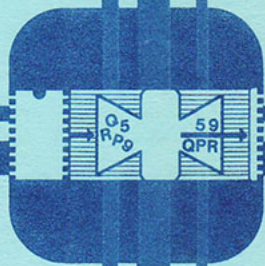


FARKASNÉ WÉBER ZSUZSA,
TÓBIÁS ÁGNES,
WÉBER GUSZTÁV

ADATKEZELÉS
SZÁMÍTÓGÉPPEL
C 16-RA
ÉS C PLUS/4-RE



ORSZÁGOS PEDAGÓGIAI INTÉZET



VORKER®

Adatkezelés számítógéppel C 16-ra és C Plus/4-re

(általános iskolai fakultációs tananyag)

Készült a G1—2/1 jelű állami megbízás keretében.

Készítették:

FARKASNÉ WÉBER ZSUZSA
TÓBIÁS ÁGNES
WÉBER GUSZTÁV

Főszerkesztő:
dr. Szűcs Barna

Szerkesztő:
Borbola István

A kiadvány szerzői:
Farkasné Wéber Zsuzsa, Tóbiás Ágnes, Wéber Gusztáv

Bírálták:
dr. Almási József
dr. Csépai János

ISBN 963 682 160 7 összkiadás
ISBN 963 682 164 x

Kiadja az Országos Pedagógiai Intézet Számítástechnikai Programirodája
Felelős kiadó: Szabolcsi Miklós

TARTALOMJEGYZÉK

1. CÉL ÉS FELADAT	11
2. A FOGLALKOZÁSOK ANYAGA	13
3. A FOGLALKOZÁSOK TÉMAKÖRÖNKÉNT	14
3.1. Általános ismeretek, alapfogalmak	14
3.2. Az adatfeldolgozás folyamata	16
3.3. Az adatfeldolgozó rendszerek fejlesztése	17
3.3.1. Helyzetfelmérés, rendszerjavaslatkészítés	17
3.3.2. Rendszertervkészítés	18
3.3.2.1. Az adatok specifikációja	18
3.3.2.2. Programspecifikáció	22
3.3.2.3. A programteszt anyaga	23
3.4. Az adatfeldolgozó rendszerek alkalmazásba vétele	23
4. A FOGALMAK HIERARCHIÁJA	25
5. TANMENETJAVASLAT	27
6. AJÁNLOTT IRODALOM	29
7. A PROGRAMOK LEÍRÁSA	30
7.1. Szubrutinok	30
7.1.1. Konstansok, függvények, segédrutinok	31
7.1.2. Képernyőkezelést segítő szubrutinok	31
7.1.3. Képernyő input/output szubrutinok	32
7.1.4. Menükezelés, választás	32
7.1.5. Keresés	32
7.2. Mintaprogram	32
7.3. Adatkezelő	34
7.3.1. Rekordtervezés	35
7.3.1.1. Képernyőszerkesztés	35
7.3.1.2. További műveletek	36
7.3.2. Adattár-kezelés	36
7.3.3. Feldolgozás, kérdések az adattárból	36
8. SZUBRUTINOK LISTÁJA	39

Melléklet: 1 db programkazetta.

A SOROZAT TAGJAI

Az Országos Pedagógiai Intézet Számítástechnikai Programirodájának gondozásában 1987/88-ban 12 számítástechnikai témakörhöz készült oktatási anyagok C 16-os és C Plus/4-es Commodore házi számítógépekre:

1. Algoritmusok, játékok:

A kiadvány tömören, de közérthetően, szemléletes példákon keresztül világít rá a probléma—algoritmus—program kapcsolatra, elvi és gyakorlati tanácsokat ad a jól felépített programok készítéséhez. A hozzá tartozó kazetta tizenkét, önmagában is alkalmazható, számítástechnikai és matematikai tanulságokkal teli programot tartalmaz. A programok jórészt listázhatók, így példát mutatnak egy színvonalas programozási stílus kialakítására, a gyakorlottabb felhasználóknak lehetőséget adnak arra, hogy más programok készítésénél is hasznosítsák az itt bemutatott programozási fogásokat. *Ajánlott:* ált. isk. 7., 8. osztályosainak és középiskolásoknak.

2. A számítógép és környezete:

Mindazoknak készült, akik szeretnék összekötni saját számítógépüket a külvilággal. 8 db különböző interfész és periféria tervezése kapcsán ismerteti meg az olvasót az interfész tervezés és illesztés fogásaival. Útmutató és 8 kapcsolási rajz, kész nyomtatott áramköri tervrajz. *Ajánlott:* ált. isk. és középiskolai tanároknak, számítástechnikai amatőröknek, 8. osztályos és középiskolai tanulóknak, tanárjelölt főiskolai és egyetemi hallgatóknak.

3. Programmodulok:

A felhasználó 55 db szubrutint, programmodult, illetve önálló felhasználói programot kap. Ezek egyrésze gépi kódú rutin, melyet az átlagos felhasználó nem tud megírni, de beépítheti munkájába. A felhasználói programok ötletet adhatnak hasonló programok megírásához. Az egyes rutinok, programok alkalmazási lehetőségeinek, működésének leírását útmutató tartalmazza. *Ajánlott:* ált. iskolai tanároknak, tanulóknak, kezdő amatőr programozóknak.

4. Adatkezelés számítógéppel C—16-ra és C Plus/4-re:

A témakör feldolgozását segítő útmutató az adatkezelés, adatfeldolgozás alapfogalmaival és folyamatával foglalkozik. Megismertet az adatfeldolgozó rendszerek fejlesztésével és a rendszerek alkalmazásbavételével. A feldolgozást 5 program és 2 minta-adatállomány segíti. *Ajánlott:* ált. és középiskolai tanároknak, fakultációs tananyagként, továbbképzési anyagként.

5. Szövegfeldolgozás számítógéppel:

A témakör feldolgozásához a felhasználó megkapja a Nemzetközi ABC szövegszerkesztő programot, amely magyar, cirill, német, lengyel, görög karakterkészletet és különleges matematikai jeleket (gyökjel, integráljel, alsó és felső index stb.) is tartalmaz. A szövegek és különleges karakterek nyomtathatók — akár fűzött kézírást imitálva is. A kézírásos nyomtatás főleg alsó tagozatban számíthat sikerre. A szövegszerkesztő minden olyan szolgáltatást tud, amely a tudományos munkához, vagy a tanári munkához kell. A feldolgozást útmutató és bemutató programok is segítik. *Ajánlott:* tanároknak, kutatóknak, íróknak, adminisztrátoroknak, illetve fakultációk anyagaként.

6. Számítógépes grafika C 16-on és C Plus/4-en:

Az útmutató és a mintaprogramok a felhasználói kézikönyvből meg nem tanulható különleges grafikus lehetőségekkel ismertetik meg az olvasót. A többszínű háttérszín üzemmód mellett a bittérképes színes grafikák elkészítésének fortélyait is megtanulhatja a felhasználó. Az útmutató és a mintaprogramok áttanulmányozása után az olvasó saját maga is képes lesz a játékprogramokból ismert színes és mozgó figurák „szellemecskék” programozására. *Ajánlott:* ált. isk. 7., 8. osztályosainak, középiskolásoknak és a grafika programozását most kezdő programozóknak.

7. Hangkeltés C 16-os (és C Plus/4-es) számítógéppel:

Az anyaghoz az útmutató mellett bemutató programok is tartoznak. Ezek áttanulmányozása alkalmassá teszi a felhasználót egy és többszólamú dallamok programozására, az interrupt alatti programozási lehetőségek kihasználására. A programok és dallamok az ált. iskolai ének-zene oktatáshoz igazodnak, így felhasználhatók számítógépes motivációra. *Ajánlott:* ált. isk. 7., 8. osztályosoknak, középiskolásoknak, valamint ált. iskolai énektanároknak, kezdő programozóknak.

8. Számítógépes fogások, trükkök C 16-ra és C Plus/4-re:

Az útmutató és a hozzá tartozó 16 program azok számára mutatja be a C 16-on és a C Plus/4-en alkalmazható fogásokat, akik a BASIC programozás alapjain túljutottak. A közölt rendszerváltozók, memóriacímek és gépi kódú programok megismertetik az olvasót az alkalmazások különleges lehetőségeivel. *Ajánlott:* kezdő programozóknak, általános és középiskolásoknak, tanároknak.

9. Számítógép és videó:

Az anyaghoz útmutató, videófilm és bemutató programok tartoznak. A videófilm a hozzá tartozó számítógépes programokkal egy megvalósítható rendszert és egy példát mutat be a képmagnó és a házi számítógép összekapcsolására, az oktatásban történő alkalmazására. Az anyag az elveken túl egy interfész elkészítésével is megismerteti az olvasót. Elsősorban tanároknak, oktatástechnikai szakembereknek ajánlott!

10. Számítógéppel vezérelt mérések:

Az útmutató megismerteti az olvasót a legfontosabb mérési fogalmakkal, a mérések tervezésének, elvégzésének, értékelésének elméleti és műszaki-, mérési alapjaival. A mérőberendezések és mérési összeállítások jelátvivő funkcionális egységként a Tudományszervezési és Informatikai Intézetnél kapható TechnoMIR moduláris interfészrendszer berendezéseit mutatja be és használja fel az anyag. A témakörhöz tartozó 21 db professzionális program egyrészt az oktatómunkában és a bemutató kísérletekben alkalmazható számítógép-vezérlésű intelligens műszert szimulál (pl. tároló oszcilloszkópot, 8 csatornás logikai analizátort, digitális multimétert stb.), másrészt az iskolai kísérletekhez szükséges mérési összeállításokat vezérli. *Ajánlott:* általános és középiskolai tanároknak, speciális szakképző intézeteknek, pedagógusképző intézeteknek bemutatói és oktatási célra, pedagógiai intézeteknek a továbbképzéshez.

11. Számítógépes irányítástechnika:

vez anyaghoz útmutató, az útmutatóban ismertetett berendezésekhez 11 db minta-Aogram tartozik. Ezek áttanulmányozása megismerteti az olvasót a számítógépes prérlekés és szabályozások alapfogalmaival, a mechatronikus modelleket működtető berendezések (kapcsolómodulok, interfészek) elkészítésével. A mintaprogramok elemzése példát mutat az irányítási algoritmusok programozására. Az irányító rendszerek jelátvivő funkcionális egységként a Tudományszervezési és Informatikai Intézetnél kapható TechnoMIR interfészrendszer elemeit, érzékelő-, erősítő berendezéseknek és beavatkozó szerveknek, berendezéseknek pedig saját készítésű, a felhasználó által is reprodukálható eszközöket használ az anyag. Ezek leírását és elkészítési módját is tartalmazza. *Ajánlott:* az általános és középiskolai technika tárgy oktatásához, fakultációkhoz, szakköri feldolgozáshoz.

12. Számítógépek összekapcsolása, helyi oktatóhálózatok:

Az anyag egy hazánkban eddig nyomtatásban még meg nem jelent területtel, a Commodore házi számítógépek (és a VIDEOTON TV COMPUTER) összekapcsolásának gyakorlati megvalósításával és e számítógépekből összeállított helyi oktatóhálózatok programozásával foglalkozik. Megismerteti a felhasználóval a számítógé-

gépek közötti kapcsolat hardver és szoftver eszközeit, a kapcsolatteremtés és az adatátvitel alapfogalmait. Megépített és reprodukálható kapcsolásokkal és kész felhasználói programokkal segíti, hogy az anyagot feldolgozó olvasó saját maga is összekapcsolhasson számítógépeket. A megszerzett ismeretek segítségével a minimális elektronikai ismerettel és készüléképítési gyakorlattal rendelkező felhasználó is meg tud valósítani számítógépek közötti adatátvitelt. Az útmutató második része az oktatásban bevált és sikerrel alkalmazott, a szegedi VORKER Kiszövetkezet által kifejlesztett TC—NET oktatóhálózatokkal, azok programozási fogásaival foglalkozik. Az anyag 6 db kapcsolást 1 db nyomtatott áramköri rajzot és 12 db professzionális felhasználói programot tartalmaz. *Ajánlott:* általános és középiskolai tanároknak, 3., 4. osztályos középiskolai tanulóknak, számítástechnikai amatőröknek, középfokú szakképző intézeteknek, tanárképző intézeteknek, pedagógiai intézeteknek, továbbképző intézeteknek stb.

