



**BME**

Budapesti Műszaki és Gazdaságtudományi Egyetem



**OKJIT**

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

# Algoritmusok Tervezése

4. Előadás

Visual Basic 1.

Dr. Bécsi Tamás

# Bevezetés

- A BASIC (Beginner's All-purpose Symbolic Instruction Code) programnyelvet oktatási célokra hozták létre 1964-ben. Az általános célú felhasználhatóság és a könnyű tanulhatóság elsődleges szempont volt.
- A nyelvnek többféle változata létezett, kezdve az iskola-számítógépek beépített BASIC értelmezőjétől, a 80-as években elterjedt személyi számítógépek Qbasic-jén keresztül, az 1991-ben megjelent Microsoft Visual Basic (röviden VB) nyelvig. A Visual Basic for Applications (röviden VBA) az MS Office szoftverek makrónyelve, a Visual Basic Script a Windows operációs rendszer scriptnyelve, a 2002-ben megjelent Visual Basic .NET pedig a .NET keretrendszer programozási nyelve.

# MS Visual Basic for Applications

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Csak Visual Basic nyelven programozhatunk, (a Visual Studio-hoz képest) korlátozott fejlesztési eszköztár mellett.
- Csak az adott szoftverrel (pl. Excel) együtt használható.
- Önállóan futtatható (\*.exe) fájlok nem készíthetők.
- Az adott szoftverhez igazodó, „készen kapott” objektumrendszer tartalmaz.

# Típusok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Adattípus	Tárolási méret	Magyarázat
Byte	1	0..255
Integer	2	-32768..32767
Long	4	-2,147,483,648 .. 2,147,483,647
LongLong	8	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807 (64bit only)
Single	4	Egyszeres pontosságú Lebegőpontos
Double	8	Kétszeres pontosságú lebegőpontos
Boolean	2	True/False
Char	2	Karakter
String	*	Szöveges típus
Date	8	Dátum
Variant	*	
Object	*	

# Változók definíciója

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

A változók deklarálásának (egyszerűsített) szintaktikája:

Dim *varname* [*As type*] [...]

*varname* A változó (betűvel kezdődő) azonosítója (neve).

*type* A változó típusa (ha hiányzik, a változó Variant típusú lesz).

Pl.

Dim i As Integer 'Egy Integer típusú változó deklarációja

Dim v 'Egy Variant típusú változó

Dim a,b As Single 'Egy Variant és egy Single típusú változó

Az Option explicit direktíva

# Operátor-precedencia

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Aritmetika	Relációs	Logikai
Hatványozás (^)	Egyenlő (=)	Not
Negáció (-)	Nem egyenlő (<>)	And
Szorzás (*), osztás (/)	Kisebb (<)	Or
Egész osztás hányadosa (\)	Nagyobb (>)	Xor
Egész osztás maradéka (Mod)	Kisebb vagy egyenlő (<=)	Eqv
Összeadás (+), kivonás (-)	Nagyobb vagy egyenlő (>=)	Imp
*String összefűzés (&)	Like	
	Is	

# Aritmetikai operátorok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

$10+3$	13
$10-3$	7
$10*3$	30
$10/3$	3,333333
$10\backslash 3$	3
$10 \text{ Mod } 3$	1
$10 \wedge 3$	1000

$10,6+3$	13,6
$10,6-3$	7,6
$10,6*3$	31,8
$10,6/3$	3,533333
$10,6\backslash 3$	3
$10,6 \text{ Mod } 3$	2
$10,6 \wedge 3$	1191,016

# Logikai Operátorok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

A	Not A
IGAZ	HAMIS
HAMIS	IGAZ

A	B	A And B	A Or B	A Xor B	A Imp B	A Eqv B
IGAZ	IGAZ	IGAZ	IGAZ	HAMIS	IGAZ	IGAZ
IGAZ	HAMIS	HAMIS	IGAZ	IGAZ	HAMIS	HAMIS
HAMIS	IGAZ	HAMIS	IGAZ	IGAZ	IGAZ	HAMIS
HAMIS	HAMIS	HAMIS	HAMIS	HAMIS	IGAZ	IGAZ



# Matematikai függvények

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Abs(X)	X abszolút értéke.
Exp(X)	Exponenciális függvény (ex)
Log(X)	Természetes alapú logaritmus
Sin(X)	X szinusza (X radiánban adott).
Cos(X)	X koszinusza (X radiánban adott).
Tan(X)	X tangense
Sqr(X)	X négyzetgyöke.
Int(X)	X lefelé kerekítve.
Round(X[,numdec])	X kerekítve, numdec tizedesre
Rnd()	Egy véletlen szám a [0, 1) intervallumból

# Konverziós függvények

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Asc(X) Az X karakter ASCII kódja.

