát a *függvény* program segítségével mutathatjuk be.

Ha időnk engedi, beszéljük meg a Munkafüzet 8. feladatát.

## Megjegyzések

Ezen az órán kezdünk először „igazi” programokat írni, melyek már helyettünk végeznek el feladatokat (kifejezések kiértékelését). Ne vesszünk el a kifejezések és az értékadó utasítások szintaxisának részletes taglalásában, inkább példákon keresztül mutassuk be az alkalmazásukat. Az Algoritmusok és adatok fejezet nem igényli bonyolult értékadó utasítások alkalmazását. Csak a tanult algoritmusok szemléltetéséhez van szükségünk rájuk. Ezt a tankönyvi leckét az előző leckével összevonva is tárgyalhatjuk.

## Házi feladat

Tankönyv 115. oldal 2., 4., 5. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Lásd a tankönyv 113. oldalán szereplő definíciót.
2. Az eredmény 3,5.
3. Írassuk ki például a  $2+3\cdot 4$  kifejezés értékét.
4. A kiírási utasításokban használjuk fel a formázott megjelenítés eszközeit.
5. Az adatokat a *readln* utasítással (Pascal), illetve az *inputbox* függvénnyel (VBScript) kérjük be.

6. A program algoritmus:

*Életkor*

Be név, szülév, jelenév

Ki: név, " ", jelenév – szülév, " éves."

*Életkor vége*

## Feltételes elágazások programozása

### Az óra célja

Feltételes elágazások kódolása, a logikai műveletek és relációk felhasználása az elágazások feltételeiben. Többirányú elágazások alkalmazása.

### Fontos fogalmak, definíciók

Relációk és logikai műveletek jelölése.

Pascal kulcsszavak: *if ... then ... else, not, and, or, xor*.

VBScript kulcsszavak és függvények: *if ... then ... else ... end if, strcmp* (sztringek összehasonlítása a magyar ábécének megfelelően), *not, and, or, xor*.

### Tevékenységek

- Feltételes elágazások kódolása.
- Összetett feltételek megfogalmazása és kódolása.
- Logikai műveletek alkalmazása a feltételekben.
- Többirányú elágazás kódolása.

### Az óra menete

Ismételjük át a feltételes elágazásról tanultakat.

Nyissuk meg a *korzet* programot. Elemezzük a szerkezetét, majd futtassuk a programot.

Oldjuk meg a tankönyv 118. oldalán az 1. feladatot. Lehetőség szerint oldjuk meg a 3. feladatot is.

Mutassuk be a relációk kódolását. Hívjuk fel a figyelmet a sztringek összehasonlításának sajátosságaira (kisbetűk-nagybetűk, ékezetes karakterek).

Ismételjük át a logikai műveletek igazságtáblázatát (lásd például a tankönyv 117. oldalán).

A *dijak* program algoritmusát és folyamatábráját alapján (tankönyv 118. oldal) mutassuk be a többirányú elágazások kódolását.

Nyissuk meg és elemezzük a *dijak* program forráskódját. Futtassuk a programot. Végezzünk kisebb módosításokat az algoritmuson.

Oldjuk meg a tankönyv 118. oldalán a 6. feladatot. Hívjuk fel a tanulók figyelmét az áttekinthető program-szerkezet kialakítására, az elágazások egyszerűsítésére.

Oldjuk meg a munkafüzet 7. feladatát.

### Megjegyzések

Ebben a leckében találkozunk először döntésekkel a programokban. Már nem egyszerűen csak képletek eredményeit határozzuk meg, hanem megtanítjuk a számítógépet az alkalmazkodásra, a különböző esetek szétválasztására.

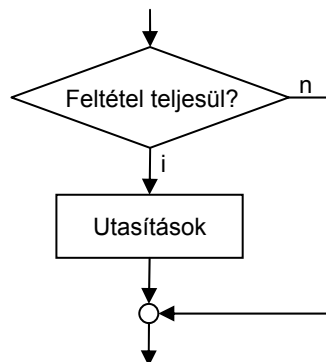
Ha van rá mód, célszerű a lecke anyagát két tanórán tárgyalni. A Munkafüzet számos feladatot biztosít a gyarkolásra.

### Házi feladat

Tankönyv 118. oldal 1. és 4. feladat. Munkafüzet 7. feladat.

### Az ellenőrző kérdések és feladatok megoldása

1. A módosított folyamatábra:



2. A program algoritmus:

*Jelszóellenőrzés*

jelszó = "xxx"

Be: próba

HA próba = jelszó AKKOR

Ki: helyes jelszó

EGYÉBKÉNT

Ki: hibás jelszó

ELÁGAZÁS VÉGE

*Jelszóellenőrzés vége*

3. A program algoritmus:

*Legnagyobb a háromból*

Be: a, b, c

legnagyobb = a

HA b > legnagyobb AKKOR

legnagyobb = b

ELÁGAZÁS VÉGE

HA c > legnagyobb AKKOR

legnagyobb = c

ELÁGAZÁS VÉGE

Ki: legnagyobb

*Legnagyobb a háromból vége*

Felhívjuk a figyelmet arra, hogy ezt a feladatot így KELL megoldani! Ez az algoritmus könnyen általánosítható tetszőlegesen sok szám közül a legnagyobb kiválasztására. A néhány könyvben látható, páronkénti összehasonlítással történő kiválasztás feleslegesen bonyolult, áttekinthetetlen programszerkezetet eredményez. Vegyük észre, hogy az előzetes értékadás (legnagyobb = a) nagymértékben egyszerűsíti az algoritmust.

