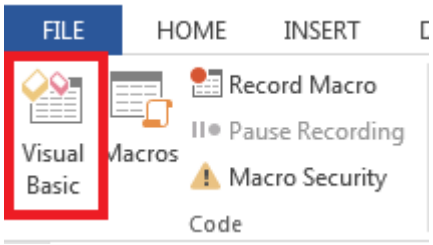
Laborator VBA pentru Excel 1

Adăugarea unei funcții simple

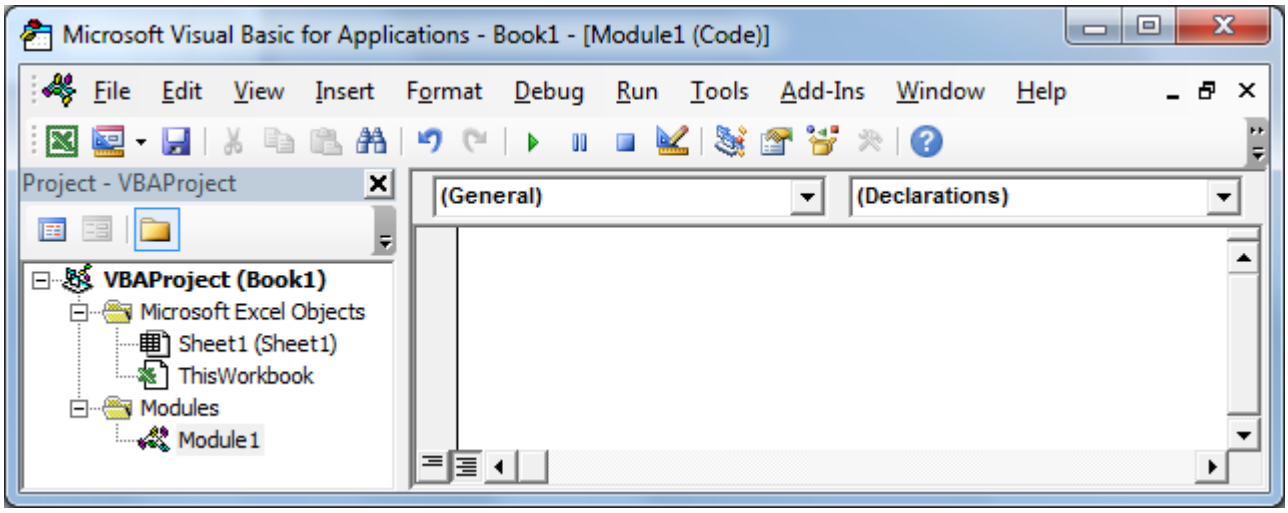
1. Lansați Microsoft Excel
2. Completați foaia 1 ca in figura alăturata

	A	B	C	D	E
1	a	2.378			
2	b	1.356			
3					

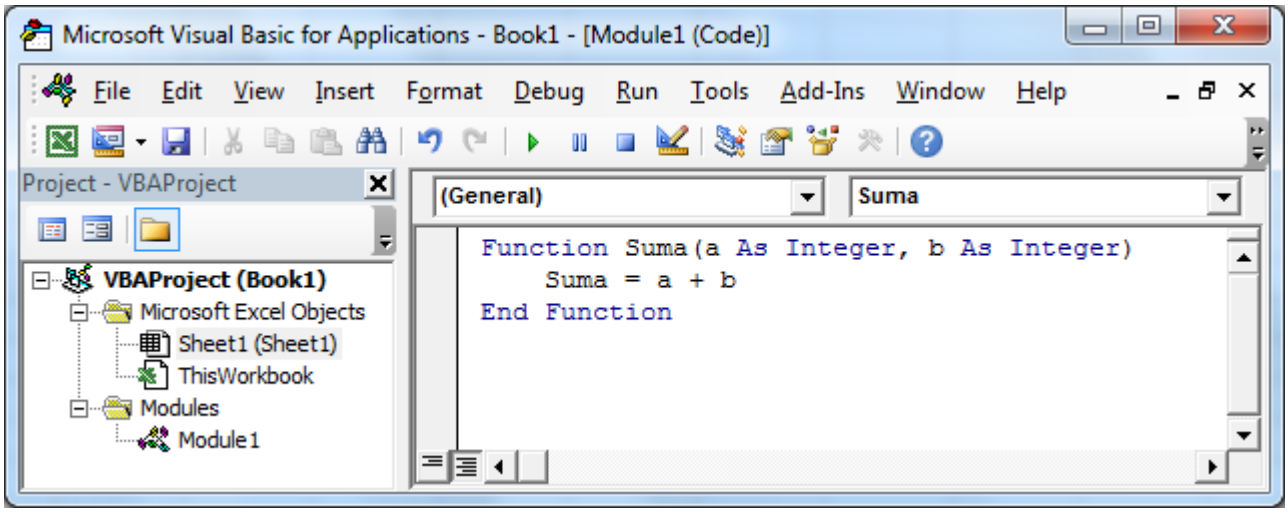
3. Lansați Visual Basic Editor (comanda **Visual Basic** din fila **DEVELOPER**, grupul **Code**)



