Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék

Számítástechnika I.

BMEKOKAA152

BMEKOKAA119

Infokommunikáció I.

BMEKOKAA606

Dr. Bécsi Tamás

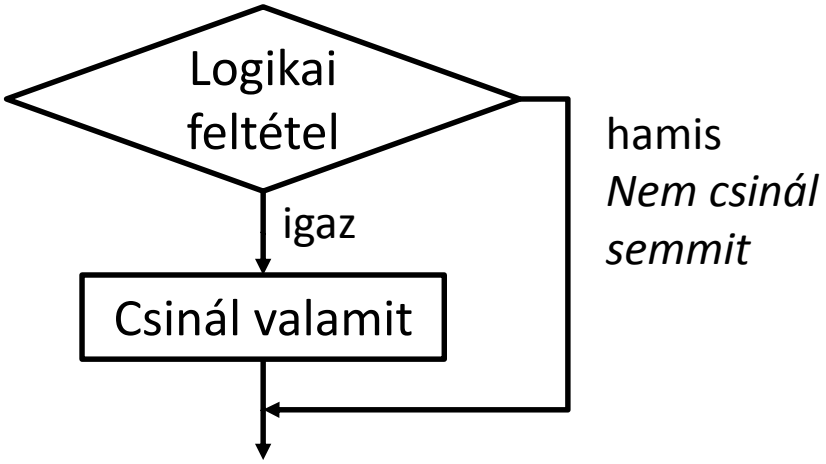
3. előadás

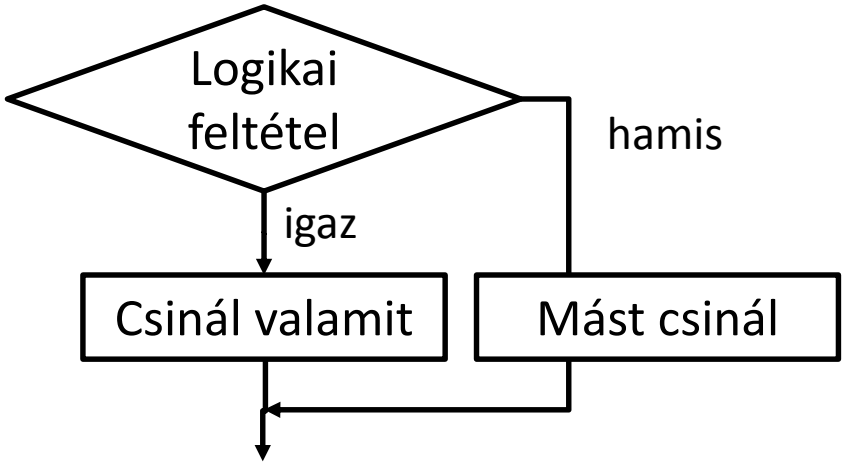
A Math osztály (System.Math)

Metódus	Művelet
Math.Sin(x)	sin(x), ahol az x szög értékét radiánban kell megadni
Math.Cos(x)	cos(x)
Math.Tan(x)	tg(x)
Math.Exp(x)	e^x
Math.Log(x)	ln(x)
Math.Sqrt(x)	x négyzetgyöke
Math.Abs(x)	x abszolút értéke
Math.Round(x)	kerekítés a matematikai szabályok szerint
Math.Ceiling(x)	felfelé kerekítés
Math.Floor(x)	lefelé kerekítés
Math.Pow(x,y)	hatványozás, x^y
Math.PI	a PI konstans (3.14159265358979323846)
Math.E	az e konstans (2.71828182845904523 54)

