Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedés- és Járműirányítási Tanszék

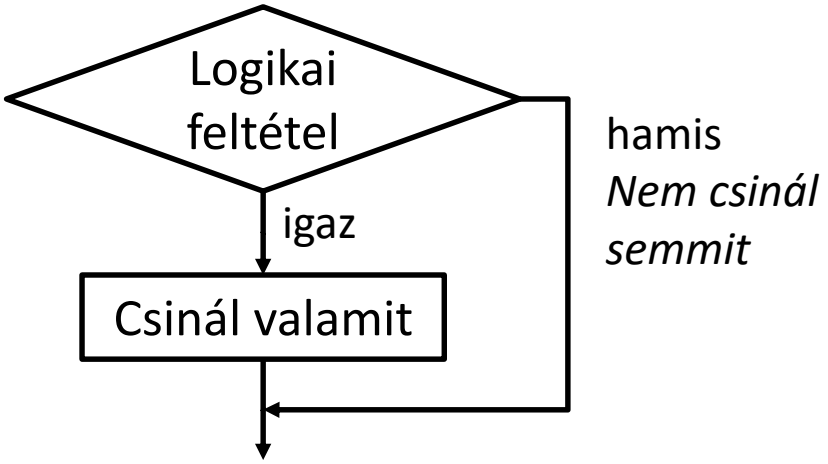
Programozás
BMEKOKAA146

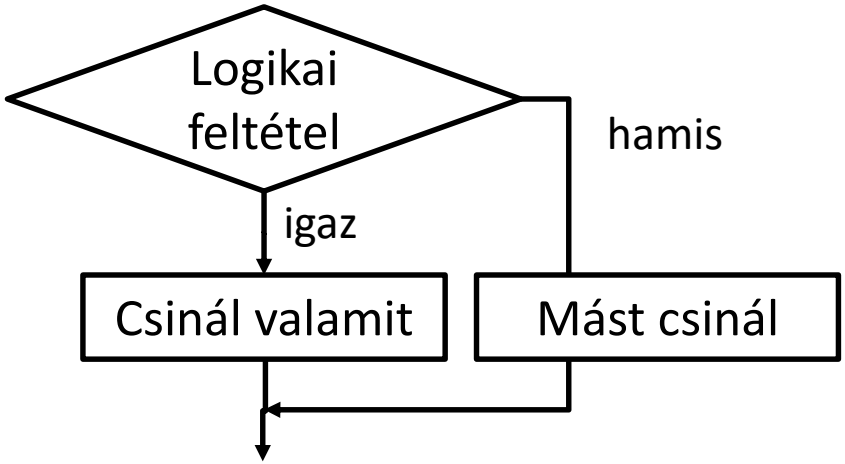
Dr. Bécsi Tamás

3. előadás

Vezérlési szerkezetek
Elágazás

- Gyakran előfordul, hogy meg kell vizsgálnunk egy állítást, és attól függően, hogy igaz vagy hamis, a programnak más-más utasítást kell végrehajtania.





Vezérlési szerkezetek
Elágazás szintaktika


```
if (<log. Kif.>)
<utasítás>
```

```
if (<log.kif.>)
<utasítás>
else
<utasítás>
```

Vezérlési szerkezetek
Elágazás


```
int i=1;
if (i < 10)
   i = 10;
else
   i -= 1;
```

```
int i=1,a;
if (i < 10)
{
    i = 10;
    a = 2;
}
else
{
    i -= 1;
    a = 21;
}
```

