

CIKLUSOK

FOR

készítette: Vastag Attila

2019

A program egyik legfontosabb tulajdonsága, hogy ismétléseket képes végrehajtani.

A **ciklus (iteráció) ismétlést** jelent, ilyenkor egy vagy több utasítás újra és újra végrehajtódik.

Vannak olyan esetek amikor az iterációk számát előre el tudjuk dönteni, de van azonban amikor a futási körülményektől függ, ezért megkülönböztetünk:

- **Számláló ciklus** (*for*)
- **Iterátor** (*foreach*)
- **Előtesztelő ciklust** (*while*)
- **Hátultesztelő ciklust** (*do - while*)

Számláló ciklus

(*for*)

```
for(int i = kezdőérték; i < ismétlések száma; i++)  
{  
    a ciklus teste  
}
```

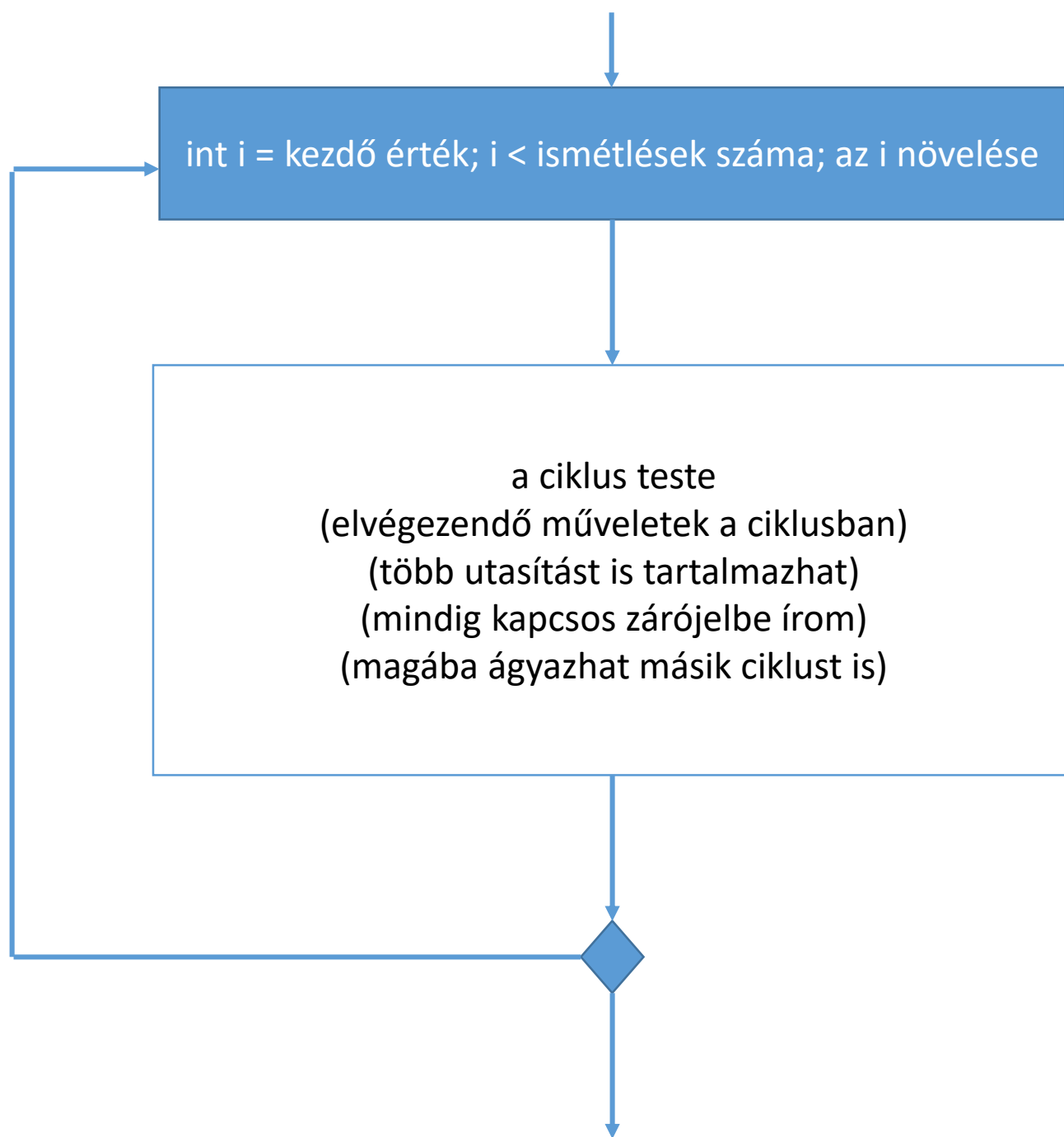
```
for(int i = 0; i < 10; i++)  
{  
    Console.WriteLine(i);  
}
```

A *for* utáni zárójelben találjuk az ún. ciklusfeltételt, ez minden ciklus része lesz, és azt adjuk meg benne, hogy hányszor fusson le a ciklus. A számlálós ciklus feltétele első ránézésre eléggé összetett, de ez ne tévesszen meg minket, valójában nem az. Mindössze három kérdésre kell választ adnunk:

Honnan?

