

```

1  Program Osszegfokozatok; {MEGOLDÁS}
2  {
3      Feladat:
4      Az alábbi szerkezetű file beolvasása és kilistázása úgy, hogy
5      közben a megfelelő "struktúraváltásokkor" ki kell egészíteni
6      az ún. összegfokozat rekorddal. Vagyis egy új tanszék indulásakor
7      a lezárt tanszék összegfokozata zárja a tanszéki sorokat; egy új
8      kar esetében nemcsak az lezárt tanszék, hanem a kar összegfokozat
9      rekordja is generálandó. Azaz a "záró" rekordok száma vagy 1, vagy
10     2, vagy 3.
11     Mind a nyitó, mind az aktuális struktúrához tartozó összegfokozati
12     rekord egy lapon kell megjelenjen. (Feltesszük, hogy ez lehetséges.)
13     Összegfokozat tartalmazza:
14     Az aktuális struktúrát azonosító mezők értékeit (de csak azokat)
15     és az ebbe a struktúrába tartozó elemek számát v. összegét ...
16     (hogy mit, az feladatfüggő).
17     Bevezető, "előfeladatok":
18     1. Síma, lapozott lista.
19     2. A file-t úgy kilistázni, hogy az egy tanszéken dolgozók
20     kiferjenek egy lapra (a tanszék nyitórekorddal együtt).
21     Ef: nincs tanszék, ahol többen dolgoznának, mint a képernyősorok
22     száma-1.
23     File-szerkezet:
24     1. szint: EGYETEM
25         szintkód, név, rektori hivatal cím, tel.szám
26         karok
27
28     2. szint: KAR
29         szintkód, név, dékáni hivatal cím, tel.szám
30         tanszékek (tanszéki egységek)
31
32     3. szint: TANSZÉKEK
33         szintkód, név, tanszékcím, tel.szám
34         dolgozók
35
36     4. szint: DOLGOZÓ
37         szintkód, név, lakcím, tel.szám, fizetés, ...
38
39     Feltételezés:
40     A file strukturálisan helyes, tanszékenként nem üres, de
41     az adott listázási feladatban kikötött számút nem haladja
42     meg.
43
44     Megjegyzés:
45     A file csupa azonos típusú elemből áll. Ezért a struktúra-
46     nyitó elemek kiegészülnek --"tartalom nélküli"-- mezőkkel,
47     amelyeket föl lehet használni az összegfokozatok utólagos
48     tárolására.
49 }
50 Uses Newdelay,Crt;
51
52 { $i Randnev.inc }
53
54 Type TSzint=(Dolgozo,Tanszek,Kar,Egyetem,Ures);
55 Const KSzint:Array [TSzint] of String=(' ','T','K','E',''); {kiirandó}
56         szintKodH=1; nevH=28; cimH=26; telH=9; fizuH=6; {kiíráshossz}
57 Type TElem=Record
58         szintKod:TSzint;
59         nev:String[nevH]; cim:String[cimH]; tel:String[telH];
60         fiz:LongInt;
61 End;
62 Type TFile=File of TElem;
63         TMegnyitasMod=(Letrehozás,Olvasás);
64 Const LaphOssz=20;
65

```

```

66  { $i osszfok.inc --
67      ebben található a lényeg szemponjából kevésbé fontos
68      általános konstansok, változók, eljárások, függvények.
