

# Egy lehetséges megoldás az algoritmizálás oktatására a kerettanterv óraszámában: Visual Basic

## Miért?

### Az algoritmizálás oktatása a közoktatásban

A ma már középkorú számítástechnika szakos kollégák még minden bizonnyal emlékeznek a hőskorszakra, amikor Commodore, Spectrum, ABC-80, stb. számítógépeken, szakkörökön tanítottuk a számítástechnikát. Ez akkoriban lényegében azonos volt a programozás oktatásával, az alkalmazói programoknak legfeljebb kezdeményei léteztek ezekre az egyszerű eszközökre.

A PC-k megjelenésével ugyan megjelent a Turbo Pascal és más, akkor modernnek tekinthető fejlesztői környezet, azonban a piaci igény (itt gondoljunk arra is, hogy a legtöbb tanár esti tanfolyamokkal egészítette ki a fizetését, tehát maga is közvetlen piaci szereplővé vált) hamarosan az iskolák zömében általánossá tette az alkalmazói programok oktatását.

Ennek ellenére egyértelmű, hogy aki nem ismeri a programozás alapjait, az nem tudja igazából megérteni a számítógép működését s így birtokolni számítógépes tudását. Ez a ma már hivatalosan is létező informatika tantárgy óratervében is szerepel: a középiskolai kerettanterv szerint a 9. évfolyamon a 74 órából kb. 5-10 óra jut rá, míg az emelt szintű érettségien 45 pontot kaphat érte a tanuló a 120-ból, ha a két rokon területet (SQL, HTML) nem számítjuk.

### Az algoritmizálás oktatás szerepe

Milyen érvek szolgálnak az algoritmizálás oktatás mellett? Először is algoritmizálást vagy programozást tanítsunk-e? Vagyis sarkítva pusztán elméleti hátteret, vagy olyat, ami kézzelfogható eredményre vezet?

Szerintem a programozás oktatásának célja a közoktatásban elsősorban ugyan elméleti, de olyan gyakorlati célokat szolgál mint: Hogyan működik (szoftver oldalról) a számítógép? Melyek a felhasználás határai? Mik az operációs rendszer felületének sajátosságai? Hogyan is készülnek a felhasználói programok? Mit és miért csinálnak a programozók?

Világos tehát, hogy lehetőleg nyelvfüggetlenül kell dolgoznunk, ám csak elméletben programozni nem lehet. Ha a fenti kérdésekre választ akarunk adni a tanulónak, akkor eszközként olyan nyelvet kell választanunk, amelyet a valóságban a programozók ténylegesen használnak, ám alkalmas kezdőnyelvnek is.

### Egy könnyen kezelhető nyelv

Természetesen több reális választási lehetőségünk van. Nem kívánok érveket és ellenérveket a Pascal vagy a C mellett és ellen felsorakoztatni, hiszen a klasszikus vicc szerint az igazi programozó Fortranban dolgozik ☺.

Egy reális alternatívának tartom középiskolában Visual Basicet, ezen belül kerettanterv szerinti óraszámában a 9. évfolyamon ennek VB Script verzióját.

*Érvek:*

1. A kerettanterv szerinti 5-10 órában komoly környezetet elkezdni nem érdemes, a 14 éves gyerekek meg már túl vannak a Logo szintjén.
2. Nem igényel előkészületeket (pl. telepítés), hiszen a Windows 98 óta minden Windows rendszerbe be van építve.
3. Minimális a fejlesztői környezete, elég hozzá a Jegyzettömb ismerete.

4. Bemutathatók az alapvető struktúrák: szekvencia, elágazás, ciklus.

5. Múdunk van tehát az algoritmus és a program fogalmának, fejlesztési lépéseinek bemutatására.

6. Isméri az objektumokat is, bár ez a 9. évfolyam 5-10 órájába biztosan nem fér bele.

7. Megalapozza a továbblépést a VB.net felé, így az átadott tudást az érdeklődő tanulók azonnal kamatoztathatják, a többiek pedig megkapják azt a „feelinget”, hogy igazi programozást tanultak. Persze azt se felejtjük el, hogy a VB.net-től már tényleg csak egy kicsi lépés a profik egyik új fejlesztőeszköze a C#...

*Ellenérvek:*

1. Sokunkba sulykolták annak idején tanulmányaink során a Basic lenézését. Nos azt javaslom, aki ezt ma is hiszi, egyszer üljön le a VB.net elé és próbálja ki.

2. A VB Script valóban kissé mostohán kezeli a típusokat. De ha 5-10 órában precíz oktatást akarunk tartani, akkor a diáknak legfeljebb a kedve megy el tőle.

3. A VB Script interpreter, és nagyobb lélegzetű programok készítésére nem alkalmas. Ha több időnk van, érdemes mielőbb a VB újabb verzióira (VB6 vagy VB.net) továbblépni.