Chr(X) Az X ASCII kódú karakter.

Str(X) Az X numerikus adat szöveggé.

Val(X) Az X számot tartalmazó szöveg numerikus értéke.

Pld.

Asc("A") → 65 Chr(65) → "A" Str(2.3) → " 2.3" Val("2.3") → 2.3

# Szövegkezelő függvények

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Len(X)	Az X sztring hossza (karaktereknek száma).
Left(X,Y)	Az X sztring elejéről Y darab karakter.
Right(X,Y)	Az X sztring végéről Y darab karakter.
Mid(X,Y[,Z])	Az X sztring Y-adik karakterétől Z darab karakter.
Trim(X)	Az X sztring vezető és záró szóközeinek levágása.
InStr([X,]Y,Z)	A Z sztring megkeresése az Y sztringben (az X-edik karaktertől kezdődően).

Pld.

Len("Alma") → 4

Right("Alma",2) → "ma"

Mid("Alma",3) → "ma"

InStr("alma","a") → 1

InStr("alma","A") → 0

Left("Alma",2) → "Al"

Mid("Alma",3,1) → "m"

Trim(" a b ") → "a b"

InStr(2,"alma","a") → 4

# If-Then-Elseif-Else-End If

## SYNTAX:

**If** *condition* **Then**

[*statements*]

[**Elseif** *condition-n* **Then**

[*elseifstatements*]

[**Else**

[*elsestatements*]

**End If**

If a = 1 Then

b = 2

End If

If a = 1 Then

b = 2

Else

b = 3

End If

If a = 1 Then

b = 2

Elseif a = 2 Then

b = 4

...

Else

b = 3

End If

# Select-Case

## Syntax

**Select Case** *testexpression*

[**Case** *expressionlist-n*

[*statements-n*]]

[**Case Else**

[*elasticsearch*]]

**End Select**

```
Dim Number As Integer
```

```
Number = 8
```

```
Select Case Number
```

```
Case 1 To 5
```

```
    MsgBox ("Between 1 and 5")
```

```
Case 6, 7, 8
```

```
    MsgBox ("Between 6 and 8")
```

```
Case 9 To 10
```

```
    MsgBox ("Greater than 8")
```

```
Case Else
```

```
    MsgBox ("Not between 1 and 10")
```

```
End Select
```

# While Loop

## Syntax

**While** *condition*  
*[statements]*

## Wend

```
Dim a, b, sum As Integer
```

```
a = 1
```

```
b = 10
```

```
sum = 0
```

```
While a <= b
```

```
    sum = sum + a
```

```
    a = a + 1
```

```
Wend
```

```
    MsgBox (sum)
```

# Do Loop

## Syntax

**Do** [{**While** | **Until**} *condition*]

[*statements*]

[**Exit Do**]

[*statements*]

## Loop

**Do**

[*statements*]

[**Exit Do**]

[*statements*]

**Loop** [{**While** | **Until**} *condition*]

# Do Until példa

a = 1

Do Until a >= 1

a = a + 1

Loop

a=1

a = 1

Do

a = a + 1

Loop Until a >= 1

a=2



# Do While példa

$a = 1$

Do While  $a < 1$

$a = a + 1$

Loop

$a=1$

$a = 1$

Do

$a = a + 1$

Loop While  $a < 1$

$a=2$

# For-Next

## Syntax

**For** *counter* = *start* **To** *end* [**Step** *step*]

[*statements*]

[**Exit For**]

[*statements*]

