# Visual Basic.NET 2. gyakorlat

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

## Solution Explorer

Egy solution több alkalmazást tartalmazhat, akár több nyelven is.

## Module, Main

Ebbe írjuk alkalmazásunkat.

Main függvény indul el futtatáskor.

## Hello Világ

Nincs pontosvessző a sor végén. Helyette van a kettőspont, de nem kötelező.

Module Module1

 Sub Main**()**

 Console.WriteLine**(**"Hello World"**)**

 End Sub

End Module

Futtatás CTRL + F5-tel, hogy ne tűnjön el.

## Beolvasás

Console.ReadLine**()**

CTRL+F5 nem kell, ha ez van a program végén. Enterrel kilépünk.

## Komment

Aposztróf vagy a $rem$ kulcsszó.

'This is a comment.

rem This is a comment too.

## Változódeklarálás

Illik változódeklarálás elé kiírni a $Dim$ kulcsszót.

Console.WriteLine**(**"What is your name?"**)**

Dim name As String **=** "initial value is optional"

name **=** Console.ReadLine**()**

Console.WriteLine**(**"Hi " **+** name**)**

### Szám, konvertálás

Dim number As Integer

Console.WriteLine**(**"Tell me a number"**)**

number **=** Convert.ToInt32**(**Console.ReadLine**())**

Console.WriteLine**(**number**)**

Console.Write**(**number.ToString **+** " + 1 = "**)**

number **+=** **1**

Console.WriteLine**(**number**)**

## Gondoltam egy számot…

Több projekt egy solutionben. Váltás köztük:



### Véletlen szám

Dim rng As Random **=** New Random**()**

Dim min As Integer **=** **10**

Dim max As Integer **=** **100**

Dim n As Integer **=** rng.Next**(**min**,** max**)**

### Tipp beolvasása

Console.Write**(**"Guess: "**)**

guess **=** Convert.ToInt32**(**Console.ReadLine**())**

### Ellenőrzés (kisebb, nagyobb, vagy egyenlő) + Ciklus

Dim guess As Integer **=** **0**

While n **<>** guess

 Console.Write**(**"Guess: "**)**

 guess **=** Convert.ToInt32**(**Console.ReadLine**())**

 If guess **<** n Then

 Console.WriteLine**(**"Too small!"**)**

 End If

 If guess **>** n Then

 Console.WriteLine**(**"Too big!"**)**

 End If

End While

Console.WriteLine**(**"Congratulations! You guessed right."**)**

### Más ciklus lehetőségek

Do While n **<>** guess

 '...

Loop

Vagy:

Do

 '...

Loop While n **<>** guess

Vagy:

Do

 '...

Loop Until n **<>** guess

Vagy:

Do Until n **<>** guess

 '...

Loop

A sokféle Basic verzió miatt van ötféle feltételes ciklus.

### Lehessen háromszor játszani

For i **=** **1** To **3** Step **1** 'Step 1 is optional

Next

### Teljes kód

Module Module1

 Sub Main**()**

 Dim rng As Random **=** New Random**()**

 Dim min As Integer **=** **10**

 Dim max As Integer **=** **100**

 Dim n As Integer

 Console.WriteLine**(**"I'm thinking of a number between " **+** min.ToString **+** " and " **+** max.ToString **+** "..."**)**

 'Console.WriteLine(n)

 Dim guess As Integer

 Dim i As Integer

 For i **=** **1** To **3** Step **1** 'Step 1 is optional

 Console.WriteLine**(**"Game #" **+** i.ToString **+** "/3"**)**

 guess **=** **0**

 n **=** rng.Next**(**min**,** max**)**

 While n **<>** guess

 Console.Write**(**"Guess: "**)**

 guess **=** Convert.ToInt32**(**Console.ReadLine**())**

 If guess **<** n Then

 Console.WriteLine**(**"Too small!"**)**

 End If

 If guess **>** n Then

 Console.WriteLine**(**"Too big!"**)**

 End If

 End While

 Console.WriteLine**(**"Congratulations! You guessed right."**)**

 Next

 Console.WriteLine**(**"Game over, press enter to exit"**)**

 Console.ReadLine**()**

 End Sub

End Module

### Egy játékmenet

I'm thinking of a number between 10 and 100...

Game #1/3

Guess: 50

Too small!

Guess: 70

Too small!

Guess: 80

Too small!

Guess: 90

Too big!

Guess: 85

Too small!

Guess: 86

Too small!

Guess: 87

Congratulations! You guessed right.

Game #2/3

Guess: 50

Too big!

Guess: 40

Too small!

Guess: 45

Too big!

Guess: 43

Congratulations! You guessed right.

Game #3/3

Guess: 50

Too big!

Guess: 30

Too small!

Guess: 40

Too big!

Guess: 34

Too big!

Guess: 33

Congratulations! You guessed right.

Game over, press enter to exit

### Hibakezelés

Nem fogunk hibát kezelni egyelőre.

Számot várunk, szöveget kapunk ⇒ Megáll a program hibaüzenettel.

$Try$ - $Catch$ blokk van már. Régen $onerror$ és $goto$ volt, ami sírba vitte a programozókat.

## Függvények

Minden függvénynek ($Function$) kell visszatérési értéket adnia. Ha $void$ függvényt szeretnénk, az az eljárás ($Sub$)

Paraméter átadás: $ByRef$ vagy $ByVal$.

Régen a ByVal volt opcionális, ma a ByRef.

### Csere

Sub swap**(**ByRef a As Integer**,** ByRef b As Integer**)**

 Dim tmp As Integer **=** a

 a **=** b **:** b **=** tmp

End Sub

### Max

Function maximum2**(**ByVal a As Integer**,** ByVal b As Integer**)** As Integer

 Dim helper As Integer

 If a **<** b Then

 helper **=** b

 Else

 helper **=** a

 End If

 Return helper

End Function

Én verzióm:

Function maximum**(**ByVal a As Integer**,** ByVal b As Integer**)** As Integer

 If a **<** b Then

 Return b

 End If

 Return a

End Function

## Hanoi tornyai

Sub hanoi**(**ByVal n As Integer**,** ByVal source As Char**,** ByVal target As Char**,** ByVal helper As Char**)**

 If n **>** **0** Then

 hanoi**(**n **-** **1,** source**,** helper**,** target**)**

 Console.WriteLine**(**source **+** " → " **+** target**)**

 hanoi**(**n **-** **1,** helper**,** target**,** source**)**

 End If

End Sub

Sub Main**()**

 Console.WriteLine**(**"Hanoi (Recursion)"**)**

 Console.WriteLine**(**"Enter number of rings"**)**

 Dim n As Integer **=** Console.ReadLine

 hanoi**(**n**,** "a"**,** "b"**,** "c"**)**

 Console.WriteLine**(**"Hanoi completed. Press enter to exit."**)**

 Console.ReadLine**()**

End Sub

### Kimenet 3-ra

Hanoi (Recursion)

Enter number of rings

3

a → b

a → c

b → c

a → b

c → a

c → b

a → b

Hanoi completed. Press enter to exit.

Gyakorlat vége.