69      Pl. Var sor:Byte, ami a képernyőre kiírt sorok számát tartalmazza
70      Procedure ElemKi kiír egy elemet a fejléchez igazodva (izlésesen)
71      Procedure Lapoz, ami lapot töröl, és fejlécet ír ki, sor-t nulláz
72      Procedure BillreVar, ami billentyű-lenyomásra vár
73      Procedure Megnyit file-t, megfelelő módon
74      Procedure General adatfile-t a fenti szerkezetben
75      Procedure Lezar file-t
76      Procedure Inic, ami inicializálja a kiíróst (~MegnyitLista)
77  }
78
79  {
80      Listázó rutinok:
81      1. Listaz
82          -- egyszerű (természetes) virtuális típus bevezetésével
83      2. ListazMaskent
84          -- (általánosítható) virtuális típus bevezetésével
85          (útban a tanszékenkénti listázás felé)
86      3. TanszTordListaz
87          -- listázás 'tanszék dologozók együtt maradásával' feltétellel
88  }
89  Procedure Listaz (Var f:TFile);
90  (*
91      Virtuális típusok:
92      Input:  TBeLap=TFile
93              BeNyit, BeVége?, Beolvas, Bezár
94              (A reprezentáció egyszerűsége miatt:)
95              * BeNyit=Megnyitas,
96              * BeVége?=Eof,
97              * Beolvas=Read,
98              * Bezár=Lezar
99      Output: TKiLap="Rekord" (sor:Egész, képernyő:TKépernyő)
100              =sor:Egész
101              KiNyit, Kiir, Kizár
102              * KiNyit=Lapoz,
103              * Kiir=ElemKi,
104              * Kizár=BillreVar, ha kell; képernyőt töröl
105  *)
106  {
107      Típus-reprezentációk:
108      Var sor:Byte; <-- osszfok.inc
109  }
110  {
111      Típus-implementációk:
112  }
113  Procedure Kizar {képernyő:TKilap};
114  Begin
115      If sor>0 then BillreVar;
116      ClrScr
117  End;
118  {
119      A megoldás:
120  }
121  Var
122      elem:TElem;
123  Begin {Listaz}
124      Megnyit(f, Olvasas); Lapoz {képernyő};
125      While not Eof(f) do
126      Begin
127          Read(f, elem);
128          ElemKi ({képernyő}, elem);
129      End;
130      Lezar(f); Kizar {képernyő}

```

```

131   End{Listaz};
132
133   (* ----- *)
134
135   Procedure ListazMaskent (Var f:TFile);
136   (*
137      Virtuális típusok:
138      Input:  TBeLap=Rekord (db:Egész, sorok:Tömb (1..LapHossz:TElem))
139             BeNyit, BeVége?, BeOlvas, BeZár
140             * BeNyit=
141               a) Megnyitas+belap-puffer inicializálás,
142               b) Megnyitas+semmi
143             * BeVége?=Eof,
144             * BeOlvas=
145               a) pufferbe olvasás betelésig,
146               b) pufferbe olvasás 1-től, amíg lehet
147             * BeZár=Lezar
148      Output: TKiLap="Rekord" (sor:Egész, képernyő:TKépernyő)
149             =sor:Egész
150             KiNyit, KiIr (belap-puffer!), KiZár
151             * KiNyit=Lapoz,
152             * KiIr=
153               a) belap-puffer kiírása
154                  + várakozás
155                  + Lapoz
156                  + puffer inicializálása
157               b) belap-puffer kiírása
158                  + várakozás
159                  + Lapoz
160                  + (a puffert nem változtatja meg!)
