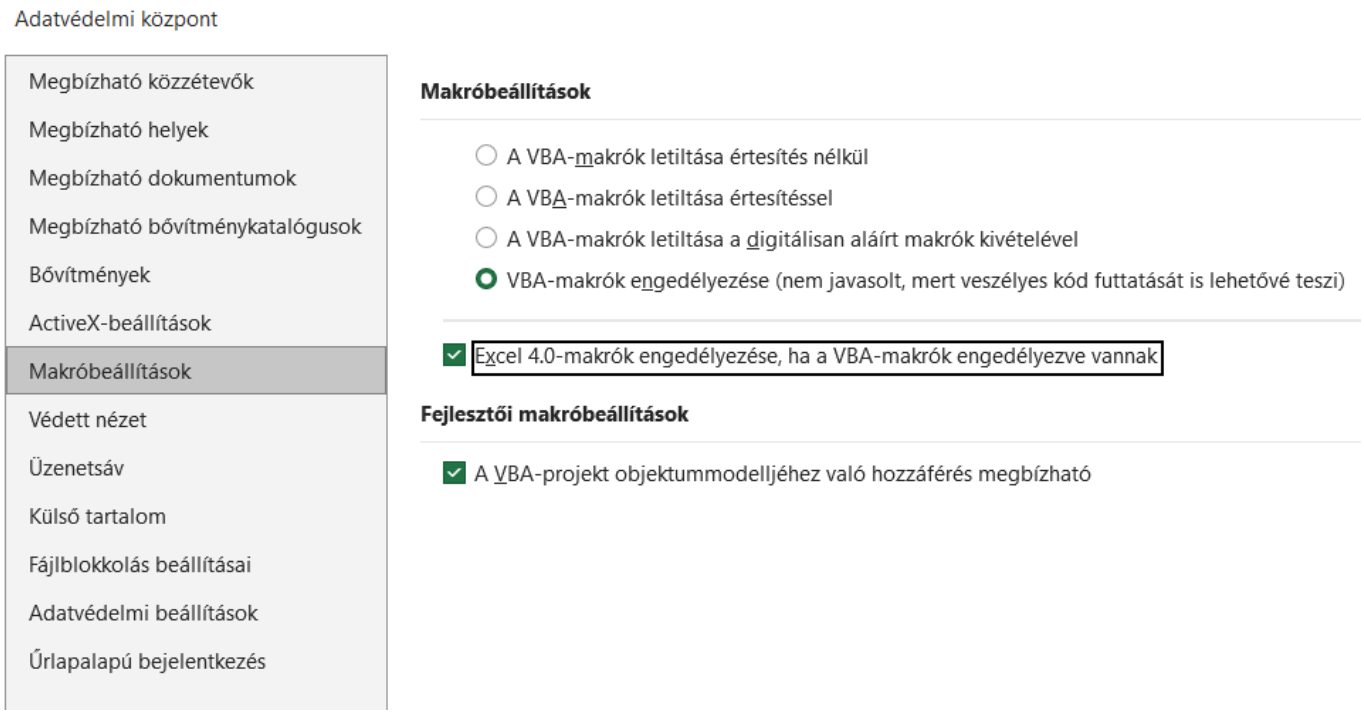
Excel VBA alapjai

Az Excel VBA (Visual Basic for Applications) egy eseményvezérelt programozási nyelv, amely lehetővé teszi az Excel alkalmazások automatizálását és testreszabását. Ebben a részben az Excel VBA alapvető nyelvi elemeit mutatjuk be, beleértve a változók deklarálását, az alapvető típusokat és a vezérlési elemeket.