Valamennyi itt felsorolt anyag (útmutató és program) valamint a TC—NET megrendelhető a szegedi VORKER Kiszövetkezettől. Cím:

VORKER Kiszövetkezet
6724 SZEGED, Zoltán u. 12.
Telefon: H 06-62-26-144.
Telex: 82-688

A VORKER Kiszövetkezet vállalja — megrendelés alapján — a felsorolt témakörökben bemutatók, alap- és továbbképzések tartását, szervezését is.

ELŐSZÓ

Az olvasó a szegedi Tarjánvárosi IV. számú Általános Iskolában fejlesztett 12 pedagógiai program (tanári kézikönyv, szoftver és/vagy beültetési rajzok) egyikét tartja a kezében. A munkában több mint hatvanan vettek részt: általános és középiskolai tanárok, főiskolai, egyetemi oktatók, tudományos intézetek munkatársai.

Az Országos Pedagógiai Intézet Számítástechnikai Programirodája és a szegedi iskola közös vállalkozása összefügg a számítástechnikai alpműveltség körülhatárolására irányuló erőfeszítésekkel. Kétségtelen, hogy a számítástechnikai-informatikai alapismeretek és az alpműveltség kapcsolatának feltárása elsősorban teoretikus jellegű munkát igényel. Az 1970-es években induló tartalmi korszerűsítés tapasztalatai azonban arra hívják fel a figyelmünket, hogy az iskolák nélkül, az iskolai hagyományok és a pedagógia öntörvényeinek mellőzésével nem hozhatók létre olyan dokumentációk, amelyeket a gyakorlat magáénak érez, s amelyekkel a pedagógusok többsége közösséget vállal.

Az alpműveltség témakörében már eddig is több tanulmány látott napvilágot. Viszonylag kevesebb szó esett azonban arról, hogy miként lehet pedagógiailag szervezett ismeretrendszerre formálni az itthon is és külföldön is már megfogalmazott elképzeléseket (amelyek között meglepően szoros összefüggés lehetséges fel).

Az elképzelésekről szóló következő ismertetés nem tekinthető állásfoglalásnak abban a kérdésben, hogy mi a jobb: ha a számítástechnikai-informatikai alapismeretekhez a diákok az *óratervben*, különféle tantárgyakban jutnak hozzá; *szakköri vagy fakultációs* programok keretében, esetleg *külön tantárgyként* biztosítják ezt számukra.

A fejlesztők munkáját a következő feltételezések irányították:

1. A számítástechnikai-informatikai ismeretek és az ez iránti érdeklődés nehezen tagolható az eddig megszokott módon, vagyis életkor és iskolatípus szerint. Az elkészült anyagoknak tehát egyaránt szolgálniuk kell az általános iskolákat és a középfokú iskolákat.
2. Az iskolák fogadókészsége eltérő (pl. más-más a felszerelés, a tanárok felkészültsége, a szoftverellátottság stb.). Ez aligha teszi lehetővé az eddig megszokott, a mindenki számára kötelező tantervi előírásokat, sémákat. Az igényekhez tehát csak a többféle lehetőség egyidejű bemutatásával kerülhetünk közelebb.
3. A 12 pedagógiai program és ezek járulékei alapul szolgálhatnak egy majdan országos érvényű informatika tantárgy kidolgozásához. A témák ilyen rendszere a már ma megvalósítható iskolai tevékenységeket tükrözi. Az informatika több

más témájáról az iskolák jórésze számára ma még csupán leírás adható (pl. országos, nemzetközi hálózatok). A későbbiekben elképzelhető, hogy az iskolákban is lehetőség lesz az informatika teljes skálájának bemutatására.

4. A 12 rész között biztosíthatók „átjárások”: a részek egymáshoz illeszthetők és kombinálhatók egymással.
5. A pedagógiai programokban megfogalmazott elképzelések érvényesítése érdekében az elinduláshoz szükséges a megfelelő segédletek elkészítése.
6. A pedagógiai programokhoz készült járulékok (beültetési rajzok, szoftverkészlet) egyik részének tantárgyi alkalmazásokra kell épülnie (pl. matematika, mérések), másik részük viszont az informatikának a mindennapi életben megjelenő gyakorlatát tükrözze (pl. adatkezelés szövegszerkesztés).

Reméljük, hogy munkánk segíteni fogja a pedagógiai gyakorlatot és ezáltal hozzájárul e téma elméleti kérdéseinek tisztázásához.

Örömmel és tisztelettel fogadjuk munkánk olvasójának, használójának észrevételeit.

Dr. Szűcs Barna

1. CÉL ÉS FELADAT

A számítógépes adatkezelés témakörében tartandó foglalkozások célja, hogy az érdeklődő tanulókkal megismertesse az adatfeldolgozás alkalmazási lehetőségeit és — alapfokon — a megvalósítás technikáját annak érdekében, hogy a fakultáció elvégzése után a komolyabb adatfeldolgozó rendszer használata, illetve annak megtanulása, működési logikájának megértése ne okozzon gondot. Egyúttal adjon segítséget az adatfeldolgozással kapcsolatos pályák felé irányuló tanulóknak a pályaválasztáshoz.

Az anyag terjedelme nem teszi lehetővé a tankönyvszerű feldolgozást, nem tanári kézikönyv, csupán áttekintést ad az adatkezelés, adatfeldolgozás témakörének tanításánál szükséges fogalmakról. A számítógépes adatkezelésben nem jártas pedagógus nem nélkülözheti az ajánlott szakirodalom tanulmányozását. Az anyag mellékleteként megadott tanmenetjavaslat és fogalmi hierarchia azt a szemléletet tükrözi, hogy egy kétéves oktatási időszak esetén a bonyolultabb fogalmak első évi tartalmi körvonalazása után azok a második évben ismét előtérbe kerülnek, és pontosabb megfogalmazásukra is akkor kerül sor. A megfogalmazott tanmenetjavaslat egy olyan ajánlás, mely az adott szakkör vagy fakultáció tartalmától, céljától függően szűkíthető, illetve bővíthető.

Az általános iskolákban általában a Commodore 16-os és a Commodore plus/4 gépek és kazettás egységek találhatók. Ezen a technikai bázison komoly adatfeldolgozó rendszer nem valósítható meg, lényegében csak adatkezelésről beszélhetünk. Törekvésünk az volt, hogy az adatkezelés, adatfeldolgozás során előforduló alapvető fogalmakat és adatszerkezeteket bemutassuk Commodore 16 és Commodore plus/4 gépeken. Az anyag oktatása során feltétlenül szükséges hangsúlyozni, hogy az alapul vett számítógép a valódi értelemben vett adatfeldolgozás megvalósítását nem teszi lehetővé, sőt bizonyos — alapvető — adatszerkezetek (nem szekvenciális adatszerkezetek) megvalósítása hajlékony lemezes háttértár nélkül nem lehetséges (ezek lényegét legfeljebb tárbeli tömbök kezelésével modellezhetjük).

A foglalkozások anyagának elsajátítása egy konkrét iskolai feladat lépésenkénti megoldásával történik. A jobb megértés érdekében választottunk ilyen feladatot, ami azonban nem zárja ki, hogy más, a tanulókhöz közelebb álló kérdéskört ne lehessen találni.

Az oktatást a mellékletben részletesebben ismertetett programok segítik.

A FELADAT

Az iskolában gyakran kell az osztályok, illetve a tanulók tanulmányi munkáját értékelni. Ez az értékelés történhet hetente, havonta, évente.

Képzeljünk el egy olyan adatfeldolgozó rendszert, ahol a tanulókat név-osztály szerint tartjuk nyilván és a tanulmányi eredmény értékelését havonta, tantárgyanként kell elvégezni. Az ehhez szükséges adatok : a tanulók neve, osztálya, osztályzata, a tantárgy neve, és az értékelés időszaka.

A témával kapcsolatos fogalmak és az adatfeldolgozó rendszerek fejlesztési technológiájának egységes értelmezése érdekében a megjelölt szakirodalmi források közül a KSH — magyar szabványon alapuló — módszertani segédleteit vettük alapul (1.13. szakirodalmi hivatkozás).

2. A FOGLALKOZÁSOK ANYAGA

2.1. Általános ismeretek, alapfogalmak:

- az adatkezelés, adatfeldolgozás feladata, a kézi adatfeldolgozó rendszerek működése,
- az adatfeldolgozás típusai,
- általános adatkezelési alapfogalmak,
- a számítástechnika alkalmazási lehetőségei az adatfeldolgozásban,
- információ,
- külső és belső adatábrázolás,
- egyszerűbb adatszerkezetek, tárolási módszerek,
- adat, törzsadat, változóadat, eredményadat,
- adatszerkezet, mező, adatmondat, adatállomány,
- adathordozók, lyukkártya, mágnesszalag, mágneslemez.

2.2. Az adatfeldolgozás folyamata:

- a kézi és a számítógépes adatfeldolgozás közti különbségek,
- adatbevitel, rögzítés, tárolás, feldolgozás, kivitel.

2.3. Az adatfeldolgozó rendszerek fejlesztése:

- adatfeldolgozási problémák felismerése,
- cél- és funkciómeghatározás,
- helyzetfelmérés, elemzés,
- logikai szerkezet felvázolása,
- az adatfeldolgozó rendszer megtervezése, részekre (modulokra) bontása,
- a program megtervezése,
- elkészítése,
- tesztelése,
- dokumentálása.

2.4. Az adatfeldolgozó rendszerek alkalmazásba vétele:

- előkészítés,
- alkalmazók betanítása,
- a rendszer karbantartása.

3. A FOGLALKOZÁSOK ANYAGA TÉMAKÖRÖNKÉNT

3.1. Általános ismeretek, alapfogalmak

A kereskedelemben, a háztartásban, az egészségügyben, a könyvkiadásban és az élet számos más területén gyakran találkozunk az adatfeldolgozás alkalmazásával. Az adatfeldolgozás kiterjedhet mérési adatok feldolgozására, nyilvántartások készítésére is.

A felmerült fogalmakat az A pontban ismertetett feladatokon keresztül magyarázzuk meg.

Ezek közül elsősorban az információról, az adatról és a köztük levő kapcsolatról kell beszélni.

Információn egy adott rendszer számára, annak működését befolyásoló új ismereteket nyújtó jelek tartalmi jelentését értjük.

Az *adat* az információnak olyan megjelenített formája, amely kisebb egységekre már nem osztható fel anélkül, hogy az általa hordozott információ ne torzulna. Például a tanulmányi átlag kiszámításakor adatnak tekintjük a tanuló nevét, érdemjegyeit stb.

Attól függően, hogy az adatok a feldolgozás során milyen gyakran változnak, különböző tulajdonságú adatokról beszélünk. A feldolgozási folyamatban állandó, vagy ritkán változó adatok a *törzsadatok*. Pl. a tanuló neve, a tantárgy megnevezése. Ha az adat tartalma gyakran változik, vagy minden alkalommal új értéket vesz fel, *változóadatról* beszélünk. Pl. egy tanuló osztályzatai. A törzsadatok és a változóadatok közötti műveletek során kapott adatot *eredményadatnak* nevezzük (pl. a tanulmányi átlag).