161             * KiZár=Várakozás (ha kell) + ClrScr
162   *)
163   {
164      Típus-reprezentációk:
165   }
166   Type TBeLap=Record
167       db:Byte; sorok:Array [1..LapHossz] of TElem;
168   End;
169   Var   egyTanszek:TBeLap;
170         {sor:Byte; <-- osszfok.inc}
171   {
172      Típus-implementációk:
173   }
174   Procedure BeNyit (Var f:TFile);
175   Begin
176       Megnyit (f, Olvasas);
177       egyTanszek.db:=0;
178   End;
179   Procedure Beolvas (Var f:TFile {; Var egyTanszek:TBeLap});
180   Begin
181       While Not Eof (f) and (egyTanszek.db<LapHossz) do
182       Begin
183           Inc (egyTanszek.db);
184           Read (f, egyTanszek.sorok[egyTanszek.db]);
185       End;
186   End;
187   Procedure Kiir {Var képernyő:TKépernyő, Var egyTanszek:TBeLap};
188   Var i:Integer;
189   Begin
190       For i:=1 to egyTanszek.db do
191       Begin
192           ElemKi ({képernyő,} egyTanszek.sorok[i]); {nem használom ki, hogy
193                                                       figyeli a sorok számát}
194       End;
195       egyTanszek.db:=0;

```

```

196     BillreVar; Lapoz;
197 End;
198 {
199     A megoldás:
200 }
201 Begin{ListazMaskent}
202     Benyit(f); Lapoz{képernyő};
203     While not Eof(f) do
204     Begin
205         Beolvas(f{,egyTanszek});
206         Kiir{képernyő,egyTanszek};
207     End;
208     Lezar(f); ClrScr
209 End{ListazMaskent};
210
211 (* ----- *)
212
213 Procedure TanszTordListaz(Var f:TFile);
214 (*
215     Virtuális típusok:
216     Input:  bemeneti egység vagy egy tanszék összes dolgozója a nyitóval
217             vagy valami más elem
218             a bementi eleme határát csak előreolvasási technikával
219             tudjuk megtalálni
220     TBeEgyseg=Rekord(db:Egész,
221                     e:TElem [az előreolvasott],
222                     sorok:Tömb(1..MaxTszDolg+1:TElem))
223     Előfeltétel => MaxTszDolg+1=LapHossz
224     BeNyit,BeVége?,Beolvas,Bezár
225     * BeNyit=Megnyitas+belap-puffer inicializálás+előreolvasás
226     * BeVége?=Eof,
227     * Beolvas=egységolvasás előreolvasással,
228     * Bezár=Lezar
229     Output: TKiLap="Rekord"(sor:Egész,képernyő:TKépernyő)
230             =sor:Egész
231     KiNyit,Kiir(belap-puffer!),Kizár
232     * KiNyit=Lapoz,
233     * Kiir=belap-puffer kiírása+sor-állítás, ha még kifér
234             ha nem fér ki, akkor
235             + várakozás
236             + Lapoz
237             + puffer inicializálása
238     * Kizár= várakozás + ClrScr
239 *)
240 {
241     Típus-reprezentációk:
242 }
243 Type TBeEgyseg=Record
244     db:Byte;
245     e:TElem;{az előreolvasott}
246     elemek:Array [1..LapHossz] of TElem;
247 End;
248 Var egyTanszek:TBeEgyseg;
249     {sor:Byte; <-- osszfok.inc}
250 {
251     Típus-implementációk:
252 }
253 Procedure BeNyit(Var f:TFile);
254 Begin
255     Megnyit(f,Olvasas);
256     egyTanszek.db:=0;
257     Read(f,egyTanszek.e);
258 End;

```

```

259 Procedure Beolvas(Var f:TFile {;Var egyTanszek:TBeLap});
260 {Ef: NOT Eof(f) AND e=előreolvasott elem}
261 Begin
262   egyTanszek.db:=1; egyTanszek.elemek[1]:=egyTanszek.e;
263   Read(f, egyTanszek.e);
264   While Not Eof(f) and (egyTanszek.e.szintKod=Dolgozo) do
265     Begin
266       Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
267       Read(f, egyTanszek.e);
268     End;
269     {Uf: Eof(f)      => egyTanszek[1..egyTanszek.db]+e=egy egység AND
270      NOT Eof(f) => e=következő nem dolgozó elem}
271     If Eof(f) then
272       Begin
273         Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
274       End;
