

Demográfiai modell

0. A modell rövid ismertetése

A modellünkkel megvizsgáljuk, hogy hogyan függ egy populáció **egyedszáma** és **korcsoport-eloszlása** a korcsoportokhoz rendelt numerikus jellemzőktől. A jellemzők a következők: a **születési „hajlandóság”** (1 utód születésének valószínűsége, utódok várható száma, mindezt időegységre vonatkoztatva), **„halandóság”**. Az egyes korcsoportokba az azonos korúakat soroljuk, a kort a modell választott időegységében mérjük.

Az **időegység** a populáció életében a ciklushossz. Egy időegység alatt minden egyeddel 3 féle esemény történhet: **meghalhat**, **szülhet**, egy időegységnyit **öregedhet**. E mellett a populáció egészére „globális” hatások is vonatkozhatnak. Pl. járványok, mesterséges beavatkozások.

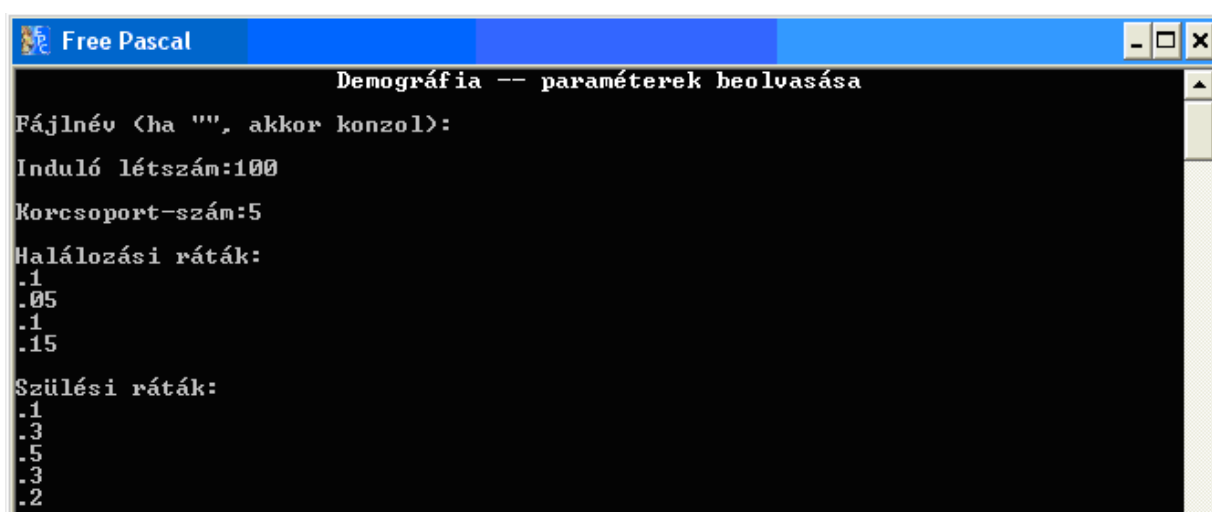
A tapasztalatokhoz igazodva modellünkkel vizsgálni fogjuk azt az esetet is, amelyben a **korcsoportjellemezők** (születési és halálozási ráták) **függenek az összegyedszámtól**. Vegyük majd észre, hogy ezen adatok más paraméterektől is függővé tehetők, pl. –szintén a valós helyzetnek megfelelően– az időtől.

1. Elvárások

A modellnek nagyszámú paramétere lehet, hiszen csak korcsoportonként kétféle biztos szükség lesz. Ezért hasznos eleve úgy tervezni a programunkat, hogy legyen képes akár **klaviatúráról**, akár **fájlból** beolvasni az adatokat.

Mivel az **utolsó korcsoportból** továbblépés nem lehetséges, ezért a hozzá tartozó **halálozási ráta** 1 kell, legyen. Ezt tehát **nem kell beolvasni!**

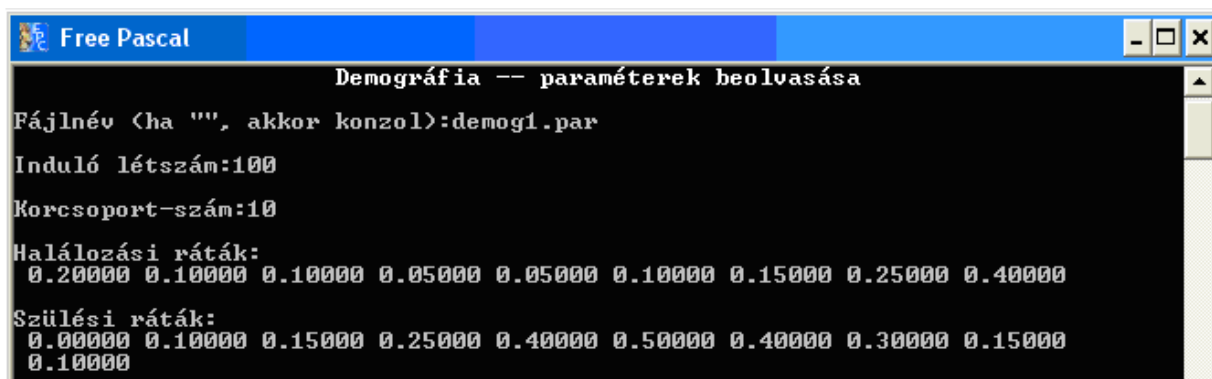
A programunkkal szembeni használati elvárásainkat legjobban az alábbi, futás során keletkezett ábrásor foglalja össze.



```
Free Pascal
Demográfia -- paraméterek beolvasása
Fájlnev <ha "", akkor konzol>:
Induló létszám:100
Korcsoport-szám:5
Halálozási ráták:
.1
.05
.1
.15
Születési ráták:
.1
.3
.5
.3
.2
```

1. ábra. Egy futási kép – „Kézi” paraméterezés, a fájlnev: üres vagy „CON”.