4. A program algoritmus:

*Ruhapróba*

konstans olcsó = 5000

konstans drága = 15000

Be: ár

HA ár ≤ olcsó AKKOR

Ki: "olcsó"

EGYÉBKÉNT HA ár < drága AKKOR

Ki: "átlagos"

EGYÉBKÉNT

Ki: "drága"

ELÁGAZÁS VÉGE

*Ruhapróba vége*

5. A program algoritmus:

*Percdíjak számítása*

Be: körzet, időszak

HA körzet = 30 AKKOR

HA időszak = "csúcs" AKKOR

percdíj = 24

EGYÉBKÉNT

percdíj = 12

ELÁGAZÁS VÉGE

EGYÉBKÉNT

HA időszak = "csúcs" AKKOR

percdíj = 36

EGYÉBKÉNT

percdíj = 18

ELÁGAZÁS VÉGE

ELÁGAZÁS VÉGE

*Percdíjak számítása vége*

6. Lásd az 5. feladat megoldását.



# Szövegdozók, parancsgombok és társaik (kiegészítő anyag)

## Az óra célja

Elemi ismerkedés az objektumokkal. A grafikus felhasználói felület input/output objektumainak kezelése. Eseménykezelő eljárások készítése.

## Fontos fogalmak, definíciók

Objektumok, az objektumok azonosítása, tulajdonságaik (attribútumaik) és metódusai. A *window* és a *document* objektum.

VBScript függvények és utasítások: *msgbox*, *sub*, *end sub*.

HTML-elem: *input* (*text* és *button*).

HTML-objektumok tulajdonságai: *id*, *innertext*, (*innerHTML*), *bgcolor*, *text*, *type*, *value*.

A *window* objektum metódusai: *resizeto*, *close*.

Események: *onclick*, *ondblclick*, *onkeydown*, *onmouseover*, *onmouseout*.

## Tevékenységek

- A szkriptek beillesztése a HTML-kódba.
- HTML-objektumok attribútumainak megadása a HTML-kódban és a szkriptekben.
- A *window* objektum metódusainak az alkalmazása.
- Eseménykezelő eljárások készítése, hozzárendelés a parancsgombhoz.
- INPUT-objektumok használata az adatbeolvasáshoz.
- Kiírás az *msgbox* függvénnyel.

## Az óra menete

Ismételjük át a HTML-kód felépítését és alapelemeit. Határozzuk meg a szkriptek helyét a HTML-kódban.

Vezessük be az objektumok fogalmát, mutassuk be tulajdonságaikat, a tulajdonságok megadásának a módját. A *szinez* program segítségével mutassuk be, hogyan lehet a tulajdonságokat a szkriptekben megadni, illetve módosítani.

Az *ablak* program segítségével definiáljuk az objektumok (*window*) metódusait, mutassuk be a használatukat.

Vezessük be az események és eseménykezelő eljárások fogalmát. A *kattint* program segítségével elemezzük az eseménykezelő eljárások működését. Töltsük be és futtassuk az *ideoda* programot. Végezzünk módosításokat a forráskódon.

Ismertessük az INPUT-objektumokat. A *masol* program segítségével mutassuk be a szövegdozossal történő adatbeolvasást, az *uzenet* programmal pedig az *msgbox* használatát.

Ha marad időnk, mutassunk további példákat eseménykezelő eljárásokra (lásd a tankönyv 121. oldalán).

## Megjegyzések

A VBScript igazi előnyét a HTML-kód objektumainak elérése, az eseménykezelő eljárások alkalmazása mutatja meg. Ezen eszközök nélkül, az algoritmusok megvalósítása, az adatbeolvasás, az eredmények megjelenítése nem sokkal kényelmesebb és látványosabb a Pascalnál. Ezért feltétlenül szükségesnek tartjuk a grafikus felhasználói felület input/output objektumainak a bemutatását, az eseményvezérelt programozás bevezetését. A parancsgombok és szövegdozók használata (utalva a dinamikus weblapokkal való kapcsolatukra) nagymértékben motiválja a diákokat a programozás elemeinek az elsajátításában.

A tankönyvi lecke meglehetősen sok új fogalmat és tevékenységet tartalmaz. Nem kell azonban elmélyedni a részletekben. Célunk csupán az, hogy a parancsgombhoz rendelt eseménykezelő eljárások segítségével be tudjuk olvasni egy szövegdozók tartalmát. A szövegdozók kényelmesebb adatbevitelt tesznek lehetővé, mint az input-függvény. A parancsgombok és eseménykezelő eljárások segítségével sokkal barátságosabb programokat írhatunk.

Mivel az INPUT-objektumok és eseménykezelő eljárások nem szerepelnek a tantervben, a tankönyv további fejezetei nem hivatkoznak ezekre az eszközökre. Ha bemutatjuk használatukat, célszerű a további leckékben szövegdozókkel végezni a beolvasást, és eseménykezelő eljárásokkal az algoritmusok megvalósítását.