Az adatfeldolgozás során szükséges, hogy a feldolgozás alkalmával az előforduló elemekre egyértelműen hivatkozhatassunk. Azt az adatot, amellyel ez a hivatkozás történik, azonosítónak nevezzük. A tanuló *azonosítója* lehet például a neve vagy sorszám az osztálynaplóban.

A fogalom megértése érdekében vegyünk az adattípusokra néhány példát.

Numerikus adat

- az osztály létszáma (törzsadat),
- az osztály tanulmányi átlaga (eredményadat),
- osztályzatok (változóadat),

Alfabetikus adat

- az osztály egy tanulójának neve (törzsadat),
- egy tantárgy neve (törzsadat).

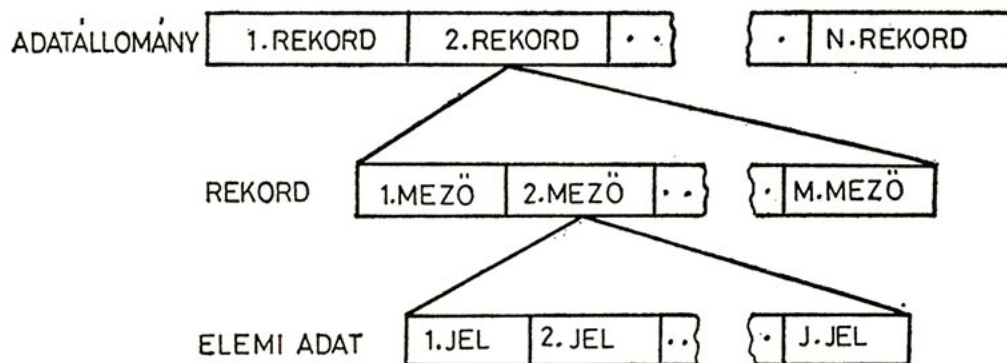
Alfanumerikus adat

— egy adott osztály neve, pl.: 3.c (törzsadat).

A feladat során az egymással kapcsolatban levő adatokból állítható össze az *adat-szerkezet*. Az azonos típusú adatokból álló rendezett adatsort *tömbnek* nevezzük, pl. a tanulók névsora.

A meghatározott adatok egy egységgé való összekapcsolása a *rekord* (adatmondat). A rekordnak azt a részét, amely az adatfeldolgozás szempontjából önálló egységnek tekinthető, *adatmezőnek* nevezzük. A rekordokat sorszámozzuk, a mezőknek nevet adunk. Pl. a tanuló neve, születési adatai, címe, melyik osztályba jár, osztályzatai stb.

Ha egy feldolgozás céljaira a feldolgozás adott szempontjai szerint valamennyi objektum rekordját összegyűjtjük, olyan adathalmazhoz jutunk, mely halmaznak egy-egy elemét egy-egy rekord alkotja. Az ilyen halmazokat *adatállománynak* nevezzük (angolul: file). Pl. egy osztály tanulóinak összes rekordja.



1. ábra

Az 1. táblázat a rekordok és mezőknek egy lehetséges ábrázolását mutatja a memóriában. (Lásd a 16. oldalon.)

Az állományokat *adathordozókon* rögzítjük, tároljuk. Az adathordozók olyan eszközök, melyeken a szükséges ideig tárolhatók az adatok. Az adathordozók lehetnek *címezhetők* és *nem címezhetők*. Címezhetők a mágneslemezek vagy más néven diszkek. Hasonló módon a tömbök kezelése a memóriában szintén egy címezhető megoldás. Nem címezhetők például a lyukkártyák, a mágnesszalagok és a kazetták.

Az ebben a témában tartott foglalkozásokon a már meglevő programozási ismeretek felhasználásával olyan feladatokat oldjanak meg a számítógéppel, mint pl. a tanulók névsorának, osztályzatainak bekérése képernyőről, ezen adatok különböző formájú tárolása és kiírása. Ennek kapcsán lehet bemutatni az előzőekben részletezett fogalmakat.

1. táblázat

	1. tömb tanulói név	2. tömb születési hely	. . .	n. tömb lakcím
1. rekord név	1. tömb 1. elem	2. tömb 1. elem		n. tömb 1. elem
2. rekord név	1. tömb 2. elem	2. tömb 2. elem		n. tömb 2. elem
.				
.				
.				
i. rekord név	1. tömb i. elem	2. tömb i. elem		n. tömb i. elem
⋮				
m. rekord név	1. tömb m. elem	2. tömb m. elem		n. tömb i. elem

3.2. Az adatfeldolgozás folyamata

A feldolgozást a számítógép központi egysége végzi. A feldolgozáshoz szükséges adatokat először be kell vinni a számítógépbe. Ezt nevezzük *bevitelnek* (input). A feldolgozás során az adatokat átmenetileg a gép memóriájában tároljuk. Az adatokat az előre meghatározott algoritmusok szerint *feldolgozzuk*. A feldolgozást segédváltozók, ún. *paraméterek* irányítják. Az eredményadatokat *kivisszük* (output) mágnesszalagra, diszkre, vagy sornyomatóval papírra nyomtatjuk ki. Ebből következően az adatfeldolgozás, adatkezelés lépései:

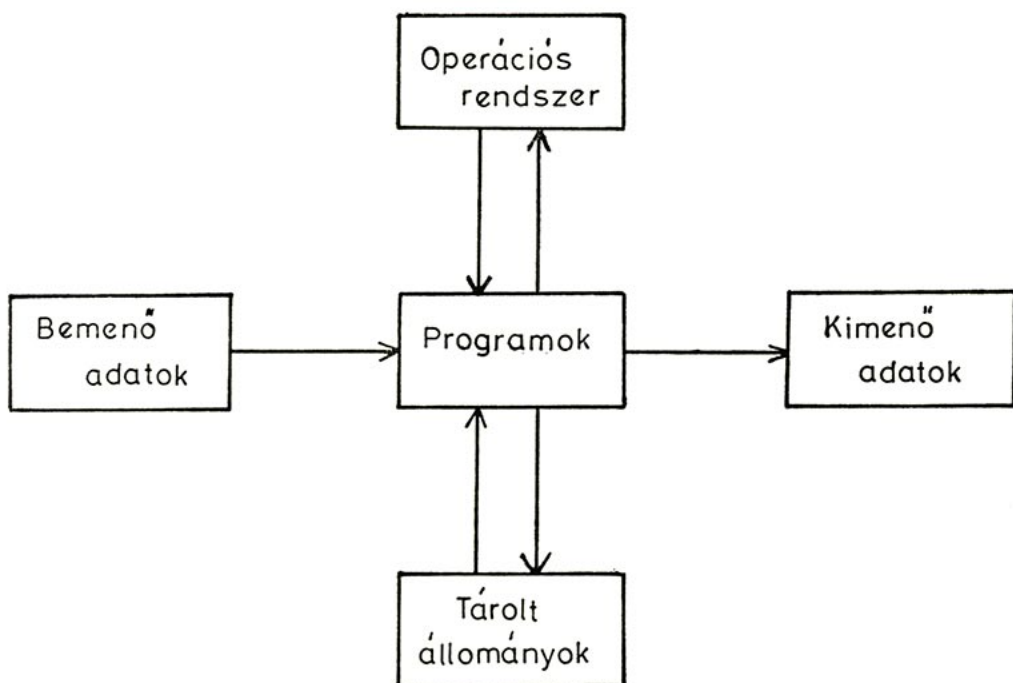
bevitel — feldolgozás — kivitel.

Például, ha egy osztály tanulmányi eredményére vagyunk kíváncsiak, a feldolgozás során a tanulók osztályzatait bevisszük a számítógépbe, mely a program által meghatározott módon kiszámítja a tanulónkénti átlagot és ezt utasításunkra a képernyőn megjeleníti, esetleg mágneses háttértárolóra kiviszi.

A foglalkozásokon megoldandó feladat: adatfeldolgozás az osztálynapló adataiból hagyományos módon és számítógéppel — egyezőségek és különbségek megvilágítása.

3.3. Az adatfeldolgozó rendszerek fejlesztése

Az adatfeldolgozó rendszert speciális információfeldolgozó rendszernek tekintjük. Az adatfeldolgozó rendszer nagy adattömeget használ. Olyan információfeldolgozó rendszer, amelyben programok, vezérlő programrendszer és adatállományok vannak.



2. ábra

A rendszer bemenetét adatok képezik, melyeket a programok feldolgoznak, ezzel kimenő adatokat és új adatállományokat állítva elő.

A rendszerfejlesztés főbb szakaszai a jelenleg érvényes szabványok szerint:

- helyzetfelmérés, rendszerjavaslat-készítés,
- rendszerterv- és programtervkészítés,
- programozás, tesztelés,
- bevezetés.

3.3.1. Helyzetfelmérés, rendszerjavaslat-készítés

A fejlesztési igény felmerülése után először a fejlesztési célt, a megvalósítandó rendszer által szolgáltatandó információk körét és a rendszer alapfunkcióit kell meghatározni.

Ehhez szükséges:

- Az adatok hol, milyen formában keletkeznek?
- Milyen és mekkora adatmennyiséget kell bevinni?
- Milyen és mekkora adatmennyiséget kell tárolni?
- Hogy kell feldolgozni a bevitt és tárolt adatokat?
- Milyen adatokat kell kiadni?
- Az eredményeket hol és mire használják?

Feladatunkhoz az adatokat a naplóból vesszük. Fel kell mérni, hogy az iskolában max. hány osztály van, egy osztályon belül mennyi a max. tanulói létszám, mennyi a tanított tantárgyak száma, hány osztályzatot kaphatnak a tanulók stb. Meg kell határozni, hogy az adatbevitel, illetve a feldolgozás milyen időszakonként történjen, mik a lekérdezés szempontjai, a bemenő adatokból hogyan, milyen formában keletkeznek az eredmények, az adatok közül melyeket tároljuk háttértárolón.

A tervezésnek ebben a fázisában el kell döntenünk, hogy milyen adatfeldolgozó típust alkalmazunk. A feldolgozás történhet az adatok táblázatokba való felvitelével hagyományos módon vagy számítógéppel. A hagyományos vagy más néven kézi adatfeldolgozásnál az adatok rögzítése, tárolása kartoték rendszerben történik, ami rendkívül körülményes és időigényes. Pl. Az osztályok közötti tanulmányi verseny esetén az adatok kigyűjtése és az adathalmaz táblázatban történő feldolgozása után következik csak az átlag kiszámítása.

Javaslatot kell adni még a szükséges eszközökre is (számítógép és perifériák). Előfordulhat, hogy a számítógép a feladat megfogalmazása előtt már megvan. Ekkor szükségmegoldásként a feladatot kell a géphez igazítani.

3.3.2. Rendszertervkészítés

A rendszerterv részei:

- adatspecifikáció (bemenő, kimenő és tárolt adatok),
- programspecifikáció,
- a programteszt anyaga.

A rendszertervezés tárgyát általában az adott feladat ellátására alkalmas elemek kiválasztása, az elemek közti kapcsolatok és a számítási algoritmusok meghatározása képezi. Egy rendszer megtervezésekor követelmény, hogy a megtervezett rendszer működését mindig valamilyen meghatározott célnak megfelelően lássa el.

3.3.2.1. Az adatok specifikációja

Meg kell határozni azoknak az adatoknak a tulajdonságait, amelyekkel dolgozni fogunk. Pl. melyek a változó és állandó adatok, milyentípusúak és melyek lesznek a kimenő, illetve a bemenő adatok.