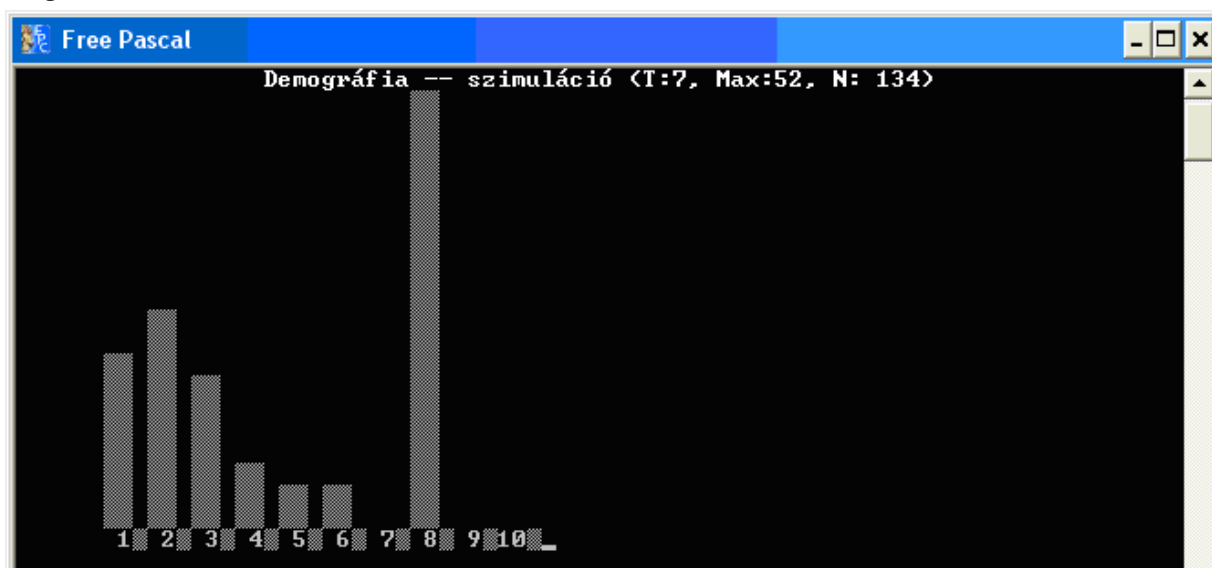
Fájlból történő indításkor a beolvasott paramétereket –ellenőrzési célból– kilistázzuk.



```
Free Pascal
Demográfia -- paraméterek beolvasása
Fájlnev (ha "", akkor konzol):demog1.par
Induló létszám:100
Korcsoport-szám:10
Halálozási ráták:
0.20000 0.10000 0.10000 0.05000 0.05000 0.10000 0.15000 0.25000 0.40000
Szülési ráták:
0.00000 0.10000 0.15000 0.25000 0.40000 0.50000 0.40000 0.30000 0.15000
0.10000
```

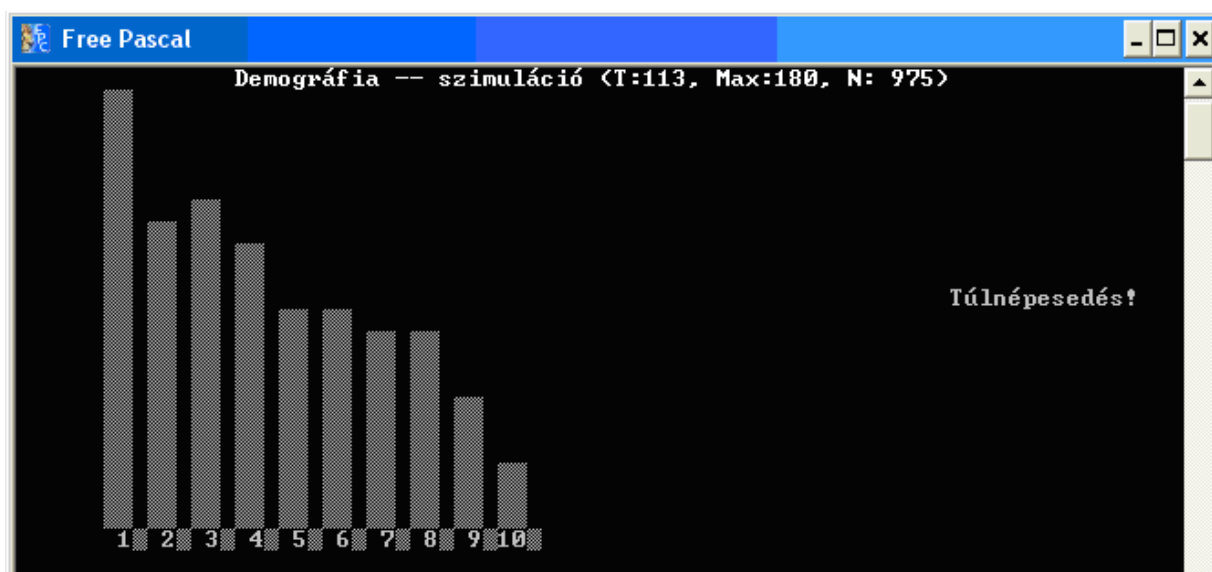
2. ábra. Egy futási kép – Indítás egy paraméterfájl megadásával (demog1.par).