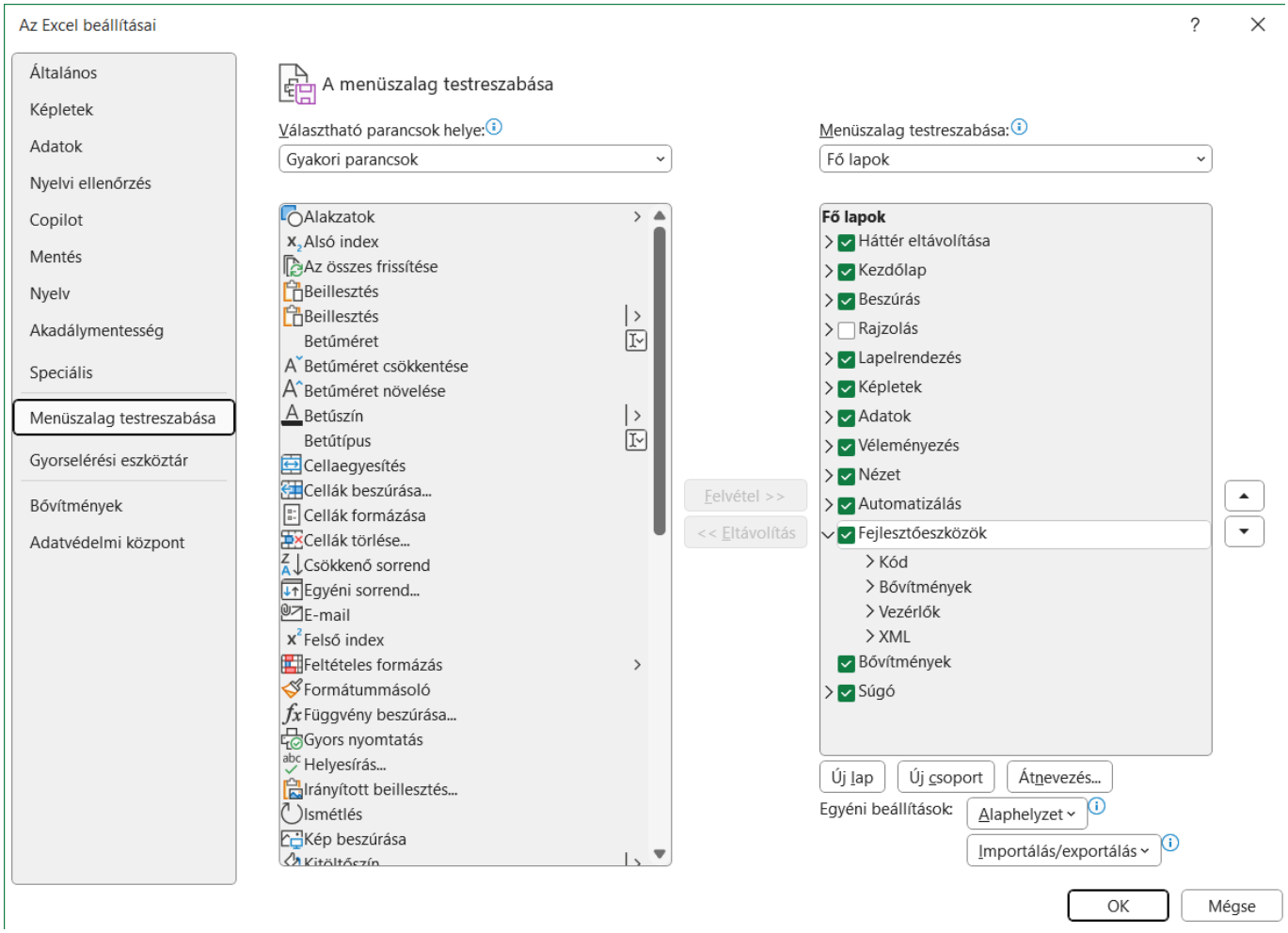
Makrók engedélyezése

A Fájl/Beállítások/Adatvédelmi központ-nál állítsuk be a makrók engedélyezését és utána zárjuk be a munkafüzetet és indítsuk újra.