275       {Uf: egyTanszek[1..egyTanszek.db]=egy egység AND
276        Eof(f)      => e=üres AND
277        NOT Eof(f) => e=következő nem dolgozó elem}
278     End;
279 Procedure Kiir(Var képernyő:TKépernyő, Var egyTanszek:TBeLap);
280   Var i:Integer;
281   Begin
282     If sor+egyTanszek.db>LapHossz then
283       Begin
284         BillreVar; Lapoz;
285       End;
286     For i:=1 to egyTanszek.db do
287       Begin
288         ElemKi({képernyő,}egyTanszek.elemek[i]); {nem használom ki, hogy
289                                                    figyeli a sorok számát,
290                                                    csak, hogy kiírja és a
291                                                    sor-t növeli}
292       End;
293       egyTanszek.db:=0;
294     End;
295 Procedure Kizar(képernyő:TKilap);
296   Begin
297     If sor>0 then BillreVar;
298     ClrScr
299   End;
300 {
301   A megoldás:
302 }
303 Begin{TanszTordListaz}
304   BeNyit(f); Lapoz;
305   While not Eof(f) do
306     Begin
307       Beolvas(f,{egyTanszek});
308       Kiir(képernyő, egyTanszek);
309     End;
310     Kizar;
311   End{TanszTordListaz};
312
313 (* ----- *)
314
315 Procedure OsszFokosListaz(Var f:TFile);
316 (*
317   Virtuális típusok:
318   Input: bemeneti egység vagy egy tanszék összes dolgozója a nyitóval
319          vagy valami más elem
320          a bementi eleme határát csak előreolvasási technikával
321          tudjuk megtalálni

```

```

322         TBeEgyseg=Rekord(db:Egész,
323                         e:TElem [az előreolvasott],
324                         elemek:Tömb(1..MaxTszDolg+6:TElem))
325     Előfeltétel => MaxTszDolg+6=LapHossz
326     (a "+6"-hoz:
327         legrosszabb esetben 3 nyitó rekord +
328         legrosszabb esetben 3 összfoz rekord)
329     BeNyit,BeVége?,Beolvas,Bezár
330     * BeNyit=Megnyitas+belap-puffer inicializálás+előreolvasás
331     * BeVége?=Eof,
332     * Beolvas=egységolvasás előreolvasással,
333     * Bezár=Lezar
334     Output: TKiLap="Rekord" (sor:Egész,képernyő:TKépernyő)
335             =sor:Egész
336     KiNyit,Kiir(belap-puffer!),Kizár
337     * KiNyit=Lapoz,
338     * Kiir=belap-puffer kiírása+sor-állítás, ha még kifér
339             ha nem fér ki, akkor
340             + várakozás
341             + Lapoz
342             + puffer inicializálása
343     * Kizár= várakozás + ClrScr
344 *)
345 {
346     Típus-reprezentációk:
347 }
348 Const UresTElem:TElem=(szintKod:Ures;nev:'';cim:'';tel:'';fiz:0);
349 Type TBeEgyseg=Record
350     db:Byte;
351     e:TElem;{az előreolvasott}
352     elemek:Array [1..LapHossz] of TElem;
353 End;
354 Var egyTanszek:TBeEgyseg;
355     {sor:Byte; <-- összfoz.inc}
356     összFok:Array [TSzint] of TElem;
357 {
358     Típus-implementációk:
359 }
360 Procedure BeNyit(Var f:TFile);
361 Begin
362     Megnyit(f,Olvasas);
363     Read(f,egyTanszek.e); {Előreolvasás:egyetem}
364 End;
365 Procedure Beolvas(Var f:TFile {;Var egyTanszek:TBeLap});
366 {Ef: NOT Eof(f) AND e=előreolvasott vmilyen szintű nyitó elem}
367     Var i:TSzint;
368 Begin
369     egyTanszek.db:=0;
370     {tanszéki dolgozókat megelőző, "felsőbb" szintű nyitó elemek
371     beolvasása, inicializálás:}
372     For i:=egyTanszek.e.szintKod downto Tanszek do
373     Begin
374         Inc(egyTanszek.db);
375         egyTanszek.e.fiz:=0; {nyitó elem fizu-ja 0-zandó}
376         egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
377         összFok[i]:=egyTanszek.e; {összfok[i].fiz:=0;}
378         Read(f,egyTanszek.e); {előreolvasás}
379     End;
380     {Uf: egyTanszek.e=előreolvasott dolgozó}