A program érdemi működése során **kiírja** a modell legfontosabb **állapothatározóit**: a populáció méretét (N) és a legnépesebb korcsoport egyedszámát. A teljesebb kép kedvéért kirajzolja a korcsoport-eloszlást is.



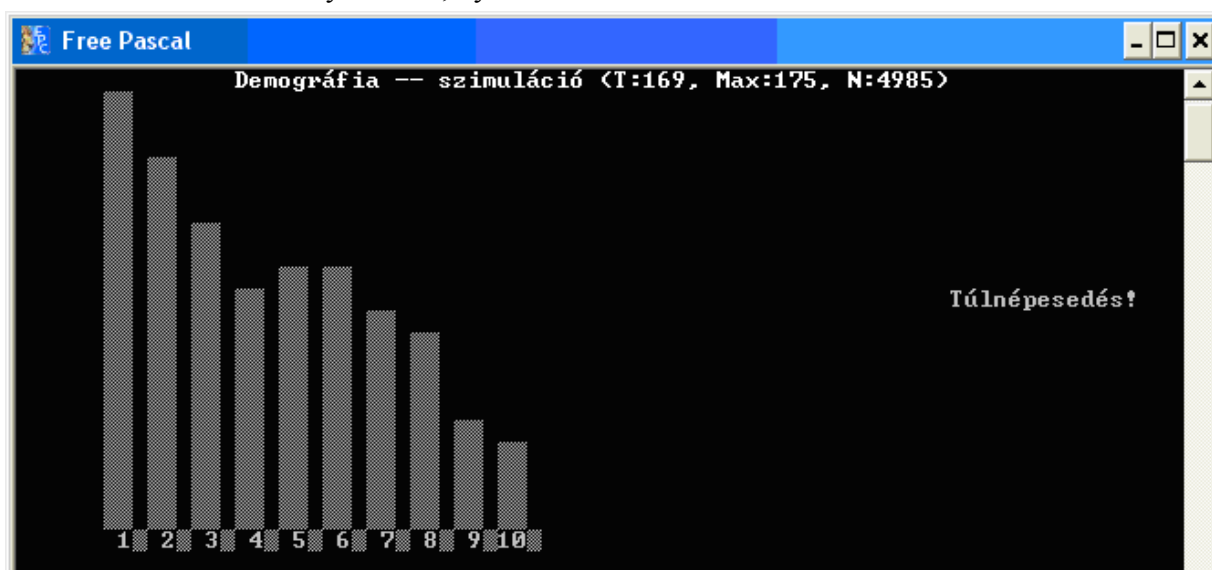
3. ábra. Egy futási kép – Valamikor a kezdetén, távol az egyensúlytól.

A modell működése során beleütközhet memóriakorlátokba. Ez akkor következik be, amikor az egyedeket tároló tömb mérete túlnő a deklarációkor megadott maximumon. Ez a „**túlnépesedés**” jelensége.



4. ábra. Egy futási kép – Egy „túlnépesedési” szituáció. (MaxN=1000 belső paraméter esetén.)

Ekkor azonban a futás folytatódhat; nyilván: némi csalás árán.

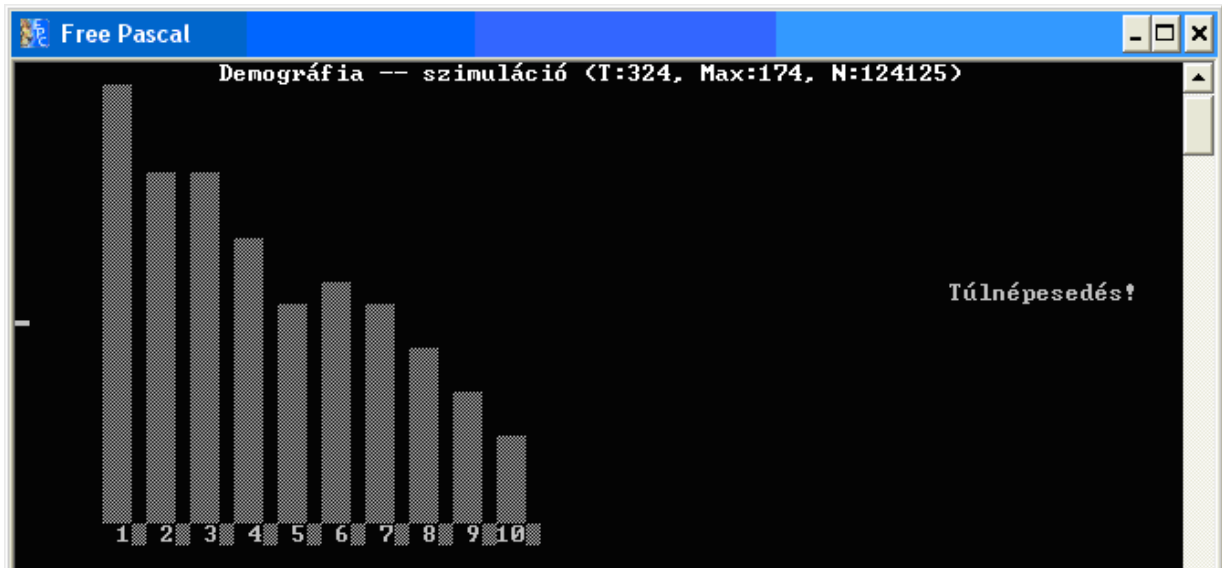


5. ábra. Egy futási kép – Egy újabb „túlnépesedési” szituáció – egyensúly közelében.