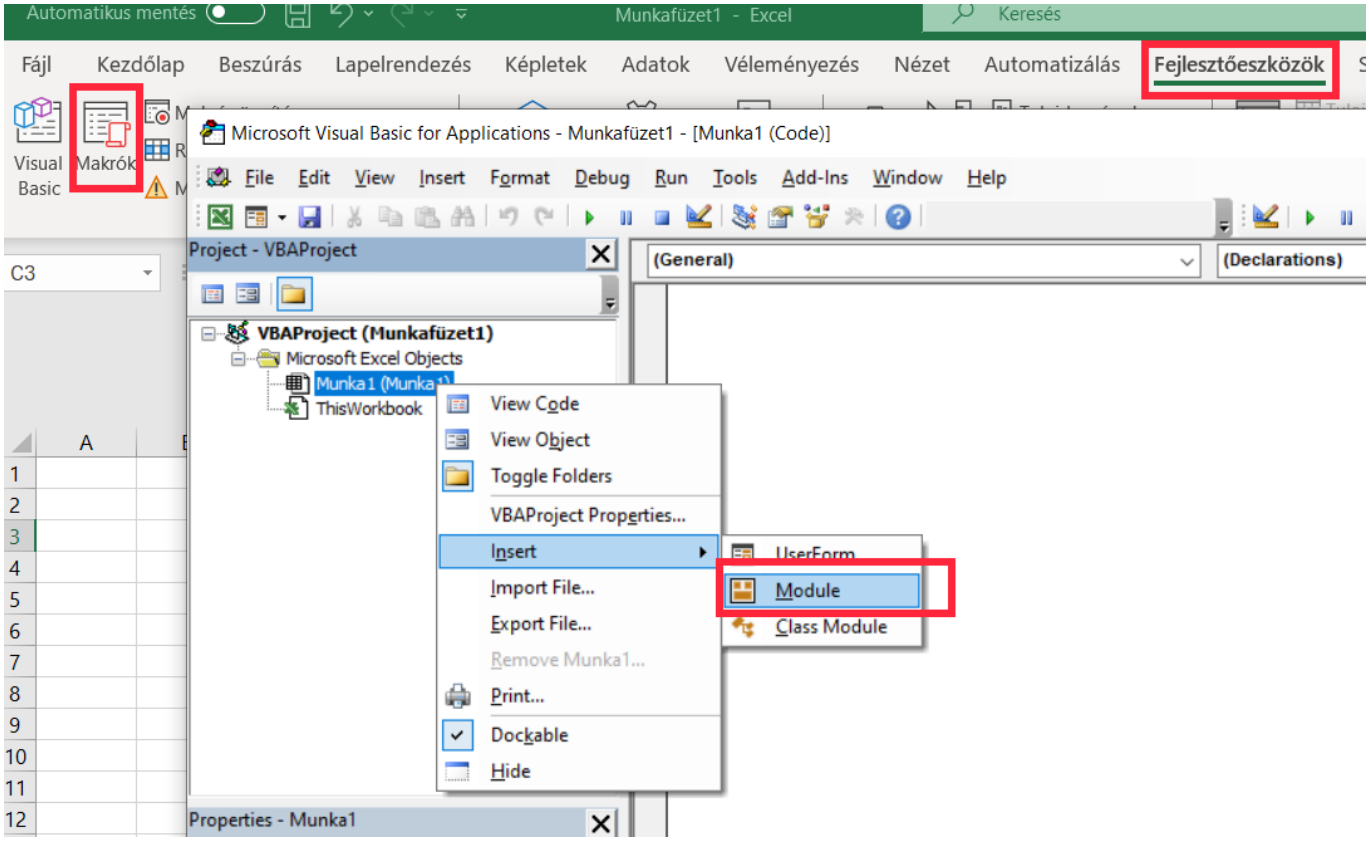


Fejlesztőeszközök (Developer) bekapcsolása:

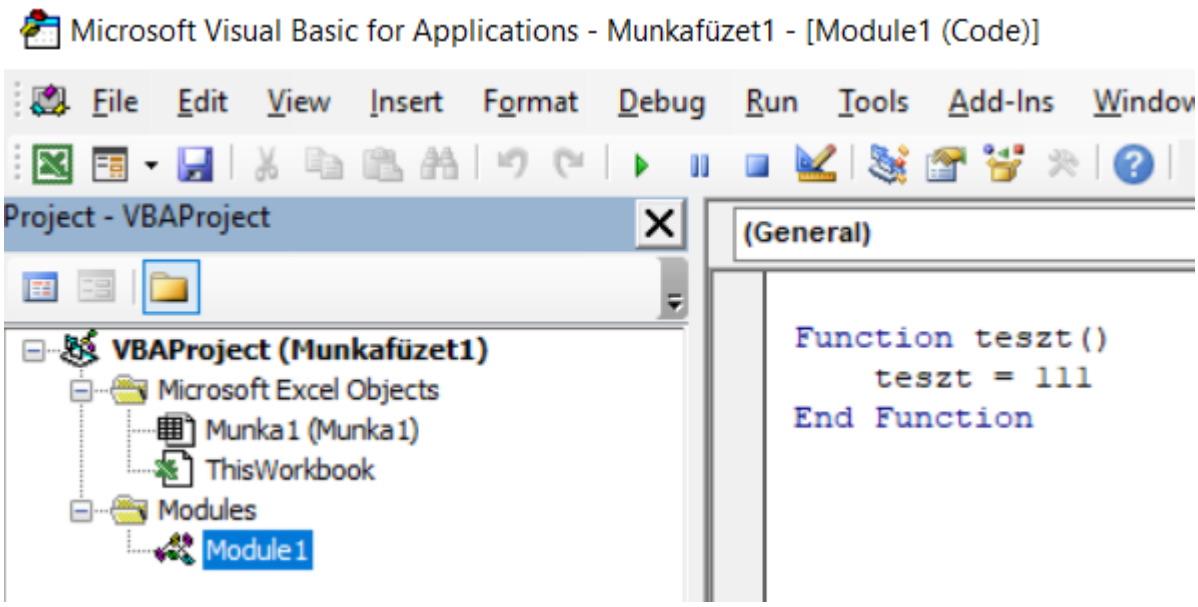
Fájl → Beállítások → Menüszalag testreszabása → Pipáljuk be a **Fejlesztőeszközök**-et.



Válasszuk ki a “Fejlesztőeszközök” fület, majd a “Makrók” gomb megnyomása utána felugró ablakban a jobb egérgombbal megjelenő menüben illesszünk be egy új modult:



Definiáljunk egy teszt függvényt ami visszaad egy konstans értéket.




```
Function teszt()
   teszt = 1
End Function
```

használata: írjuk be egy cellába **=teszt()**

Változók Deklarálása

A VBA-ban a változók deklarálása a `Dim` kulcsszóval történik. A változó típusát is megadhatjuk a deklaráció során, ami segít a kód olvashatóságában és a típushibák elkerülésében.

- **Példa**: Egyszerű változó deklaráció

```
Dim szam As Integer
Dim szoveg As String
```

Alapvető Típusok

A VBA számos alapvető adattípust támogat, többek között:

- **Integer**: Egész számok
- **Long**: Nagyobb egész számok
- **Single**: Lebegőpontos számok (egyszeres pontossággal)
- **Double**: Lebegőpontos számok (kétszeres pontossággal)
- **String**: Szöveges típus
- **Boolean**: Logikai típus (Igaz vagy Hamis)

Kör területének számítása

```
Sub Test()
   Dim r As Double
   Dim area As Double
   r = InputBox("Kérem a kör sugarát")
   area = r * r * Application.WorksheetFunction.Pi
   MsgBox area
End Sub
```

Vezérlési Elemek

A VBA vezérlési szerkezetei lehetővé teszik a program ágának irányítását a különböző feltételek alapján.

- **If...Then...Else**

```
If szam > 10 Then
    MsgBox "A szám nagyobb, mint 10."
```

```
Else
    MsgBox "A szám 10 vagy annál kisebb."
End If
```

- **For Next Ciklus**

```
For i = 1 To 5
    MsgBox "Szám: " & i
Next i
```

Itt fontos megjegyezni hogy a szövegeket & jellel adjuk össze, nem a plusz operátorral!

- **Do While Loop**

```
i = 1
Do While i <= 5
    MsgBox "Szám: " & i
    i = i + 1
Loop
```

Sztringek kezelése

A VBA-ban a stringek kezelésének alapjai közé tartozik a változók deklarálása, értékadás, és az összehasonlítás.

```
  // Deklarálás
  Dim exampleString As String
  // Értékadás
  exampleString = "Hello World"
```

Műveletek Sztringekkel

Összefűzés:

A & operátort használva fűzhetünk össze sztringeket.

```
  Dim firstName As String
  Dim lastName As String
  Dim fullName As String
```

```
firstName = "John"
lastName = "Doe"
fullName = firstName & " " & lastName // "John Doe"
```

A **Len** függvény visszaadja a string hosszát.

```
Dim textLength As Integer
textLength = Len("Hello World") // 11
```

Részsztring Kinyerése

A **Mid** függvényt használva kinyerhetünk egy részsztringet egy adott pozíciótól kezdődően, adott hosszúságban.

```
Dim subString As String
subString = Mid("Hello World", 7, 5) // "World"
```