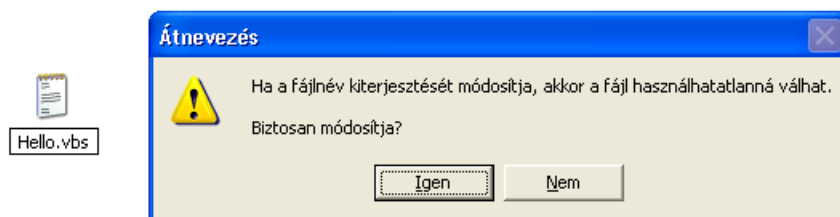
## Hogyan?

A továbbiakban néhány mintapéldát mutatok be a VB Scriptek oktatásához. Ha akarjuk, minden következő alfejezet egy-egy tanítási óra anyaga. Ezt a gyakorlatban magam is kipróbáltam, s ha programozók nem is lettek a gyerekek, azt azért megértették, hogy mi a programozás. Néhányan közülük persze azóta szerver oldali webes alkalmazásokat fejlesztenek .net környezetben, de ez nem az én érdemem.

Mivel kezdjük? Világos, hogy egy kevés elmélettel: mi a program, hogyan működik, stb. Ez 10 percen összefoglalható, több elméletet meg a tanuló nem bír el. Beszéljünk-e az algoritmusok lényegéről ezen a szinten? (Hogyan megyünk át a zebrán? Hogyan telefonálunk?) Szerintem bevezetőként nem, mert a matematika tanár feladatmegoldásnál, a magyar tanár mondatelemzésnél, a fizika és kémia tanár a kísérletek bemutatásánál már ennél magasabb szinten megtette. A számítógépes algoritmus meg jön elő menetközben.

## Hello World!

Tehát kattintsunk az asztalra egyet a jobb gombbal, majd a helyi menüből válasszuk ki az *Új* majd *Szöveges dokumentum* lehetőséget. (Ugye nem felejtettük el még az operációs rendszer ismereteknél bekapcsolni, hogy látszódjának a kiterjesztések is?)



Aztán nevezzük át *Hello.vbs*-re. (Persze, hogy komolyan gondoltuk.) Ezzel a program kész van, ha rákattintunk kettőt, el is indul. Ha szerencsénk van, valaki úgymint megkérdezi: „Tanár Úr! Hiába kattintok rá kettőt, nem történik semmi!” Ennél nagyobb sikerünk nem is lehet, hiszen *Mondtuk neki, hogy mit csináljon?*

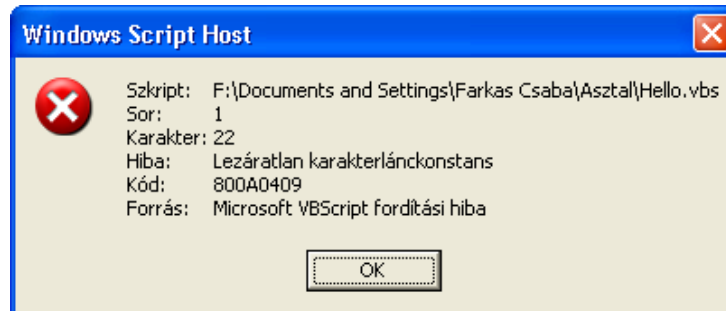
Tehát jobb gomb és *Szerkesztés*. Bejön egy ismerős fejlesztői környezet, a Jegyzettömb.

```
MsgBox("Hello World!")
```

A *MsgBox* (message box, üzenet doboz a mögötte lévő szöveget egy kis ablakban kiírja a képernyőre. A tanulók számára tehát ismerős dolog történik, a programunk lényegében elküld a felhasználónak egy sms-t.

Remélhetőleg lesz olyan, aki hibázik:

```
MsgBox("Hello World!")
```



*Lezáratlan karakterkonstans.* Egyszerűsége ellenére a VB Script lekezeli a hibákat is. Ami fontos: kezd látszani, hogy meddig tart az alkalmazásfejlesztés, és hol kezdődik a felhasználás.

### Egyszerű számítások: Négyzet

A következő lépés természetesen egyszerű számítások végzése, így határozzuk meg pl. egy négyzet területét. Amit a programunktól várunk:

```
Be: x
k = 4 * x
Ki: k
```

Az  $x$  illetve  $k$  változók fogalma már matematika, fizika óráról ismerős. A bevétel pedig a *MsgBox*-hoz hasonlóan, az *InputBox* segítségével történik:

```
x=InputBox("Oldal")
k=4*x
MsgBox(k)
```

Ha a program már jól működik, érdemes egy kis ravaszsággal más megoldást is keresni:

```
x=InputBox("Oldal")
k=x+x+x+x
MsgBox(k)
```

Ha a bevitt érték 16, akkor az első esetben 64-et, a másodikban 16161616-ot kapunk. Az eltérés lehetőséget ad a típus fogalmának bevezetésére, pl. úgy, hogy hivatkozunk a matematikai egyenleteknél megszokott *értelmezési tartomány* fogalomra.

