

Ciklusok - iteráció

A ciklusok olyan programvezérlési szerkezetek, ahol a program bizonyos szakaszt (sorait) többször is végrehajthatjuk. Ezt a szakaszt ciklus magnak nevezzük.

Fajtái:

- előírt lépésszámú ciklusok (számlálós, for)
- elől tesztelős ciklusok (while)
- hátul tesztelős ciklusok (do – while)

Előírt lépésszámú ciklusok (számlálós, for)

Ennél a ciklusnál előre tudjuk, hogy hányszor kell végrehajtani az adott utasítássorozatot.

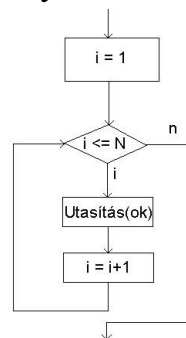
Algoritmusa:

ciklus $i=1$ -től n -ig egyesével
utasítások
ciklus vége

C# - ban:

```
for (int i=1; i<=n;i++)  
{  
    utasítások  
}
```

Folyamatábrája:



i : ciklusváltozó

$\text{int } i=1$: a ciklusváltozó kezdőérték adása, az int elhagyható, ha a ciklusváltozót korábban létrehoztuk.

$i \leq n$: a ciklusban maradás feltétele

$i++$: a léptető utasítás ($i++$ az $i=i+1$ rövidítése)

Elől tesztelős ciklusok

Az elől tesztelős ciklusoknál általában nem tudjuk, hogy a ciklusmag hányszor fut le. A feltételvizsgálat a ciklus elején történik, ezért elképzelhető, hogy a ciklusmag egyszer sem fog lefutni.

Algoritmusa:

Folyamatábrája:

ciklus amíg feltétel
utasítások
ciklus vége

A feltétel egy logikai kifejezés.

Az elől tesztelő ciklusnak lehet ciklusváltozója, de ebben az esetben gondoskodnunk kell a kezdőérték adásáról és a léptetésről.

c# megvalósítása:

```
while (feltétel)
{
    utasítások
}
```

Hátul tesztelő ciklusok

A hátul tesztelő ciklusoknál általában nem tudjuk, hogy a ciklusmag hányszor fut le. A feltételvizsgálat a ciklus végén történik, ezért a ciklusmag egyszer mindenképpen le fog futni.

Algoritmusa:

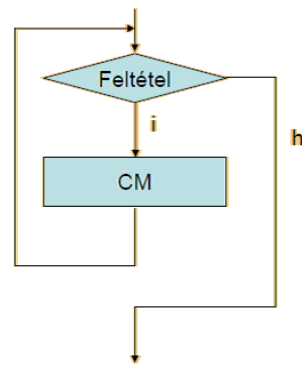
ciklus
 utasítások
amíg feltétel

A feltétel egy logikai kifejezés.

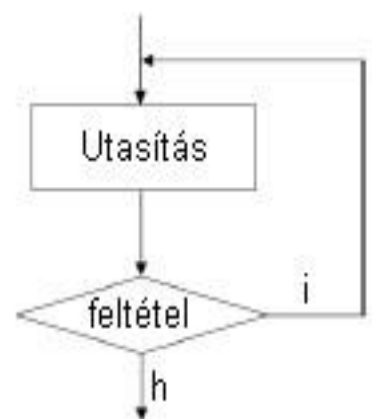
A hátultesztelő ciklusnak lehet ciklusváltozója, de ebben az esetben gondoskodnunk kell a kezdőérték adásáról és a léptetésről.

c# megvalósítása:

```
do
{
    utasítások
} while (feltétel);
```



Folyamatábrája:



foreach ciklus

Végigmegy a tömbön, és az egymást követő tömbelemeket teszi a ciklusmagon belül elérhetővé. Először az első tömbelem értéke lesz a kódblokkon belül elérhető a változóban, majd a második... végül az utolsó. Nem kell ismernünk a tömbelemek számát, és nem fordulhat elő, hogy nem létező tömbelemmel végzünk műveletet.

A leggyorsabb megoldást kínálja ismeretlen méretű tömbök kezelésére.

c# megvalósítása:

```
foreach(var item in collection)
{
    utasítások
}
```