

Laborator 13

Automatizarea procesului de modelare utilizând API

10.1 Scop

Lucrarea prezintă descrierea amănunțită a etapelor pentru începerea programării cu modulul API (Application Programming Interface) din Solidworks. Modulul API este utilizat cu scopul automatizării diferitelor operații redundante care intervin pe parcursul proiectării produselor.

10.2 Obiective

- Formarea abilităților de creare și editare a macrocomenzilor și aplicării și utilizării seturilor de instrumente bazate pe librăria Microsoft Visual Basic for Applications (VBA):
- Crearea de interfețe de tip UserForm și integrarea lor în meniul de bază al softului;
- Aplicarea metodelor de lucru învățate prin înregistrarea și editarea macrourilor pentru modele de piese industriale.

10.3 Noțiuni introductive

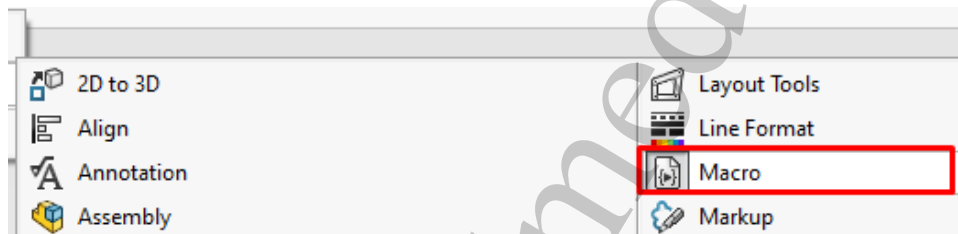
În zilele noastre integrarea proceselor de automatizare a diferitelor operații care sunt utilizate în cadrul proiectării asistate de calculator a componentelor industriale reprezintă un domeniu de interes. Softul SolidWorks conține un modul de lucru cu macrocomenzi și cod care utilizează sintaxă de programare sub trei limbaje de programare principale: Visual Basic Application Programming Language (VBA), C# și C++. Datorită flexibilității utilizării și a avantajului că permite utilizatorilor să scrie programe în Windows, VBA este recomandat pentru utilizatori începători. Un mare avantaj al creării de aplicații în modulul API îl constituie faptul că operațiile aferente creării modelului grafic sunt înregistrate și salvate în cod specific limbajului de programare. De asemenea, rularea codului se poate face pas cu pas pentru fiecare comandă realizată pentru modelarea piesei, iar în situația în care apar linii de cod care nu influențează compilarea corectă a codului, acestea pot fi înlăturate (de exemplu linii de cod care sunt generate în urma operației de rotire a piesei). În cadrul acestei lucrări se prezintă toate etapele de realizare

parametrizată a unui model de șaibă (secțiune inelară), respectiv crearea unei interfețe de comunicare între utilizator și soft în care se introduc parametri dimensionali ai modelului: diametrul exterior, diametrul interior și grosimea. La finalul lucrării interfața realizată este integrată în cadrul instrumentelor din meniul softului.

10.4 Modelarea tridimensională a unei piese și crearea macrourilor

În această parte a lucrării se prezintă pașii necesari, instrucțiunile utilizate pentru înregistrarea macrocomenzilor și generarea codului specific limbajului de programare VBA.

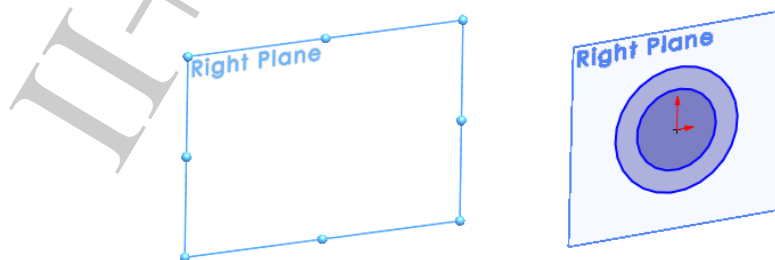
1. Punctul de start al lucrării este dat de introducerea comenzilor specifice modulului **Macro** din SolidWorks. După deschiderea softului SolidWorks cu click mouse stânga pe bara de instrumente, din meniul deschis se alege modulul Macro.



