

```

Program Osszegfokozatok; {KERET}
{
  Feladat:
  Az alábbi szerkezetű file beolvasása és kilistázása úgy, hogy
  közben a megfelelő "struktúraváltásokkor" ki kell egészíteni
  az ún. összegfokozat rekorddal.
  Összegfokozat tartalmazza:
  Az aktuális struktúrát azonosító mezők értékeit (de csak azokat)
  és az ebbe a struktúrába tartozó elemek számát v. összegét ...
  (hogy mit, az feladatfüggő).
  Bevezető, "előfeladatok":
  1. Síma, lapozott lista.
  2. A file-t úgy kilistázni, hogy az egy tanszéken dolgozók
     kiférjenek egy lapra (a tanszék nyitórekorddal együtt).
     Ef: nincs tanszék, ahol többen dolgoznának, mint a képernyősorok
        száma-1.
  File-szerkezet:
  1. szint: EGYETEM
     szintkód, név, rektori hivatal cím, tel.szám
     karok

  2. szint: KAR
     szintkód, név, dékáni hivatal cím, tel.szám
     tanszékek (tanszéki egységek)

  3. szint: TANSZÉKEK
     szintkód, név, tanszékcím, tel.szám
     dolgozók

  4. szint: DOLGOZÓ
     szintkód, név, lakcím, tel.szám, fizetés, ...

  Feltételezés:
  A file strukturálisan helyes, tanszékenként nem üres, de
  az adott listázási feladatban kikötött számút nem haladja
  meg.

  Megjegyzés:
  A file csupa azonos típusú elemből áll. Ezért a struktúra-
  nyitó elemek kiegészülnek "--tartalom nélküli"-- mezőkkel,
  amelyeket föl lehet használni az összegfokozatok utólagos
  tárolására.
}

Uses Newdelay,Crt;

{$i Randnev.inc}

Type TSzint=(Dolgozo,Tanszek,Kar,Egyetem,Ures);
Const KSzint:Array [TSzint] of String=(' ','T','K','E',''); {kiirandó}
szintKodH=1; nevH=28; cimH=26; telH=9; fizuH=6; {kiíráshossz}
Type TElem=Record
szintKod:TSzint;
nev:String[nevH]; cim:String[cimH]; tel:String[telH];
fiz:LongInt;
End;
Type TFile=File of TElem;
TMegnyitasMod=(Letrehozás,Olvasás);

```

```

Const LaphOssz=20;

{$i osszfok.inc --
  ebben található a lényeg szemponjából kevésbé fontos
  általános konstansok, változók, eljárások, függvények.
  Pl. Var sor:Byte, ami a képernyőre kiírt sorok számát tartalmazza
  Procedure ElemKi kiír egy elemet a fejléchez igazodva (izlésesen)
  Procedure Lapoz, ami lapot töröl, és fejléctet ír ki, sor-t nulláz
  Procedure BillreVar, ami billentyű-lenyomásra vár
  Procedure Megnyit file-t, megfelelő módon
  Procedure General adatfile-t a fenti szerkezetben
  Procedure Lezar file-t
  Procedure Inic, ami inicializálja a kiÍrβst (~MegnyitLista)
}

{
  Listázó rutinok:
  1. Listaz
    -- egyszerű (természetes) virtuális típus bevezetésével
  2. ListazMaskent
    -- (általánosítható) virtuális típus bevezetésével
    (útban a tanszékenkénti listázás felé)
  3. TanszTordListaz
    -- listázás 'tanszék dologozók együtt maradásával' feltétellel
}

Procedure Listaz(Var f:TFile);
(*
  Virtuális típusok:
  Input: TBeLap=TFile
         BeNyit,BeVége?,Beolvas,Bezár
         (A reprezentáció egyszerűsége miatt:
         * BeNyit=Megnyitas,
         * BeVége?=Eof,
         * Beolvas=Read,
         * Bezár=Lezar)
  Output: TKiLap="Rekord" (sor:Egész,képernyő:TKépernyő)
          =sor:Egész
          KiNyit,Kiir,Kizár
          (A reprezentáció egyszerűsége miatt:
          * KiNyit=Lapoz,
          * Kiir=ElemKi,
          * Kizár=BillreVar, ha kell; képernyőt töröl)
*)
{
  Típus-reprezentációk:
  Var sor:Byte; <-- osszfok.inc
}
{
  Típus-implementációk:
}
Procedure Kizar(képernyő:TKilap);
Begin
  If sor>0 then BillreVar;
  ClrScr
End;