Vezérlési szerkezetek
Elágazás összetett példa

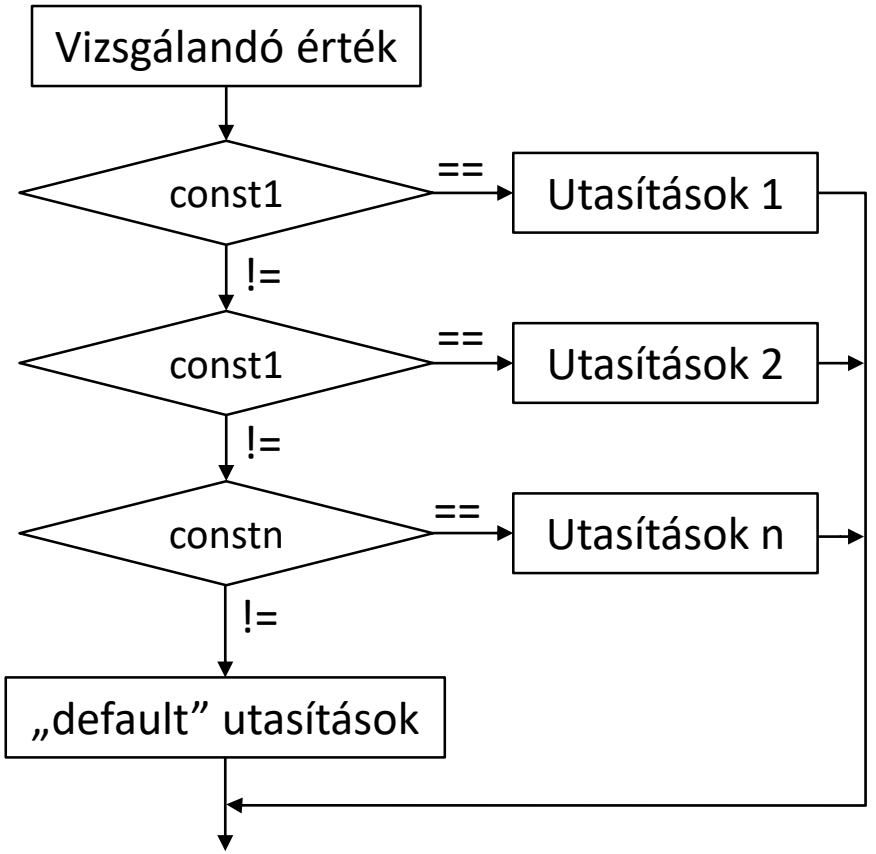

```
int a,b;
bool asiker,bsiker;
string s;
Console.Write("Kérek egy számot:");
s=Console.ReadLine();
asiker = int.TryParse(s, out a);
Console.Write("Kérek még egy számot:");
s = Console.ReadLine();
bsiker = int.TryParse(s, out b);
```

Vezérlési szerkezetek
Elágazás összetett példa 2


```
if (asiker && bsiker)
    if (a > b)
        Console.WriteLine("a nagyobb");
    else if (b>a)
        Console.WriteLine("b nagyobb");
    else Console.WriteLine("a két szám egyenlő");
else
    Console.WriteLine("nem megfelelő a bemenet");
```

Értékkereső elágazás switch

- A feladat ebben az esetben megvizsgálni egy kifejezés értékét, és annak megfelelően különböző utasításokat végrehajtani.
- A switch utasítás egy kifejezést vizsgál, majd több esetre (case) ad vizsgálati lehetőséget.
- Lehetőség van egy ún. default ág definiálására, amely akkor hajtódik végre, amennyiben egyik addigi érték sem egyezett



Értékkereső elágazás
switch

- Minden case labelhez tartozik egy konstans érték
- Ezt követhetnek utasítások
- Az utasításokat minden esetben le kell zárni egy break; paranccsal, így a különböző ágak nem folytatódhatnak egymásban (kivéve üres case label)