Logikai Elágazás
if-else

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





Logikai Elágazás
if-else


```
if (<log. Kif.>)
<utasítás>
```

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

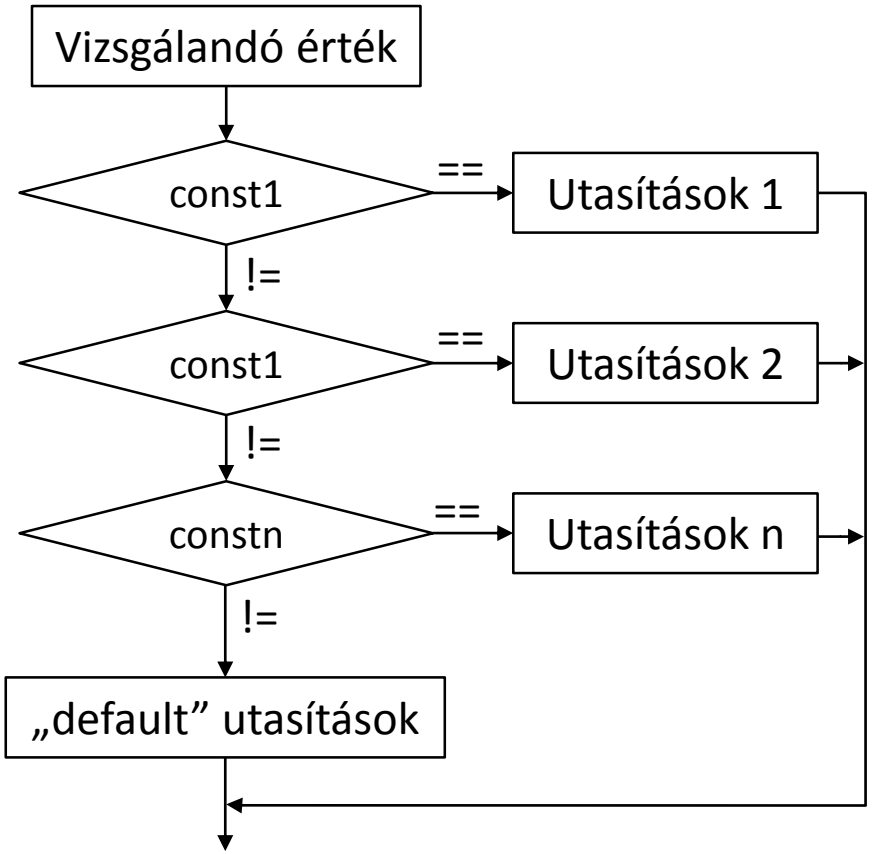
Logikai Elágazás
if-else


```
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;
}
```

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)

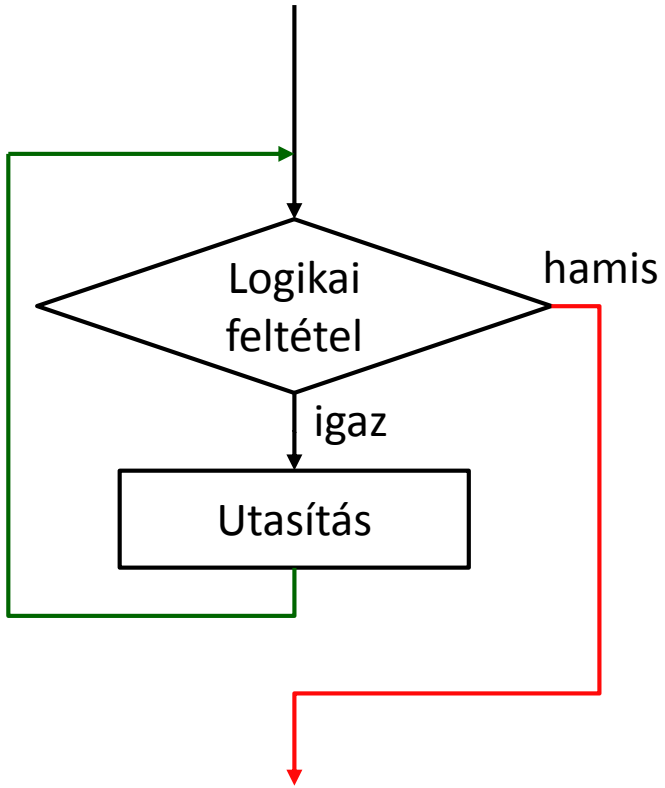
```
int caseSwitch = 1;
switch (caseSwitch)
{
  case 0: case 1:
      Console.WriteLine("Case 0 or 1");
      break;
  case 2:
      Console.WriteLine("Case 2");
      break;
  default:
      Console.WriteLine("Default case");
      break;
}
```