## Házi feladat

Munkafüzet 5. és 7. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Cseréljük fel a *bgcolor* és *text* tulajdonságok értékét.
2. Hajtsuk végre a megadott weblap megnyitását.
3. Helyezzünk el szövegdobozokat és egy parancsgombot a *body* elembe, majd a parancsgomb *onclick* eseménykezelő eljárásában rendeljük hozzá a megfelelő változóhoz a szövegdobozok *value* tulajdonságát. Ne feledkezzünk meg a típuskonverzióról (például *cint*). Az algoritmus további utasításait is az eseménykezelő eljárás tartalmazza. Példaként a tankönyv forráskódjai között megtaláljuk a *dijak* program szövegdobozos változatát (*dijakszoveg.hta*).
4. Lásd a tankönyv forrásfájljai között található *kattintaszoveg.hta* fájlt.
5. Lásd a tankönyv forrásfájljai között található *minmax.hta* fájlt.
6. Lásd a tankönyv forrásfájljai között található *bezar.hta*, illetve *bezar.htm* fájlt. A *htm* fájl bezárásakor a böngésző rákérdez az engedélyezésre.

## Telefonköltség

### Az óra célja

Feltételes ciklusok kódolása. Az összegezés algoritmusának és alkalmazása.

### Fontos fogalmak, definíciók

Az összegezés algoritmusának.

Pascal kulcsszavak: *while ... do, repeat ... until*.

VBScript kulcsszavak: *do, while, loop*.

### Tevékenységek

- Feltételes ciklusok kódolása.
- Kilépési és ismétlési feltételek negálása.
- Az összegezés algoritmusának felírása, alkalmazása.

### Az óra menete

Ismételjük át a feltételes ciklus fogalmát, típusait (elől/hátultesztelő, ismétlési/kilépési feltétel).

Vezessük be az összegezés algoritmusának célját, fogalmazzuk meg az algoritmust.

A Munkafüzet 2. feladatának a segítségével beszéljük meg a ciklus előtti értékadások szerepét az összegezés algoritmusában (lásd a tankönyv 122. oldalán).

Készítsük el az összegezés algoritmusának kódját a tanult programozási nyelven (*osszeg* program).

Mutassuk be a feltételes ciklus típusainak kódolását a tanult programozási nyelven.

Írjuk fel az összegezés algoritmusát hátultesztelő ciklussal (*hatul* program). Hasonlítsuk össze a kétféle megvalósítást.

Írjuk fel az összegezés algoritmusát kilépési feltétellel (*kilepesi* program). Gyakoroltassuk az ismétlési és kilépési feltételek átalakítását.

A Munkafüzet 3. és 6. feladatához hasonló példákkal érdekesebbé tehetjük az óra menetét.

### Megjegyzések

Ebben a leckében tárgyaljuk az első, úgynevezett programozási tételt. Bár még nem állnak rendelkezésünkre tömbök, az összegezés algoritmusának egy valós, és könnyen eltéveszthető folyamatot (sok szám összeadása) automatizál, színesebbé téve a fejezet tárgyalását.

Hívjuk fel a diákok figyelmét az előzetes értékadások szerepére, kapcsolatukra a rájuk következő ciklus típusával. Beszéljük meg az egyes típusok alkalmazásának feltételeit, mikor melyiket célszerű felhasználni.

Emeljük ki, hogy a tanult programozási nyelvben milyen ciklusokat lehet közvetlenül kódolni, illetve hogyan valósíthatjuk meg a többi típust.

### Házi feladat

Tankönyv 124. oldal, 3. feladat; Munkafüzet 4. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Írjuk be ellentétes előjellel, és adjuk hozzá az eddigiek összegéhez.
2. Az elől/hátultesztelő átalakítás mintáját a tankönyv 123. oldalán látjuk. A kilépési/ismétlési feltétel átalakításához tagadjuk a feltételt, és cseréljük fel az igen/nem ágakat.
3. A program algoritmus:

*Maradék díj*

Be: érték

*ide kerül az összegezés algoritmus*

Ki: érték – összeg

*Maradék díj vége*

4. A program algoritmus:

*Átlagos magasság*

összeg = 0

db = 0

CIKLUS

Be: magas

HA magas  $\neq$  0 AKKOR

összeg = összeg + magas

db = db + 1

ELÁGAZÁS VÉGE

AMÍG magas  $\neq$  0

CIKLUS VÉGE

Ki: összeg/db

*Maradék összeg vége*

5. Lásd az *osszeg.hta* és az *osszeg2.hta*, illetve a *kilepesi.hta* és a *kilepesi2.hta* programokat.

## Összetett adattípusok

### Az óra célja

A tömbök megismerése és kezelése a programokban. Az összetett adattípus fogalma, néhány fajtája.

### Fontos fogalmak, definíciók

Tömb, tömbelem, index, dimenzió. Típusos konstans (Pascal). Összetett adattípus (esetleg sor, verem).

Pascal kulcsszavak: *array ... of*.

VBScript függvény: *array*.

### Tevékenységek

- Tömbök deklarálása, hivatkozás a tömbelemekre, értékadás a tömb elemeinek.
- Tömbök használata a programokban, a tömbelemek feldolgozása ciklussal (beolvasás, kiírás, módosítás).

### Az óra menete

Vezessük be a tömb fogalmát, mutassunk rá a szerepére. Definiáljuk a tömbbel kapcsolatos fogalmakat.