Keresés a Sztringben

A **InStr** függvény visszaadja egy substring első előfordulásának pozícióját egy másik stringben.

```
Dim position As Integer
position = InStr("Hello World", "World") // 7
```

Sztring Nagy- és Kisbetűssé Alakítás

A **UCase** és **LCase** függvényeket használva nagybetűssé, illetve kisbetűssé alakíthatunk egy sztringet.

```
Dim upperCaseString As String
Dim lowerCaseString As String
upperCaseString = UCase("Hello World") // "HELLO WORLD"
lowerCaseString = LCase("Hello World") // "hello world"
```

Sztring Összehasonlítás

A **StrComp** függvényt használva összehasonlíthatunk két stringet, figyelembe véve a nagy- és kisbetűk különbségét is.

```
```

```
  Dim result As Integer
  // 0, ha a stringek egyenlőek
  result = StrComp("Hello", "hello", vbTextCompare) // egyenlő, hiába H != h
  result = StrComp("Hello", "Hello", vbBinaryCompare) // Egyenlő
```

Vektorok, tömbök kezelése

```
  Dim szamok(0 To 4) As Integer
```

Ez egy 5 elemű tömb (indexei: 0, 1, 2, 3, 4), amely egész számokat tárol.

```
Sub FixMeretuTomb()
    Dim szamok(0 To 2) As Integer
    szamok(0) = 10
    szamok(1) = 20
    szamok(2) = 30

    MsgBox szamok(1)  ' Kiírja: 20
End Sub
```

Változó méretű tömb:

```
Sub DinamikusTomb()
    Dim tomb() As String
    ReDim tomb(1 To 3)

    tomb(1) = "alma"
    tomb(2) = "banán"
    tomb(3) = "citrom"

    MsgBox tomb(2)  ' banán
End Sub
```

Tömb különböző típusú elemekkel:

```
Sub FixMeretuTomb()
   Dim arr(5)
   arr(0) = "1"           'Number as String
   arr(1) = "VBScript"    'String
```

```
   arr(2) = 100           'Number
   arr(3) = 2.45          'Decimal Number
   arr(4) = #10/7/2033#   'Date
   arr(5) = #12:45:00 PM# 'Time
   MsgBox ("Value stored in Array index 0 : " & arr(0))
   MsgBox ("Value stored in Array index 1 : " & arr(1))
   MsgBox ("Value stored in Array index 2 : " & arr(2))
   MsgBox ("Value stored in Array index 3 : " & arr(3))
   MsgBox ("Value stored in Array index 4 : " & arr(4))
   MsgBox ("Value stored in Array index 5 : " & arr(5))
End Sub
```

Tömb bejárása ciklussal:

```
Sub TombBejarasa()
    Dim i As Integer
    Dim szamok(1 To 5) As Integer

    For i = 1 To 5
        szamok(i) = i * 10
    Next i

    For i = 1 To 5
        Debug.Print szamok(i)
    Next i
End Sub
```

megjegyzés: ilyenkor az immediate ablakban jelenik meg a kimenet a szerkesztőben.

A tömb határainak lekérdezése:

```
Dim tomb(3 To 7) As Integer
MsgBox LBound(tomb)  ' alsó határ (3)
MsgBox UBound(tomb)  ' felső határ (7)
```

Ez alapján a TombBejarasa() szubrutint dinamikusabbá tehetjük:

```
Sub TombBejarasa()
    Dim i As Integer
    Dim szamok(1 To 5) As Integer

    For i = LBound(szamok) To UBound(szamok)
```

```
        szamok(i) = i * 10
    Next i

    For i = LBound(szamok) To UBound(szamok)
        Debug.Print szamok(i)
    Next i
End Sub
```

Gyakorló feladatok

Adjunk össze két számot.

```
Function Osszeadas(szam1 As Double, szam2 As Double) As Double
    Osszeadas = szam1 + szam2
End Function
```

használata: **=Osszeadas(A1;B1)**

Írjunk egy függvényt, amely egy sztringet vesz bemenetként, és visszaadja annak fordított változatát.

```
Function SzovegForditas(szoveg As String) As String
    Dim i As Integer
    For i = Len(szoveg) To 1 Step -1
        SzovegForditas = SzovegForditas & Mid(szoveg, i, 1)
    Next i
End Function
```