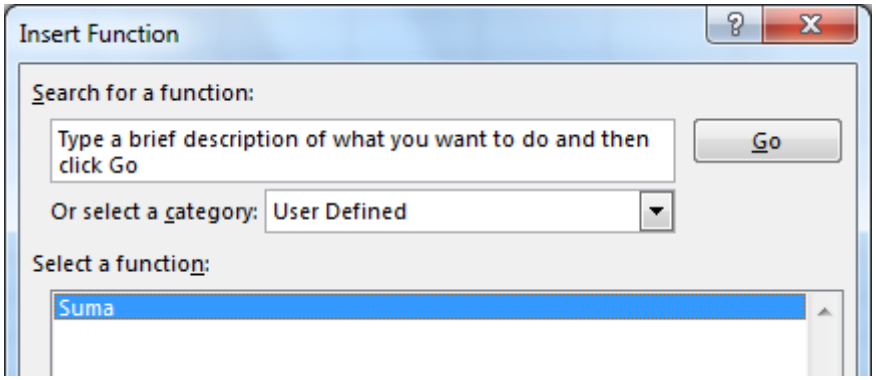
4. In VBE adăugați un modul (**Insert > Module**)



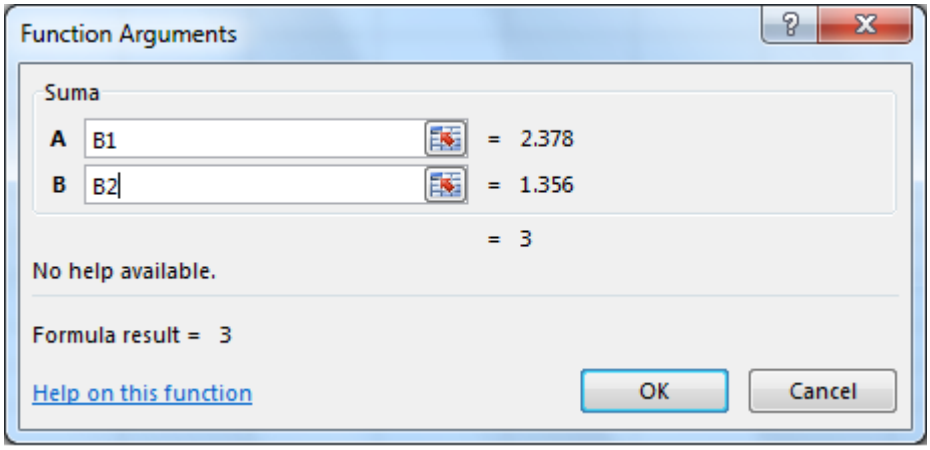
5. Tastați codul de mai jos in modulul **Module1**



6. Închideți VBE și activați foaia Excel
7. Mutați cursorul in celula C2
8. Lansați comanda Insert Function
9. In fereastra de dialog Insert Function, selectați categoria User Defined, respectiv funcția Suma in lista Select a function:



10. In fereastra **Function Arguments** adăugați ca argumente celulele B1 si B2 ca in figura alăturata



Depanarea si inspectarea variabilelor

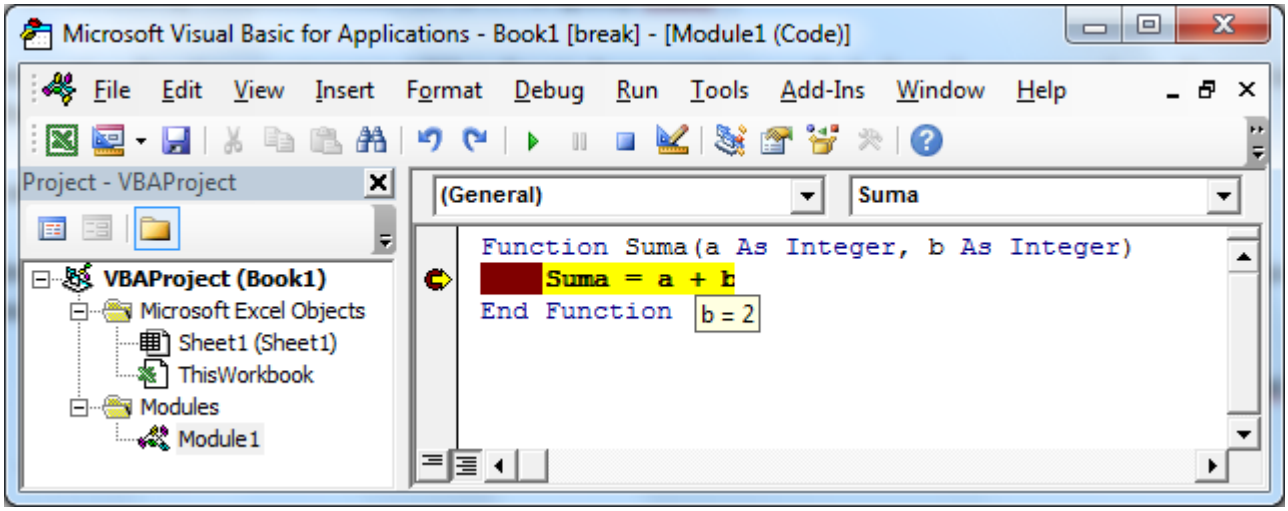
1. Redeschideți VBE
2. In linia Suma = a + b adăugați un punct de oprire (**Debug > Toggle Breakpoint**, sau apăsați tasta **F9**)
3. Activați aplicația **Excel** selectând butonul **Microsoft Excel** in **Taskbar**.
4. Modificați conținutul celulei B2 in 1.5 si apăsați **Enter**

Remarcați ca Excel trece automat in VBE in funcția **Suma** pentru a recalcula formula cu noua valoare. Pentru ca am adăugat un punct de oprire, execuția se oprește în acest punct.

In acest moment suntem in modul **Depanare** (**Debug**)

Putem inspecta valorile variabilelor mutând pointerul mausului peste numele unei variabile si așteptând apariția unui **Tooltip** care afișează valoarea variabilei.

Putem executa pas cu pas liniile de cod apăsând repetat tasta **F8** sau (**Debug > Step Into**)



1. Modificați funcția Suma în **VBE** de forma:

```
Function Suma(a As Double, b As Double)
    Suma = a + b
End Function
```

2. Reveniți in foaia Excel si forțați recalcularea formulei apăsând **CTRL+ALT+F9**
3. Interpretați noile rezultate obținute

Algoritmi ramificați

Vom scrie o funcție VB care evaluează funcția de mai jos (schema logică a funcției a fost prezentată in laboratorul **Algoritmi si scheme logice**):

$$f(x) = \begin{cases} x^2 + 2 \cdot x & \text{pentru } x \leq 0 \\ x + 3 & \text{pentru } 0 < x < 1 \\ 2 \cdot x & \text{pentru } x \geq 1 \end{cases}$$

1. Activați **VBE** si in modulul **Module1** adăugați următorul cod:

```
Function Functie(x As Double) As Double
    If (x <= 0) Then
        Functie = x ^ 2 + 2 * x
    Else
        If (x >= 1) Then
            Functie = 2 * x
        Else
            Functie = x + 3
        End If
    End If
End Function
```

2. Adăugați un punct de oprire la primul **If** si executați funcția pentru diferite valori ale lui x pentru a parcurge fiecare dintre ramurile de execuție.

Algoritmi ciclici cu număr cunoscut de pași

Vom scrie o funcție VB care evaluează factorialul unui număr (schema logică a funcției a fost prezentată in laboratorul **Algoritmi si scheme logice**):

1. Activați **VBE** si in modulul **Module1** adăugați următorul cod:

```
Function Factorial(n As Long)
    Dim i As Long

    Factorial = 1
    For i = 1 To n
        Factorial = Factorial * i
    Next i
End Function
```

2. Adăugați un punct de oprire in linia **Factorial = Factorial * i** si executați aceasta funcție pentru diferite valori ale lui n.

Obs. Vom remarca ca funcția Factorial poate calcula valori ale factorialului pentru numere cuprinse intre 1 si 12. Daca numărul este mai mare decât 12 se produce o depășire a capacității de stocare a variabilelor de tip **Long**.