Példákon keresztül mutassuk meg a tömbelemekre történő hivatkozást, az indexek megadásának a módját, a tömbelemek módosítását.

A Munkafüzet 3. és 4. feladatának segítségével gyakoroljuk az indexek használatát.

Mutassuk meg, hogyan egyszerűsíthetjük a tömbelemek előzetes értékadását (Pascal: típusos konstans, VBScript: *array* függvény).

A *beolvas* program elemzésével mutassuk be a tömbök beolvasását. Hívjuk fel a figyelmet az utasítások megfelelő sorrendjére, a beolvasott elemek számának a meghatározására.

Keressünk példákat további összetett adattípusokra.

## Megjegyzések

Eddigi leckéink példáit erősen korlátozta a tömb hiánya. Most ezt a hiányosságot pótoljuk.

A több dimenziós tömböket megemlíjtük, de nem alkalmazunk ilyen típust a példáinkban.

Visual Basic Scriptben a tömbelemeket 0-tól kezdve indexeljük. A tankönyvi leckékben azonban általában nem használtuk fel a 0 indexű elemet.

A diákok az előző fejezetekből már ismerik az összetett adattípus fogalmát (például az Excelben két dimenziós táblázatok celláira hivatkozunk, az adatbázis-kezelésnél rekordokkal dolgozunk stb.).

## Házi feladat

A tankönyv 127. oldalán lévő 4. és 5. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Hasonlít például: több elem tárolására használható. Különbözik például: a halmaznál nincs értelmezve az elemek sorrendje.
2.  $3 \cdot i - 1$  ( $i = 1$ -től  $n/3$ -ig)
3. A program algoritmusa:

*Tömbelemek összege*

*értékkadás az elemeknek*

összeg = 0

CIKLUS  $i = 1$ -től 5-ig

összeg = összeg + tömb( $i$ )

CIKLUS VÉGE

Ki: összeg

*Tömbelemek összege vége*

4. A név beolvasásánál írjuk ki az  $n$  értékét is.
5. Az algoritmusban szereplő feltételes elágazás módosítása:

HA név( $n$ )  $\neq$  "" AKKOR

Be: szám

HA szám = 30 AKKOR

telefonszám( $n$ ) = szám

EGYÉBKÉNT

Ki: hibaüzenet

ELÁGAZÁS VÉGE

ELÁGAZÁS VÉGE

Hibás szám esetén egy feltételes ciklussal újra bekérhetjük a számot.

6. Lásd a *betuz.pas* programot.

# Számlálós ciklusok

## Az óra célja

Számlálós ciklusok kódolása és felhasználása a programokban. Ciklusok egymásba ágyazása. A megszámlálás algoritmusának felírása és alkalmazása.

## Fontos fogalmak, definíciók

A megszámlálás algoritmus.

Pascal kulcsszavak: *for ... to/downto ... do.*

VBScript kulcsszavak: *for ... to ... step ... next.*

## Tevékenységek

- Számlálós ciklusok kódolása és alkalmazása.
- A ciklusváltozó értékének felhasználása különböző értékek képzéséhez és a tömbök indexeléséhez.
- Visszafelé számláló ciklus felírása és alkalmazása.
- Ciklusok egymásba ágyazása.
- A megszámlálás algoritmusának felírása és alkalmazása.

## Az óra menete

Ismételjük át a számlálós ciklus fogalmát és alkalmazását.

Mutassuk be a számlálós ciklus kódolását a tanult programozási nyelven.

A *negyzetek* program segítségével mutassuk be a ciklusváltozó felhasználását a kifejezésekben. Végezzünk kisebb módosításokat a programon (lásd például a tankönyv 128. oldalán). Térjünk ki arra az esetre, amikor a léptetés értéke nagyobb, mint 1, illetve negatív (visszafelé számláló ciklus). Felhasználhatjuk a *paros* és a *vissza* programok forráskódját.

A *lotto* program segítségével mutassuk be a ciklusok egymásba ágyazását. Hívjuk fel a figyelmet a ciklusváltozók elnevezésére, illetve a strukturált írásmód szerepére.

A tankönyvben szereplő példa alapján építsük fel a megszámlálás algoritmusát. Használjuk fel a *megszamlal* programot.

A Munkafüzet 2. feladata alapján elemezzük a megszámlálás algoritmusának szerkezetét.

A Munkafüzet 3., 4. és 6. feladatával színesíthetjük az óra menetét.

## Megjegyzések

Hasonlítsuk össze a számlálós és feltételes ciklusokat. Hívjuk fel a figyelmet a tanult programozási nyelv sajátosságaira (a számlálós ciklus elől- vagy hátultesztelő, mikor lép ki a ciklusból az előre és visszafelé számláló ciklus, a ciklusfejből szereplő értékek, illetve a ciklusváltozó módosításának lehetősége és következménye a cikluson belül stb.).

## Házi feladat

Tankönyv 129. oldal, 2., 3., 4. feladat.

## Az ellenőrző kérdések és feladatok megoldása

1. Az előre számláló ciklus esetén a ciklusváltozó kezdőértéke legyen nagyobb, mint a végérték.
2. Egyszer sem, egyszer, kétszer, hatszor (!).
3. A program algoritmus:

*Héttel osztható számok*

CIKLUS  $i = 105$ -től 7-ig visszafelé 7-esével

Ki:  $i$

CIKLUS VÉGE

*Héttel osztható számok vége*

Pascalban futtassuk a ciklust  $i = 35$ -től 1-ig visszafelé, és írassuk ki a  $7 \cdot i$  értékét (vagy alkalmazzunk feltételes ciklust).

