4 Egymásba ágyazott ciklusok

- egymásba ágyazott ciklusok
- csökkenő ciklusváltozó (**TO** helyett **DOWNTO**)

4.1 Egymásba ágyazott ciklusok

Programjaink készítésekor sokszor elő fog fordulni, hogy nem lesz elég egy ciklust használnunk, hanem vagy egymás után, vagy akár egymáson belül fogunk több ciklust használni. Lássunk most az utóbbi esetre egy példát. Mielőtt azonban belekezdenénk, készítsük egy egyszerűbb programot, melyben még csak egy ciklusunk lesz.

Feladat: Készítsünk programot, amely beolvas egy egész számot, majd kiír a képernyőre egymás mellé ennyi darab * (csillag) karaktert.

Megoldás: A programban deklarálni fogunk egy **n** (integer típusú) változót, melybe beolvassuk a program elején a kiírandó csillagok számát. Ezen kívül szükségünk lesz még egy ciklusváltozóra is - ez legyen i. Magában a programban egy egyszerű write('*') parancsot fogunk megismételtetni **n**-szer egy **for** ciklus segítségével (a ciklus most 1-től n-ig fog menni). Programunk így néz ki:

```
program Pelda14a;
uses crt;
var n,i:integer;
begin
 clrscr;
 write('Kerem a kiirando csillagok szamat: ');
 readln(n);
 for i:=1 to n do write('*');
 readln;
end.
```

Miután lefuttattuk a programot, a következő jelent meg a képernyőn:

```
Kerem a kiirando csillagok szamat: 8
********_
```

Módosítsunk most a fenti programon úgy, hogy ne csak egy sornyi csillagot írjon ki, hanem a csillagok segítségével rajzoljon ki egy négyzetet. Például n=8 esetre a következő jelenjen meg a képernyőn:

```
********
********
********
********
********
********
********
********
```

Tehát ne csak egy sorba írjon ki n drb. csillagot, hanem írjon ki n darab sort, melyek mindegyikében n darab csillag legyen. Ezt úgy érhetjük el, hogy a fenti programban levő ciklus után (amely kiír egy sornyi csillagot) teszünk egy writeln utasítást (ezzel a kiírt sor után egy új sorba kerülünk), majd a sor kiírására szolgáló ciklust az utána következő **writeln** paranccsal együtt (begin..end kulcsszavakkal összekapcsolva) megismételjük **n**-szer (egy külső ciklusban).

Programunk így néz ki:

```
program Pelda14b;
uses crt;
var n,i,j:integer;
begin
 clrscr;
 write('Kerem a kiirando csillagok szamat: ');
 readln(n);
 for j:=1 to n do
  begin
  for i:=1 to n do write('*');
  writeln;
  end;
 readln;
end.
```

A program a következőt fogja kiírni a képernyőre:

```
Kerem a kiirando csillagok szamat: 8
********
********
********
********
********
********
********
********
_
```

Fontos, hogy ha ilyen és ehhez hasonló egymásba ágyazott ciklusokat használunk, akkor a külső ciklusnak egy másik ciklusváltozót kell választanunk, mint a belső ciklusnak. A mi példánkban a külső ciklus ciklusváltozója j, a belső ciklusé pedig i. A két ciklus valójában a következő képen fut le:

- a külső ciklusnál a j kezdeti értéke 1 lesz (j=1),

  - a belső ciklusnál az i kezdeti értéke 1 lesz (i=1), majd végrehajtódik a write('*') parancs,
  - a belső ciklusnál az i értéke növekszik (i=2), majd végrehajtódik a write('*') parancs,
  - ...
  - a belső ciklusnál az i értéke növekszik (i=n), majd végrehajtódik a write('*') parancs,

  - a writeln parancs lefutásával a kurzor egy új sorba kerül,

- a külső ciklusnál a j értéke növekszik (j=2),

  - a belső ciklusnál az i kezdeti értéke 1 lesz (i=1), majd végrehajtódik a write('*') parancs,
  - a belső ciklusnál az i értéke növekszik (i=2), majd végrehajtódik a write('*') parancs,
  - ...
  - a belső ciklusnál az i értéke növekszik (i=n), majd végrehajtódik a write('*') parancs,

  - a writeln parancs lefutásával a kurzor egy új sorba kerül,

- a külső ciklusnál a j értéke növekszik (j=3),

  - a belső ciklusnál az i kezdeti értéke 1 lesz (i=1), majd végrehajtódik a write('*') parancs,
  - a belső ciklusnál az i értéke növekszik (i=2), majd végrehajtódik a write('*') parancs,
  - ...
  - a belső ciklusnál az i értéke növekszik (i=n), majd végrehajtódik a write('*') parancs,

  - a writeln parancs lefutásával a kurzor egy új sorba kerül,
- ...

Ez így folytatódik mindaddig, amíg a külső ciklus ciklusváltozója (j) nem éri el az n értékét, majd ezen belül a belső ciklus ciklusváltozója (i) is nem éri el az n értékét.

Ebből is látszódik, hogy ha a két ciklusban ugyanaz a ciklusváltozó lenne (pl. mindkettőben i), akkor a belső ciklus változtatná a külső ciklus által beállított értéket, ami hibához vezetne.

4.2 Csökkenő ciklusváltozó (TO helyett DOWNTO)

Néha előfordulhat, hogy olyan ciklusra van szükségünk, melyben a ciklusváltozó nem növekszik (pl. 1-től 10-ig), hanem csökken (pl. 10-től 1-ig). Ilyenkor egyszerűen a **to** helyett a **downto**-t fogjuk használni:

```
for ciklusváltozó := kifejezés1 downto kifejezés2 do utasítás ;
```

Ebben az esetben *ciklusváltozó* felveszi először a *kifejezés1* értékét. Végrehajtja az *utasítás*-t, majd a *ciklusváltozó* csökken eggyel és ismét végrehajtja az *utasítás*-t. Ezután ismét csökken eggyel és végrehajtja az *utasítás*-t. Mindezt addig fogja csinálni, amíg a *ciklusváltozó* nem lesz egyenlő a *kifejezés2* értékével. Ekkor még utoljára végrehajtja az utasítást.

Ha a *kifejezés2* értéke nagyobb, mint a *kifejezés1* értéke, akkor az utasítást egyszer sem hajtja végre.

Ha a *kifejezés1* értéke egyenlő a *kifejezés2* értékével, akkor az utasítást csak egyszer hajtja végre.

Feladat: Készítsünk programot, amely bekér egy **N** természetes számot, majd kihagy egy üres sort, és kiírja egymás mellé N-től 0-ig az összes egész számokat (mindegyik szám után egy szóközt rak).

Megoldás: Mivel a számokat csökkenő sorrendben kell kiírnunk, ezért a ciklusban a fent említett downto-t fogjuk használni:

```
program Pelda15;
uses crt;
var i,n:integer;
begin
 clrscr;
 write('Kerem az N szamot: ');
 readln(n);
 writeln;
 for i:=n downto 0 do write(i,' ');
 readln;
end.
```

Lássuk mit írt ki a programunk, ha a program futásakor az N számnak 15-öt írunk be:

```
Kerem az N szamot: 15

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 _
```

Próbáljuk programunkat lefuttatni többször is, mindig az N-nek más számot megadva. Hogy a következő futtatásnál ne a 0 után kezdődjön a kiírás a képernyőre, kiegészíthetjük ezt a programot még úgy, hogy a végére a ciklus után beírunk egy **writeln;** parancsot (esetleg kettőt, ha szeretnénk hogy programunk után egy üres sort is kihagyjon).