Kimenő adatok megtervezésekor figyelembe kell venni, hogy milyen adatokat várunk a géptől, milyen formában és mely időpontban (üzemmód). Ezen túlmenően

meg kell határozni, hogy mi az, amit azonnal meg akarunk jeleníteni a képernyőn és mi az, amit papírra akarunk kiíratni. Mindkét esetben meg kell tervezni a megjelenítés formáját. Esetünkben kimenő adatnak tekintjük pl. a tanulónkénti és tantárgyankénti tanulmányi átlagot.

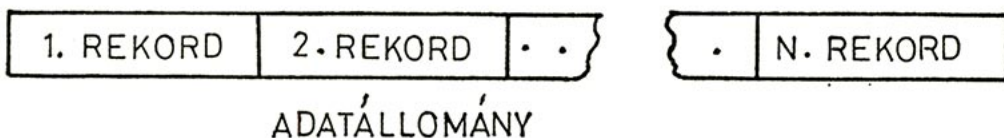
A bemenő adatok, melyekre a feldolgozás során szükségünk van, többnyire papíron, bizonylaton vagy esetünkben az osztályzatok az osztályozási naplóban vannak. Ezeket a forrásokat bizonylatoknak is nevezzük.

Azokat az adatokat, amelyekre a feldolgozás egy későbbi szakaszában is szükségünk van, tárolni kell. Feladatunkban tároljuk pl. a tanulók nevét, a tantárgyak megnevezését.

A tervezés következő fázisában — a tároláshoz és a további feldolgozáshoz — meg kell határozni az adatállomány szerkezetét. Ennek részfeladatai a következők: meg kell határozni a szükséges adatokat, azt, hogy hogyan tagolódik rekordokra ez az állomány, mi az a tulajdonság, amivel azonosítani tudunk egy objektumot és melyik az a mező — *rendezési kulcs* —, mely meghatározza egy rekord helyét az állományban, végül a rekord mezőinek nevét, hosszát, formátumát és tartalmát.

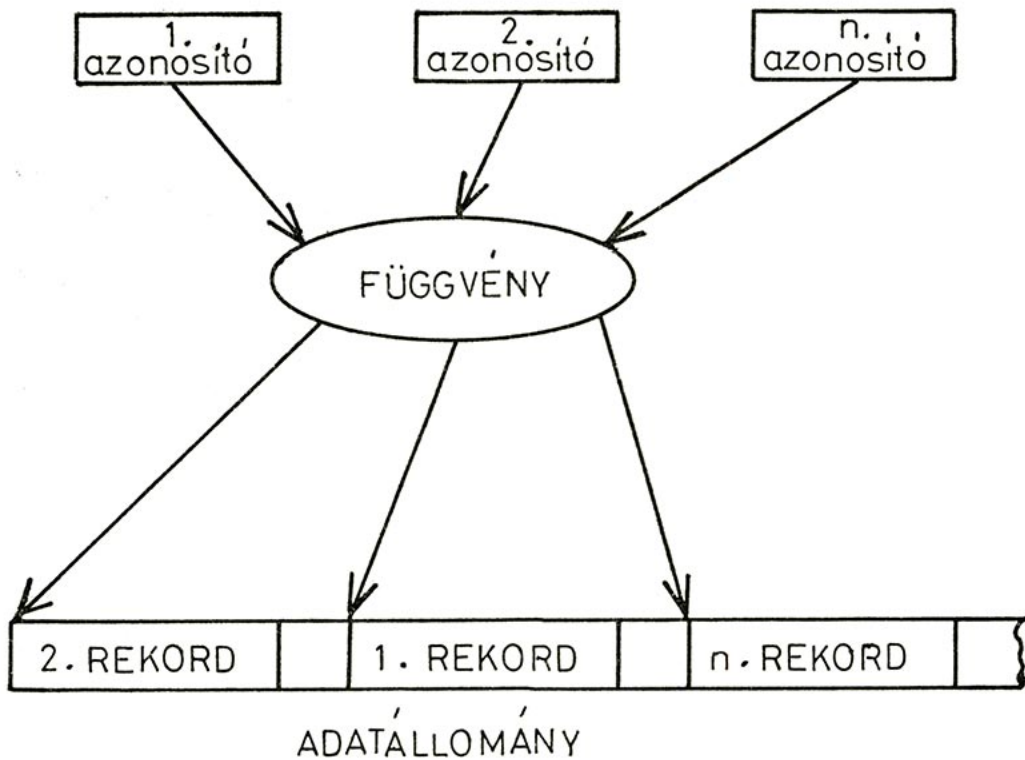
Ezzel azonban nem zárult le az adattárolás megszervezésével kapcsolatos tevékenység. Elő kell írni a *tárolási módot* is. A tárolási móddal határozzuk meg, hogy az egyes rekordok hogyan legyenek elrendezve az adatállományban. Többféle lehetőség van:

— Az egyes rekordok az állományban folytonosan a következő szabad helyre kerülnek. Ezt *soros vagy más néven szekvenciális* tárolásnak nevezzük. A szekvenciális tárolás esetén a rekordok egy megadott rendezési kulcs szerint növekvő vagy csökkenő sorrendben is következhetnek.



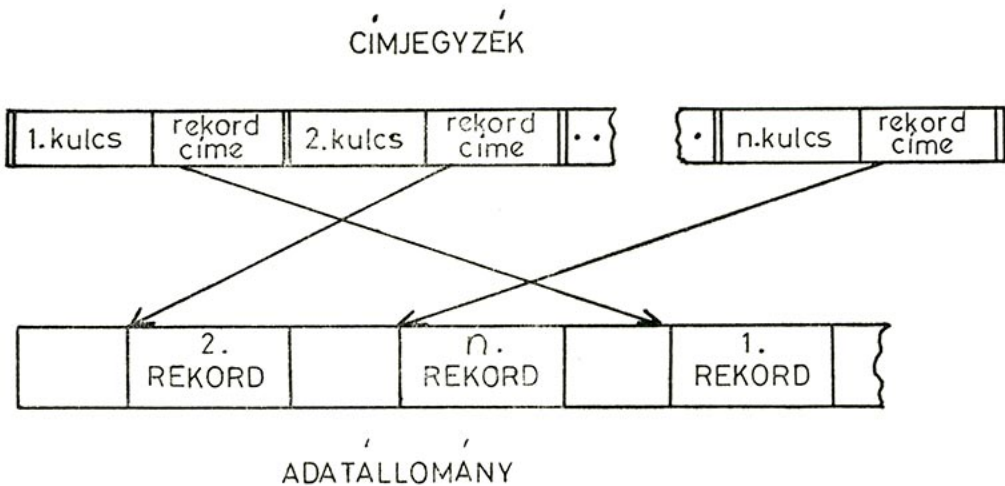
3. ábra

— Amennyiben a rekord azonosítója és a rekord állománybeli helye között egyértelmű összefüggés van, a rekord *direkt* módon is tárolható. Ekkor a rekord állománybeli helyét az azonosító határozza meg.



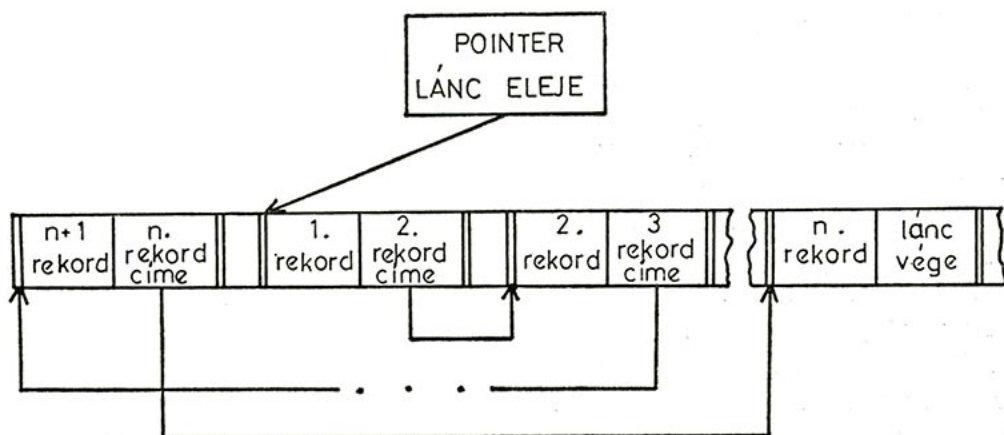
4. ábra

— Egy másik tárolási módnál az állományt egy ún. címjegyzékkel látjuk el, és ez határozza meg közvetlenül a rekord helyét az állományban. Ezt a módot nevezzük *index szekvenciális* tárolásnak. Itt egy olyan segéd táblázat készül, ahonnan leolvasható, hogy melyik rekord hol van a tároló eszközön és ezzel a keresés ideje lerövidül.



5. ábra

— További tárolási forma a *láncolt tárolás*. Láncolt tárolás esetén a rekord egy külön mezője (*pointer*) mutatja, hogy az őt logikailag követő rekord az állományban hol helyezkedik el. Ennél a tárolási módnál szükséges még a legelső rekord (lánc eleje) helyét is tárolni.



6. ábra

Azt az időt, ami az adat kérése és a megérkezése között telik el, *hozzáférési időnek* nevezzük.

Mágnesszalagra csak soros tárolási módot lehet alkalmazni, míg a mágneslemezes adathordozókon az ismertetett tárolási módok mindegyike alkalmazható.

A bemenő adatok leírása:

Itt kell meghatározni a bizonylatokat, amiről az adatokat bevisszük, az adatbevitel folyamatát, az adatok *külső* (ahogyan az adatot bevisszük, illetve az adat megjelenik) és *belső* (gépi) *ábrázolási formáit*, az adatok tartalmát, a kódrendszereket, az adatok ellenőrzéséhez az azok közötti logikai összefüggéseket.

A kimenő adatok leírása:

Itt kell a felhasználónak szánt kimenetek (képernyők, táblák) rajzait és jegyzékét megadni.

A tárolt adatok leírása:

Ez tartalmazza az adattárak közötti logikai és fizikai összefüggéseket, az adatállományok, rekordok leírását.

Az adatspecifikáció megadásánál célszerű előre megszerkesztett táblázatot használni:

Sorsz.	Megnevezés	Adatnév	Hossz	Típus	Megjegyzés
1.	napló sorsz.	NSOR	2	N	kulcs
2.	tanuló neve	NÉV	30	A	—

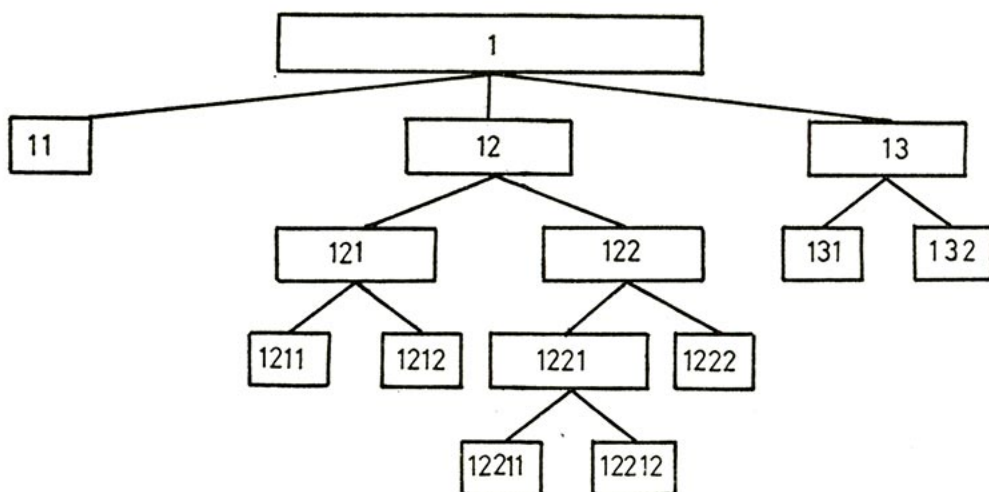
Amennyiben a tanulók adatait osztályonként külön állományba szervezzük, úgy a naplósorszám elégséges az azonosításhoz. Ha az iskola összes tanulóját egy állományban tartjuk nyilván, akkor többszörös kulcselőfordulás elkerülése miatt a naplósorszámot ki kell egészíteni pl. az osztályra utaló megkülönböztető kóddal is.