A VB Script alapértelmezésben csak a *Variant* típust ismeri, aminek altípusai a szöveges, egész (pl. Integer), valós (pl. Single) stb., tehát ami a programozási nyelvekben szokásos. Beolvasáskor szöveggként kezeli, de értékadásnál mindig arra az altípusra konvertálja, amivel az adott művelet elvégezhető. Második példánkban a + jelentheti a konkatenációt, de az első példa azért működött „helyesen”, mert a \* szövegekre nincs értelmezve, csak számokra.

Természetesen lehetőségünk van a típuskonverziós függvények használatára. Az  $x$  változó értékét a *CInt(x)* egészre, a *CStr(x)* karakterlánccra, a *CSng(x)* egyszeres pontosságú valósra konvertálja. A „helyes” megoldás:

```
x=InputBox("Oldal")
x=CSng(x)
k=x+x+x+x
MsgBox(k)
```

### Elágazás: Súlyteszt

Kedvenc feladatomban az elágazás bevezetéséhez a súlyteszt. Internetről hamar megkereshetjük, hogy egy ember mikor számít elhízottnak, orvosi vagy egészséges életmóddal foglalkozó lapokon hamar megtaláljuk az algoritmust (*ti* neve: testtömegindex):

```
Be: kg, m
ti = kg/m2
Ha ti 20 és 25 közé esik, akkor normális
Ha ti 25 és 30 közé esik, akkor túlsúlyos.
Ha ti 30-nál több, akkor elhízott.
```

Az elágazás a Visual Basicben a következő szerkezettel valósítható meg:

```
If <feltétel> Then
    <utasítások>
End If
```

Ennek alapján programunk kódja a következő lehet:

```
kg = InputBox("Tömege (kg)=")
m = InputBox("Magassága (m)=")
ti = kg / (m*m)

If ti < 25 Then
    MsgBox("Ön nincs elhízva!")
End If

If ti >= 25 Then
    MsgBox("Ön el van hívva!")
End If
```

Természetesen a rendelkezésünkre álló időtől függően további sávokat is hozzávehetünk.

További szerkezetek:

```
If <feltétel> Then <egy darab utasítás>
```

illetve

```
If <feltétel> Then
    <igaz ág utasításai>
Else
    <hamis ág utasításai>
End If
```

### Ciklus bevezetése: a Manhattan-project

1626-ban az indiánok 24\$ értékű üveggyöngyért eladták Manhattan szigetét a hollandoknak. Ha az akkori törzsfőnök ezt a pénzt betette volna a *Bank of New York*-ba, mennyit érne ez a pénz ma?

Tanulságos végigszámolni, hogyan nő a betét az első pár évben, pl. 10 %-os kamatláb mellett:

24; 26,4; 29,04; 31,944; 35,1384; 38,65224; 42,517464; 46,7692104; sít.

A kamatlábat bekérhetjük pl. egy *InputBox* segítségével, az összeget pedig egy *For* ciklus segítségével követhetjük nyomon:

```
k = InputBox("Kamatláb=")
p = 24
For ev = 1626 To 2004
    p=p+k*p/100
Next
MsgBox("Mai értéke=" & p)
```

A ciklus bevezetése illetve oktatása az egyik legkomolyabb lépés, erre gyakorlatilag minden kolléga egyéni módszereket használ. Látható azonban, hogy egyszerű számolási eljárásokkal (pl. a kamatos kamat, az első  $n$  szám összege, átlaga, prímszámkeresés) a programozási tételekig is eljuthatunk, jóllehet a vektorok kezelése értelemszerűen az 5-10 órába nem fér, és nem is férhet bele.

Természetesen a VB Script ismeri a *While* ciklust is:

```
Do While <ciklusfeltétel>
    <utasítások>
Loop
```

Hátultesztelő változatban a *While* <ciklusfeltétel> a *Loop* utasítást követi.

Egy 2-nél nagyobb szám prím-e:

```
p = InputBox("Prím-e")

i = 2
l = False
Do While i < Sqr(p) And l = False
    If p/i=CInt(p/i) Then l = True
    i = i + 1
Loop

If l = False Then MsgBox("Prím")
If l = True Then MsgBox ("Összetett")
```

## Ajánlott irodalom

*A programozás alapjai Visual Basicben* (VB6, VBA, VB Script), Jedlik Oktatási Stúdió, Budapest, 2004. Internet: [www.jos.hu/Konyv/VB/VB6.htm](http://www.jos.hu/Konyv/VB/VB6.htm)

*Makró-sarok*: Hasznos kódrészletek a [www.msiskola.jedlik.hu/pages/Makro.aspx](http://www.msiskola.jedlik.hu/pages/Makro.aspx) internet címen. Szerkeszti Szabó Marcell.

Farkas Csaba (farkascs@jedlik.hu)  
Jedlik Ányos Gimnázium, Budapest