

## Elágazás

### Példa: súlyteszt

Felnőtt emberek esetén az elhízás mértékét az ún. testtömegindex segítségével szokták meghatározni. Ha az ember magassága  $m$  méter, tömege pedig  $kg$  kilogramm, akkor bmi testtömegindexét a  $bmi = kg / m^2$  összefüggésből számolhatjuk. Ha ez a szám 25 fölött van, akkor az illető felnőtt túlsúlyos, míg 30 fölött elhízott.

Foglaljuk össze a fenti eljárás algoritmusát!

Be: kg, m

bmi =  $kg / (m * m)$

Ha bmi 25-nél kisebb, akkor Ki: Ön nincs elhízva.

Ha bmi 25 és 30 közé esik, akkor Ki: Ön túlsúlyos.

Ha bmi 30-nál több, akkor Ki: Ön elhízott.

Az *elágazás* olyan vezérlési szerkezet, amely az utasítások egy adott csoportját attól függően hajtja végre, hogy egy adott logikai feltétel teljesül-e.

### Az elágazás megvalósítása

Az elágazás legegyszerűbb változatát a következő algoritmus adja meg:

Ha *logikai feltétel* akkor  
*utasítások*

Elágazás vége

Az *utasítások* rész akkor és csak akkor hajtódik végre, ha a *logikai feltétel* teljesül. A fenti szerkezet egy egységet alkot, ennek egy része csupán az *utasítások* blokk. Az *utasítások* blokk beljebb kezdésével ez külön is hangsúlyozható, és jobban áttekinthető magá a programkód is.



A vasúti elágazásnál a váltó dönt

```
If feltétel Then
    utasítások
End If
```

```
If feltétel Then
    Begin
        utasítások;
    End;
```

Az elágazás legegyszerűbb megvalósítása Visual Basicben, illetve Turbo Pascalban

## Algoritmizálás

Az előző, gyakran kétirányú elágazásnak nevezett megoldás mellett használják az ún. *többirányú elágazást* is. Ekkor *egy kifejezés értékétől függően különböző utasításcsoportok végrehajtására kerülhet sor*. Tipikus alkalmazása, amikor egy kereskedő a megvásárolt árucikkek számától függően különböző mértékű kedvezményt ad. Ezzel az esettel most részletesebben nem foglalkozunk.

### Összetett feltétel megadása. Logikai műveletek

Az elágazás kritikus pontja a logikai feltétel helyes megfogalmazása, főleg összetett állítások esetén. Az **összetett állítások képzésére a programozási nyelvek a matematikai logikában megismert logikai műveleteket használják**: többnyire a NOT (negáció), az OR (diszjunkció) és az AND (konjunkció) kulcsszavakkal.

A több relációt tartalmazó összetett állításokat általában a programozási nyelvekben az állítások konjunkciójára kell bontani. Pl. a *bmi értéke 25 és 30 között van* állítás így adható meg: *bmi* >= 25 AND *bmi* <= 30. A két elemi kifejezést a programozási nyelv szabályaitól függően vagy zárójelbe kell tenni, vagy nem.

```
Option explicit
```

```
Dim kg, m, bmi
```

```
kg = InputBox("Tömege (kg)=")
```

```
m = InputBox("Magassága (m)=")
```

```
bmi = kg/(m*m)
```

```
If bmi<25 Then
```

```
    MsgBox("Ön nincs elhízva!")
```

```
End If
```

```
If bmi>=25 And bmi<=30 Then
```

```
    MsgBox("Ön túlsúlyos!")
```

```
End If
```

```
If bmi>30 Then
```

```
    MsgBox("Ön elhízott!")
```

```
End If
```

```
Program Sulyteszt;
```

```
Var m, kg, bmi:Real;
```

```
Begin
```

```
    Write('Tömege (kg)=');
```

```
    Readln(kg);
```

```
    Write('Magassága (m)=');
```

```
    Readln(m);
```

```
    bmi := kg/(m*m);
```

```
If bmi<25 Then
```

```
    Begin
```

```
        Writeln('Ön nincs elhízva');
```

```
    End;
```

```
If (bmi>=25) And (bmi<=30) Then
```

```
    Begin
```

```
        Writeln('Ön túlsúlyos');
```

```
    End;
```

```
If bmi>30 Then
```

```
    Begin
```

```
        Writeln('Ön elhízott');
```

```
    End;
```

```
    Readln;
```

```
End.
```

A súlyteszt program kódja Visual Basicben és Turbo Pascalban