

Juhász Tibor – Kiss Zsolt:  
**Programozási ismeretek**  
(Műszaki Könyvkiadó, 2011, MK-4462-3)

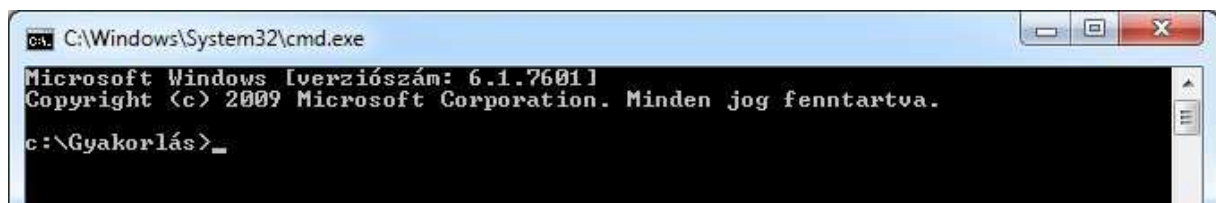
**Visual Basic 2010 Express**  
Kiegészítés a tankönyvhöz

## Parancssori argumentumok beolvasása a Visual Basicben

### Az operációs rendszer parancssora

Az operációs rendszerek általában rendelkeznek egy úgynevezett parancssorral, melynek segítségével szöveges parancsokkal vezérelhetjük a működésüket. A Windowsban a parancssor a *Start/Minden program/Kellékek* csoportban található.

A parancssor általában a sor elején látható tájékoztató karaktersorozat, az úgynevezett prompt után villogó kurzorral jelzi, hogy várja a parancsainkat. Az alábbi ábrán például megjeleníti az aktuális meghajtót (c:) és könyvtárat (\Gyakorlás), majd a > jel utal a parancsra való várakozásra.



Az operációs rendszerek számos parancsot ismernek. Ezek listáját a Súgóban találjuk. A Windowsban keressünk rá a *parancssori utasítások* (Windows XP), illetve a *parancssori referencia* (Windows 7) kifejezésre! A *dir* parancs például kilistázza az aktuális könyvtár tartalmát. Egy meghajtó betűjelének (például *d:*) begépelésével pedig háttértárat válthatunk.

Megjegyezzük, hogy a parancssori ablak megnyitása óta kiadott parancsok a ↑, illetve ↓ kurzormozgató billentyűkkel hívatók elő. A bal felső sarokban található ablakmenü *Tulajdonságok* menüpontjával pedig az ablakméret, betűméret, betűszín stb. állítható be. Ugyanitt találjuk a vágólap tartalmának beillesztését szolgáló menüparancsot is (*Szerkesztés/Beillesztés*).

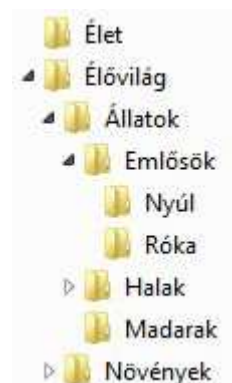
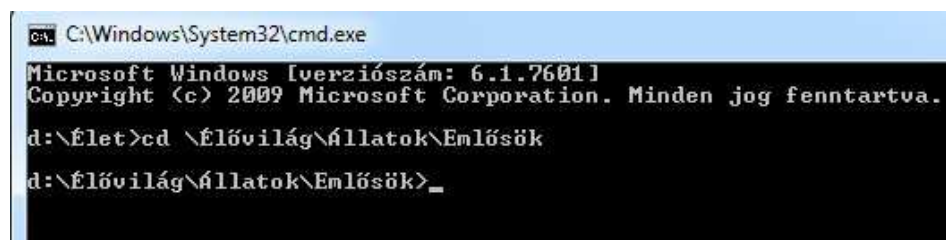
### Parancssori argumentumok

Sok parancs további információkat, úgynevezett argumentumokat vár a végrehajtáshoz. Az argumentumokat a parancssorban a parancs neve után írjuk, a névtől és egymástól általában szóközzel elválasztva.

A *cd* parancssal például meg tudjuk változtatni az aktuális könyvtárat (change directory). A parancs neve után meg kell adnunk a másik könyvtár relatív vagy abszolút elérési útját. Ha például a *d:* meghajtó *Élet* mappájában vagyunk (lásd a mellékelt ábrát), akkor a következő parancs:

```
cd \Élővilág\Állatok\Emlősök
```

az *Emlősök* mappát teszi aktuálissá.



## A parancssori argumentumok elérése a Visual Basicben

Visual Basicben általában konzolalkalmazásokat szoktunk a parancssor segítségével futtatni. Elegendő belépni az exe-fájlt tartalmazó mappába, majd beírni a parancssorba az exe-fájl nevét (vagy megadni az elérési útját).

