

Az elágazás megvalósítása

Szemétdíj

Egy településen a szemétdíj a kuka méretétől (liter) és a családtagok számától (fő) függ a következő módon:

Az alapdíj 1000 Ft
 Ha a kuka 100 liternél nagyobb, akkor
 a 100 liter feletti részre további 10 Ft/liter fizetendő
 Ha a családtagok száma 6-nál több, akkor
 a család a fenti összegből 20% kedvezményt kap

A fenti eljárásban többször is szerepel a Ha...akkor szerkezet, ez az Excel HA() függvényének felel meg, a programozásban **elágazásnak** nevezik. A makrónyelvben a kicsit „bőbeszédűbb” If...Then szerkezet felel meg neki:

```
Function Szemétdíj(liter As Integer, fő As Integer)
    Dim fizetendő As Integer
    Dim többlet As Integer
    fizetendő = 1000
    If liter > 100 Then
        többlet = liter - 100
        fizetendő = fizetendő + többlet * 10
    End If
    If fő > 6 Then fizetendő = fizetendő * 0.8
    Szemétdíj = fizetendő
End Function
```

Az If...Then szerkezet kétféleképpen írható be. Ha több utasítás követi, akkor azok külön sorba kerülnek, és az End If kulcsszó zárja a szerkezetet, ha pedig csak egy, akkor az közvetlenül a Then után is írható.

A munkalapfüggvényeknél megismert HA() függvény három paraméteres: a harmadik paraméter azt adja meg, mit kell tenni, ha a feltétel nem teljesül. Ezt a makrónyelvben az Else ág valósítja meg. Például az abszolútérték függvény kódja:

```
Function Abszolút(x As Double)
    If x >= 0 Then
        Abszolút = x
    Else
        Abszolút = -x
    End If
End Function
```

Megjegyzés. A fenti kódrészletekben az elágazás „törzsét” beljebb kezdtük. Ez nem kötelező, de ajánlott: így ugyanis a programkód szerkezete jobban áttekinthető, ez hosszabb programok esetén megkönnyíti a hibakeresést. A továbbiakban mindig követjük ezt a gyakorlatot, és az Olvasónak is ajánljuk.

Ciklus

Gyakran előfordul, hogy egy műveletet sokszor kell egymás után elvégezni. Példánkban egy bank a betéteseinek minden évben annyi százalék kamatot fizet, ahány éve a pénz a bankban van. Például, ha beteszünk 5000 Ft-ot, akkor

az első év végén $5000 \text{ Ft} + 5000 * 1/100 \text{ Ft} = 5050 \text{ Ft}$,

a második év végén $5050 \text{ Ft} + 5050 * 2/100 \text{ Ft} = 5151 \text{ Ft}$,

a harmadik év végén $5151 + 5151 * 3/100 \text{ Ft} = 5305,53 \text{ Ft}$...

lesz a bankszámlánkon. A megoldás során minden lépésben ugyanazt a művelet-sort ismételjük, az ilyen szerkezetet **ciklusnak** nevezik.

A megoldás algoritmus a következő. Ha a p jelöli a betét összegét, i pedig az évet (vagyis értéke rendre 1, 2, 3...) akkor az első n évben pénzünk változása:

```
Ciklus i=1-től n-ig
    p = p + p*i/100
Ciklus vége
```

A fenti algoritmus kódja:

```
Function Kamatos(p As Double, n As Integer)
    Dim i As Integer
    For i = 1 To n
        p = p + p * i / 100
    Next
    Kamatos = p
End Function
```

A példánkban szereplő az ún. *számláló típusú ciklus* a For és Next sorok közötti részben lévő *ciklusmag* utasításait rendre megismétli az i *ciklusváltozó* értékeire 1-től n -ig.

A ciklusmag természetesen összetettebb is lehet. A következő függvény az x egész szám osztóit számolja meg a db nevű változóban. A változó kezdőértéke 0, majd minden megtalált osztónál értéke 1-gyel nő:

```
Function Osztdb(x As Integer)
    Dim db As Integer
    db = 0
    Dim i As Integer
    For i = 1 To x
        If x Mod i = 0 Then db = db + 1
    Next
    Osztdb = db
End Function
```

Megjegyzés. Ciklus létrehozására több lehetőség is van, a For...Next csak egy ezek közül. A következő oldalakon megismerkedünk egy másik, tartományok bejárása esetén kényelmesebb megoldással.