Az **egyensúlyi eloszlást** akkor közelíti meg a szimuláció, amikor már keveset változik a korcsoport-eloszlás diagramja. Megfigyelhetjük, hogy még ekkor is az összlétszám folyamatos változásban (általában: növekedésben) van. Érdeemes megtippelni: a változás trendjét! ¹

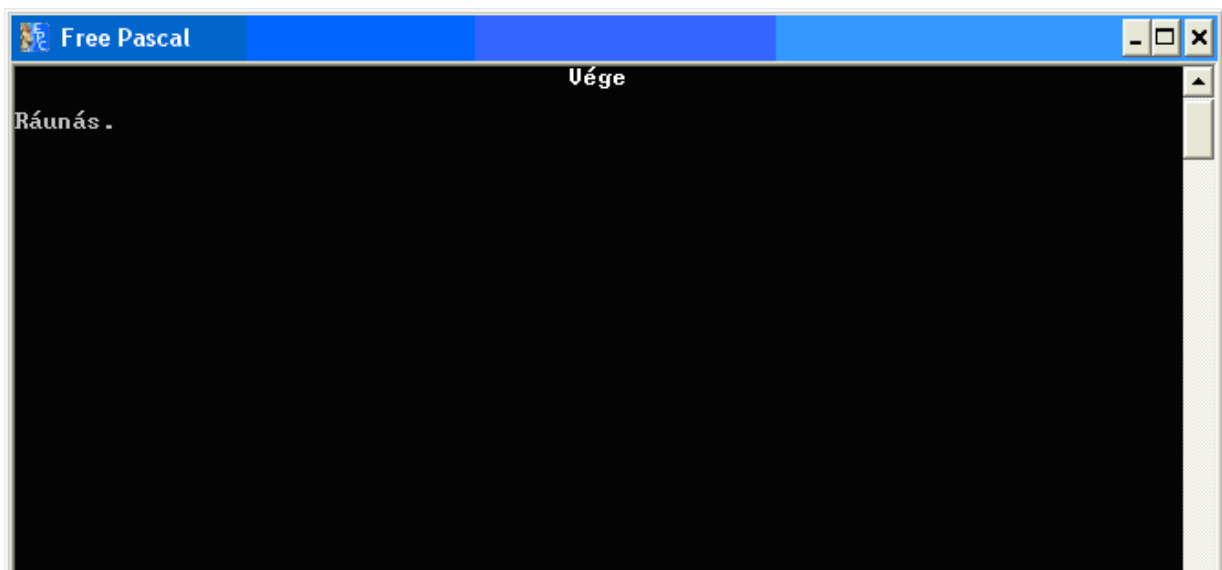
Az egyensúlyhoz közel kerülés érzékelését akár automatizálhatjuk is! E programbővítés a használó szempontjából komoly segítség lehet.

¹ Ellenőrizze pl. egy táblázatkezelővel, hogy exponenciális ez a változási trend. Ehhez épeszü bővítés a használó szempontjából komoly segítség lehet.



6. ábra. Egy futási kép – Egy újabb „túlnépesedési” szituáció – egyensúly közelében.

Egy tipikus végkifejlet a futásban: amikor exponenciálisan növekszik a populáció mérete, és már megítélésünk szerint elértük a (dinamikus) egyensúlyi helyzetet: kilépünk.



7. ábra. Egy futási kép – Egy kilépési ok a program közvetlen megállása előtt.

A továbblépés előtt lássunk egy „in vivo” próbát! ([demogr.exe](#)) Keressünk eltérő végkimenetelű paraméter-együtteseket! Milyen csoportokba sorolhatók a paraméterek, felfedezhetők-e valamilyen szabályszerűségek?

2. A problémák

A kezelés és a látvány, valamint a szimuláció specifikus problémái és megoldásötletei.

2.1. Képernyő

A modell működésének legjellemzőbb állapotátározója a *korcsoport-eloszlás* és az *összlétszám*. Így ezek kelljenek, hogy minden időegységben megjelenjenek a képernyőn. Mivel a korcsoportok száma a gyakorlatban legfeljebb 10-20 közötti, megelégedhetünk a *konzolos* programok szokásos képernyőjével.²

2.2. Billentyűfigyelés

Ha már nem óhajtjuk a szimulációt, kiléphessünk egy billentyű lenyomásával, ill. lehessen időlegesen felfüggeszteni a futást.

2.3. Tájékoztató(k)

Egyszerű szerkezetű text-fájlból, a karakteres képernyőre, lapozottan írandó. Szükséges ismeretek: text-kezelési minimum.

2.4. Szimulációs keret

Alapfeltételezés: az egyes korcsoportok *azonos* hosszúságú időintervallumot ölelnek föl. A szimuláció ebben az időintervallumos egységben dolgozik, azaz egy lépés során a túlélők pontosan egy korcsoporttal jutnak „előbbre”.

Alapelképzelés: minden *egyedet* tárolunk, és a *korcsoportindexével* azonosítunk.

A szimuláció során az *összlétszám* –többnyire– *exponenciálisan* nő (vagy csökken), alig található olyan paraméterezés, amely mellett az *összlétszám* stabil marad. Ez komoly fejtörést okoz. Megoldás ötlete: a gyermekeket nem helyezzük el rögvest a táblában, hanem csak számláljuk. Majd ellenőrizzük, hogy hozzávehető-e az egyed-táblához a „túlcsordulás” veszélye nélkül. Ha nem, akkor *korcsoportlétszám-arányos tömörítést* hajtunk végre.

Modellezés alapadatai:

- T – idő
- N,K – egyedek és korcsoportok száma (*modell-specifikus*)
- egyedek – kora, korcsoport-sorszám (i>modell-specifikus)
- szulRatak, halRatak – születési és halálozási valószínűségek korcsoportonként (*modell-specifikus*)
- kCsElo – korcsoport-eloszlás (*modell-specifikus*)
- felSzoroz – a tárolt és a „feltételezett” össznépeség közötti szorzó (*modell-specifikus*)
- Tulnepesedes – kezelhető-e még a létszám (*modell-specifikus*)

² Megjegyzem: érdemes az eddig használt szimulációs keretprogramunk felhasználásával az igényesebb felületű programot elkészíteni.

