

# A ciklus és a konzol felület

## Konzol:

A konzol karakteres felület. Ha konzol alkalmazást készítünk, akkor a kiíró és beolvasó utasítások: `Console.Write()`, `Console.WriteLine()`, `Console.Read()`, `Console.ReadLine()`, `Console.ReadKey()`. A „Line”-os változatok kiírás/beolvasás után új sort kezdenek.

Példa:

```
double a, t;
//Nem kezd új sort, a beolvasás az = jel után kezdődik:
Console.Write("A négyzet oldala=");
//A ReadLine mindig szöveget olvas be ezért konvertálni kell:
a = Convert.ToDouble(Console.ReadLine());
t = a * a;
//A {} helyére a vessző utáni 0, az {1} helyére az 1. változó kerül:
Console.WriteLine("Az {0} oldalú négyzet területe: {1}", a, t);
//Vár egy karakter lenyomására:
Console.ReadKey();
```

## A ciklus fogalma:

Ciklus: A ciklus egy olyan programozási szerkezet, amely egy adott műveletsort (*ciklusmag*), addig hajt végre, amíg egy adott feltétel (*ciklusfeltétel*) teljesül.

## Példa: Adjuk össze a számokat 1-től $n$ -ig

```
be: n
s = 0
ciklus i=1-től n-ig
    s = s + i
ciklus vége
ki: s
```

## Működése:

Legyen  $n$  értéke 3.

A ciklusba való belépés pillanatában  $s=0$ ,  $n=3$

A ciklusmag végrehajtása során először  $i=1$ .

A ciklusmagban  $s$  értéke változik:  $s = 0 + 1 = 1$  lesz.

A ciklusmag következő végrehajtásakor  $i=2$

A ciklusmagban  $s$  értéke változik:  $s = 1 + 2 = 3$  lesz.

A ciklusmag következő végrehajtásakor  $i=3$

A ciklusmagban  $s$  értéke változik:  $s = 3 + 3 = 6$  lesz.

(Mivel  $i$  értéke elérte  $n$  értékét, ami szintén 3, többször nem hajtja végre.)

A kimeneten  $s=6$  jelenik meg

## Megvalósítása C# nyelven konzol felületen:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Összegezés
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Meddig adjuk össze a számokat?");
            int n = Convert.ToInt32(Console.ReadLine());
            int s = 0;
            for (int i = 1; i <= n; i++)
                s += i;
            Console.WriteLine(s.ToString());
            Console.ReadKey();
        }
    }
}
```

### A névtér fogalma:

A családtagokhoz hasonlóan, pl. Baradlay Ödön, Baradlay Richárd, Baradlay Jenő, az utasításokat is „családokba, úgynevezett névterekbe szervezik.

A mintaprogramban szereplő *Console.WriteLine* eljárás például a *System* névtérben van, a pontos neve: *System.Console.WriteLine*.

Ha használt névteret a *using* kulcsszóval a kód elején megadjuk:

```
using System;
```

akkor a névtér megnevezése a kódban elhagyható.

(Ahogy a Baradlay családban az ebédlőasztalnál sem Baradlay Jenőt, hanem egyszerűen Jenőt mondanak.)

A programunkban használt *n* változó pontos neve ugyanígy *Összegezés.n*. Itt a névtér megnevezése azért hagyható el, mert az *Összegezés* névtérben vagyunk:

```
namespace Összegezés
{
```

### A Main eljárás szerepe:

Programunk mindig a *Main* eljárással indul.