Vezérlési szerkezetek

Tartalomjegyzék

Készítette: Gál Tamás

Elágazás

Ha a feltétel teljesül, akkor az igaz ágon található utasítást a program végrehajtja, egyébként átlépi.

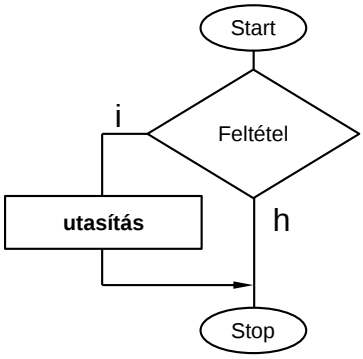
Mondatszerű leírás:

```
ha feltétel akkor utasítás
```

C# és Java kód

```
int a = 6;
int b = 5;

if (a>b) Console.WriteLine("Az a nagyobb: "+a);
```



Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

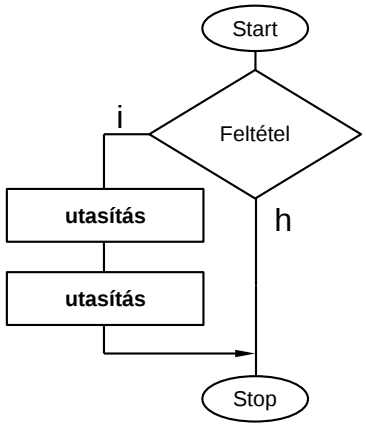
```
program Elagazas;
var a, b: integer;
begin
 a := 6;
 b := 5;
 if (a>b) then writeln('Az a nagyobb: ',a);
end.
```

Elágazás utasításblokkal

Ha a feltétel teljesülésekor több utasítást kívánunk végrehajtani, akkor utasítás zárójeleket használunk. (C alapú és Java nyelveknél '{}' Pascal nyelvben 'begin end')

Mondatszerű leírás:

```
ha feltétel akkor
  utasítás
...
  utasítás
ha vége
```



C# és Java kód

```
int a = 6;
int b = 5;

if (a>b) {
  Console.Write("Az a nagyobb: ");
  Console.WriteLine(a);
}
```

Java esetén:
- Console.Write helyett **System.out.print**
- Console.WriteLine helyett **System.out.println**

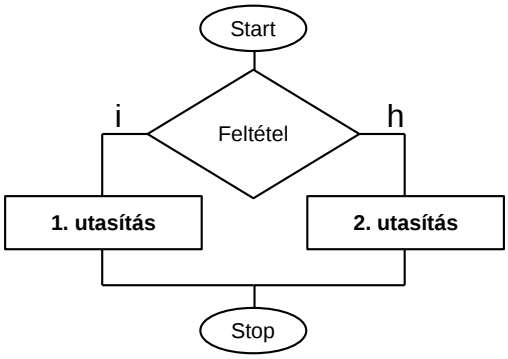
Pascal esetén:

```
program Elagazas;
var a, b: integer;
begin
 a := 6;
 b := 5;
 if (a>b) then begin
   write('Az a nagyobb: ');
   writeln(a);
 end;
end.
```

Elágazás else ággal

Ha a feltétel teljesül, akkor az igaz ágon található utasítást hajtja végre a program, egyébként az else után állót.

Mondatszerű leírás:




```
ha feltétel
  akkor 1. utasítás
  egyébként 2. utasítás
```

C# és Java kód

```
int a = 6;
int b = 5;

if (a>b)
  Console.WriteLine("Az a nagyobb: "+a);
else
  Console.WriteLine("Az a nem nagyobb: "+a);
```

Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

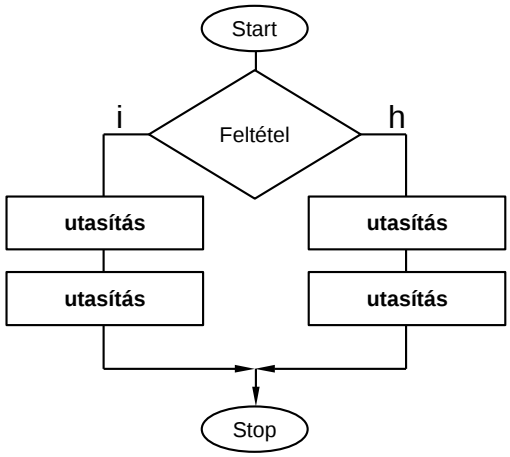
```
program Elagazas;
var a, b: integer;
begin
 a := 6;
 b := 5;
 if (a>b) then writeln('Az a nagyobb: ',a) else writeln('Az a nem nagyobb: ',a);
end.
```