3.3.2.2. Programspecifikáció

A programozási feladatok megoldásához olyan módszert használunk, amely az elemzési, tervezési és programozási szakaszban is hasznosan alkalmazható. Az alapmódszer a *moduláris programozás*. A moduláris programozás lényege, hogy a feladatot részfeladatokra — modulokra — kell bontani, ezeket önállóan kell programozni, majd az egyes modulokból össze kell állítani a kész programot.

Mintafeladatunkból kiindulva a tervezés első lépéseként pontosan le kell írni a tanulmányi eredmények nyilvántartása feladat alapján azt, hogy a kiinduló adatokból az eredményadatok hogyan állíthatók elő.

A feladatot részfeladatokra, majd ezeket további részekre bontjuk mindaddig, míg el nem jutunk a résztevékenységekig. A részek — *modulok* — egymásra épülését a következő ábra szemlélteti:



7. ábra

A modulok a feladatmegoldásban betöltött szerepük szerint három csoportba sorolhatók: vezérlőmodul, adatmodul, eljárásmodul.

- a *vezérlőmodul* irányítja az egész program végrehajtását,
- az *adatmodul* a program adatainak kezelésére szolgál, a feladat érdemi megoldásához szükséges műveleteket nem tartalmaz, csupán az adatok beolvasását, ellenőrzését és kiírását végzi el,
- az *eljárásmodul* tényleges műveletet tartalmazó modul.

Az egyes modulok között kapcsolat van, ezek a kapcsolatok biztosítják, hogy a modulok végrehajtása helyes sorrendben történjék.

A programtervezés fázisában tehát le kell írni a programrészek egymásra épülését, a programok működésének pontos algoritmusát és az ellenőrzési szempontokat az adatspecifikációs részekre (pl. ha numerikus adatot vár a gép, ne fogadjon el mást, kulcsmező esetén vizsgálja meg, hogy az adott kulcs nem létezik-e már).

Ha a bemenő adatokból az eseményekkel egy időben eredményadatokat várunk, *valós idejű* (real-time) üzemelésre van szükség. Pl. olimpiai eredmények, iskolai sportverseny eredményei. Ha az adatokat először összegyűjtjük és az összegyűjtött adathalmazon együttesen hajtunk végre egy programot vagy programsorozatot, akkor *kötegelt* (batch) üzemmódról beszélünk. Pl. az év végi tanulmányi átlag egy-egy kiszámítása a havi tanulmányi átlagokból történik. Hátránya, hogy a feldolgozási időszakok között az eredmény nem áll rendelkezésre.

A tervezési szakasz végére, az adatleírástól kezdve az üzemmódig, minden lépést ki kell dolgozni és a rendszertervben kell rögzíteni.

3.3.2.3. A programteszt anyaga

A program hibátlan működésének bevizsgálásához egy olyan kisméretű próbaanyagra van szükség, melynek lefuttatásával megállapítható, hogy az a rendszerterv előző fejezeteiben leírt módon működik-e, elvégzi-e a szükséges ellenőrzéseket.

Le kell írni tehát

- a tesztelés folyamatát,
 - meg kell adni hibátlan és hibás bemenő adatokat,
 - a várt eredményeket,
- törekedve a rendszer valamennyi logikai ágának ellenőrzésére.

3.4. Az adatfeldolgozó rendszerek alkalmazásba vétele

A rendszer alkalmazásba vételének feltétele, hogy az alkalmazó részére megfelelő kezelési utasítást adjunk. Ebben — nem számítástechnikai fogalmakkal — le kell írni, hogy

- a rendszer milyen célra készült,
- milyen adatokból milyen eredményeket ad,
- milyen fő funkciói vannak,
- alkalmazásának milyen feltételei vannak,
- kezelésekor milyen tevékenységeket kell ellátni!

Képernyős rendszereknél ismert az ún. ismertető és az ún. segítő (HELP) funkció. Az ismertető a program egy önálló ága, amely képernyőn megadja a működtetéshez szükséges ismereteket. A segítő funkció esetén — ha a program a kezelőtől választ,

döntést vár — részletesebb magyarázat hívható a képernyőre a programfutás megszakítása nélkül.

Végül az alkalmazásba vétel további fontos feltétele, hogy az adatfeldolgozó rendszer alkalmazói részére egy bevezető ismertetést (oktatást) tartsunk. Ehhez szükséges egy mintaanyag összeállítása, melynek feldolgozásával bemutatjuk a rendszer működését és a kézi feldolgozástól való eltéréseket.

4. FOGALMAK HIERARCHIÁJA

Jelölések: A — alapfogalmak,
S — speciális fogalmak,
K — kapcsolt fogalmak.

Fogalom	7. oszt.	8. oszt.	közép.
	A S K	A S K	A S K
1. információ	A . .
2. adat	A
3. adatszerkezet	A
4. logikai adatszerkezet	. . K
5. fizikai adatszerkezet	. . K
6. adatnév	. . K
7. adattartalom S
8. adattípus	. S
9. numerikus adat	. . K
10. alfabetikus adat	. . K
11. alfanumerikus adat	. . K
12. törzsadat K	. . .
13. változóadat K	. . .
14. eredményadat K	. . .
15. adatmező	A
16. azonosító	. . .	A
17. kulcs	. . .	A
18. reláció K	. . .
19. tömb	. S
20. rekord	A
21. adatállomány	A
22. adathordozó	A
23. hagyományos adathordozók	. . K
24. számítástechnikai adathordozók	. . K
25. nemcímezhető adathordozó K	. . .
26. címezhető adathordozó K	. . .
27. bevitel	. S
28. feldolgozás	. S
29. paraméterek S
30. kivitel	. S
31. rendszer	A . .
32. alrendszer K
33. rendszerhierarchia S .

Fogalom	7. oszt.			8. oszt.			közép.		
	A	S	K	A	S	K	A	S	K
34. programmodulok	S
35. vezérlőmodul	S	.
36. adatmodul	S	.
37. eljárásmodul	S	.
38. adatfeldolgozó rendszer	A
39. rendezettség	A
40. rendezési kulcs	.	.	K
41. rendezési algoritmusok	S	.
42. tárolás	A
43. k.resés	A
44. keresési algoritmusok	.	S
45. tárolási mód	A
46. soros tárolás	.	S
47. szekvenciális tárolás	.	S
48. direkt tárolás	.	S
49. index szekvenciális tárolás	S
50. láncolt tárolás	S	.
51. pointer	K
52. adatelérés	.	.	.	A
53. hozzáférési idő	K	.	.	.
54. adatábrázolás	.	.	.	A
55. külső adatábrázolás	K	.	.	.
56. belső adatábrázolás	K	.	.	.
57. feldolgozási mód	.	.	.	A
58. valósidejű feldolgozás	K	.	.	.
59. kötegetelt feldolgozás	K	.	.	.

Feltételezett ismeretek:

- az adott szintnek megfelelő matematikai fogalmak,
- számítógépezési alapismeretek,
- BASIC nyelv alapfokon,
- ciklusszervezés,
- háttértár kezelési alapismeretek,
- szubrutinkezelés.

5. TANMENETJAVASLAT

(7. osztály)

1. A számítástechnikai alapfogalmak ismételése. BASIC utasítások, parancsok. Egy egyszerű program írása önállóan.
2. Az adat fogalmának bevezetése. Az adatfeldolgozás, mint folyamat értelmezése és vizsgálata. Az adatfeldolgozás típusai. Kézi adatfeldolgozó rendszerek. Feladat: hagyományos adatfeldolgozás.
3. A számítástechnika alkalmazási lehetőségei az adatfeldolgozásban. A gépi adatfeldolgozás. Általános adatfeldolgozási alapfogalmak. Adattípusok. Feladat: egyszerű programok készítése során különböző adattípusok alkalmazása.
4. Az adatszerkezet fogalma. Tömb. Adattárolás a programban. Feladat: adatbekérés képernyőről, tárolás, kiíratás.
5. Adatszerkezet: logikai, fizikai. Azonosító, adatmező, rekord, adatállomány fogalma.
6. Adathordozók: hagyományos és számítástechnikai. Tárolási módok: soros és direkt. Adattárolás kazettás egységen. A számítógépes adatfeldolgozás folyamata: bevitel, tárolás, feldolgozás, kivitel.
- 7—8. Az adatfeldolgozó rendszer fogalma. A rendszerfejlesztés főbb szakaszainak megbeszélése: helyzetfelmérés, rendszertervkészítés, programozás, tesztelés, bevezetés. (A továbbiakban konkrét feladatokon keresztül történik az ismeretek elsajátítása.)
9. Helyzetfelmérés, rendszerjavaslat-készítés. Rendszertervkészítés, a rendszerterv részei. Adatspecifikáció.
- 10—11. Programspecifikáció: a feladat részekre bontása. Az algoritmus meghatározása és folyamatábra-készítés.
- 12—13. A rendszerterv alapján program készítése önállóan.
 14. A programteszt elkészítése.
- 15—16. A rendszer bevezetése, a működés közben adódó hibák kijavítása.
 17. Üzemlátogatás.
 18. Az évi munka értékelése.

(8. osztály)

1. Ismétlés: általános adatfeldolgozási alapfogalmak, gépi adatfeldolgozás.
 2. Ismétlés: a rendszerterv készítésének lépései.
 - 3—4. Adatszerkezetek: logikai, fizikai. Azonosító, adatmező, rekord, adatállomány fogalmainak ismétlése és ezek létrehozása.
 5. Adatábrázolás: külső, belső. Adathordozók: címezhető, nem címezhető.
 6. Tárolási módok: soros, direkt (ismétlés). Indexszekvenciális tárolás. Hozzáférfési idő.
 7. Programszefikáció, programmodulok. Üzemmodok: valós idejű és köteget.
 - 8—15. Egy adatfeldolgozó rendszer fejlesztése az elmúlt, valamint ebben az évben megszerzett ismeretek felhasználásával.
 16. Üzemlátogatás.
 17. Ismétlés. Az adatfeldolgozással kapcsolatos ismeretek összefoglalása.
 18. Az évi munka értékelése.
- A foglalkozásokhoz további ajánlások találhatók a mintaprogram leírásánál.