4. A kiírást végző ciklus módosítva:

db = 0

CIKLUS i = 5-től 1-ig visszafelé

Ki: név(i)

HA elsőbetű(név(i)) = "K" AKKOR

db = db + 1

ELÁGAZÁS VÉGE

CIKLUS VÉGE

Ki: db

Az *elsőbetű* függvény Pascalban: `nev[i][1]` vagy `nev[i,1]`, VBScriptben: `left(nev(i),1)`.

5. A program algoritmus:

*M-szavak*

CIKLUS i = 1-től 5-ig

CIKLUS

Be: temp

AMÍG elsőbetű(temp) ≠ "m"

CIKLUS VÉGE

szó(i) = temp

CIKLUS VÉGE

*M-szavak vége*

6. A belső ciklust egészítsük ki a következő utasításokkal:

HA szám < 1 vagy szám > 90 AKKOR

Ki: hibaüzenet

ELÁGAZÁS VÉGE

## Keresés a tömbben

### Az óra célja

A lineáris és bináris keresés algoritmusának megismerése, alkalmazása.

### Fontos fogalmak, definíciók

Lineáris keresés, bináris (logaritmikus) keresés.

### Tevékenységek

- A lineáris keresés algoritmusának felírása és alkalmazása.
- A bináris keresés algoritmusának felírása és alkalmazása.

### Az óra menete

Ismertessük a keresési algoritmusok szerepét.

Írjuk fel a lineáris keresés algoritmusát. A *keres* program segítségével alkalmazzuk az algoritmust.

Utaljunk a keresés irányára, több egyforma elem megtalálásának módjára. Beszéljük meg a tankönyv 131. oldalán lévő 2. és 3. feladatot.

Térjünk ki a keresés hatékonyságának fontosságát.

Írjuk fel a bináris keresés algoritmusát. A *binaris* program segítségével alkalmazzuk az algoritmust. Oldjuk meg a tankönyv 131. oldalán lévő 5. feladatot.

Becsüljük meg az egyes keresések átlagos lépésszámát.

### Megjegyzések

A keresés, főleg a bináris keresés algoritmus már jártasságot igényel az algoritmusok felírásában és értelmezésében. A rendezés mellett ez a legbonyolultabb algoritmus, amit ebben a fejezetben tárgyalunk.

A lineáris keresés algoritmus kissé eltér a szokásos felírástól, mert nem minden programozási nyelv teszi lehetővé a logikai kifejezések rövidre zárását, így túlléphetjük a megengedett indexhatárt. Ennek elkerüléséhez a ciklust az utolsó előtti elemig futtatjuk, és szükség esetén az utolsó elemet a kilépés után vizsgáljuk meg.

*Lineáris keresés rövidzár nélkül*

```
i = 1
CIKLUS AMÍG i < n és nem T(tömb(i))
  i = i + 1
CIKLUS VÉGE
HA T(tömb(i)) AKKOR
  Ki: i, van megfelelő elem
EGYÉBKÉNT
  Ki: nincs megfelelő elem
ELÁGAZÁS VÉGE
```

*Lineáris keresés rövidzár nélkül vége*

*Lineáris keresés rövidzárral*

```
i = 1
CIKLUS AMÍG i ≤ n és nem T(tömb(i))
  i = i + 1
CIKLUS VÉGE
HA i ≤ n AKKOR
  Ki: i, van megfelelő elem
EGYÉBKÉNT
  Ki: nincs megfelelő elem
ELÁGAZÁS VÉGE
```

*Lineáris keresés rövidzárral vége*

Didaktikai szempontból sem tartjuk helyesnek a rövidzár kihasználását, mert

- megszünteti a logikai műveletek kommutativitását;
- a ciklust követő feltételes elágazás feltételénél nem jelzi, hogy milyen elemet kerestünk;
- a logikai műveletek egyéb alkalmazási területein (például az adatbázis-lekérdezéseknél) sem használhatjuk fel;
- formai szempontból sem helyesek az ilyen algoritmusok, hiszen látszólag nem létező tömbelemre hivatkoznak.

A rövidzár elkerülésének egyéb lehetőségeit a Visual Basic Script programozási összefoglalóban mutatjuk be.

A bináris keresést gyakran logaritmikus keresésnek nevezik. A diákok matematikai ismereteinek hiánya miatt a bináris elnevezést alkalmaztuk, amely utal az algoritmus működésére (felezi a tömböt).

### Házi feladat

Munkafüzet 4. feladat.

### Az ellenőrző kérdések és feladatok megoldása

1. Lásd a tankönyv 131. oldalát.

2. A módosított ciklus:

```
i = n
CIKLUS AMÍG i > 1 és tömb(i) nem rendelkezik a vizsgált tulajdonsággal
  i = i - 1
CIKLUS VÉGE
```

3. A módosított algoritmus (kiválasztás):

```
i = 1
CIKLUS AMÍG tömb(i) nem rendelkezik a vizsgált tulajdonsággal
  i = i + 1
CIKLUS VÉGE
Ki: i
```

4. Az átalakítást a tankönyv 123. oldalán lévő folyamatábra mintájára végezhetjük el.

5. Cseréljük fel egymással a < és > relációkat a ciklus feltételes elágazásainak feltételeiben.

6. Lásd a Munkafüzet 4. feladatát.

## A telefonkönyv rendezése

### Az óra célja

A beillesztéses rendezés algoritmusának megismerése és alkalmazása.