Elágazás else ággal utasításblokkal

Ha a feltétel teljesül, akkor az igaz ágon található utasításokat hajtja végre a program, egyébként az else után állókat. Több utasítás esetén utasítás zárójeleket használunk. (C alapú és Java nyelveknél '{}' Pascal nyelvben 'begin end')

Mondatszerű leírás:

```
ha feltétel akkor
  utasítás
...
  utasítás
egyébként
  utasítás
...
  utasítás
ha vége
```



C# és Java kód

```
int a = 6;
int b = 5;

if (a>b)
{
  Console.Write("Az a nagyobb: ");
  Console.WriteLine(a);
}
else
{
  Console.Write("A b nagyobb vagy egyenlő:");
  Console.WriteLine(b);
}
```

Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

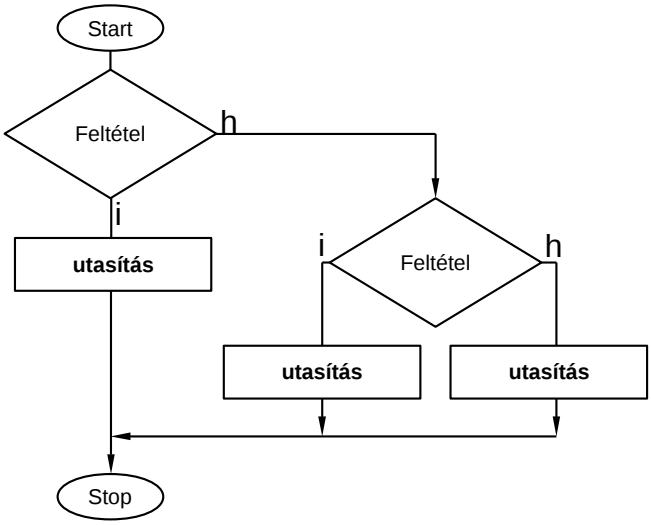
```
program Elagazas;
var a, b: integer;
begin
 a := 6;
 b := 5;
 if (a>b) then begin
   write('Az a nagyobb: ');
   writeln(a);
 end else begin
   write('A b nagyobb vagy egyenlő: ');
   writeln(b)
  end;
end.
```

Egymásba ágyazott elágazások

Egy elágazás akár igaz, akár hamis ágába egyaránt további elágazás illeszthető.

Mondatszerű leírás:

```
ha feltétel akkor
  utasítás
egyébként
  ha feltétel akkor
    utasítás
  egyébként
    utasítás
  ha vége
ha vége
```



C# és Java kód

```
int a = 6;
int b = 5;

if (a==b)
{
  Console.Write("Egyenlő: ");
  Console.WriteLine(a);
}
else if (a>b)
{
  Console.Write("a nagyobb:");
  Console.WriteLine(a);
}
else
{
  Console.Write("b nagyobb:");
  Console.WriteLine(b);
}
```

Java esetén:

- Console.WriteLine helyett **System.out.println**

Pascal esetén:

```
program Elagazas;
var a, b: integer;
begin
 a := 6;
 b := 5;
 if (a=b) then begin
   write('Egyenlő: ');
   writeln(a);
 end else begin
   if a>b then begin
     write('a nagyobb: ');
     writeln(a)
    end else begin
      write('b nagyobb: ');
      writeln(b)
    end;
  end;
end.
```

