

```

Program SzekvencialisFileListazasok; {KERET}
    {Az összfokozatos listázás elé}
{
    Feladat:
    Az alábbi szerkezetű file beolvasása és kilistázása
    1. lapozottan, ill. úgy, hogy
    2. az egy tanszéken dolgozók kiferjenek 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, nem üres.

    Megjegyzés:
    A file csupa azonos típusú elemből áll. Ezért a struktúra-
    nyitó elemek kiegészülnek --"tartalom nélküli"-- mezőkkel,
    amelyeket esetleg föl lehet használni az összefokozatok
    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-nyomásra vár
    Procedure Megnyit file-t, megfelelelo módon
    Procedure General adatfile-t a fenti szerkezetben
    Procedure Lezar file-t
    Procedure Inic, ami inicializálja a kiírást (~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=Várakozás + ClrScr
*)
Begin{Listaz}
  (* ide kell valami *)
End{Listaz};

(* ----- *)

Procedure ListazMaskent(Var f:TFile);
(*
  Virtuális típusok:
  Input:  TBeLap=Rekord(db:Egész, sorok:Tömb(1..LapHossz:TElem))
          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
*)
Begin{ListazMaskent}
  (* ide kell valami *)
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
  TBeEgység=Rekord(db:Egész,
                    sorok:Tömb(1..MaxTszDolg+1:TElem),
                    e:TElem [az előreolvasott])
  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 kifér
           ha nem fér ki, akkor
           + várakozás
           + Lapoz
           + belap-puffer kiírása+sor-állítás
           + puffer inicializálása
  * KiZár=Várakozás + ClrScr
*)
Begin{TanszTordListaz}
  (* ide kell valami *)
End{TanszTordListaz};

Var
  egy:TFile;

Begin
  Inic;

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

  ClrScr;
  HighVideo; Writeln(Kozepre('Sima listázás',80)); NormVideo;
  {Sima listázás:}
  Listaz(egy);

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

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

```