```

```

{
  A megoldás:
}
Var
  elem:TElem;
Begin {Listaz}
  Megnyit(f,Olvasas); Lapoz{képernyő};
  While not Eof(f) do
  Begin
    Read(f,elem);
    ElemKi({képernyő},elem);
  End;
  Lezar(f); Kizar{képernyő}
End {Listaz};

(* ----- *)

Procedure ListazMaskent(Var f:TFile);
(*
  Virtuális típusok:
  Input:  TBeLap=Rekord(db:Egész, sorok:Tömb(1..MaxTszDolg+1:TElem))
          Előfeltétel => MaxTszDolg+1=LapHossz
          BeNyit,BeVége?,Beolvas,Bezár
          * BeNyit=Megnyitas+belap-puffer inicializálás,
          * BeVége?=Eof,
          * Beolvas=puffernyi olvasás,
          * Bezár=Lezar)
  Output: TKiLap="Rekord"(sor:Egész,képernyő:TKépernyő)
          =sor:Egész
          KiNyit,Kiir(belap-puffer!),Kizár
          (A reprezentáció egyszerűsége miatt:
          * KiNyit=Lapoz,
          * Kiir=belap-puffer kiírása
            + várakozás
            + Lapoz
            + puffer inicializálása
          * Kizár=ClrScr)
*)
{
  Típus-reprezentációk:
}
Type TBeLap=Record
  db:Byte; sorok:Array [1..LapHossz] of TElem;
End;
Var  egyTanszek:TBeLap;
     {sor:Byte; <-- osszfok.inc}
{
  Típus-implementációk:
}
Procedure BeNyit(Var f:TFile);
Begin
  Megnyit(f,Olvasas);
  egyTanszek.db:=0;
End;

```

```

Procedure Beolvas(Var f:TFile {;Var egyTanszek:TBeLap});
Begin
  While Not Eof(f) and (egyTanszek.db<LapHossz) do
    Begin
      Inc(egyTanszek.db);
      Read(f,egyTanszek.sorok[egyTanszek.db]);
    End;
  End;
Procedure Kiir(Var képernyő:TKépernyő, Var egyTanszek:TBelap);
  Var i:Integer;
Begin
  For i:=1 to egyTanszek.db do
    Begin
      ElemKi({képernyő,}egyTanszek.sorok[i]); {nem használom ki, hogy
                                                figyeli a sorok számát}
    End;
    egyTanszek.db:=0;
    BillreVar; Lapoz;
  End;
{
  A megoldás:
}
Begin{ListazMaskent}
  Benyit(f); Lapoz{képernyő};
  While not Eof(f) do
    Begin
      Beolvas(f{,egyTanszek});
      Kiir{képernyő,egyTanszek};
    End;
    Lezar(f); ClrScr
End{ListazMaskent};

(* ----- *)

Procedure TanszTordListaz(Var f:TFile);
(*
  Virtuális típusok:
  Input: bemeneti egység vagy egy tanszék összes dolgozója a nyitóval
        vagy valami más elem
        a bementi eleme határát csak előreolvasási technikával
        tudjuk megtalálni
        TBeEgyseg=Rekord(db:Egész,
                        e:TElem [az előreolvasott],
                        sorok:Tömb(1..MaxTszDolg+1:TElem))
  Előfeltétel => MaxTszDolg+1=LapHossz
  BeNyit,BeVége?,Beolvas,Bezár
  * BeNyit=Megnyitas+belap-puffer inicializálás+előreolvasás
  * BeVége?=Eof,
  * Beolvas=egységolvasás előreolvasással,
  * Bezár=Lezar)
  Output: TKiLap="Rekord" (sor:Egész,képernyő:TKépernyő)
         =sor:Egész
  KiNyit,Kiir (belap-puffer!),Kizár
  (A reprezentáció egyszerűsége miatt:
  * KiNyit=Lapoz,
  * Kiir=belap-puffer kiírása+sor-állítás, ha még kifer