**Next** [*counter*]



# For példa

```
Dim a, i As Integer
```

```
a = 1
```

```
For i = 1 To 6
```

```
    a = a * i
```

```
Next
```

```
MsgBox (a)
```

```
Dim a, i As Integer
```

```
a = 0
```

```
For i = 0 To 10 Step 2
```

```
    a = a + i
```

```
    If i = 4 Then Exit For
```

```
Next
```

```
MsgBox (a)
```

# For példa 2.

```
Dim a, i As Integer
```

```
a = 0
```

```
For i = 100 To 0 Step -1
```

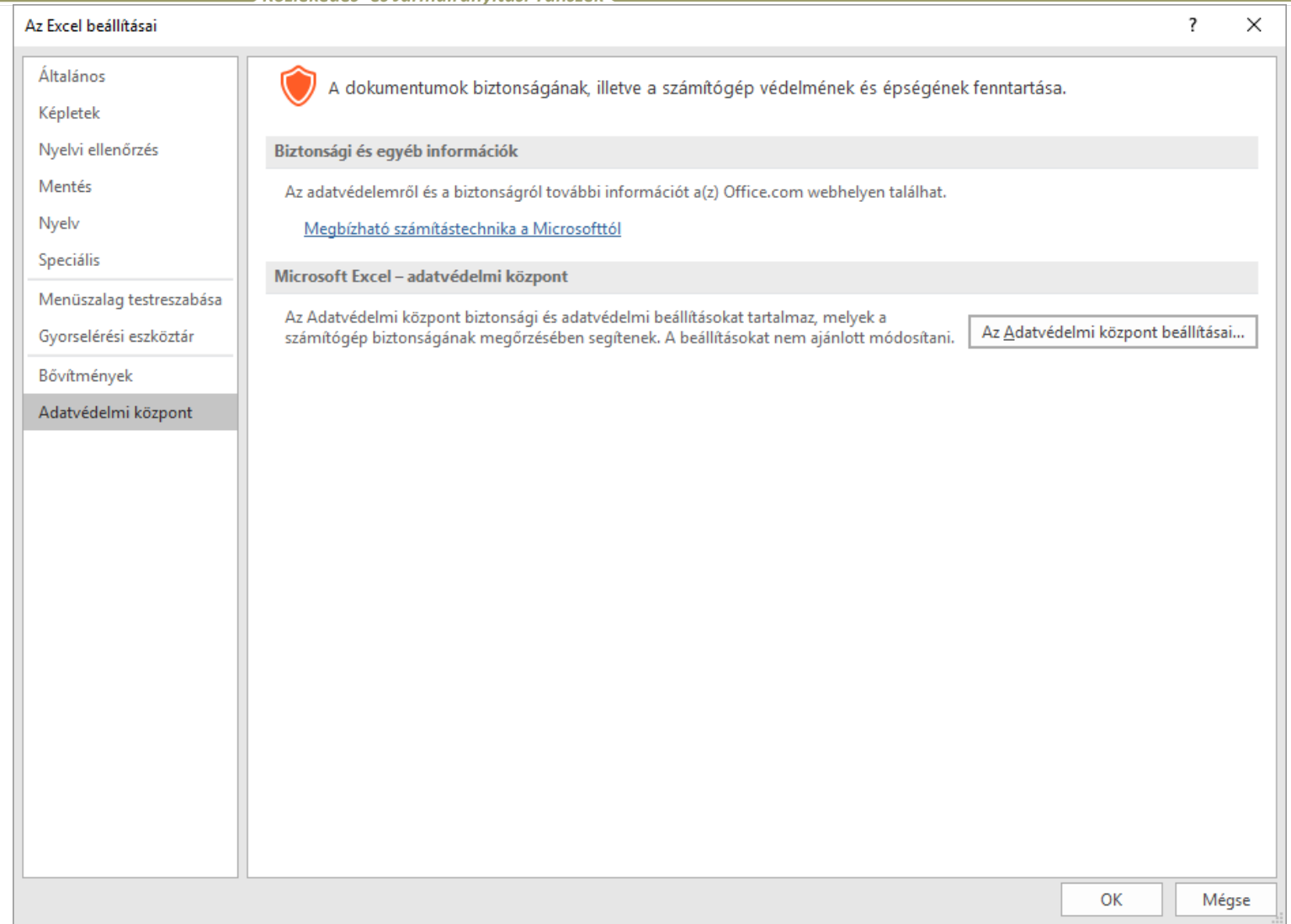
```
    If i Mod 7 = 0 Then a = a + 1
```

```
Next
```

```
MsgBox (a)
```