3. Modificați funcția astfel încât pentru numere mai mari de 12, funcția factorial să returneze un mesaj de eroare.

```
Function Factorial(n As Long)
    Dim i As Long
    If (n > 12) Then
        Factorial = "Depasire"
     Else
        Factorial = 1
        For i = 1 To n
            Factorial = Factorial * i
        Next i
    End If
End Function
```

Inspectarea variabilelor cu Inspectorul

1. Adăugați doua puncte de oprire in liniile Factorial = "Depasire" si Factorial = CLng(1).
2. Mutați cursorul in interiorul variabilei **Factorial** si din meniul **Debug** lansați comanda **Add Watch**.
3. In fereastra de dialog **Add Watch** asigurați-vă ca in câmpul **Expression** apare numele variabilei Factorial
4. Apăsați **Enter.**

Obs: Remarcați deschiderea ferestrei **Watches** in VBE.

5. Re-executați codul funcției Factorial pentru valori ale variabilei n mai mici sau mai mari decât 12 si urmăriți valoarea si tipul variabilei **Factorial** in fereastra **Watches**.
6. Adăugați un bloc **IF** pentru a verifica daca n este negativ. Daca n este negativ returnați din funcția **Factorial** mesajul "Negativ".

```
Function Factorial(n As Long)
    Dim i As Long

    If (n < 0) Then
        Factorial = "Negativ"
    Else
        If (n > 12) Then
            Factorial = "Depasire"
        Else
            Factorial = 1
            For i = 1 To n
                Factorial = Factorial * i
            Next i
        End If
    End If
End Function
```

Algoritmi ciclici cu număr necunoscut de pași

Vom scrie o funcție VB care calculează e^x cu eroarea absoluta ε.

Indicație: Pentru calculul lui e^x vom folosi dezvoltarea in serie:

$$e^x = \sum_{i=1}^{\infty} u_i = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} \quad \text{, cu} \quad u_i = \frac{x^i}{i!} = \frac{x}{i} \cdot u_{i-1}$$

Vom inițializa variabilele e cu zero, i (variabila contor) cu 1, respectiv variabila u cu 1.

1. Adăugați in modulul Module1 codul sursa de mai jos.
2. Inserați in foaia Excel aceasta funcție si calculați valoarea exponentului pentru diverse valori ale lui x.
3. Verificați valoarea obținuta de funcția noastră cu valoarea rezultata prin folosirea funcției din librăria Excel de funcții **EXP**.
4. Adăugați puncte de oprire pentru a parcurge algoritmul pas cu pas in mod **Debug**.

```
Function ExpX(x As Double)
    Const Eps = 0.0001
    Dim u As Double
    Dim i As Integer

    ExpX = 0
    u = 1
```

```
    i = 0
    Do While (u > Eps)
         ExpX = ExpX + u
         i = i + 1
         u = u * x / i
    Loop
End Function
```

Obs. Observați ca am adăugat o constanta declarata cu cuvântul cheie **Const**. Pentru ciclul cu număr necunoscut de pași condiționat anterior am folosit instrucțiunea **Do While … Loop**.

Vom scrie o funcție VB care extrage rădăcina pătrata dintr-un număr a, cu o precizie constantă ε.

Indicație: Se știe că șirul (x_n) definit prin

$$x_1 = a,\ \ x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right)$$

converge la $\sqrt{a}$. Limita $\sqrt{a}$ se aproximează prin acel termen al șirului x_n pentru care $\lfloor x_n - x_{n-1} \rfloor < \varepsilon$.

```
Function Radical(a As Double) As Double
    Const Eps = 0.0001
    Dim xn1 As Double
    Dim xn As Double
    xn = a
    Do
        xn1 = xn
        xn = (xn1 + a / xn1) / 2
    Loop While (Abs(xn - xn1) >= Eps)
    Radical = xn
End Function
```

Obs. Pentru calculul modulului unei valori, folosim funcția VB **Abs**. Pentru ciclul cu număr necunoscut de pași condiționat posterior am folosit instrucțiunea **Do … Loop While.**

1. Adăugați in modulul Module1 codul sursa de mai sus.
2. Inserați in foaia Excel aceasta funcție si calculați valoarea radicalului pentru diverse valori ale lui a.
3. Verificați valoarea obținuta de funcția noastră cu valoarea rezultată prin folosirea funcției din librăria Excel de funcții **SQRT**.
4. Adăugați puncte de oprire pentru a parcurge algoritmul pas cu pas in mod Debug.