```

```

        ha nem fér ki, akkor
        + várakozás
        + Lapoz
        + puffer inicializálása
    * Kizár= várakozás + ClrScr)
*)
{
    Típus-reprezentációk:
}
Type TBeEgyseg=Record
    db:Byte;
    e:TElem;{az előreolvasott}
    elemek:Array [1..LapHossz] of TElem;
End;
Var egyTanszek:TBeEgyseg;
    {sor:Byte; <-- osszfok.inc}
{
    Típus-implementációk:
}
Procedure BeNyit(Var f:TFile);
Begin
    Megnyit(f,Olvasas);
    egyTanszek.db:=0;
    Read(f,egyTanszek.e);
End;
Procedure Beolvas(Var f:TFile {;Var egyTanszek:TBeLap});
{Ef: NOT Eof(f) AND e=előreolvasott elem}
Begin
    egyTanszek.db:=1; egyTanszek.elemek[1]:=egyTanszek.e;
    Read(f,egyTanszek.e);
    While Not Eof(f) and (egyTanszek.e.szintKod=Dolgozo) do
    Begin
        Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
        Read(f,egyTanszek.e);
    End;
    {Uf: Eof(f) => egyTanszek[1..egyTanszek.db]+e=egy egység AND
      NOT Eof(f) => e=következő nem dolgozó elem}
    If Eof(f) then
    Begin
        Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
    End;
    {Uf: egyTanszek[1..egyTanszek.db]=egy egység AND
      Eof(f) => e=üres AND
      NOT Eof(f) => e=következő nem dolgozó elem}
End;
Procedure Kiir(Var képernyő:TKépernyő, Var egyTanszek:TBelap);
    Var i:Integer;
Begin
    If sor+egyTanszek.db>LapHossz then
    Begin
        BillreVar; Lapoz;
    End;
    For i:=1 to egyTanszek.db do
    Begin
        ElemKi({képernyő,}egyTanszek.elemek[i]); {nem használom ki, hogy
            figyeli a sorok számát,
            csak, hogy kiírja és a

```

```

                                                                    sor-t növeli}
    End;
    egyTanszek.db:=0;
End;
Procedure Kizar{képernyő:TKilap};
Begin
    If sor>0 then BillreVar;
    ClrScr
End;
{
    A megoldás:
}
Begin{TanszTordListaz}
    BeNyit(f); Lapoz;
    While not Eof(f) do
        Begin
            Beolvas(f{,egyTanszek});
            Kiir{képernyő,egyTanszek};
        End;
    Kizár;
End{TanszTordListaz};

(* ----- *)

Procedure OsszFokosListaz(Var f:TFile);
(*
    Virtuális típusok:
    Input: bemeneti egység vagy egy tanszék összes dolgozója a nyitóval
           vagy valami más elem
           a bementi eleme határát csak előreolvasási technikával
           tudjuk megtalálni
           TBeEgység=Rekord(db:Egész,
                           e:TElem [az előreolvasott],
                           elemek:Tömb(1..MaxTszDolg+1:TElem))
           Előfeltétel => MaxTszDolg+1=LapHossz
           BeNyit,BeVége?,Beolvas,Bezár
           * BeNyit=Megnyitás+belap-puffer inicializálás+előreolvasás
           * BeVége?=Eof,
           * Beolvas=egységolvasás előreolvasással,
           * Bezár=Lezar)
    Output: TKiLap="Rekord"(sor:Egész,képernyő:TKépernyő)
            =sor:Egész
            KiNyit,Kiir(belap-puffer!),Kizár
            (A reprezentáció egyszerűsége miatt:
            * KiNyit=Lapoz,
            * Kiir=belap-puffer kiírása+sor-állítás, ha még kifér
              ha nem fér ki, akkor
              + várakozás
              + Lapoz
              + puffer inicializálása
            * Kizár= várakozás + ClrScr)
*)
Begin{OsszFokosListaz}
    (* ide kell valami *)
End{OsszFokosListaz};

```

```
Var
    egy:TFile;

Begin
    Inic;

    ClrScr;
    HighVideo; Writeln(Kozepre('File-létrehozás',80)); NormVideo;
    Megnyit(egy,Letrehozás);
    General(egy);
    Lezar(egy);

    ClrScr;
    HighVideo; Writeln(Kozepre('Síma listázás',80)); NormVideo;
    {Sima listázás:}
    Listaz(egy);
    (*
    ClrScr;
    HighVideo; Writeln(Kozepre('Síma listázás -- másként',80)); NormVideo;
    {Sima listázás, másként:}
    ListazMaskent(egy);

    ClrScr;
    HighVideo; Writeln(Kozepre('Tanszékre tördelt listázás',80)); NormVideo;
    {TanszékreTördeltListázás:}
    TanszTordListaz(egy);
    *)
    ClrScr;
    HighVideo; Writeln(Kozepre('Összefokozatos listázás',80)); NormVideo;
    {Összfokos listázás:}
    OsszFokosListaz(egy);
End.
```