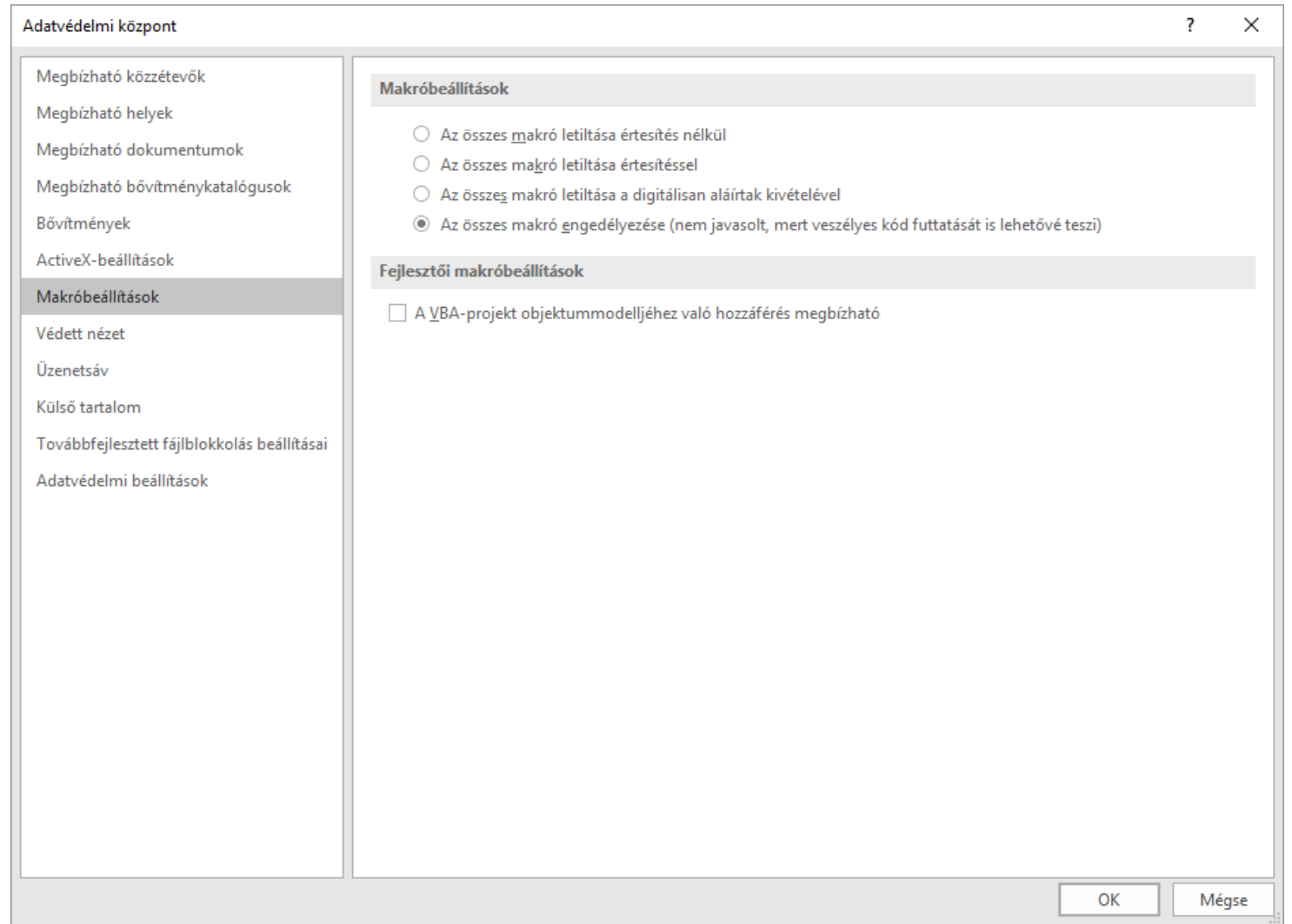
# Excel beállítások 1.