Rendezett adatbevitel.

### Fontos fogalmak, definíciók

Beillesztéses rendezés.

### Tevékenységek

- A beillesztéses rendezés algoritmusának megismerése és alkalmazása.
- A rendezési algoritmus hatékonyságának elemzése.

### Az óra menete

A tankönyv példája alapján írjuk fel és elemezzük egy új név felvételének módját a telefonkönyv memóriájába. Futtassuk a *bovit* programot.

Bővítsük az előző algoritmust úgy, hogy több elemmel is bővíthessük a listát (rendezett adatbevitel). Elemezzük és futtassuk a *nevsor* programot.

Alakítsuk át a rendezett adatbevitel algoritmusát úgy, hogy már meglévő elemeket rendezzen (beillesztéses rendezés). Elemezzük és futtassuk a *rendez* programot.

Vizsgáljuk meg a beillesztéses rendezés hatékonyságát (lásd a tankönyv 133. oldalán). Hívjuk fel a figyelmet a hatékony rendezési algoritmusok fontosságára.

Ha marad rá időnk, a tankönyv 133. oldalán lévő 4. és 5. feladat alapján vonjuk össze a beillesztéses rendezésben szereplő két ciklust.

### Megjegyzések

A rendezéssel elérkeztük a fejezet utolsó, egyben legnehezebb algoritmusához. A lehetséges algoritmusok közül a beillesztéses rendezést választottuk ki a rendezés bemutatására. A beillesztéses rendezés algoritmusa:

- szemléletes, hétköznapi tevékenységhez kapcsolódik (kézben tartott kártyalapok rendezése);
- felbontható egymásra épülő elemekre (egy elem beillesztése a már meglévő sorozatba, rendezett bevitel, meglévő elemek rendezése);
- az első lépés (egyetlen elem beillesztése) is felbontható egymástól elkülönülő tevékenységekre (az elem helyének megkeresése, az elem helyének felszabadítása, az elem beillesztése);
- gyakran előfordul más algoritmusokban (Shell-rendezés, keverési algoritmusok stb.).

Az általános iskolai tankönyvek általában a buborékos vagy az egyszerű cserés rendezést ismertetik, így célszerű volt egy másik rendezési algoritmust tárgyalni.

A lépésekre bontás miatt a beillesztéses rendezésnek nem a leghatékonyabb módszerét választottuk. Az algoritmusban szereplő két ciklust össze lehet vonni (lásd a tankönyv 133. oldalán lévő 4. és 5. feladatot).

### Házi feladat

Munkafüzet 3. és 4. feladat.

### Az ellenőrző kérdések és feladatok megoldása

1. A beolvasást tegyük egy feltételes elágazásba:

HA  $n < \text{maxindex}$  AKKOR

Be: újnév

ELÁGAZÁS VÉGE

2.  $n(n-1)/2$  összehasonlítást és  $2(n-1)+n(n-1)/2$  mozgatót.

3. A telefonszámokat együtt kell mozgítani a hozzájuk tartozó neveikkel.



4.-5. A módosított algoritmus (VBScriptben kihasználjuk, hogy létezik 0 indexű elem):

*Beillesztéses rendezés*

CIKLUS  $i = 2$ -től  $n$ -ig

$j = i - 1$

újelem = tömb( $i$ )

CIKLUS AMÍG  $j > 0$  és tömb( $j$ ) > újelem

tömb( $j+1$ ) = tömb( $j$ )

$j = j - 1$

CIKLUS VÉGE

tömb( $j+1$ ) = újelem

CIKLUS VÉGE

*Beillesztéses rendezés vége*

6. A folyamatábra a fenti algoritmusnak megfelelően felrajzolható.

7. Lásd például a Munkafüzet 6. feladatát.

## Rendszerezés

### Az óra célja

Az Algoritmusok és adatok fejezet összefoglalása. Gyakorlás, a dolgozat előkészítése.

### Az óra menete

A tankönyv 134. oldala alapján foglaljuk össze a legfontosabb tudnivalókat.

Oldjuk meg a 134. oldalon lévő 4. feladatot.

### Házi feladat

Válogatás a Munkafüzet feladataiból.

## Számonkérés

Dolgozat a Feladatlapok feladatsorai alapján.

# Könyvtárhasználat

## A könyvtárak története

<b>Tanítási cél</b>	A könyvtárak történetének rövid bemutatása.
<b>Fogalmak, definíciók</b>	Könyvtár, agyagtábla, papirusztekercs, nyomtatás, corvina, kódex, elektronikus könyvtár.
<b>Tevékenységek</b>	Tanári bemutató, egy híres könyvtár virtuális vagy valóságos meglátogatása.
<b>Megjegyzések</b>	Az interneten elektronikus könyvtárakat kereshetünk fel.
<b>Házi feladat</b>	Beiratkozás egy könyvtárba. Munkafüzetből feladat.