6. AJÁNLOTT IRODALOM

1. Dr. Arató Péter—Dr. Stauder Ernő: Számítógépek és alkalmazásuk. Bp. Tankönyvkiadó, 1983.
2. Bánkfalvi Judit—Dettrich Árpád—Turny Zoltán: Számítógépek és adatfeldolgozó rendszerek programjainak dokumentációja. Bp. KSH, 1984.
3. Benkő Tiborné—Jávor András: Bevezetés a számítástechnika alkalmazásába (Különböző területek szakemberei részére). Bp. BME, 1978.
4. Bocsi Éva—Hunyady Lászlóné: Kommunikáció a számítógéppel. Bp. KSH, 1980.
5. D. Busse: Kulcs a számítógéphez (Gyakorlat, Hardver, Szoftver, Szervezés). Bp. Műszaki Könyvkiadó, 1984.
6. C—16, PLUS/4 programozói útmutató. Bp. Novotrade Rt., 1987.
7. Csépai János: A számítástechnika alapjai. Bp. Műszaki Könyvkiadó, 1985.
8. Demetrovics J.—J. Denev—R. Pavlov: A számítástudomány matematikai alapjai. Bp. Tankönyvkiadó, 1985.
9. Gyürki József: Modern számítógépes adatbázis kezelő software rendszerek és nyelvek. Bp. BME, 1982.
10. G. F. Hice—W. S. Turner—L. F. Cashwell: Számítógépes rendszerek fejlesztésének módszertana. Bp. Műszaki Könyvkiadó, 1983.
11. Dr. Kiss Imre: A számítógépes információrendszerek szervezésének alapjai. Bp. KGTMI, 1982.
12. Krajcsovits Márton: Elektronikus adatfeldolgozási rendszerek (Adatkezelés, programcsomagok, rendszertervezés). Bp. Műszaki Könyvkiadó, 1975.
13. Dr. Mészáros Tamásné—Morvay János—Weidl Lajos—Zentai Tamás: Információrendszerek számítógépes adattárainak tervezése és kezelése. Bp. KSH, 1975.
14. Mi micsoda magyarul a számítástechnikában? (Mikroszámítógépes értelmező szótár). Bp. Tömegkommunikációs Kutatóközpont, 1986.
15. C. N. Prague—J. E. Hammit: Adatbáziskezelés, programozás. dBASE III. Bp. Műszaki Könyvkiadó, 1987.
16. Rendszerfejlesztési dokumentáció. Bp. KSH, 1984.
17. Dr. Úry László: Commodore C—16, BASIC és felhasználói kézikönyv. Bp. LSI, 1985.
18. Wirth Niklaus: Algoritmusok + adatstruktúrák = programok. Bp. MK., 1982.

7. A PROGRAMOK LEÍRÁSA

A készlet egy kazettát is tartalmaz 5 programmal és 2 minta-adatállománnyal az alábbi sorrendben:

- SZUBRUTINOK (program),
- MINTAPROGRAM (program),
- NÉVSOR (mintaállomány a mintaprogramhoz),
- REKORDTERV (az adatkezelő 1. programja),
- KARBANTARTÓ (az adatkezelő 2. programja),
- FELDOLGOZÓ (az adatkezelő 3. programja),
- VERSENY (mintaállomány az adatkezelő 2. és 3. programjához).

Hardverszükséglet:

— COMMODORE 16 vagy plus/4 (ékezetes betűkkel).

Az első két program mindkét gépen, az utolsó három plus/4 gépen futtatható (ha a rekordot plus/4 gépen megterveztük, a **KARBANTARTÓ** és a **FELDOLGOZÓ** programok már C=16-os gépen is futtathatók).

— Kazettás háttértár.

A programok működését az ékezetes betűk hiánya nem befolyásolja, csupán a magyarázó szövegek ékezetes betűi helyett lesznek grafikus jelek. Ebben az esetben a BASIC nyelvű programokban ezeket a jeleket célszerű kicserélni ékezet nélküliekre.

7.1. Szubrutinok

A program az adatkezelést segítő 13 szubrutint tartalmaz a használatukhoz szükséges — a szubrutinokba beírt — kommentekkel együtt.

Mindegyik előtt megjegyzések adnak magyarázatot a szubrutin

- funkciójára,
- a hívásához szükséges paraméterekre,
- az esetleges eredményváltozókra és megadják
- a hívott szubrutincímeket, függvényeket és
- a használt munkaváltozókat.

Valamennyi szubrutin X, Y és Z betűvel kezdődő változókat használ, és az azonos funkciójú változók minden szubrutinban azonosak. (Pl. egy adott képernyőpozíció meghatározására mindig az X%, Y% változók szolgálnak.)

Bizonyos változók (pl. a REVERSE kiírást vezérlő XR, a karaktervillogatást bekapcsoló XF vagy az üzeneteket közvetítő X\$ változók) a szubrutinból való kilépés előtt alapértelmezésre állnak vissza.

A *szubrutinok célja*, hogy az oktatás során készülő programok írásakor alapvető funkciókra ne kelljen külön programrészeket írni, ezek *beépíthetők*. Felhasználhatók még a programszegmentálás és paraméterezés szemléltetésére is.

A szubrutinok az első három programsorban beállított fekete keretű háttérszín előtt, kisbetű/nagybetű üzemmódban működnek és általában kezelői kommunikációra a RETURN, SHIFT+RETURN és a funkció (F) billentyűket használják.

COMMODORE 16-os gép alkalmazása esetén — ha ezeket a szubrutinokat használni akarjuk — a szubrutinok elején található megjegyzéseket törölni kell azok nagy tárigénye miatt.

7.1.1. *Konstansok, függvények és segédrutinok*

Ezek a többi szubrutin által használt állandó értékek, a képernyő input ellenőrzéséhez szükséges konstansok, a rugalmas képernyő outputot segítő függvénydefiníciók.

A program elején — ezt a részt is mint szubrutint meghíva — az értékadások, definíciók megtörténnek.

Ez a szubrutin írja át a funkcióbillentyűk alapértelmezéseit is a többi szubrutin működéséhez szükséges értékekre.

7.1.2. *Képernyőkezelést segítő szubrutinok*

Az egész programfutást meghatározó egységes képernyőszerkezet, gép és kezelő közötti párbeszéd esztétikus megvalósítását segítő szubrutinok:

- kurzorbeállító,
- képernyőfej (fejléc) kezelő,
- ablakkezelő,
- üzenőmező,
- ablak-lábjegyzetkezelő.

7.1.3. Képernyő input/output szubrutinok

Ellenőrzött képernyő input és paraméterezett, illetve szerkesztett képernyőkiírást segítő szubrutinok:

- inputkezelő (karakteres és numerikus),
- karakteres output,
- numerikus output,
- kerekítő.

7.1.4. Menükezelés, választás

A menükezelést, valamint a gép és kezelője párbeszédének megvalósítását segítő szubrutinok:

- menükezelő,
- kérdező,
- eldöntendő kérdés,
- válasz-kiértékelő.

7.1.5. Keresés

A tömbön belül egy kulcs helyét kereső szubrutin.

7.2. Mintaprogram

Az előzőekben ismertetett SZUBRUTIN csomag használatát és egyszerű adatkezelési alapfunkciókat bemutató program.

Célja a fentiekén túl, hogy az oktatás során a tanulók bevonásával — a keretprogram átalakításával és bővítésével — lehessen további adatkezelési funkciókat egyszerűen megvalósítani.

A program bemutatja a

- menükezelést,
- az állandó képernyőfej és üzenőmező használatát,
- a kulcsok kezelését,
- a rekordbeszúrás, a rekordmódosítás és a rekordtörlés technikáját,
- az adatok kazettára mentését és kazettáról történő betöltésének technikáját.

A mintaprogram egy névsor felállítását teszi lehetővé. A mintaprogram mögött elhelyezett — *névsor* nevű — adatállománnyal lehet a program szolgáltatásait kipróbálni.

Az adatokat egy $tn\$(ts, 3)$ méretű füzértömbben (stringtömb) tároljuk, melyben az i -edik tanuló adatai:

- $tn\$(i, 0)$ — a tanuló neve
(20 karakter),
- $tn\$(i, 1)$ — a lakóhelye: város, község stb.
(20 karakter),
- $tn\$(i, 2)$ — utca neve
(20 karakter),
- $tn\$(i, 3)$ — házszám
(5 hosszú numerikus).

A kulcsokat egy $nk\%(ts)$ numerikus tömbben tároljuk, melyben $nk\%(i)$ az i -edik tanuló azonosítására szolgáló kulcs.

A ts paraméter a névsorba maximálisan felvehető, a ta változó pedig a névsorban pillanatnyilag nyilvántartott tanulók számát adja.

A program használ még egy $tx\$(3)$ méretű munkatömböt, ahol $tx\$(j)$ a feldolgozás alatt álló rekord j -edik mezőjét tartalmazza az állományba történő beépítést megelőzően.

A mintaprogram ajánlott továbbfejlesztési lehetőségei a 3. pontban javasolt problémák megoldásán túl:

— a képernyőfej-kezelő rutin átírásával a pillanatnyi rekordszám (ta) és a még névsorba felvehető tanulók számának ($ts-ta$) megjelenítése.

A főprogram érintett szubrutinjai:

- = új rekord
- = törlés

— a rekord kibővítése egy mezővel (pl. hányadik osztályba jár).

A főprogram szükséges módosítása:

- = $tn\$, tx\$$ tömbök méretének megváltoztatása,
- = tanulói adatok — képernyőszubrutin: ablakméret-megnövelés és mezőnévkiírás,
- = az összes olyan főprogramszubrutin, melyben az eddigi 4 mező (0—3) konstansként szerepel.

További fejlesztési lehetőség: a főprogram szubrutinjaiban a mezők számát változók segítségével megadni.

— A tanulók állomány ($tn\$$ tömb) névsorba rendezése.

Új szubrutint kell írni, mely a $tn\$$ tömböt a 0 -adik oszlopa szerint növekvőbe rendezi.

— A program kérdéságába egy új funkció illesztése, amely nem az összes rekordot, hanem csak megadott szempontnak megfelelő rekordot listázza a képernyőre. Pl. csak azokat, akik Szegeden laknak.

Szükséges módosítások:

= a főprogram kérdezés ágába egy almenút kell beiktatni,

= egy új szubrutin kell, amely a tn\$ tömb 1. indexű elemei közül csak azokat engedi a képernyőre, melyeknek tartalma megfelel az előbbi szempontnak.

A mintaprogramnak az előzőekben ajánlott továbbfejlesztései természetesen a lehetőségeknek csak egy részét mutatják be.

7.3. Adatkezelő

Az általános célú adatkezelő program feladata, egy — az alkalmazó által megfogalmazott — adattár (táblázat)

— létrehozása,

— karbantartása,

— az adattár különböző szempontok szerinti lekérdezése.

Alkalmas egy osztály tanulóinak, tanulmányi eredményének, egy kisebb könyvtár könyveinek vagy akár történelmi események nyilvántartására, tehát minden olyan nyilvántartás vezetésére, amely táblázatos formában megfogalmazható.

A létrehozott adattárat úgy kell elképzelni, mint egy táblázatot, melynek minden sora egy nyilvántartásba vett elemet (objektumot) ír le, oszlopai tehát az elemek jellemzői.

A program használatával jól demonstrálhatók az adatfeldolgozás lépései, egy egyszerű adatfeldolgozó program működése.

A kazettán a program mögött elhelyezkedő *verseny* nevű adatállománnyal lehet a program szolgáltatásait kipróbálni.

A program szolgáltatásai:

A funkciókat három önálló program látja el, melyeket külön-külön kell betölteni:

— a képernyő és a rekord mezőinek megtervezése

(REKORDTERV),

— adattár létrehozás és karbantartás

(KARBANTARTÓ),

— feldolgozás, keresés, válogatás az adattárból

(FELDOLGOZÓ).

A programok működését vezérlő billentyűk általában:

— az adott funkció befejezése: **SHIFT + RETURN**,

— rekordok között előre-, visszamozgás a kurzor „le”, illetve „fel” billentyűkkel,

— egy rekord mezői közötti mozgás a kurzor „jobbra”, illetve „balra” billentyűkkel,

— egy mező értékének módosítása **DEL** billentyűvel.

7.3.1. Rekordtervezés

A feladat megfogalmazása után, ekkor kell a *képernyőt megszerkeszteni* és a program számára a nyilvántartásba kerülő elemek *rekordjait* — adatmezőit — *leírni*.

