5.A FOR ciklus

5.1 A FOR ciklus

Gyakran előfordul, hogy a programunkban valamit többször meg szeretnénk ismételni. Ilyenkor ciklusokat használunk. Ezeknek több fajtájuk van, most a **for** ciklussal fogunk foglalkozni, melynek a következő szerkezete van:

for *ciklusváltozó* **:=** *kifejezés1* **to** *kifejezés2* **do** *utasítás* **;**

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

Ha a *kifejezés2* értéke kisebb, 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.

Példaként készítsünk egy programot, amely 1-től 10-ig kiírja az összes egész számot és mellé a szám négyzetét is. Ehhez szükségünk lesz egy ciklusváltozóra - jelöljük ezt most `i`-vel. Mivel ez a változó egész számértékeket fog felvenni, ezért ezt **integer** típusú változónak deklaráljuk. Ennek a változónak az értéke először 1 lesz, majd 2, 3,... egészen 10-ig. Ezt a folytonos növekedést 1-től 10-ig ciklus segítségével fogjuk megoldani. A ciklusmagban mindig ki fogjuk íratni az `i` változó értékét és hozzá a szám négyzetét is, tehát az `i*i` értékét. Programunk így nézni ki:

```
program negyzetes;
var i:integer;
begin
for i:=1 to 10 do writeln(i,' negyzete = ',i*i);
end.
```

Ez a program a következőt fogja kiírni a képernyőre:

```
1 negyzete = 1
2 negyzete = 4
3 negyzete = 9
4 negyzete = 16
5 negyzete = 25
6 negyzete = 36
7 negyzete = 49
8 negyzete = 64
9 negyzete = 81
10 negyzete = 100
```

A **négyzetgyök** kiszámítására is létezik egy **függvény**, ez pedig az **sqrt()**.

5.2 A szám ill. szöveg kiírása előre megadott hosszúságú helyre

A kiírásnál megfigyelhettük, hogy a program az 1, 2, ... 9 számokat egymás alá írta, de a 10 számnál (mivel ez kétjegyű), a szöveg további része egy hellyel arrébb tolódott. Szöveg kiírásánál van arra is lehetőségünk, hogy egy bizonyos szöveget, számot előre megadott hosszúságú helyre írjunk ki. Tehát például az 1, 2, ... 9 számokat is két helyre írjuk ki, még ha ezek csak egyet foglalnak is el. Ezt a **writeln** parancsban adhatjuk meg a kiírandó szám mögé írva a **:2**-t, ahol a 2 azt jelenti, hogy az előtte levő számot 2 helyre szeretnénk kiírni. Programunk ez után a módosítás után így néz ki:

```
program negyzetes;
var i:integer;
begin
for i:=1 to 10 do writeln(i:2,' negyzete = ',sqr(i));
end.
```

Az egyjegyű számokat is most már két helyre írja ki a program. Mivel most a legnagyobb szám háromjegyű, ezért itt minden számot három helyre íratunk ki:

```
program negyzetes;
var i:integer;
begin
for i:=1 to 10 do writeln(i:2,' negyzete = ',sqr(i):3);
end.
```

ÍA számhoz hasonlóan egy szöveges változónál vagy egy szövegnél is megadhatjuk, hogy azt milyen hosszú helyre akarjuk kiíratni, például így:

```
writeln('Hova kerul ez?':20);
```

Ilyenkor a szám kiírásához hasonlóan a szöveg elé megfelelő számú szóközt tesz ki a program. Néha ügyelnünk kell arra, hogy a szöveges képernyő felbontása 80 x 25, tehát egy sorban 80 karakter fér el. Ha ennél többet írunk ki, akkor a maradékot (ami nem fér ki) már a következő sor elejére fog kerülni.

5.3 A BEGIN ... END; kulcsszavak használata

for *ciklusváltozó* **:=** *kifejezés1* **to** *kifejezés2* **do**

begin
első parancs **;**
második parancs **;**
utolsó parancs **;**
end ;

A **begin**..**end;** parancsot akkor használjuk, ha a cikluson belül egy parancs helyett többet szeretnénk összekapcsolni.