381     {egy tanszék beolvasása, elkönyvelése:}
382     While Not Eof(f) and (egyTanszek.e.szintKod=Dolgozo) do
383     Begin
384         Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
385         Inc(összfok[Tanszek].fiz,egyTanszek.e.fiz);
386         Read(f,egyTanszek.e);

```

```

387      End;
388      {Uf: Eof(f)      => egyTanszek[1..egyTanszek.db]+e=egy egység AND
389        NOT Eof(f) => e=következő nem dolgozó elem}
390      If Eof(f) then
391      Begin
392        Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=egyTanszek.e;
393        Inc(osszFok[Tanszek].fiz, egyTanszek.e.fiz);
394      End;
395      {Uf: egyTanszek[1..egyTanszek.db]=egy egység AND
396        Eof(f)      => e=üres AND
397        NOT Eof(f) => e=következő nem dolgozó elem}
398      End;
399      Procedure Kinyit{Var képernyő:TKépernyő};
400      Begin
401        egyTanszek.db:=0;
402        Lapoz;
403      End;
404      Procedure Kiir{Var képernyő:TKépernyő, Var egyTanszek:TBelap};
405      Var i:Integer;
406      Begin
407        If sor+egyTanszek.db>LapHossz then
408        Begin
409          BillreVar; Lapoz;
410        End;
411        For i:=1 to egyTanszek.db do
412        Begin
413          ElemKi({képernyő,}egyTanszek.elemek[i]); {nem használom ki, hogy
414                                                    figyeli a sorok számát,
415                                                    csak, hogy kiírja és a
416                                                    sor-t növeli}
417        End;
418        egyTanszek.db:=0;
419      End;
420      Procedure Kizar{képernyő:TKilap};
421      Var i:TSzint;
422      Begin
423        If sor>0 then BillreVar;
424        ClrScr
425      End;
426      {
427      A megoldás:
428      }
429      Var
430        i, ig:TSzint;
431      Begin{OsszFokosListaz}
432        Benyit(f); Kinyit{képernyő};
433        {összfokozatok generálása a beolvasott tanszékhez;}
434        While not Eof(f) do
435        Begin
436          Beolvas(f{, egyTanszek});
437          If Eof(f) then
438          Begin
439            ig:=Egyetem
440          End
441          else
442          Begin
443            ig:=egyTanszek.e.szintKod
444          End{If Eof};
445          For i:=Tanszek to ig do
446          Begin
447            Inc(osszFok[Succ(i)].fiz, osszFok[i].fiz); {összfokozat göngyölítés}
448            {összfokozat hozzávétele;}
449            Inc(egyTanszek.db); egyTanszek.elemek[egyTanszek.db]:=osszFok[i];
450            osszFok[i]:=UrestElem;
451          End;

```

```
452         {Az egység kiírása:}
453         Kiir{képernyő,egyTanszek};
454     End;
455     Lezar(f); Kizar{képernyő};
456 End{OsszFokosListaz};
457
458 Var
459     egy:TFile;
460
461 Begin
462     Inic;
463
464     ClrScr;
465     HighVideo; Writeln(Kozepre('File-létrehozás',80)); NormVideo;
466     Megnyit(egy,Letrehozás);
467     General(egy);
468     Lezar(egy);
469
470     ClrScr;
471     HighVideo; Writeln(Kozepre('Síma listázás',80)); NormVideo;
472     {Sima listázás:}
473     Listaz(egy);
474     (*
475     ClrScr;
476     HighVideo; Writeln(Kozepre('Síma listázás -- másként',80)); NormVideo;
477     {Sima listázás, másként:}
478     ListazMaskent(egy);
479
480     ClrScr;
481     HighVideo; Writeln(Kozepre('Tanszékre tördelt listázás',80)); NormVideo;
482     {TanszékreTördeltListázás:}
483     TanszTordListaz(egy);
484     *)
485     ClrScr;
486     HighVideo; Writeln(Kozepre('Összegfokozatos listázás',80)); NormVideo;
487     {Összfokos listázás:}
488     OsszFokosListaz(egy);
489 End.
```