A lehetséges mezőtípusok:

- karakteres (alfanumerikus és egyéb jelek),
 - numerikus,
 - dátum
- típusú mezők, melyek közül egy kötelezően a kulcs.

7.3.1.1. Képernyőszerkesztés

A program *saját kurzora* a „*”, melyet a kurzormozgató billentyűvel mozgathatunk a kívánt képernyő-pozícióra.

Bármely képernyő-pozícióban az alábbi műveletek lehetségesek:

- rekordképrajzolás: a „*” **billentyű** lenyomása után mintegy megrajzolhatjuk az input/output képernyőt. Ekkor a képernyőre magyarázó szövegeket, táblázatot rajzolhatunk a később meghatározásra kerülő adatmezőkhöz. Minden rekord később ebben a képernyőrajzban jelenik meg.
- Az „A” **billentyű** lenyomása után karakteres mezőt határozhatunk meg: az „A” betűt annyiszor kell lenyomni, amennyi karakter hosszúra tervezzük a mezőt.
- Az „E” **betűvel** numerikus egész típusú mezőt lehet megadni, hossza 2—6 karakter (E betű) lehet.
- A „V” **betű** valós mezőt jelent, hossza tartalmazza az előjel, a tizedespont és a két tizedes jegy helyét, így a mezőhossz 4—10 karakterrel definiálható.
- A „D” **betű** egy dátum típusú mezőt jelöl ki, hossza kötelezően 10 karakter (4 az évnek, 2—2 a hónapnak és a napnak, közöttük két pont lesz).

Egy mező végét a RETURN billentyű lenyomásával kell jelezni. Ekkor a program visszaadja a kurzort (*) és utána vihetjük a kurzort a következő mező kezdetére. A mezők között kötelező egy üres helyet hagyni.

Mezőt javítani vagy törölni a DEL billentyűvel lehet úgy, hogy előzőleg a kurzorral a mezőre állunk. Ha a kurzort egy mező belsejébe visszük, az automatikusan eltűnik, és az adott mező típusának megfelelő karakterekkel lehet a mező hosszát módosítani vagy törölni. *A képernyő szerkesztését a SHIFT+RETURN billentyű együttes lenyomásával* fejezzük be.

Ekkor a kijelölt adatmezők *egy alapértelmezést kapnak:*

- numerikus mezők: 0,
- dátummező: a program elején bekért dátum,
- karakteres mező: annyi pont, amilyen hosszúra jelöltük ki az adott mezőt.

7.3.1.2. További műveletek

A rekordkép megszerkesztése után a program automatikusan az alábbi tevékenységek elvégzését kéri:

- *Kulcsmező-kijelölés:* a **REVERSE** módban megjelenő mező, **SHIFT+RETURN** lenyomása esetén, kulcsmező lesz. Ha nem ez a kulcsmező, akkor a **RETURN** billentyűt kell lenyomni. Ezt a funkciót nem lehet addig elhagyni, amíg egy kulcsmezőt nem jelöltünk ki.
- A továbbiakban *a mezők lehetséges legkisebb, legnagyobb értékeit és az alapértelmezés* módosítását kéri a program úgy, hogy ezek elvi alapértelmezéseként a mezőben ábrázolható legkisebb és legnagyobb értékeket kínálja. Ezeket a funkciókat is a **SHIFT+RETURN** lenyomásával lehet elhagyni.

A rekordtervezés végén — a másik két program részére — a rekordszerkezetet kazettára kell menteni. *Célszerű az adatállományok tárolására külön kazettát használni.*

7.3.2. Adattár-karbantartás

A program indítás után adatállomány nevet kér. Ekkor kell az adatkazettát betenni, amelyről a program a rekordszerkezetet, illetve adatállományt beolvassa.

A képernyőn a 7.3.1. pont szerint megtervezett képernyőkép jelenik meg.

Új rekord bevitele esetén az adatmezőkben az alapértelmezések jelennek meg. Módosításnál megvizsgálja a program, hogy a mező értéke a lehetséges legkisebb és legnagyobb értékek közé esik-e.

Karbantartási műveleteket vezérlő billentyűk:

- f1: rekordmódosítás,
- f2: új rekord előállítása,
- f3: rekordkeresés,
- f4: rekordtörlés,
- f5: adatállomány-tárolás,
- f6: adatállomány-beolvasás,
- f7: program vége,
- **HELP:** tájékoztató.

7.3.3. Feldolgozás, kérdések az adattárból

Az adattárból képernyőre lehet kérni:

- egy rekordot valamennyi mezőjével (a táblázat egy sorát),
- sorban a rekordokat valamely mező értéke szerint növekvő sorrendben,

- valamely mező értéke szerint kiválogatott rekordokat (sorokat),
- egy állományhoz egy másik — azonos rekordszerkezetű — állományt lehet hozzáfésülni akkor, ha kulcsaik különbözök.

Karbantartási műveleteket vezérlő billentyűk:

- f1: hozzáfésülés,
- f2: keresés mező szerint,
- f3: rendezés,
- f4: rekordtörlés,
- f5: adatállomány-tárolás,
- f6: adatállomány-beolvasás,
- f7: vége,
- **HELP:** tájékoztató.

Úgy a karbantartó, mint a feldolgozó programnál az elvégezni kívánt tevékenységet az ott leírt funkcióbillentyűkkel lehet kiválasztani.

A programok futása során a HELP billentyűvel *segítség kérhető*, amely leírja, hogy melyik funkciót melyik billentyűvel érhetjük el.

Képernyőkezelésre ugyancsak mindkét programnál a RETURN, SHIFT+RETURN, DEL és a kurzormozgató billentyűk használhatók az ismertető elején megadott módon.


```

1620 " |-----+
1630 " |KURZOR BEALLITO |
1632 " |-----+
1650 " |Parameterai:
1660 " | x% - kepernyo sor [0,24]
1670 " | y% - kepernyo oszlop[0,39]
1680 " |Hasznalt konstans:
1690 " | xl$
1700 " |-----+
1710 print " |" left$(xl$,x%)tab(y%):return
1720 goto1840
1730 " |-----+
1740 " |KEPERNYOFEJLEC IRO |
1742 " |-----+
1760 " |Parameter
1770 " | ci$ - aktualis funkcio es
1780 " | ad$ - adatallomany nev
1790 " |Hasznalt konstans:
1800 " | sp$,vo$
1810 " |Hivott szubrutin:
1820 " | gosub1600
1830 " |-----+
1840 print " |" left$(vo$,39):
1850 print " | N vsor ";
1860 xi=int((30-len(ci$))/2)
1870 print tab(9) left$(sp$,xi);
1872 print ci$left$(sp$,30-xi-len(ci$))
1880 x$=ad$:gosub 1600
1890 print " |" left$(vo$,xi)ad$;
1892 print left$(vo$,39-xi-len(ad$))
1900 return
1910 goto2140
1920 " |-----+
1930 " |ABLAK KEZELO |
1940 " |Szines ablakot nyit s cimez, |
1950 " |vagy torli az ablakot. |
1952 " |-----+
1970 " |Parameterai:
1980 " | xa% - ablak 1.sora [0,24]
1990 " | ya% - ablak 1.oszlopa[0,39]
2000 " | xs% - ablak sorai [0,23]
2010 " | yo% - ablak oszlopai [0,38]
2020 " | xr = 1 ablakot nyit
2030 " | 0 ablakot torol
2040 " | xc% - ablak szinkodja
2050 " | x$ - alakcim (opcionalis)
2060 " |Munkavaltozok, konstansok:
2070 " | xi,xj,sp$
2100 " |Hivott szubrutin, fuggveny
2110 " | gosub1610,2510,fnxr(xr)
2130 " |-----+
2140 x%=xa%:y%=ya%:gosub 1610
2150 if x$="" then xj=0: goto 2210
2160 : xj=1:color1,xc%,5
2170 : xi=int((yo%-len(x$))/2)
2180 : printtab(ya%)chr$(fnxr(xr));
2190 : printleft$(vo$,xi)x$;
2200 : printleft$(vo$,yo%-xi-len(x$))
2210 color1,xc%
2220 for xi=xj+1 to xs%-xj
2230 : print tab(ya%)chr$(fnxr(xr));
2240 : print left$(sp$,yo%)
2250 next xi
2260 if xr=1 then xv%=1:gosub 2510
2270 return
2280 goto2440

```

```

2290 "+-----+
2300 " IUZENOMEZO
2310 " IKepernyo utolso soraba
2320 " Iuzenetet ir, vagy torol
2322 "+-----+
2340 " IParametere:
2350 " | x$ - uzenet (ha ures:torol)
2360 " | xf - 1 villogtat
2370 " | 0 nem
2380 " IHasznalt konstans:
2390 " | sp$
2400 " IHivott szubrutin, fuggveny:
2410 " | gosub1580,1590,1610,fnxf(xf)
2430 "+-----+
2440 gosub 1580
2450 x%=25:y%=0:gosub 1610
2460 gosub 1600
2470 print " "left$(sp$,xi);
2480 print " "chr$(fnxf(xf))x$" ";
2482 print left$(sp$,39-xi-len(x$))" ";
2490 gosub 1590:xf=0:x$=""
2500 return
2510 goto 2680
2520 "+-----+
2530 " IABLAK LAB
2540 " IAz aktualis ablak utolso
2550 " I sorat lezarja, vagy az
2560 " I utolso sorba 'RENDBEN?'
2570 " Iuzenetet ir ms valaszt var.
2572 "+-----+
2590 " IParameterei:
2600 " | xv% - 1 nem kell valasz
2610 " | 0 kell j rhagyrs
2620 " | az aktualis ablakot meg-
2630 " | hatarozo parameterek
2640 " I Eredmenyvaltozo:
2650 " | xv% - 0 ha 'Return'
2660 " | 1 ha 'Shift+Return'
2662 " IHivott szubrutin, fuggveny
2664 " | gosub 1610,3480,4340
2666 " | fnxr(xr)
2670 "+-----+
2680 x%=xa%+xs%-1:y%=ya%:gosub1610
2690 color1,xc%,5
2700 print chr$(fnxr(xr));
2702 print left$(vo$,yo%);
2710 if xv%=1 then color1,xc%:x$="":xr=0:return
2720 x$="RENDBEN?":xf=1:xz=xr
2730 y%=ya%+int((yo%-8)/2):gosub3480
2740 x$=" ":gosub 4340:xr=xz
2750 if xv%=1 then 2680
2760 return
2770 goto3100

```

```

2780 "+-----+
2790 " |ALTALANOS INPUTKEZELO |
2800 " |Adott kepernyopoziciotol kezdve |
2810 " |megadott hosszusagu numerikus, |
2820 " |vagy karakteres adatot ker. |
2830 " |Mezo vege: 'Return' |
2840 " |Karakter torles: 'Del' |
2842 "+-----+
2860 " |Parameterei:
2870 " | xt = 1 karakteres
2880 " | 2 numerikus
2890 " | x%,y% ua. mint 'KURZOR'-nal
2900 " | x$ - kiirando mezonev (opc)
2910 " | xh% - mezohossz
2920 " | xt% - tizedes jegyek szama,
2930 " | csak xt=2 eseten
2940 " | x$ - kiirand mezonev (opc)
2950 " | xr = 1:reverse mod
2960 " | 0:normal mod
2970 " |Eredmeny valtozok:
2980 " | x$ - a leutott karakterek
2990 " | xh$ - x$ jobbról xh% hosszura
3000 " | ures karakterekkel kieg.
3010 " | x - a beutott szam
3020 " |Munkavaltozok:
3030 " | z$,y$,xp%,xi,x1,x2
3040 " |Hasznalt konstansok
3050 " | sp$,pt$,ch$
3060 " |Hivott szubrutin, fuggveny
3070 " | gosub1610,2280,3150,3200,
3072 " | 3310,3710,fnxr(xr)
3090 "+-----+