A nagyvonalú algoritmus:

```
{inicializálás:}3 Param;  
T:=0;  
{KezdőKépernyő + EredmenyMegjelenites:} KorCsopKi;  
While {not VegeE:} Folytatas and N>0 and not Tulnepesedes do  
Begin  
  SzimulaciosLepes;  
  Inc(T);  
  {EredmenyMegjelenites:} KorCsopKi;  
End;  
{OsszefoglaloMegjelenites:}  
If  
  N=0 then Write(CrLf+'Kihalás.') else if  
  Tulnepesedes then Write(CrLf+'Túlnépesedés.')  
  else Write(CrLf+'Ráúnás.')  
{EndIf};
```

Az eljárások/függvények rövid leírása:

- Param:
Paraméterbeolvasás – a nagyszámú paraméterhez megoldandó az igény szerinti fájlból olvasás (*modell-specifikus*)
Eseménytér inicializálása – kezdő populáció generálása
- KorCsopKi
Aktuális idő kiírása
Képernyőre normálás
Korcsoport gyakoriság oszlopainak kirajzolása
- Folytatas
Billentyűfigyelés –üzemmódtól függően: várakozás–
Végállapot-figyelés (*modell-specifikus*)
- SzimulaciosLepes
A lényeg – modellváltozók módosítása (*modell-specifikus*)
Az eseménytér módosítása (ha kell)
- OsszefoglaloMegjelenites
Konklúzió levonása (*modell-specifikus*)

³ A zárójelk közöttett megjegyzés a korábbi szimulációs (keret) programban „szokásossá” vált tevékenységre utal.

3. A keretprogram

L. [DemogKer.pas](#).

```

1  Program Demografia;
2  (*
3   A demográfiai feladat kerete.
4  *)
5  Uses
6  {$IFDEF FPC -- FreePascal}
7   Crt;
8  {$ELSE -- TurboPascal}
9   Newdelay,Crt;
10 {$ENDIF}
11 {$i AltRutin.inc}
12 Const
13 {$IFDEF FPC -- FreePascal}
14   MaxN=50000;
15 {$ELSE -- TurboPascal}
16   MaxN=5000;
17 {$ENDIF}
18   MaxKorcsop=50;
19   Eps=1.0E-9;
20   bEsc=#27;
21   bSpace=' ';
22   bTeli='█';
23 Type
24   TEgyed=Byte;           {korcsoport-sorszám}
25   TEgyedSzam=LongInt;
26   TValoszinuseg=Real; {0..1}
27
28   TEgyedIndex= 0..MaxN;
29   TEgyedek=Array [TEgyedIndex] of TEgyed;
30   TKorcsopIndex=1..MaxKorcsop;
31
32   TValosKCs=Array [TKorCsopIndex] of TValoszinuseg;
33   TEgeszKCs=Array [TKorCsopIndex] of TEgyedSzam;
34 Var
35   N:TEgyedIndex;
36   K:TKorcsopIndex;
37   egyedek:TEgyedek;
38   szulRatak,
39   halRatak:TValosKCs;
40   kCsElo:TEgeszKCs;
41   T:LongInt;
42   leptek,           {a képernyőre normáláshoz}
43   felSzorz:Real; {a tároltból a tényleges létszám meghatározásához}
44   Tulnepesedes:Boolean;
45
46 Procedure Param;
47 Var
48   i:TEgyedIndex;
49   fNBe:String;
50   fBe:Text;

```

Demografia / 2010.12.13.

```
51 Begin
52   UjLap('Demográfia -- paraméterek beolvasása',0);
53   Write(CrLf,'Fájlnév (ha "", akkor konzol:'); Readln(fNBe);
54   Assign(fBe,fNBe);
55   {$i-}Reset(fBe);{$i+}
56   If (IOResult<>0) or (fNBe='') then
57     Begin
58       fNBe:='CON'; Assign(fBe,fNBe); Reset(fBe);
59     End;
60     {Most nem ellenőrünk, csak bambán olvasunk, hiszen lehet, h. fájlból jön}
61     Write(CrLf,'Induló létszám:'); Readln(fBe,N);
62     If fNBe<>'CON' then Writeln(N);
63     Write(CrLf,'Korcsoport-szám:'); Readln(fBe,K);
64     If fNBe<>'CON' then Writeln(K);
65     Writeln(CrLf,'Halálozási ráták:');
66     For i:=1 to K-1 do
67       Begin
68         Readln(fBe,halRatak[i]);
69         If fNBe<>'CON' then
70           Begin
71             If (i Mod 10)=0 then Writeln;
72             Write(halRatak[i]:8:5);
73           End;
74       End;
75     If fNBe<>'CON' then Writeln;
76     halRatak[K]:=1;
77     Writeln(CrLf,'Szülési ráták:');
78     For i:=1 to K do
79       Begin
80         Readln(fBe,szulRatak[i]);
81         If fNBe<>'CON' then
82           Begin
83             If (i Mod 10)=0 then Writeln;
84             Write(szulRatak[i]:8:5);
85           End;
86       End;
87     If fNBe<>'CON' then
88       Begin
89         Writeln; BillreVar;
90       End;
91     Close(fBe);
92     {Kezdő populáció kitöltés;}
93     For i:=1 to N do
94       Begin
95         egyedek[i]:=1;
96       End;
97     kCsElo[1]:=N;
98     For i:=2 to K do
99       Begin
100        kCsElo[i]:=0;
101      End;
102      {egyéb paraméterek;}
103      leptek:=1.0; felSzorz:=1; Tulnepesedes:=False;
104    End;
105
```