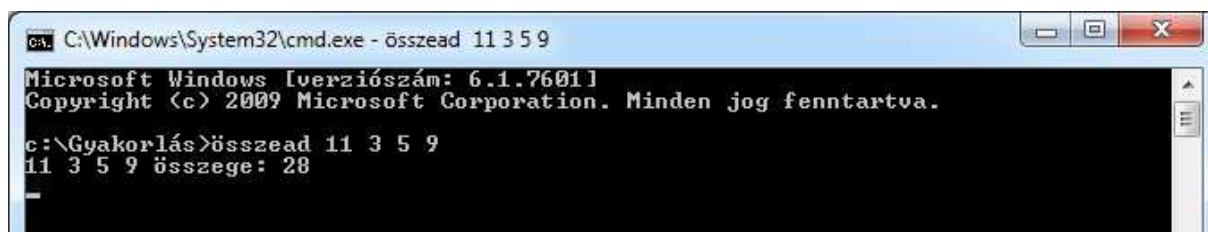
A konzolalkalmazásokban azonban kissé nehézkes a kommunikáció a felhasználóval. A munkát parancssori argumentumok alkalmazásával gyorsíthatjuk, melyek segítségével megadhatjuk a program futtatásához szükséges adatokat. Erre például az input-fájl nevének kijelölésekor van szükség.

Az argumentumok eléréséhez definiáljunk egy sztringtömböt a *Main* eljárás paramétereiként:

```
Sub Main(ByVal Argumentumok() As String)
```

A tömb elemei felveszik a parancssorban megadott, egymástól szóközzel elválasztott argumentumokat. A következő program például összeadja az argumentumként felsorolt számokat.

```
Module Module1
Sub Main(ByVal Argumentumok() As String)
Dim Összeg = 0
For Each Elem In Argumentumok
Console.Write(Elem & " ")
Összeg += Elem
Next
Console.WriteLine("összege: " & Összeg)
Console.ReadLine()
End Sub
End Module
```



```
C:\Windows\System32\cmd.exe - összead 11 3 5 9
Microsoft Windows [verziószám: 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Minden jog fenntartva.

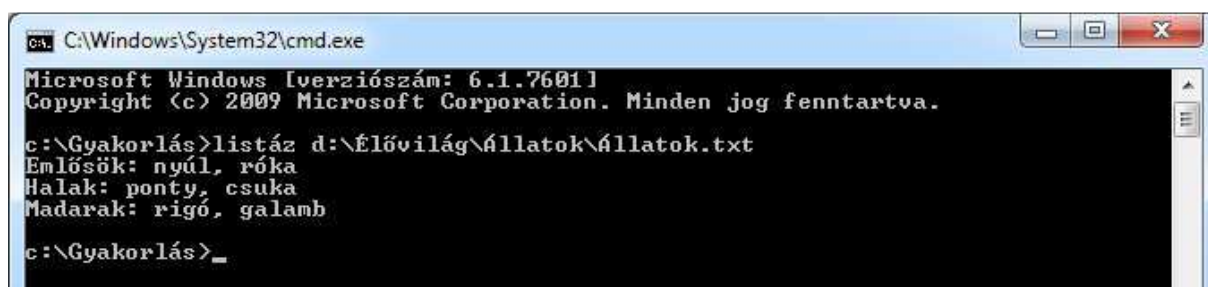
c:\Gyakorlás>összead 11 3 5 9
11 3 5 9 összege: 28
```

Vegyük észre, hogy a programban nem határoztuk meg előre az összeadandók számát!

## Input-fájl elérési útjának megadása a parancssorban

Az előző példában látható módon tetszőleges információt, például a feldolgozandó adatokat tartalmazó input-fájl elérési útját is megadhatjuk a parancssorban. Az alábbi program kilistázza a parancssorban megadott szövegfájl tartalmát a képernyőre:

```
Module Module1
Sub Main(ByVal Fájlnev() As String)
Dim Sor As String
Dim Fájlbe As New IO.StreamReader(Fájlnev())
Do While Not Fájlbe.Peek
Sor = Fájlbe.ReadLine
Console.WriteLine(Sor)
Loop
Fájlbe.Close()
Fájlbe.Dispose()
End Sub
End Module
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [verziószám: 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Minden jog fenntartva.

c:\Gyakorlás>listáz d:\Élővilág\Állatok\állatok.txt
Emlősök: nyúl, róka
Halak: ponty, csuka
Madarak: rigó, galamb

c:\Gyakorlás>_
```

