

Az „Összefokozatos listázás” téma 1. gyakorlata

Feladat:

Az alábbi szerkezetű file beolvasása és kilistázása

1. lapozottan "józanésszel",
2. úgy, hogy a típusfinomítással komolyabban élünk
(azaz ne használjuk ki, hogy az ElemKi figyel a kiírt sorok számát),
3. az egy tanszéken dolgozók kifejezjenek egy lapra
(a tanszék nyitórekorddal együtt)
Ef: nincs tanszék, ahol többen dolgoznának, mint a képernyősorok
száma-2 ("-2" = fejsor + sorszámot tartalmazó lábléc)
4. minden szervezeti egységet egy ún. összegző rekorddal kiegészítve, és
fenntartva az együttmaradás feltételét.
Ef: olyan, hogy megvalósítható legyen...

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: legalább 1 dolgozó van.

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.

Reprezentáció:

Input -

```

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;

```

Output:

```

Const MaxLapHossz=100; {egy-egy lapnyi maximum ennyi sorból állhat}
Type TKepernyo= képernyő (Pascal) | TMemo (Lazarus) | TTabSheet (Lazarus)
           | Text *
Var    lapSzam,sor:byte; {a megnyitott lapok,
                           az aktuális lapra kiírt sorok száma}

```

Implementáció:**osszfok.inc - "legalapabb alap" rutinjai**

ebben található a lényeg szemponjából kevésbé fontos általános konstansok, változók, eljárások, függvények. Például:

```

Procedure ElemKi kiír egy elemet a fejléchez igazodva (izlésesen),
           ügyel a lapra kiírt sorok számára
           TKepernyo-tól függően más és más megoldás
Procedure Lapoz, ami lapot töröl, és fejléctet ír ki, sor-t nulláz
           TKepernyo-tól függően más és más megoldás
Procedure BillreVar, ami billentyű-nyomásra vár
           TKepernyo-tól függően más és más megoldás
           (pl. a lapot megfelelő számú üres sorral tölt föl)
Procedure Megnyit file-t, megfelelő módon
Procedure General adatfile-t a fenti szerkezetben
Procedure Lezar file-t
Procedure Inic, ami inicializálja a kiírást (a Lapoz-on túliak)
           TKepernyo-tól függően más és más megoldás

```

Listázó rutinok:

1. **simaLista**
-- egyszerű, lapozott listázás (természetes) virtuális típus bevezetésével
2. **maskentLista**
-- (általánosítható) virtuális típus bevezetésével (útban a tanszékenkénti listázás felé)
3. **tanszékenkentLista**
-- listázás 'tanszék dologozói együtt maradásával' feltétellel

```

Procedure simaLista;
(*)
  Virtuális típusok:
  Input:  TBeLap=TFile
          BeNyit,BeVége?,BeOlvas,BeZár
          (A reprezentáció egyszerűsége miatt:
          * BeNyit=Megnyit,
          * BeVége?=Eof,
          * BeOlvas=Read,
          * BeZár=Lezar)

  Output: TKiLap="Rekord" (sor:Egész*,képernyő:TKepernyő)
          KiNyit,KiIr,KiZár
          (A reprezentáció egyszerűsége miatt:
          * KiNyit=inic + lapoz,
          * KiIr=ElemKi (lapozásra ügyelve kiír),
          * KiZár=ha kell, lap alja)
*)

```

* Pirossal szedem, amik output-reprezentációfüggők.

♦ a képernyőn lévő sorok aktuális száma

```

{
  Típus-reprezentációk:
}
Var elem:TElem;
      f:TFile;
      {lapSzam,sor:Byte; <-- Globális}
{
  Típus-implementációk:
}
Procedure KiNyit{output:TKiLap};
Begin
  Inic; Lapoz;
End;{KiNyit}
Procedure KiZar{output:TKiLap};
Begin
  If sor>0 then LapAlja;
End;{KiZar}
{
  A megoldás:
}
Begin{Listaz}
  Megnyit(f,Olvasas); Lapoz{output};
  While not Eof(f) do
  Begin
    Read(f,elem);
    ElemKi({output,}elem);
  End;
  Lezar(f); KiZar{output}
End{Listaz};
(* ----- *)
Procedure maskentLista;
(*
  Virtuális típusok:
  Input:  TBeLap=Rekord(db:Egész, sorok:Tömb(1..{Max}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)
          KiNyit,KiIr(beLap-puffer!),KiZár
          * KiNyit=inic,
          * KiIr=lapoz
            + beLap-puffer kiírása
            + lap alja
          * KiZár=Lezár
*)
{
  Típus-reprezentációk:
}
Type TBeLap=Record
      db:Byte; sorok:Array [1..{Max}LapHossz] of TElem;
      End;
Var egyLap:TBeLap;
      f:TFile;
      {lapSzam,sor:Byte; <-- Globális}