Demografia / 2010.12.13.

```
106 Function Folytatas:Boolean;
107   Var
108     c:Char;
109 Begin
110   If KeyPressed then
111     Begin
112       c:=ReadKey; c:=ReadKey; {újabb billre tovább}
113     End
114     Else
115       Begin
116         c:=bSpace;
117       End;
118     Folytatas:=c<>bEsc
119 End;
120
121 Procedure KorCsopKi;
122   Const
123     MaxSorDb=20;
124   Var
125     i,j,max:Integer;
126     tS,maxS,nfS:String;
127 Begin
128   Str(T,tS);
129   max:=kCsElo[1];
130   For i:=2 to K do
131     Begin
132       If kCsElo[i]>max then max:=kCsElo[i]
133     End;
134     {normálás a képernyőre;}
135     If max>0 then leptek:=MaxSorDb/max;
136     If felSzorz>Eps then
137       Begin
138         Str(N/felSzorz:4:0,nfS)
139       End
140       else
141         Begin
142           nfS:='Túl nagy'; Tulnepesedes:=True;
143         End;
144       Str(max/felSzorz:4:0,maxS);
145       {kirajzolás a képernyőre;}
146       UjLap('Demográfia -- szimuláció (T:'+tS+', Max:'+maxS+', N:'+nfS+')',0);
147       For i:=1 to K do
148         Begin
149           GotoXY(4+i*3,22); Write(i:2,bTeli);
150           For j:=1 to Round(leptek*kCsElo[i]) do
151             Begin
152               GotoXY(4+i*3,22-j); Write(bTeli+bTeli);
153             End;
154           End;
155       Delay(500);
156 End;
157
158 Procedure SzimulaciosLepes;
159   Const
160     szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
```

Demografia / 2010.12.13.

```
161     Var
162     gyDb:TEgyedSzam;
163     i,j:TEgyedIndex;
164
165     Procedure Hullatakaritas(Var N:TEgyedIndex);
166     {hullatakarítás: 0-k kihagyása az egyedek[1..N] tömbből
167       N = tömör egyedek tömb hossza
168       számlálás: kCsElo[1..K] = korcsoportok gyakorisága}
169     Var
170     i,j:TEgyedIndex;
171     Begin
172     (* ide kell a lényeg *)
173     End;
174
175     Procedure AranyosTomorites;
176     {Átlagosan minden x. marad meg, a többi "meghal";
177     felSzorz = felSzorz/x
178     most x=1/(1-szelRata)=5}
179     Var
180     i:TEgyedIndex;
181     Begin
182     GotoXY(65,11); Writeln('Túlnépesedés!'); BillreVar;
183     For i:=1 to N do
184     Begin
185     If (egyedek[i]>0) and (Random<szelRata) then egyedek[i]:=0
186     End;
187     Hullatakaritas(N);
188     felSzorz:=felSzorz*(1-szelRata);
189     End;
190
191     Begin {Szimulacios Lepes}
192     (* ide kell a lényeg *)
193     End;
194
195     Begin
196     UjLap('Demográfia',0);
197     Param; T:=0;
198     KorCsopKi;
199     While Folytatas and (N>0) and not Tulnepesedes do
200     Begin
201     Inc(T);
202     SzimulaciosLepes;
203     KorCsopKi;
204     End;
205     UjLap('Vége',-1);
206     If
207     N=0 then Write(CrLf+'Kihalás.') else if
208     Tulnepesedes then Write(CrLf+'Túlnépesedés.')
209     else Write(CrLf+'Ráunás.')
210     {EndIf};
211     BillreVar;
212     End.
```

Az alapfeladat megoldásához minden szükséges kellék: [Demogr_Gyak.zip](#).

4. A teljes anyag

4.1. A megoldás egy változata

```

Program Demografia;
(*
  A demográfiai feladat megoldása.
*)
...

Var
  N:TEgyedIndex;
  K:TKorcsopIndex;
  egyedek:TEgyedek;
  szulRatak,
  halRatak:TValosKCs;
  kCsElo:TEgeszKCs;
  T:LongInt;
  leptek,           {a képernyőre normáláshoz}
  felSzorz:Real;   {a tároltból a tényleges létszám meghatározásához}
  Tulnepesedes:Boolean;

Procedure Param;
...

Function Folytatas:Boolean;
...

Procedure KorCsopKi;
...

Procedure SzimulaciosLepes;
  Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
  Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

Procedure Hullatakaritas(Var N:TEgyedIndex);
  {hullatakarítás: 0-k kihagyása az egyedek[1..N] tömbből
    N = tömör egyedek tömb hossza
    számlálás: kCsElo[1..K] = korcsoportok gyakorisága}
  Var
    i,j:TEgyedIndex;
Begin
  For j:=1 to K do kCsElo[j]:=0;
  j:=0;
  For i:=1 to N do
  Begin
    If egyedek[i]>0 then
    Begin
      Inc(kCsElo[egyedek[i]]);
      Inc(j); egyedek[j]:=egyedek[i];
    End
  End;

