

## 5. Számlálós ciklus

A *ciklus* olyan vezérlési szerkezet, amely az utasítások egy adott csoportját (*ciklusmag*) addig ismétli, amíg egy adott feltétel (*ciklusfeltétel*) teljesül.

A számláló típusú ciklus

**Mondatszerű leírással:**

ciklus  $i=0$ -tól  $n-1$ -ig  $l$ -esével  
    *egy\_utasítás*

ciklus  $i=0$ -tól  $n-1$ -ig  $l$ -esével

*utasítás1*

*utasítás2*

    ...

ciklus vége

**Szintaxis C típusú nyelvekben:**

```
for (int i=0; i<n; i++)  
    egy_utasítás;
```

```
for (int i=0; i<n; i++)  
{  
    utasítás1;  
    utasítás2;  
    ...  
}
```

*Példa: Kettő első 20 hatványainak kiírása*

**Algoritmus:**

**C# kód:**

```
Console.WriteLine("Kettőhatványok");  
ciklus i=0-tól 20-ig 1-esével for (int i = 0; i < 20; i++)  
    {  
    ki:  $i, 2^i$  Console.WriteLine("{0} - {1}", i, Math.Pow(2, i));  
    ciklus vége }  
    }
```

*Példa: Első n páratlan szám összege*

**Algoritmus:**

**C# kód:**

```
Console.WriteLine("Páratlan számok");  
s=0 int s = 0;  
ciklus i=0-tól 20-ig lépésköz: 2 for (int i = 1; i <= 20; i=i+2)  
    s=s+i s=s+i;  
ciklus vége Console.WriteLine("Az összeg: {0}", s);  
ki:s
```

A ciklusmagban egy utasítás van, ezért a { } elhagyható.

A program működése:

    s változó kezdetben 0.

$i=1$  esetén  $s=0+1=1$  lesz;  $i$  2-vel nő

$i=3$  esetén  $s=1+3=4$  lesz;  $i$  2-vel nő

$i=5$  esetén  $s=4+5=9$  lesz...

s értéke a négyzetszámokon meg végig

### **Böhm-Jacopini tétele: (1966)**

Ha a programnak egy kezdete és egy vége van, akkor felépíthető három programozási szerkezet véges sokszori alkalmazásával, ezek:

- szekvencia (egymást követő utasítások)
- szelekció (elágazás, utasítások egy részét csak akkor hajtja végre, ha egy feltétel teljesül)
- iteráció (ciklus, utasítások egy részét addig ismétljük, amíg egy adott feltétel teljesül)

... vagyis további vezérlési szerkezeteket nem kell tanulnunk ☺.

A tétel lényege: más szerkezet pl.: direkt ugrás nem szükséges.

A Böhm-Jacopini tétel bizonyítása a gráfelmélet eszközeivel történik.