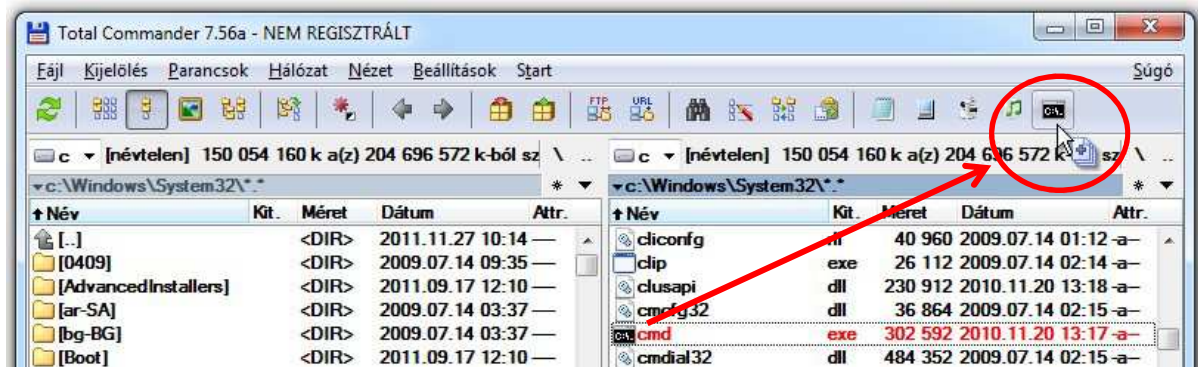
A *Main* eljárás paramétere mindenképpen tömb legyen, még akkor is, ha csak egyetlen argumentumot írunk a parancssorba.

Ügyeljünk arra, hogy a Windows 7 már Unicode-ot használ a parancssorban, így az ékezetes karakterek helyes megjelenítéséhez Unicode-ban (vagy UTF-8 kódolással) kell menteni a szövegfájlt (ha csak nem adjuk meg a fájl kódolását a megnyitáskor).

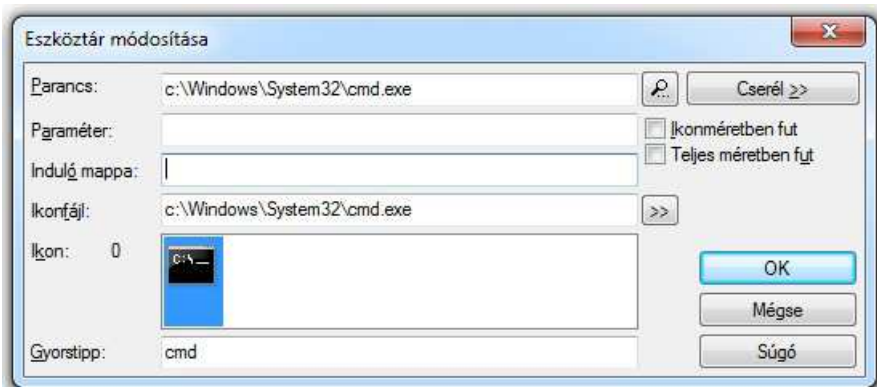
## A parancssori ablak gyors megnyitása a TotalCommander segítségével

A parancssori ablak megnyitásakor (a *Start* menü vagy a *cmd* parancs segítségével) általában a felhasználó könyvtára az aktuális könyvtár. A könyvtárak váltása esetenként nehezkesse teszi a parancssor használatát. A TotalCommander eszköztárát kiegészíthetjük a parancssor parancsikonjával. Ekkor beállíthatjuk, hogy a TotalCommanderben kiválasztott könyvtár maradjon az aktuális könyvtár.

Keressük meg a *c:\Windows\System32* mappában a *cmd.exe* fájlt, majd fogjuk meg az egérrel, és tegyük fel a TotalCommander eszköztárának jobb oldali részére:



Kattintsunk rá jobb egérgombbal az így elhelyezett ikonra, majd válasszuk a *Módosítás* parancsot. A megjelenő *Eszköztár módosítása* ablakban töröljük ki az *Induló mappa* szövegdoboz tartalmát:



Ezzel elérjük, hogy a TotalCommanderben kiválasztott könyvtár maradjon a parancssori ablakban az aktuális könyvtár.

## Egyéb lehetőségek

A teljesség kedvéért megjegyezzük, hogy a parancssori argumentumokra a *Main* eljárás paraméterezése nélkül, közvetlenül hivatkozhatunk a *My.Application.CommandLineArgs* tömb elemeiként. A tömb típusa:

*ObjectModel.ReadOnlyCollection(Of String)*

Változó deklarálása nélkül például:

```
Dim ElsőArgumentum As String
```

```
ElsőArgumentum = My.Application.CommandLineArgs(0)
```

vagy változó deklarálásával:

```
Dim Argumentumok As ObjectModel.ReadOnlyCollection(Of String) = _
    My.Application.CommandLineArgs
```

```
ElsőArgumentum = Argumentumok(0)
```

A Visual Basicben használhatjuk a *Command* függvényt is, melynek visszatérési értéke a parancssori argumentumok (egyetlen) sztringje. Több argumentum esetén a sztringet a *Split* függvénnyel vághatjuk szét.

A következő utasítás például létrehozza a parancssorban megadott fájl névhez rendelt *StreamReader* objektumot:

```
Dim Fájlbé As New IO.StreamReader(Command)
```

A Visual Basic dokumentációja szerint a *My.Application.CommandLineArgs* tömb hatékonyabban működik, mint a *Command* függvény.