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

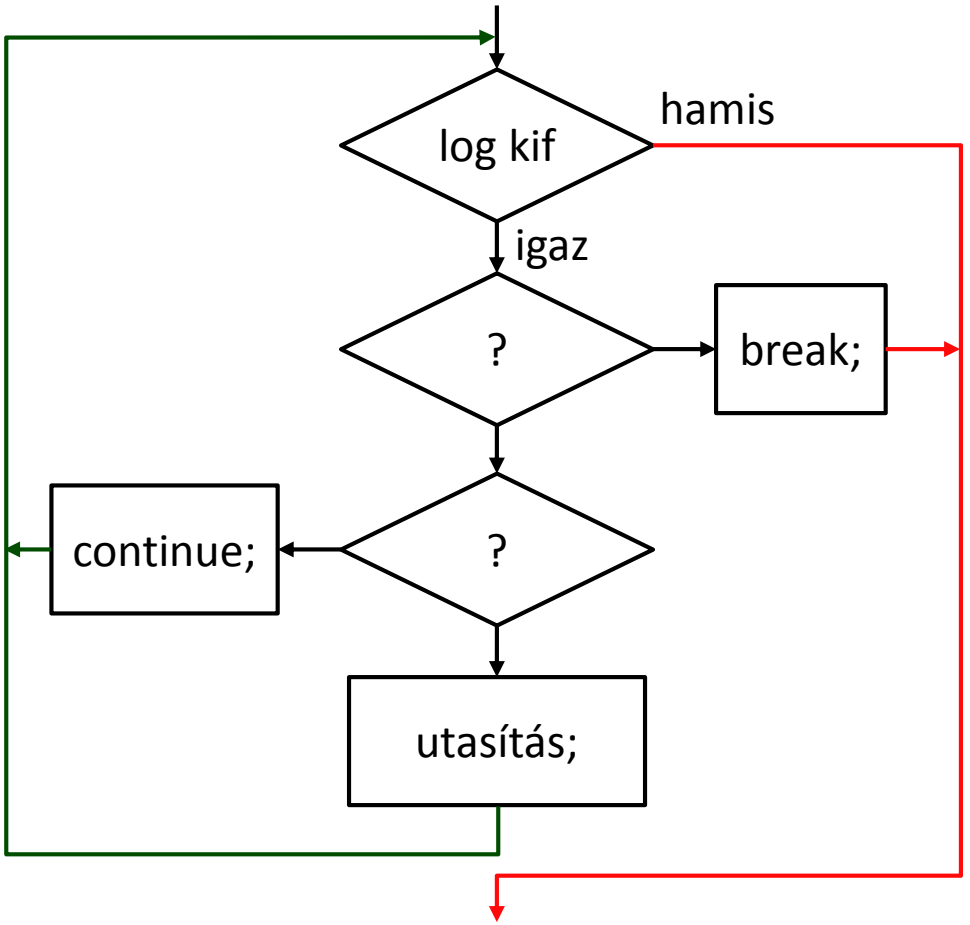
- Elöl tesztelő ciklus
- Először kielemzi a bennmaradási feltételt, majd annak függvényében végrehajtja az utasítást. Ezt addig folytatja, míg a logikai kifejezés *false* nem lesz.
- A *while* ciklushoz egy utasítás tartozik!




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

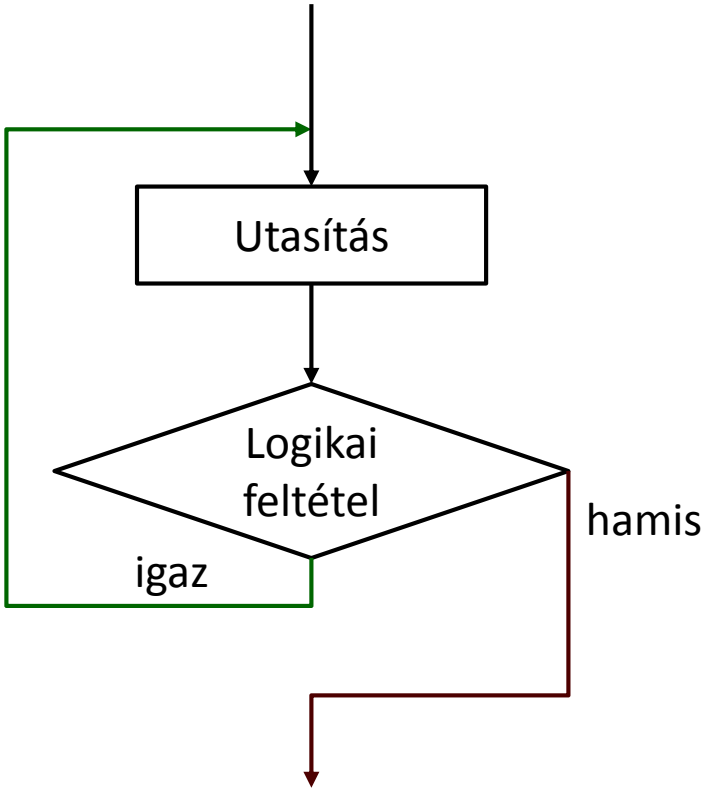
További ciklusvezérlés break, continue

- 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)



A do-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!




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

A for ciklus

- A for ciklust jellemzően számlálót alkalmazó ciklusként alkalmazzuk,

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

ugyanaz, mint:

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