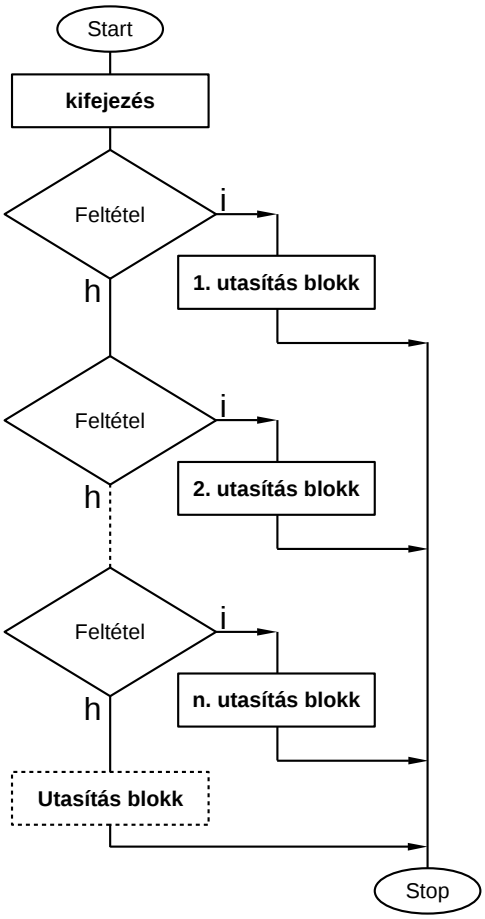
Többirányú elágazás

Többirányú más néven összetett elágazás több végrehajtási ággal rendelkezik, és egyetlen kifejezés kerül kiértékelésre, melynek eredménye alapján kerül az egyes ágakra a vezérlés.

A kiértékelendő kifejezés értéke csak megszámlálható típusú lehet.

Mondatszerű leírás:

```
Többágú elágazás
   feltétel_1 esetén utasítások_1
   ...
   feltétel_n esetén utasítások_n
   egyéb esetén utasítások_m
Elágazás vége
```



C# kód

```
int szam = 3;
switch(szam)
{
  case 1  : Console.WriteLine("Egy");   break;
  case 2  : Console.WriteLine("Kettő"); break;
  case 3  : Console.WriteLine("Három"); break;
  default : Console.WriteLine("Sok");   break;
}
```

*A **C# nyelvben** minden elágazáságat, még a default ágat is, kötelező valamilyen vezérlés átadással befejezni! (break, goto, return...)*

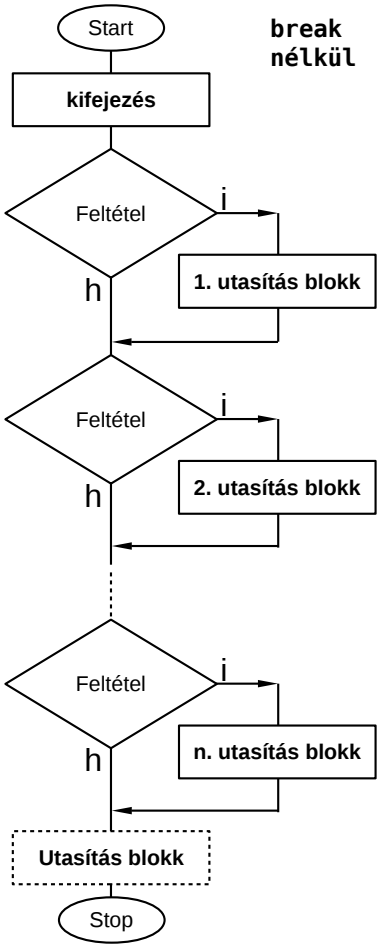
Java kód

```
int szam = 3;

switch(szam)
{
  case 1  : System.out.println("Egy");   break;
  case 2  : System.out.println("Kettő"); break;
  case 3  : System.out.println("Három"); break;
  default : System.out.println("Sok");
}
```

*A **Java nyelvben** nem kötelező minden elágazáságat vezérlés átadással befejezni!*

```
int szam = 1;
switch(szam)
{
  case 1  : System.out.println("Egy");   szam++;
  case 2  : System.out.println("Kettő"); szam++;
  case 3  : System.out.println("Három"); szam++;
  default : System.out.println("Sok");
}
```