### Ellenőrző kérdések

1. Mit tud az ókori könyvtárakról?	Tk.. 135. o.
2. Hol és milyen híres könyvtárak működtek Magyarországon az elmúlt évszázadokban?	Tk.. 135. o.
3. Mit tud Mátyás király budai könyvtáráról?	Tk.. 135-136. o.
4. Az internetet is használva derítse ki, hogy hol található ma Magyarországon corvinák (korvinák)! Mátyás díszes borítású könyvei.	Adatgyűjtés az Interneten.
5. Az internetet használva keressen magyar régi és híres könyvtárakat! Válasszon ki egyet, és gyűjtsön róla adatokat! Mit lehet megtudni a választott könyvtárról?	Adatgyűjtés az Interneten.

## A dokumentumok csoportosítása

<b>Tanítási cél</b>	A hétköznapiakból ismert dokumentumok áttekintése, csoportosítása.
<b>Fogalmak, definíciók</b>	Írásos, képi, audió, vizuális, audiovizuális és elektronikus dokumentumok. A könyv részei. E-könyv.
<b>Tevékenységek</b>	A különféle dokumentumtípusok gyakorlati megtekintése és használata. Elektronikus dokumentumok pl. multimédia anyagok, e-könyvek használata, jellemzőik.
<b>Megjegyzések</b>	Célszerű a legújabb elérhető technológiákkal foglalkozni.
<b>Házi feladat</b>	Keressünk valódi ekönyveket az interneten. Munkafüzetből.

### Ellenőrző kérdések

1. Milyen szempontok szerint csoportosítaná a dokumentumokat?	Az adathordozó típusa szerint. Digitális vagy analóg dokumentumok. Az érzékelés módja szerint.
2. Mondjon minél több példát a következő dokumentum-kategóriákra: írásos, képi, audiovizuális, elektronikus! Természetesen adathordozó típusokat kell megnevezni, nem konkrét dokumentumokat.	
3. Mit nevezünk multimédia dokumentumnak?	Egyszerre több médiát használó dokumentum (szöveget, képet, rajzot, mozgóképet, hangot, zenét), amelyek között van időben változó is (például zene is).
4. Milyen (szerkezeti) részekből áll a könyv?	Könyvborító, védőlap, előzéklap, címlap, könyvtést, tartalomjegyzék, tárgymutató, mellékletek.
5. Milyen tartalmi részekből áll a könyv?	Tk. 138. o.
6. Milyen előnyeit ismeri az online elektronikus dokumentumnak?	Tk. 138. o.

7. <i>Gyűjtsenek adatokat a településükön található egy-egy könyvtárban tárolt dokumentumok típusáról!</i>	Adatgyűjtés a könyvtárban.
8. <i>Nézzon utána, hogy az iskolai könyvtár milyen elektronikus dokumentumokat tárol! Felhasználhatók-e ezek a tanulásban? Milyen tantárgyakhoz?</i>	Adatgyűjtés a könyvtárban.

## A könyvtárak típusai és szolgáltatásai

<b>Tanítási cél</b>	Megismertetni a tanulókkal a könyvtárak fontosabb típusait és az általános könyvtári szolgáltatásokat.
<b>Fogalmak, definíciók</b>	Könyvtár, nemzeti, szakkönyvtár, közművelődési, iskolai, elektronikus könyvtár. Funkcionális terek: szabadpolc, olvasóterem, számítógépes terem, katalógusok helye, kölcsönzőhely, kézikönyvtár. Szolgáltatások: kölcsönzés, előjegyzés, olvasás, zenehallgatás, videonézés, másolatkészítés, tájékoztatás, rendezvények.
<b>Tevékenységek</b>	A könyvtárlátogatás során egyes szolgáltatások igénybe vétele, használata.
<b>Megjegyzések</b>	
<b>Házi feladat</b>	Munkafüzetből

### Ellenőrző kérdések

1. <i>Milyen könyvtártípusokat ismer és mi a jellemzőjük?</i>	Iskolai könyvtár, szakkönyvtár, közművelődési, hagyományos és elektronikus könyvtár, Tk. 139. o.
2. <i>Milyen részekből (funkcionális terekből) áll egy korszerű könyvtár?</i>	Szabadpolcos rész, olvasóterem, videotéka, fonotéka, számítógépes terem, katalógus rész, kölcsönzőhely. Tk. 139. o.
3. <i>Milyen szolgáltatásokat nyújt egy korszerű közkönyvtár?</i>	Tk. 140. o.
4. <i>Hogyan használhatjuk a könyvtárat?</i>	Tk. 140. o.
5. <i>Iratkozzon be egy közművelődési vagy egy szakkönyvtárba!</i>	
6. <i>Tájékozódjon használatának feltételeiről, módjáról a gyakorlatban!</i>	Célszerű egy könyvtárlátogatást is szervezni.

## Könyvtári katalógusok és tájékoztató eszközök

<b>Tanítási cél</b>	Megismertetni a tanulókkal a tájékoztatás eszközeit és a különféle katalógusokat.
<b>Fogalmak, definíciók</b>	Tájékoztató könyvek, közhasznú források, ETO, Cutter-szám, szakrendi jel, raktári jelzet, könyvtári ABC, bibliográfiai leírás, katalóguscédula, rendszó, ISBN, ISSN. Címszókatalógus, szerzői katalógus, sorozati katalógus, szak-katalógus, tárgyszókatalógus, elektronikus katalógus.
<b>Tevékenységek</b>	Keresés katalógusokban. Katalóguscédulával kapcsolatos feladatok megoldása. Keresés a MOKKA-ban különböző feltételek (kiinduló információk) alapján.
<b>Megjegyzések</b>	
<b>Házi feladat</b>	Munkafüzetből.