- Fájl
- Beállítások
- Adatvédelmi központ
- Az Adatvédelmi központ beállításai



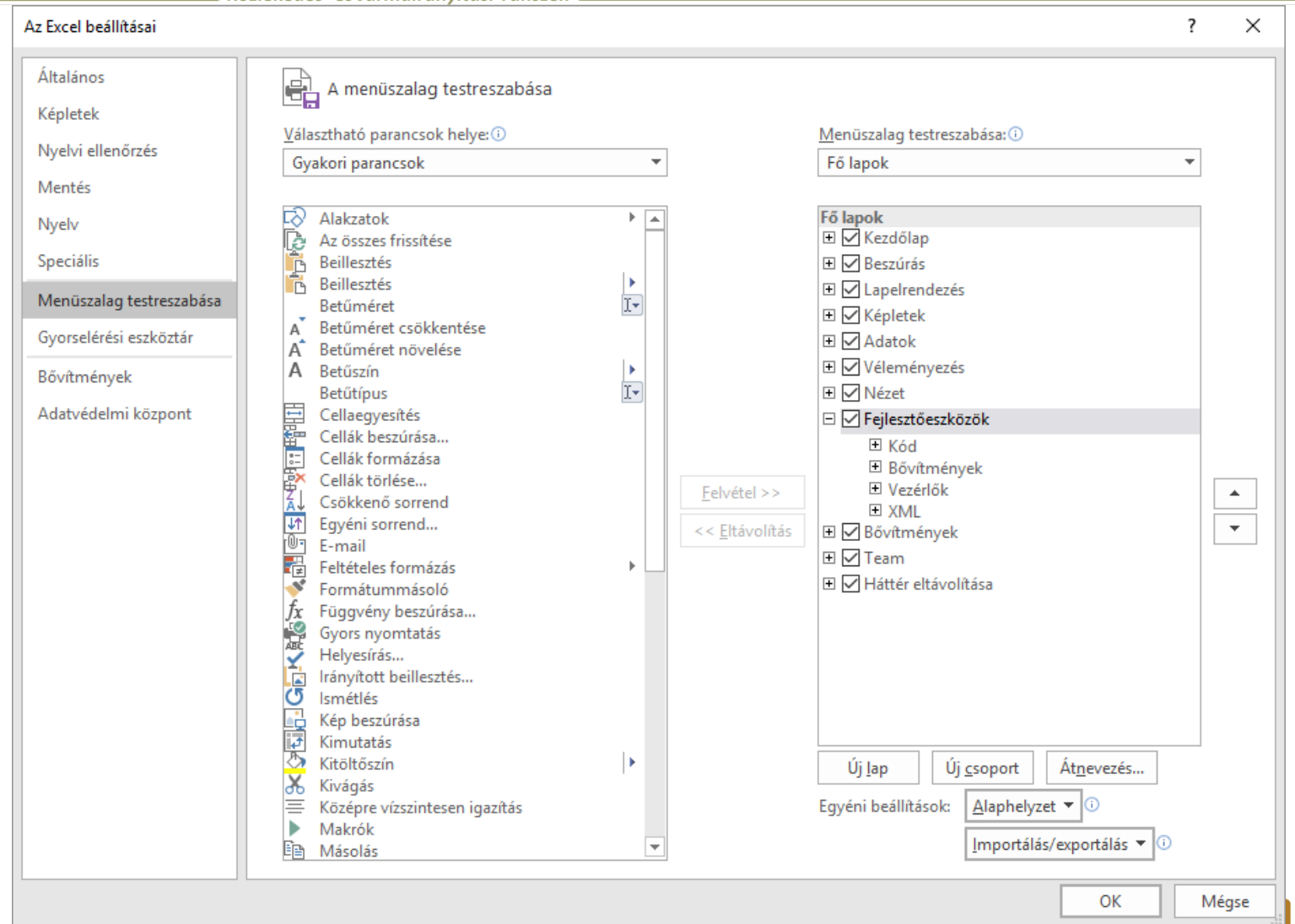
# Excel beállítások 2.