```

```

    N:=j
End;

Procedure AranyosTomorites;
{Átlagosan minden x. marad meg, a többi "meghal";
 felSzorz = felSzorz/x
 most x=1/(1-szelRata)=5}
    Var
        i:TEgyedIndex;
Begin
    GotoXY(65,11); Writeln('Túlnépesedés!'); BillreVar;
    For i:=1 to N do
        Begin
            If (egyedek[i]>0) and (Random<szelRata) then egyedek[i]:=0
        End;
        Hullatakaritas(N);
        felSzorz:=felSzorz*(1-szelRata);
    End;

Begin {SzimulaciosLepes}
    gyDb:=0;
    for i:=1 to N do
        Begin
            If Random<halRatak[egyedek[i]] then {meghalt}
                Begin
                    egyedek[i]:=0;
                End
            Else
                Begin
                    If Random<szulRatak[egyedek[i]] then {szült => gyerekszámolás}
                        Begin
                            Inc(gyDb);
                        End;
                    Inc(egyedek[i]); {öregszik}
                End;
            End;
        {hullatakarítás és számlálás:}
        Hullatakaritas(N);
        While N+gyDb>=MaxN do {túlnépesedés}
            Begin
                AranyosTomorites;
                {a gyerekek szelektálása:}
                gyDB:=Round((1-szelRata)*gyDb);
            End;
            For i:=1 to gyDb do egyedek[N+i]:=1;
            kCsElo[1]:=gyDb;
            N:=N+gyDb;
        End;

Begin
    ...
End.

```

4.2. A második megoldás egy változata

A második változatban a születés naiv megoldását helyettesítjük egy valóság húbellel, a Poisson-eloszlással leírt véletlen gyermekszámmal.

Demografia / 2010.12.13.

```
Program DemografiaP;
(*
  A demográfiai feladat megoldása.
  Poisson-eloszlással.
*)
...

Procedure SzimulaciosLepes;
  Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
  Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

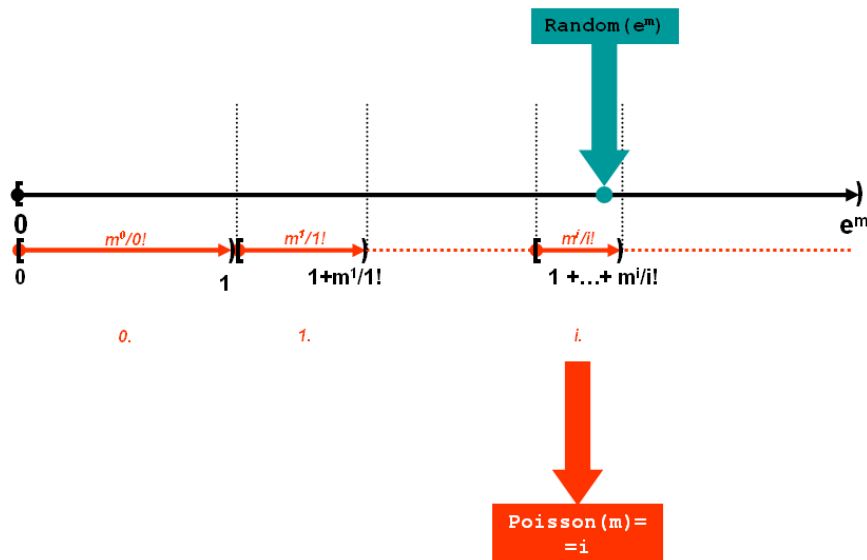
  Procedure Hullatakaritas (Var N:TEgyedIndex);
    ...

  Procedure AranyosTomorites;
    ...

Begin {SzimulaciosLepes}
  gyDb:=0;
  For i:=1 to N do
  Begin
    If Random<halRatak[egyedek[i]] then {meghalt}
    Begin
      egyedek[i]:=0;
    End
    Else
    Begin
      Inc (gyDb, Poisson (szulRatak[egyedek[i]]));
      Inc (egyedek[i]); {öregszik}
    End;
  End;
  {hullatakarítás és számlálás;}
  Hullatakaritas (N);
  While N+gyDb>=MaxN do {túlnépesedés}
  Begin
    AranyosTomorites;
    {a gyerekek szelektálása;}
    gyDB:=Round((1-szelRata)*gyDb);
  End;
  For i:=1 to gyDb do egyedek[N+i]:=1;
  kCsElo[1]:=gyDb;
  N:=N+gyDb;
End;

Begin
...
End.
```

... és a Poisson-eloszlású véletlenszám generálása:



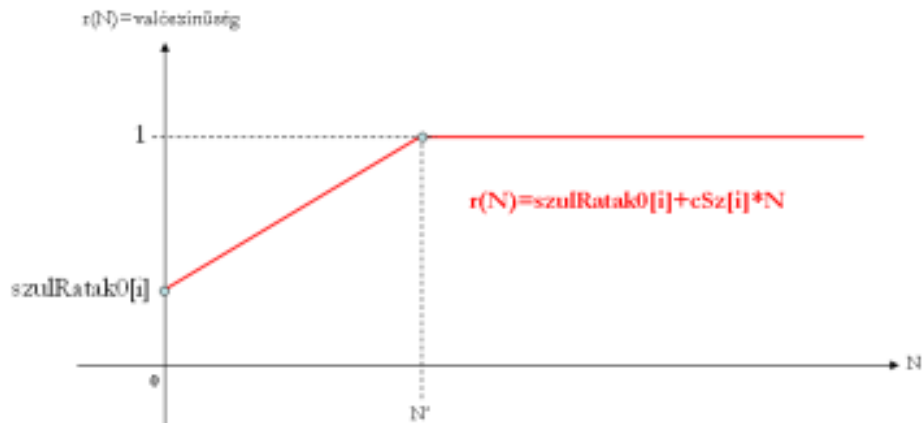
8. ábra. A Poisson-eloszlás „szimulálásának” magyarázata. (L. még: <http://people.inf.elte.hu/szlavi/InfoRendsz/Szimulacio/PoissonAbra.ppt>)

```
{
    Poisson-eloszlású véletlenszámot generáló függvény. Paramétere a
    várhatóértéke (és szórása).
    Értéke: 0..MaxInt
}
```

```
Function Poisson(m: real):Integer;
Var
    i: integer;
    r,t,s: real;
Begin
    r:=random;
    i:=0; t:=exp(-m); s:=t;
    While r>=s do
        Begin
            i:=succ(i); t:=t*m/i; s:=s+t;
        End;
    Poisson:=i;
End;
```

4.3. A harmadik megoldás egy változata

Ebben egyedszám-függővé tesszük a paramétereket. Mind a születési ráták (Poisson-várhatóértékek), mind a halálozás ráták (valószínűségek) az összlétszám lineáris függvényei lesznek.