Pascal kód

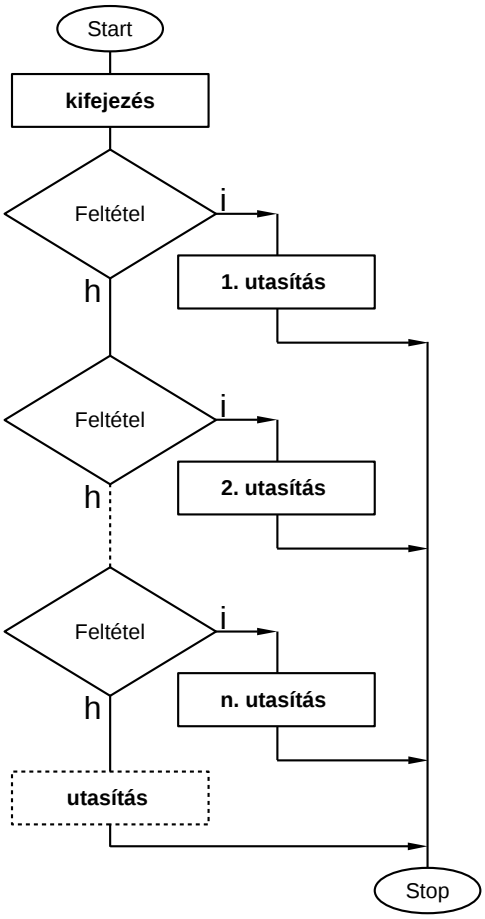
Összetett elágazás Pascal formája:

```
Case kifejezés of
értéklilsta_1: utasítás_1;
...
értéklista_n: utasítás_n;
else utasítás_m;
End;
```

```
program Tobbiranyu;
var i: integer;
begin
 i := 1;

 Case i of
  1: Writeln('Egy');
  2: Writeln('Kettő');
  3: Writeln('Három');
  else Writeln('Sok');
 end;
end.
```

Ha valamely ágon több utasítást kívánunk végrehajtani, akkor Pascal nyelvben kötelező a „Begin end;” páros.



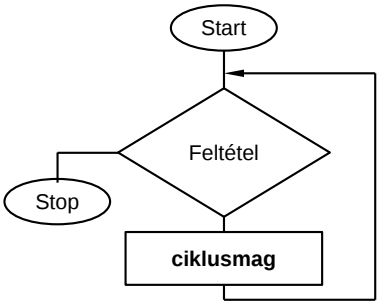
Elöltesztelős ciklus

Az elöltesztelős ciklus először megvizsgálja, hogy a feltétel teljesül-e. Ha igen, akkor végrehajtja a ciklusmagot, majd a folyamat az újból kezdődik. Ha nem, akkor a program a ciklus utáni ponton folytatódik tovább, azaz a ciklusmag kimarad.

Ha a feltétel már első alkalommal sem teljesül, akkor a ciklusmag egyszer sem lesz végrehajtva.

Mondatszerű leírás:

```
ciklus amíg feltétel
  ciklusmag utasításai
ciklus vége
```



Előltesztelős ciklus C# és Java formája:

```
while (feltétel) {
  ciklusmag
}
```

C# és Java kód

```
int i = 1;
while (i<10) {
  Console.WriteLine(i);
  i++;
}
```

Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

Előltesztelős ciklus Pascal formája:

```
while feltétel do begin
  ciklusmag
end;
```

```
program ElolTesztel;
var i: integer;
begin
 i := 1;

 while i<10 do begin
   writeln(i);
   i:=i+1;
 end;
end.
```

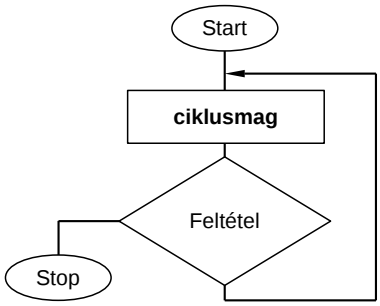
Hátultesztelős ciklus

A hátultesztelős ciklus először végrehajtja a ciklusmagot utána megvizsgálja, hogy a feltétel teljesül-e.
Java és C alapú nyelvek esetén ha a **feltétel igaz, akkor a ciklus végrehajtása folytatódik**, ellenkező esetben pedig leáll.
Pascal nyelv esetén ha a **feltétel hamis, a ciklus végrehajtása folytatódik**, ellenkező esetben pedig leáll.
A hátultesztelős ciklusban **Java és C alapú nyelvek** esetén **a folytatás,** a **Pascal** nyelv esetén a **kilépés feltételét** kell megadni!