- Makró-beállítások



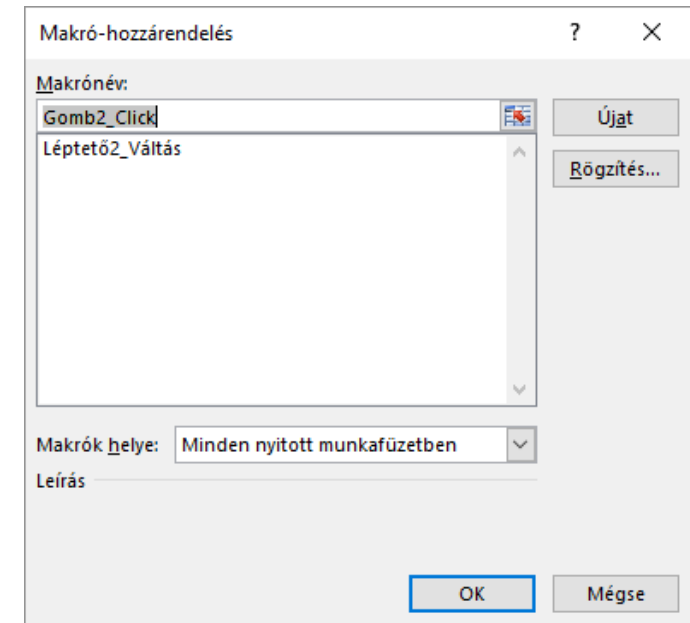
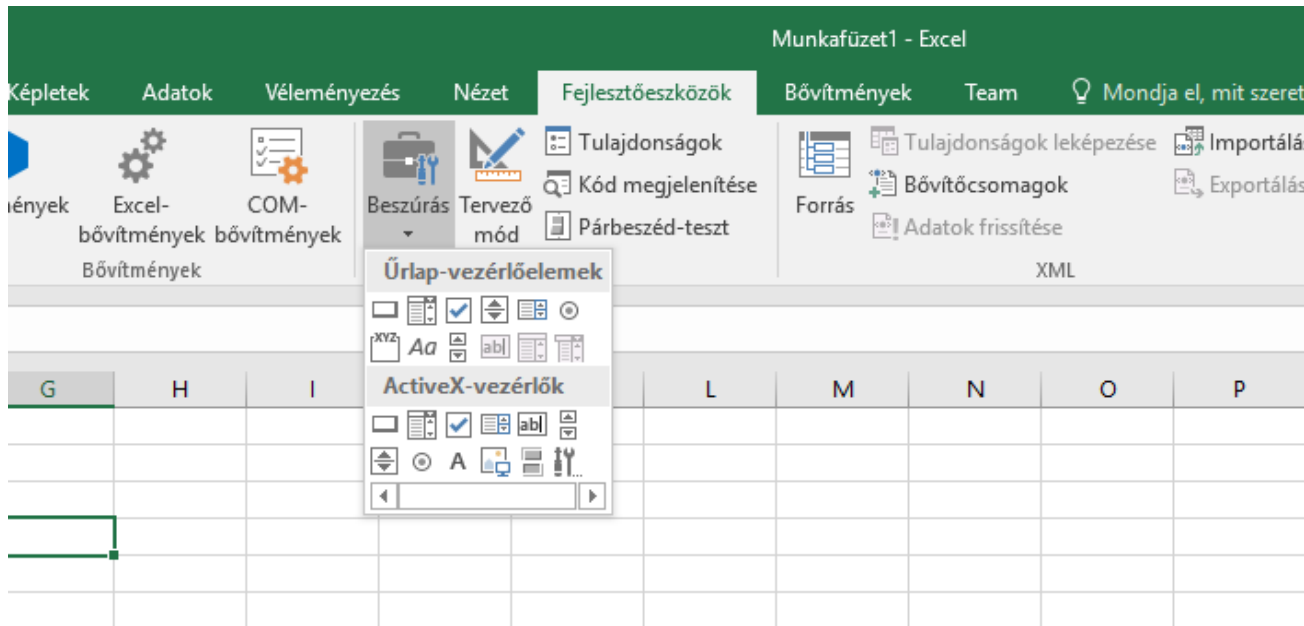
# Excel beállítások 3.

- File
- Beállítások
- Menüszalag testreszabása
- Fejlesztőeszközök



# Alap példa

- Cél: Egyszerű, gombbal vezérelt program írása
- Gomb elhelyezése, függvény hozzárendelése





# Alap példa