```



```

3100 color1,xc%:gosub 1610
3110 printchr$(fnxr(xr))x$left$(pt$,xh%)
3120 y%=y%+len(x%):gosub 1610
3130 z$="." :x$="":xp%=0
3140 on xt gosub 3150,3200
3142 xr=0:x$="":gosub 2280:return
3150 rem "karakteres input
3160 gosub 3310
3170 if y<>2 then x%=x%+y%:print y%::goto 3160
3180 xh%=x%+left$(sp$,xh%-xp%)
3190 gosub 1610:print xh%" ";:return
3200 rem "numerikus input
3210 x%=str$(xh%+(xt%<>0)-xt%)+ " eg msz"
3220 if xt%<>0 then x%=x%+ " ms"+str$(xt%)+ " tize
des"
3230 x%=x%+ " jegy ":gosub2280:gosub1610
3232 xi=3:x1=0:x2=0
3240 gosub 3310
3250 if y=2 then x=val(x%):gosub 3710:return
3260 if y<x1 or y>15 then xp%=xp%-1:goto 3240
3270 print y%::x%=x%+y%
3272 if xi>5 then 3240
3280 if x1=0thenx1=-(y>2andy<5ory>5)*xp%
3290 if x2=0 then x2=-(y=5)*xp%
3300 gosub 3470:goto 3240
3310 rem "input segedrutin
3320 if xp%=xh% then z$="*":else z$="."
3330 print " "chr$(fnxr(xr))z$" ";
3340 getkey y%:y=instr(ch$,y%)
3350 on (1-y*(y<3)-3*(y>=3)) goto 3340,3360,3450
,3430
3360 if xp%=xh% thenz$=" ":elsez$="."
3370 if xp%<1 then 3340
3380 print chr$(fnxr(xr))z$" ";
3390 if xp%=x1 then x1=0
3400 if xp%=x2 then x2=0
3410 xp%=xp%-1:x%=left$(x%,xp%)
3420 gosub 3470:goto 3320
3430 if xp%=xh% then 3340
3440 xp%=xp%+1:return
3450 print " ";
3460 return
3470 xi=-5*(x1>0andx2=0)-6*(x2>0):return
3480 goto3660

```

```

3490 " +-----+
3500 " IKARAKTERES MEJELENITES |
3510 " IEgy alfanumerikus mezot adott |
3520 " Ikepernyopoziciotol kiir |
3522 " +-----+
3540 " IParameterai:
3550 " | x%,y% ua. mint 'KURZOR'-nal
3560 " | x$ - kiirando mezo
3570 " | xr = 1 reverse mod
3580 " | = 0 normrl mod
3590 " | xf = 1 villogtat
3600 " | = 0 nem villog
3610 " IHivott szubrutin, fuggveny
3620 " | gosub1610,fnxr(xr),fnxf(xf)
3650 " +-----+
3660 color1,xc%:gosub1610
3670 printchr$(fnxr(xr))chr$(fnxf(xf));
3680 printx$;
3690 print"RD";:xr=0:xf=0:x$=""
3700 return
3710 goto3920
3720 " +-----+
3730 " INUMERIKUS MEJELENITES |
3740 " INumerikus mezot adott keper- |
3750 " Inyo poziciotol kerekitve kiir |
3752 " +-----+
3770 " IParameterai:
3780 " | x%,y%,xr,xf ua. mint az
3790 " | 'ALFANUMERIKUS'-nal
3800 " | xt% - tizedesjegyek
3810 " | szama [0,9]
3820 " | xh% - a mezo hossza [4,11]
3830 " | x - kiirand szam
3840 " IHasznalt konstans
3850 " | xq$
3860 " IHivott szubrutin, fuggveny
3870 " | gosub1610,3970
3890 " | fnxr(xr),fnxf(xf)
3910 " +-----+
3920 color1,xc%:gosub 1610:gosub 3970
3930 printchr$(fnxr(xr))chr$(fnxf(xf));:xr=0:xf=
0
3940 if xt%=0 then print using left$(xq$,xh%);y:
:goto 3960
3950 print usingleft$(xq$,xh%-xt%-1)+". "+left$(x
q$,xt%);y:
3960 print"RD":return
3970 goto 4090
3980 " +-----+
3990 " IKEREKITES |
3992 " +-----+
4010 " IParameterai:
4020 " | xt% - tizedes jegyek szama
4030 " | x - kerekitando szam
4040 " IEredmeny valtozo
4050 " | y - kerekített szam
4060 " IMunkavaltozo
4070 " | xi
4080 " +-----+
4090 xi=10↑xt%
4100 y=sgn(x)*int(abs(x)*xi+0.5)/xi
4110 return
4120 goto 4300

```

```

4130 "+-----+
4140 " IKERDEZO
4150 " I Labjegyzetben kerdez
4152 "+-----+
4170 " I Parameterei:
4180 " | x$ - kerdes (opcionalis)
4190 " I Eredmenyvaltozo
4200 " | xv$ = bilentyukod
4210 " | xv% = 0 ha 'Return'
4220 " | | ha 'Shift+Return'
4230 " | 2 ha 'F1'
4240 " | 3 ha 'F2'
4250 " | . . .
4260 " | -1 kulonben
4270 " I Hivott szubrutin
4280 " | gosub 2280,4550
4290 "+-----+
4300 xr=1:gosub 2280
4310 getkey xv$:gosub 4550
4320 x$="":xr=0:gosub 2280
4330 return
4340 goto 4490
4350 "+-----+
4360 " I ELDONTENDO KERDES
4370 " I Labjegyzetben kerdez, csak
4380 " | 'Return'-t es 'Sh+Ret'-t fogad
4382 "+-----+
4400 " I Parameterei:
4410 " | x$ - kerdes (opcionalis)
4420 " I Eredmenyvaltozo
4430 " | xv% = 0 ha 'Return'
4440 " | 1 ha 'Shift+Return'
4450 " | -1 kulonben
4460 " I Hivott szubrutin
4470 " | gosub 4120
4480 "+-----+
4490 z$=x$
4500 if x$="" then x$="rendben?"
4510 x$=x$+" igen:'Sh+Ret' nem:'Ret"
4520 xr=1:gosub 4120
4530 if xv%<0 or xv%>1 then x$=z$:goto 4500
4540 return
4550 goto 4690
4560 "+-----+
4570 " I VALASZ KIERTEKELES
4580 " I Egy bilentyukodot konvertal
4590 " | sztenderd valaszkodra
4592 "+-----+
4610 " I Parameterei:
4620 " | xv$ - bilentyukod (ASCII)
4630 " I Eredmenyvaltozo:
4640 " | xv% - a valaszbilentyu kodja
4650 " | ua. mint 'KERDEZO'-nel
4660 " I Hasznalt munkavaltozo
4670 " | xi
4680 "+-----+
4690 xi=asc(xv$)
4700 xv%=-1-(xi=13)-2*(xi=141)-(xi=130)*(xi>132a
ndxi<141)
4710 return
4720 goto 4900

```



```

4730 " |-----|
4740 " |MENUKEZELO |
4750 " |Valasztas 'F' billentyukkel |
4752 " |-----|
4770 " |Parameterei:
4780 " | x$(i) menupontok szovege
4790 " | xa% - menu elso sora [3,23]
4800 " | ya% = menu elso oszlopa[0,30]
4810 " | xs% = a menupontok szama
4820 " |Eredmenyvaltozo:
4830 " | xv% - a valasztott pont kodja
4840 " | . ua. mint 'KERDEZO'-nel
4850 " |Munkavaltozo:
4860 " | xz
4870 " |Hivott szubrutinok:
4880 " | gosub 1610,1720,1910,3480,4120
4890 " |-----|
4900 xc%=5: xr=1: x%=xa%: y%=ya%: gosub 1610
4910 for xz=1 to xs%
4920 : printtab(y%) " W F"chr$(xz+48);
4930 : print " W- "x$(xz)" o": print
4940 next xz
4950 x$="valasztas 'F.' billentyukkel"
4960 xf=1: gosub 4120: xz=xv%
4970 if xv%<2 or xv%>xs%+1 then 4950
4980 y%=ya%+6: x%=xa%+2*xv%-4
4990 ci$=x$(xv%-1)
5000 x$=ci$: xr=1: xf=1: gosub 3480
5010 gosub 4340: x$=ci$: gosub 3480
5020 if xv%=0 then 4950
5030 yo%=ya%+len(x$(1))+4
5040 xs%=2*xs%: gosub 1910
5050 gosub 1720: xv%=xz: xc%=8
5060 return
5070 goto 5220
5080 " |-----|
5090 " |FELEZESEK KERESSES |
5091 " |Kereses nk% tombben |
5092 " |-----|
5110 " |Parameterei:
5120 " | nk%(ta) tomb, szigoruan
5130 " | . monoton novekvő
5140 " | nk% - keresett szam
5150 " |Eredmenyvaltozo
5160 " | ik - ha ix>0 a megtalalt,
5170 " | ha ix<0 az elso nagyobb
5180 " | tombelem indexe
5190 " |Munkavaltozok:
5200 " | xi, xa, xf
5210 " |-----|
5220 xa=1: xf=ta
5230 if nk%<nk%(1) then ik=-1: return
5240 if nk%>nk%(ta) then ik=-ta-1: return
5250 xi=xf-xa: if xi=1 then 5300
5260 xi=xa+int(xi/2)
5270 if nk%<nk%(xi) then xf=xi: goto 5250
5280 if nk%>nk%(xi) then xa=xi: goto 5250
5290 ik=xi: return
5300 if nk%=nk%(xa) then ik=xa: return
5310 if nk%=nk%(xf) then ik=xf: return
5320 ik=-xa-1
5330 return

```

88-1635 — Szegedi Nyomda
Felelős vezető: Surányi Tibor igazgató.

A VORKER Kiszövetkezet ajánlja a Tisztelt Megrendelők figyelmébe az alábbi szoftver termékeket és — szolgáltatásokat.

1. C 16, C Plus/4, C 64 és Spectrum gépekre kifejlesztett oktató és játékprogramokat, amelyek Magyarországon a legolcsóbbak. Ezek közül az idegen és magyar nyelv, a matematika, a kémia, a biológia és a fizika tantárgyi oktatóprogramok segítik a pedagógusok és a tanulók munkáját.
2. Egy- és kétszemélyes játékok, valamint totóprogramok segítik a szabadidő jobb eltöltését.
3. IBM kompatibilis gépekre vállaljuk a munkaügyi és bérszámfejtő rendszerünk adaptálását.

Különleges hardver ajánlatunk: a világon egyedülálló fejlesztésünk a TC-NET+4 számítógép-interfész, amellyel 16 db C 16 illetve C Plus/4 számítógépet kapcsolhatunk lokális hálózatba. Ez két fontos célt szolgál: egyrészt az összes gép ilyen módon használhat egyetlen mágneslemezt és nyomtató egységet, másrészt megvalósítja számítástechnikai vonalon azokat az előnyöket, amelyeket a nyelvoktatás területén egy nyelvi labor biztosít. Ily módon számítástechnikai kabinet hozható létre kis anyagi ráfordítással.

Várjuk érdeklődésüket!

VORKER® Vállalatközi
Organizációs Ipari Szolgáltató
és Kereskedelmi Kiszövetkezet
Szeged, Pf: 711.
Telex: 82-688
Telefon: (62) 26-144
(62) 25-479