A hátultesztelős ciklus ciklusmagja egyszer mindenképpen végrehajtásra kerül.

Mondatszerű leírás:

```
ciklus
  ciklusmag utasításai
amíg feltétel
```



Hátultesztelős ciklus C# és Java formája:

```
do {
  ciklusmag
} while (ismétlés feltétele);
```

C# és Java kód

```
int i = 1;
do {
  Console.WriteLine(i);
  i++;
} while (i<10);
```

Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

Hátultesztelő ciklus Pascal formája:

```
repeat
  ciklusmag
until (kilépés feltétele);
```

```
program ElolTesztel;
var i: integer;
begin
 i := 1;

 repeat
   writeln(i);
   i:=i+1;
 until i=10;
end.
```

Növekményes ciklus

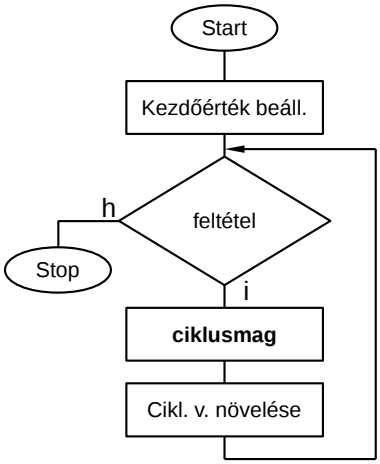
Az ismétlések száma előre ismert. A ciklus a ciklusváltozó előre megadott értékétől, előre megadott értékig fut. A ciklusváltozó egész típusú.

Mondatszerű leírás:

```
ciklus n-től m-ig
  ciklusmag utasításai
ciklus vége

Ha a ciklusváltozó nem egyesével növekszik,
akkor megadjuk a számlálás irányát és lépésközét is.


ciklus n-től m-ig lefelé egyesével
  ciklusmag utasításai
ciklus vége
```



For ciklus C# és Java formája:

```
for (utasítás1; feltétel; utasítás2) {
// Ciklusmag
...
}

utasítás1: ciklusváltozó deklarálása
feltétel: amíg a feltétel igaz addig fut a ciklus
utasítás2:  ciklusváltozó növelése/csökkentése

Mindaddig amíg a feltétel teljesül a feltételvizsgálatot követően megtörténik a
ciklusmag végrehajtása, majd a ciklusváltozó csökkentése és ismét kezdődik előről.
Ha a feltétel nem teljesül, akkor ciklust követő utasítással folytatódik a program
végrehajtása.

---------------------------------------
```

C# és Java kód

```
for (int i=0;i<10;i++)
{
  Console.WriteLine(i);
};

for (int i=9;i>=0;i--)
{
  Console.WriteLine(i);
};

---------------------------------------
```

Java esetén:
- Console.WriteLine helyett **System.out.println**

Pascal esetén:

For ciklus Pascal formája:
for ciklusváltozó := kifejezés1 to kifejezés2 do utasítás ;

kifejezés1: ciklusváltozó kezdőértéke
kifejezés2: ciklusváltozó végértéke
utasítás: az ismételni kívánt utasítás (ciklusmag)

Ha több utasítást szeretnénk ismételtetni, akkor a do után begin és end páros között soroljuk fel őket. (Utasításblokk.)

Működés:
A ciklusváltozó felveszi először a kifejezés1 értékét.
Ha a ciklusváltozó nagyobb mint a végérték, akkor a program végrehajtja az ciklusmagot, majd a ciklusváltozó növekszik eggyel és ismét megtörténik az összehasonlítás. Mindaddig ismétlődik az összehasonlítás, ciklusmag végrehajtása és a ciklusváltozó növelése, amíg a ciklusváltozó értéke nem lesz nagyobb a végértéknél. Ekkor a ciklust követő utasítással folytatódik a program végrehajtása.

```
program ForNovel;
var i: integer;
begin
  for i:=0 to 9 do begin
   writeln(i);
  end;
end.
```

```
program ForNovel;
var i: integer;
begin
  for i:=9 downto 0 do begin
   writeln(i);
  end;
end.
```