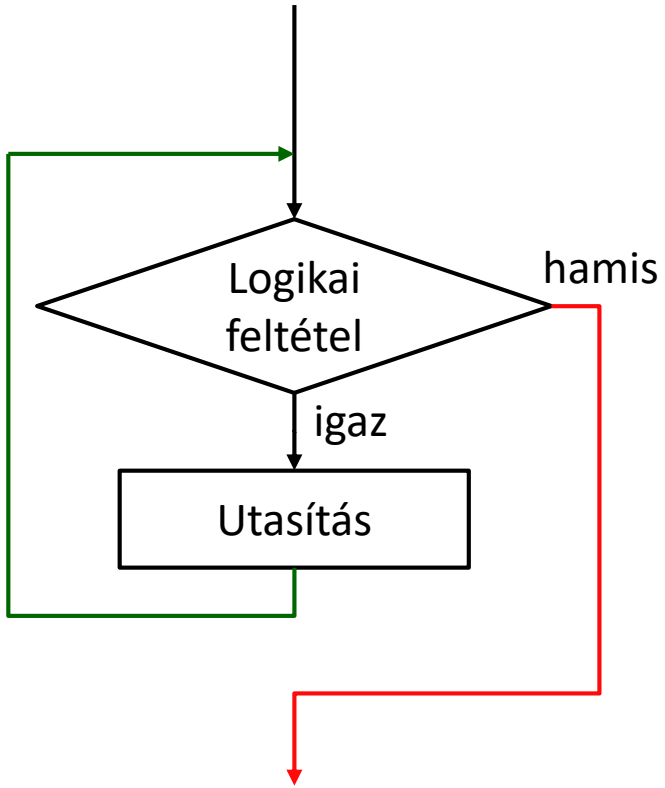
```
int caseSwitch = 1;
switch (caseSwitch)
{
  case 0: case 1:
      Console.WriteLine("Case 0 or 1");
      break;
  case 2:
      Console.WriteLine("Case 2");
      break;
  default:
      Console.WriteLine("Default case");
      break;
}
```

Ciklusok

- Amikor egy adott utasítássorozatot egymás után többször kell végrehajtanunk, akkor ciklust használunk.
- A C# négyféle ciklust biztosít számunkra. (Ebből most hármat tanulunk)

A while ciklus

- Elöl tesztelő ciklus
- Először kielemzi a bennmaradási feltételt, majd annak függvényében végrehajtja az utasítást. Ezt addig folytatja, míg a logikai kifejezés *false* nem lesz.
- A *while* ciklushoz egy utasítás tartozik!




```
while(<logikai kifejezés>)
    utasítás;
```

while példa

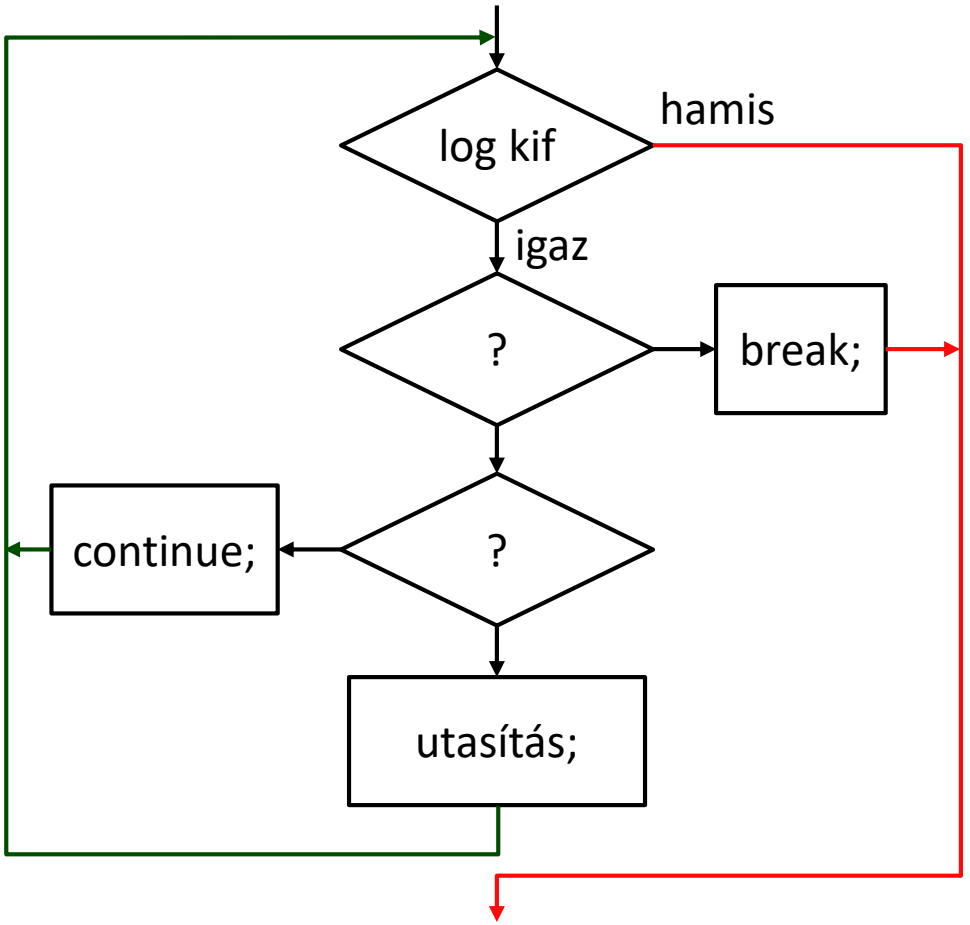

```
int c = 0;
while (++c < 10)
{

    Console.Write("{0} ", c);
}
```