9. ábra. A születési ráta függése az összlétszámtól.

```

Program DemografiaPF;
(*
  A demográfiai feladat megoldása.
  Poisson-eloszlással.
  Függvényes paraméterrel.
*)
...

Var
  ...
  szulRatak0,cSz,
  halRatak0,cH:TValosKCs;
  ...

Procedure Param;
  Var
    i:TEgyedIndex;
  Begin
    ...
    Writeln(CrLf,'Halálozási ráták (h0 és szorzó):');
    For i:=1 to K-1 do
      Begin
        Readln(fBe,halRatak0[i],cH[i]);
        If fNBe<>'CON' then
          Begin
            If (i Mod 5)=0 then Writeln;
            Write(halRatak0[i]:8:5,cH[i]:8:5);
          End;
        End;
      If fNBe<>'CON' then Writeln;
      halRatak0[K]:=1;
      Writeln(CrLf,'Szülési ráták (r0 és szorzó):');
      For i:=1 to K do
        Begin
          Readln(fBe,szulRatak0[i],cSz[i]);
          If fNBe<>'CON' then
            Begin
              If (i Mod 5)=0 then Writeln;
              Write(szulRatak0[i]:8:5,cSz[i]:8:5);
            End;
        End;
    End;
  End;

```

Demografia / 2010.12.13.

```
End;
If fNBe<>'CON' then Writeln;
If fNBe<>'CON' then BillreVar;
...
End;

Function Folytatas:Boolean;
...

Procedure KorCsopKi;
...

Procedure SzimulaciosLepes;
Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

Function szulRatak(const i:Integer):TVarhatoertek;{Poisson!}
Var
    r:Real;
Begin
    r:=szulRatak0[i]+cSz[i]*N;
    If r<0 then szulRatak:=0 else szulRatak:=r
End;

Function halRatak(const i:Integer):TValoszinuseg;
Var
    r:Real;
Begin
    r:=halRatak0[i]+cH[i]*N;
    If
        r>1 then halRatak:=1 else if
        r<0 then halRatak:=0
        else halRatak:=r
    {EndIf};
End;

Procedure Hullatakaritas(Var N:TEgyedIndex);
...

Procedure AranyosTomorites;
...

Begin {SzimulaciosLepes}
    gyDb:=0;
    For i:=1 to N do
    Begin
        If Random<halRatak(egyedek[i]) then {meghalt}
        Begin
            egyedek[i]:=0;
        End
        Else
        Begin
            Inc(gyDb,Poisson(szulRatak(egyedek[i])));
            Inc(egyedek[i]); {öregszik}
        End
    End
End;
```



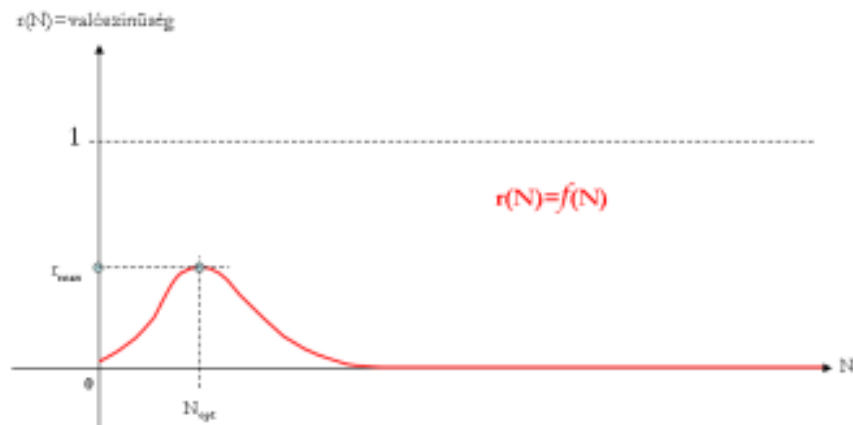
```

End;
End;
{hullatakarítás és számlálás;}
Hullatakaritas(N);
While N+gyDb>=MaxN do {túlnépesedés}
Begin
  AranyosTomorites;
  {a gyerekek szelektálása;}
  gyDB:=Round((1-szelRata)*gyDb);
End;
For i:=1 to gyDb do egyedek[N+i]:=1;
kCsElo[1]:=gyDb;
N:=N+gyDb;
End;
Begin
  ...
End.

```

A rátákra vonatkozó összefüggéssel érdemes játszani. Megfigyelték, hogy a *születési ráta* mind kis, mind nagy egyedszám mellett viszonylag kicsi, s van egy *optimális népesség*, amely mellett *maximális*. A *halálozási rátára* is hasonlóan igaz, hogy az *optimális népesség* mellett a *legkisebb*, s minél nagyobb a tőle való távolság, annál jobban nő.

A modell is arról számol be, hogy ilyen változású ráták stabilizálják a populációt, azaz egy egyensúlyi létszám felé terelik őt.



7. ábra. A születési ráta függése az összlétszámtól.