### Ellenőrző kérdések

<i>1. Mi a különbség a lexikon és az enciklopédia között?</i>	A lexikon a címszavakhoz tartozó legfontosabb ismereteket sorolja fel. Az enciklopédia témakörök szerint dolgozza fel az ismereteket.
<i>2. Soroljon fel közhasznú információforrásokat!</i>	Telefonkönyv, menetrend, évkönyv, ki kicsoda, kronológia.
<i>3. Mi az Egyetemes Tizedes Osztályozás lényege és mire használják?</i>	Tk. 141. o.
<i>4. Mi a Cutter-szám (betűrendi jel)?</i>	Tk. 141. o.
<i>5. Miért készítenek a könyvekről bibliográfiai leírásokat?</i>	A könyv azonosításához és visszakereséséhez van szükség bibliográfiai leírásra. Tk. 141. o.
<i>6. Mi a katalógus és miért kell?</i>	A bibliográfiai leírások rendezett halmaza. Tk. 142. o.
<i>7. A katalógusoknak milyen fajtáit ismeri?</i>	Tk. 141. o.

# Tartalom

Bevezetés.....	1
Segédletek a tankönyvhöz.....	1
<b>Információs társadalom (Végh András)</b> .....	2
Az informatika fejlődéstörténete I.....	2
Az informatika fejlődéstörténete II.....	3
Az informatika fejlődéstörténete III.....	4
Az informatika fejlődéstörténete IV.....	5
Az információs társadalomról.....	6
Robotok.....	7
Jog és etika az informatikában.....	8
<b>Táblázatkezelés (Devecz Ferenc)</b> .....	9
Bevezetés.....	9
A táblázatkezelők funkciói, munkakörnyezete, a táblázatok elemei.....	10
Adattípusok, formátumok, adatbevitel.....	11
Műveletek, kifejezések és képletek, formázás igazítással.....	12
Képletek másolása, automatikus kitöltés, irányított beillesztés.....	13
Képletek áthelyezése, munkalap-műveletek, nevek használata.....	14
Statisztikai függvények, táblaformázás, grafikonszerkesztés.....	15
Dátum- és időértékek, adatok másolása és áthelyezése.....	16
Logikai műveletek és függvények, táblák rendezése.....	17
Nyomtatás, részösszegek, tömbfüggvények.....	18
Grafikonszerkesztés, tesztadatok generálása.....	19
Keresőfüggvények, az adatok szűrése.....	20
Függvények ábrázolása.....	21
Adatbázisok, külső kapcsolatok.....	22
Rendszerezés.....	23
<b>Adatbázis-kezelés (Makány György)</b> .....	25
Bevezetés.....	25
Az adatbázis-kezelés fogalmai, adatbázis modellek.....	26
Programok, táblák, adattípusok, kulcsok.....	26
Rendezés, indexelés, az adatok érvényessége.....	27
Úrlapok használata.....	27
Interaktív adatkezelés.....	28
Egyszerű lekérdezések.....	28
Lekérdezések szerkesztése.....	29
Jelentés készítése, nyomtatása.....	29
Összetett adatbázis tervezése.....	30
Kapcsolatok, hivatkozások.....	30
Többtáblás lekérdezések.....	31
Összetett adatbázisok úrlapjai.....	31
Az adatbázis-kezelés közös nyelve, az SQL.....	32
Különleges lekérdezések.....	32
<b>Algoritmusok és adatok (Juhász Tibor)</b> .....	33
Bevezetés.....	33
Az Algoritmusok fejezet a tantervben.....	33
Az Algoritmusok és adatok fejezet tanítása.....	34
A tankönyv Algoritmusok és adatok fejezete.....	34
A tankönyvi fejezet és az emelt szintű érettségi.....	35
A kiválasztott programozási nyelvek.....	35
A Free Pascal programozási nyelv.....	35
A Visual Basic Script programozási nyelv.....	36

Segédanyagok a tankönyvhöz .....	38
Segédanyagok az Interneten .....	38
AI-Hvárizmi öröksége .....	39
Algoritmusleíró eszközök .....	40
Algoritmusok és a számítógép .....	41
A programozási nyelvek elemei .....	42
Elemi adattípusok .....	44
Értékadás a változóknak .....	45
Feltételes elágazások programozása .....	46
Szövegdobozok, parancsgombok és társaik (kiegészítő anyag) .....	48
Telefonköltség .....	49
Összetett adattípusok .....	50
Számlálás ciklusok .....	52
Keresés a tömbben .....	53
A telefonkönyv rendezése .....	55
Rendszerezés .....	56
Számonkérés .....	56
<b>Könyvtárhasználat (Végh András).....</b>	<b>57</b>
A könyvtárak története .....	57
A dokumentumok csoportosítása.....	57
A könyvtárak típusai és szolgáltatásaik .....	58
Könyvtári katalógusok és tájékoztató eszközök .....	59