Eredmény: 1 2 3 4 5 6 7 8 9

További ciklusvezérlés break, continue

- A ciklusok működését befolyásolhatjuk a break és a continue utasításokkal
- A *break* azonnal kilép a ciklusból
- A *continue* visszaugrik a ciklus elejére (a bennmaradási feltétel elé)



continue példa


```
int c = 0;
while (++c < 10)
{
      if (c == 3) continue;

      Console.Write("{0} ", c);
}
```

Eredmény: 1 2 4 5 6 7 8 9

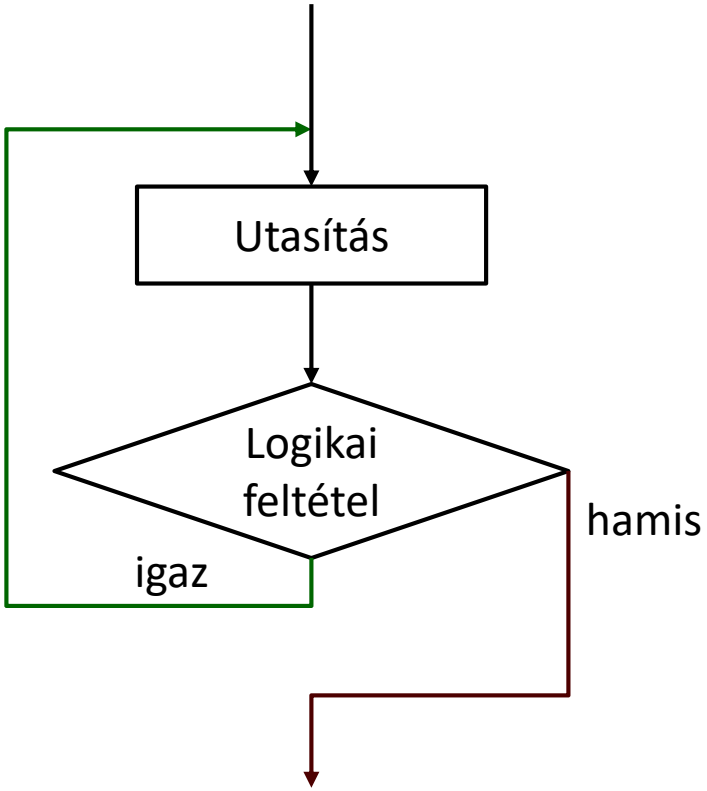
break példa


```
int c = 0;
while (++c < 10)
{
      if (c == 3) continue;
      if (c == 6) break;
      Console.Write("{0} ", c);
}
```