```

```

{
  Típus-implementációk:
}
Procedure BeNyit(Var f:TFile);
Begin
  Megnyit(f,Olvasas);
End;{BeNyit}
Procedure BeOlvas(Var f:TFile {;Var egyLap:TBeLap});
Begin
  egyLap.db:=0;
  While Not Eof(f) and (egyLap.db<LapHossz-2) do
  Begin
    Inc(egyLap.db);
    Read(f,egyLap.sorok[egyLap.db]);
  End;
End;{BeOlvas}
Procedure KiNyit{output:TKilap};
Begin
  Inic;
End;{KiNyit}
Procedure KiIr{Var képernyő:TKépernyő, Var egyLap:TBeLap};
  Var i:Integer;
Begin
  Lapoz; //a fejléc kiírása
  For i:=1 to egyLap.db do
  Begin
    ElemKi({képernyő,}egyLap.sorok[i]); {nem használom ki, hogy
                                          figyelj a sorok számát}
  End;
  LapAlja; //a lábléc kiírása
End;{KiIr}
Procedure KiZar{output:TKiLap};
Begin
  //most semmi fontos
End;{KiZar}
{
  A megoldás:
}
Begin{maskentLista}
  Benyit(f); KiNyit{output};
  While not Eof(f) do
  Begin
    BeOlvas(f{, egyLap});
    KiIr{képernyő, egyLap};
  End;
  Lezar(f); KiZar{output}
End{maskentLista};

(* ----- *)

Procedure tanszekenkentLista;
(*
  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

```

♥ TKépernyő-től függő, hiszen, ha a lapalja egy lábléct (is) jelent, akkor azt is bele kell számítani a laphosszba. A „füles lap” esetén ilyenre nincs szükség, tehát ott ez nem jelent többlet sort. 2, mert 1 fejléc + 1 lábléc

```

TBeEgyseg=Rekord (db:Egész,
                  e:TElem [az előreolvasott],
                  sorok:Tömb (1..MaxTszDolg+1:TElem))
Előfeltétel => MaxTszDolg+1=MaxLapHossz
BeNyit,BeVége?,BeOlvas,BeZár
* BeNyit=Megnyitas + előreolvasás
* BeVége?=Eof,
* BeOlvas=egységolvasás előreolvasással,
* BeZár=Lezar)
Output: TKiLap="Rekord" (sor:Egész,képernyő:TKépernyő)
KiNyit,KiIr (belap-puffer!),KiZár
* KiNyit=Inic + lapoz,
* KiIr=ha nem fér ki => lap alja
                        + lapoz
                        beLap-puffer kiírása + sor-állítás
* KiZár=ha kell, lap alja
*)
{
  Típus-reprezentációk:
}
Type TBeEgyseg=Record
    db:Byte;
    e:TElem;{az előreolvasott}
    sorok:Array [1..LapHossz] of TElem;
End;
Var egyTanszek:TBeEgyseg;
    f:TFile;
    {lapSzam,sor:Byte; <-- Globális}
{
  Típus-implementációk:
}
Procedure BeNyit(Var f:TFile);
  {Ef: NOT Eof(f)}
Begin
  //..
End;
Procedure BeOlvas(Var f:TFile {;Var egyTanszek:TBeLap});
  {Ef: NOT Eof(f) AND e=előreolvasott elem}
Begin
  //..
End;
Procedure KiIr{Var képernyő:TKépernyő, Var egyTanszek:TBeLap};
  Var i:Integer;
Begin
  //..
End;
Procedure KiZar{output:TKiLap};
Begin
  //..
End;
{
  A megoldás:
}
Begin{TanszekenkentLista}
  //..
End{TanszekenkentLista};

```