Ez a függvény egy tartomány elemeinek átlagát számítja ki.

```
Function TartomanyAtlag(tartomany As Range) As Double
    Dim osszeg As Double
    Dim db As Long
    osszeg = 0
    db = 0
    For Each cella In tartomany
        osszeg = osszeg + cella.Value
        db = db + 1
    Next cella
    TartomanyAtlag = osszeg / db
```

```
End Function
```

Töltsük fel tetszőleges értékekkel a A1 és B2 téglalapba eső cellákat:

	A	B
1	2	3
2	3	10

Egy másik cellába írjukbe: **=TartomanyAtlag(A1:B2)**

Viszont, ha az értékek között van olyan ami nem szám, akkor nem működik, ezért alakítsuk át, hogy kezelje a kivételeket is.

```
Function TartomanyAtlag(tartomany As Range) As Double
    Dim cella As Range
    Dim osszeg As Double
    Dim db As Long
    osszeg = 0
    db = 0
    For Each cella In tartomany
        If IsNumeric(cella.Value) And Not IsEmpty(cella.Value) Then
            osszeg = osszeg + cella.Value
            db = db + 1
        End If
    Next cella
    If db > 0 Then
        TartomanyAtlag = osszeg / db
    Else
        TartomanyAtlag = 0
    End If
End Function
```

Töltsük fel tetszőleges értékekkel a A1 és B2 téglalapba eső cellákat, de az egyiket akarattal elrontjuk:

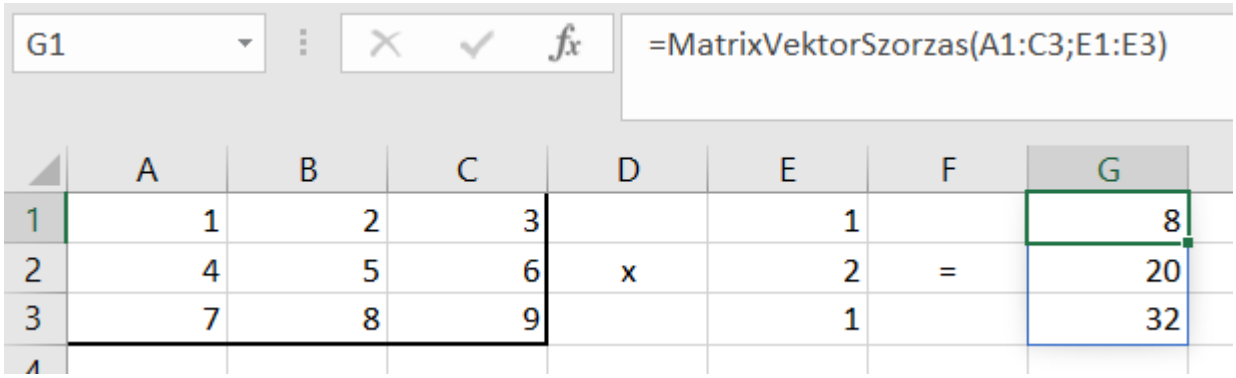
	A	B
1	2	3
2	hello	1

Egy másik cellába írjukbe: **=TartomanyAtlag(A1:B2)**

Szorozzunk össze egy mátrix-ot egy vektorral:

```
Function MatrixVektorSzorzas(matrix As Range, vektor As Range) As Variant
    Dim sorok As Long, oszlopok As Long, i As Long, j As Long
    Dim eredmeny() As Double
    sorok = matrix.Rows.Count
    oszlopok = matrix.Columns.Count
    ' Ellenőrizzük, hogy a vektor mérete megfelel-e a mátrix oszlopainak
számával
    If vektor.Rows.Count <> oszlopok Or vektor.Columns.Count > 1 Then
        MatrixVektorSzorzas = CVErr(xlErrValue)
        Exit Function
    End If
    ' oszlopvektor legyen az eredmény
    ReDim eredmeny(1 To sorok, 1 To 1) As Double
    For i = 1 To sorok
        For j = 1 To oszlopok
            eredmeny(i, 1) = eredmeny(i, 1) + matrix.Cells(i, j).Value *
vektor.Cells(j, 1).Value
        Next j
    Next i
    MatrixVektorSzorzas = eredmeny
End Function
```

Használata például ez lehet:



—