Meddig?

Hogyan?



Vajon mit ír ki a program?

Menjünk sorjában: a **honnan**ra adott válaszban megmondjuk azt, hogy milyen típust használunk a számoláshoz és azt, hogy honnan kezdjük a számolást. Tulajdonképpen ebben a lépésben adjuk meg az ún. **ciklusváltozót**, amelyre a ciklusfeltétel épül. A fenti példában egy *int* típusú ciklusváltozót hoztunk létre a ciklusfeltételen belül és nulla kezdőértéket adtunk neki.

A ciklusváltozó neve konvenció szerint **i** lesz az angol *iterate* – ismétel szóból. Több ciklusváltozó használatakor általában **i, j, k** ... sorrendet követünk.

Mivel a ciklusfeltétel után blokkot nyitunk, azt hinné az ember, hogy a ciklusváltozó a globális lesz a programra nézve, de ez nem fedí a valóságot. A ciklusfeltételen belül deklarált ciklusváltozó lokális lesz a ciklust nézve (a *for* után következő kapcsos zárójelekkel határolt részre).

Épp ezért a következő forráskód nem fordulna le: (mert az *i* változó nem létezik a ciklus blokkján kívül)

```
int main()
{
    for(int i = 0; i < 10; i++)
    {
        Console.WriteLine(i);
    }

    Console.WriteLine(i); //itt a hiba
}
```

„Meddig?”!

Most azt kell megválaszolniunk, hogy a ciklusváltozó milyen értéket vehet fel , ami kielégíti a ciklusfeltételt. Most azt adtuk meg, hogy *i*-nek kisebbnek kell lennie tíznél, vagyis kilenc még jó, de ha *i* ennél nagyobb, akkor a ciklust be kell fejezni.

„Hogyan?”

Adjuk meg a választ, vagyis azt , hogy milyen módon változtatjuk a ciklusváltozó értékét . A leggyakoribb módszer a példában is látható inkrementáló (dekrementáló) operátor használata: *i++* vagy *i--*.

FELADATOK

- 1 – Írjuk ki a számokat 2-től 15-ig növekvő sorrendbe.
- 2 – Írjuk ki a számokat 0-tól -8-ig csökkenő sorrendbe.
- 3 – Írjuk ki növekvő sorrendbe a számokat 3 és 11 között.
- 4 – Írjuk ki csökkenő sorrendbe a számokat -8 és 6 között.
- 5 – Írjuk ki csökkenő sorrendbe a páros számokat -10 és 10 között.
- 6 – Írjuk ki növekvő sorrendbe a páratlan számokat -11 és 11 között. [HF]

7 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd írjuk ki csökkenő sorrendbe a számokat az intervallumból.

8 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd írjuk ki növekvő sorrendbe a páratlan számokat az intervallumból.

9 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd írjuk ki csökkenő sorrendbe a páros számokat az intervallumból.

10 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd adjuk össze e intervallum számait és írjuk ki az összeget.

Önállóan kiegészíteni: szorzat!

11 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd adjuk össze e intervallum páros számait és írjuk ki az összege, a páratlan számok szorzatát.

12 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), számoljuk meg, hogy e intervallumban hány szám osztható hárommal.

13 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd intervallumban ellenőrizzük, hogy a páros vagy páratlan számok összege e a nagyobb.

14 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd intervallumban ellenőrizzük, hogy az öttel vagy a héttel osztható számok összege e a nagyobb.

15 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket), majd az intervallumban ellenőrizzük, hogy hány olyan páratlan szám van a két szám közt mely osztható 3-al.

16 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket) majd határozzuk meg a páros és a páratlan számok összegének az átlagát.

17 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket) majd határozzuk meg az intervallum átlagát.

18 – A felhasználótól kérjünk be egy intervallumot (kezdő és vég értéket). Az első elemet adjuk hozzá az összeghez, a másodikat vonjuk ki, a harmadikat újra adjuk hozzá az összeghez, a negyediket vonjuk ki az összegből ... és így tovább.

Példa:

$osszeg = 0;$

$\{5, 6, 7, 8, 9, 10\}$

$0 + 5 - 6 + 7 - 8 + 9 - 10$

Kérem adja meg hány elemű a számpiramis?

9

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
```

Kérem adja meg hány elemű a számpiramis?

8

```
8 7 6 5 4 3 2 1
7 6 5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
Kérem adja meg hány elemű a számpiramis?  
9  
  
          1  
        1 2 3  
      1 2 3 4 5  
    1 2 3 4 5 6 7  
  1 2 3 4 5 6 7 8  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
```