Eredmény: 1 2 4 5

A do-while ciklus

- Hátul tesztelő ciklus
- Végrehajt egy meghatározott utasítás sorozatot,
- majd kielemzi a bennmaradási feltételt, majd annak függvényében végrehajtja az utasítást. Ezt addig folytatja, míg a logikai kifejezés *false* nem lesz.




```
do
 utasítás;
while(<logikai kifejezés>);
```

do while példa


```
int c = 0;
do
{
    Console.Write("{0} ", ++c);
}
while (c < 10);
```

Eredmény: 1 2 3 4 5 6 7 8 9 10

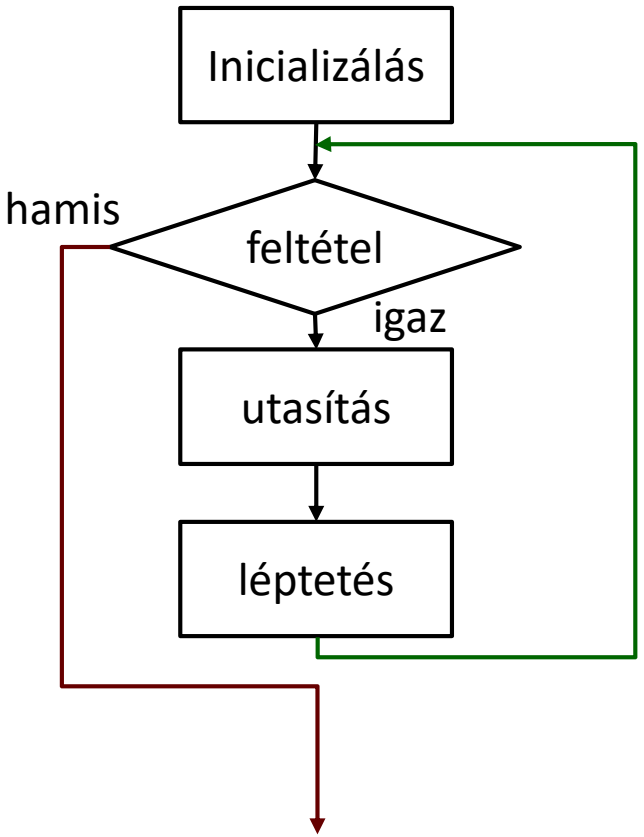
A for ciklus

- A for ciklust jellemzően számlálót alkalmazó ciklusként alkalmazzuk,

```
for (ini ; felt ; lépt)
 utasítás;

ugyanaz, mint:

ini;
while (felt)
{
 utasítás;
 lépt;
}
```



for példa


```
char c;
for (c='a';c<='z';c++)
{
    Console.Write(c);
}
```

Eredmény: abcdefghijklmnopqrstuvwxyz

Összetett példa
Faktoriális


```
int i, szam, fakt;
bool sikeres;
string s;
do
{
    Console.Write("Adj meg egy számot:");
    s = Console.ReadLine();
    sikeres = int.TryParse(s, out szam);
} while (!sikeres);
fakt = 1;
for (i = 2; i<=szam ; i++)
{
    fakt = fakt * i; //fakt*=i;
}
Console.WriteLine("{0}!={1}", szam, fakt);
```

Ciklusok egymásba ágyazása


```
int i,j,k=7;
for (i = 1; i < k ; i++)
{
    for (j = 1; j < k; j++)
        Console.Write("{0,4}", i*j);
    Console.WriteLine();
}
Console.ReadLine();
```

```
Eredmény:
   1   2   3   4   5   6
   2   4   6   8  10  12
   3   6   9  12  15  18
   4   8  12  16  20  24
   5  10  15  20  25  30
   6  12  18  24  30  36
```