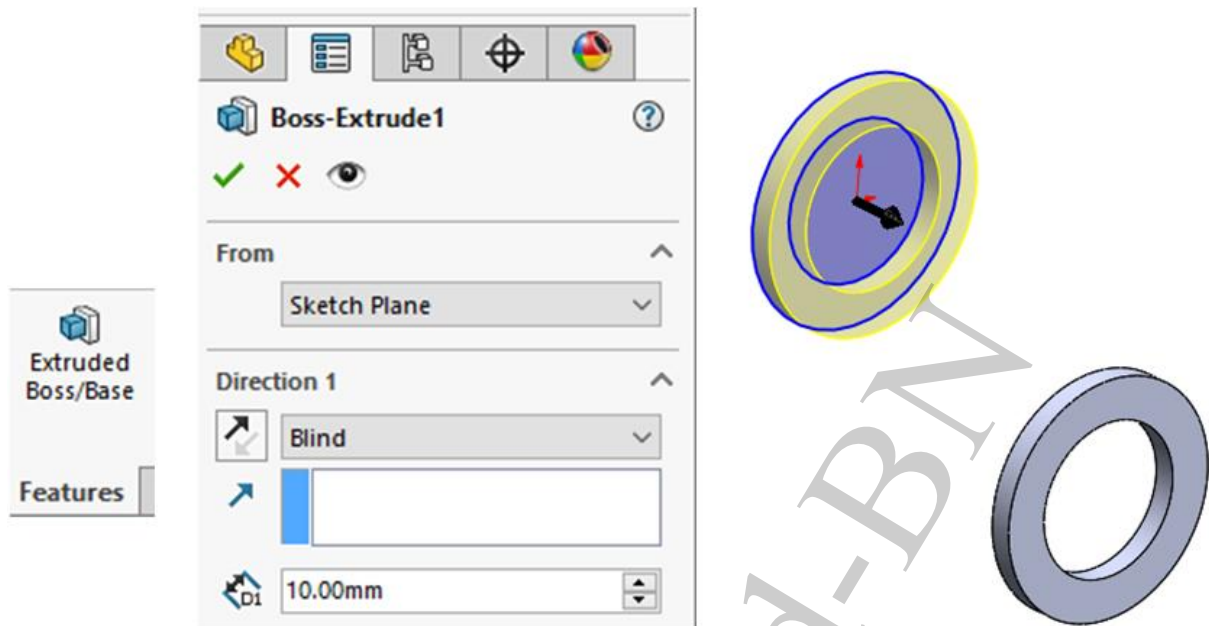
2. Se înregistrează comenzile următoare prin alegerea comenzii **Record/Pause Macro**, din bara de instrumente Macro, după care se deschide un fișier nou de tip part.



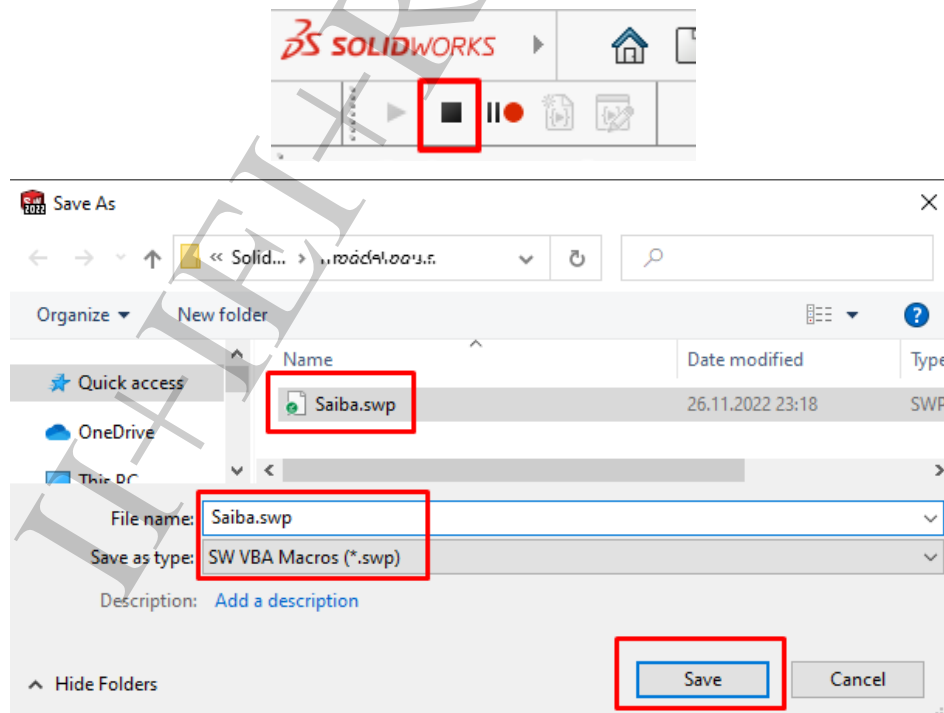
3. Schița piesei este alcătuită din două cercuri selectate din bara de instrumente **Sketch**, desenate pe planul **Right**. Diametrul celor două cercuri nu se cotează, dimensiunile fiind lăsate implicite. Nu se cotează dimensiuni în cazul în care se înregistrează operațiile cu scopul realizării de macro-uri, deoarece nu există funcții în limbajul de programare aferente operațiilor din cadru comenzii **Smart Dimension**.



4. Schița este extrudată cu comanda **Extruded Boss/Base**, din bara de instrumente **Features**. Distanța de extrudare nu este indicată, această valoare este lăsată implicit.

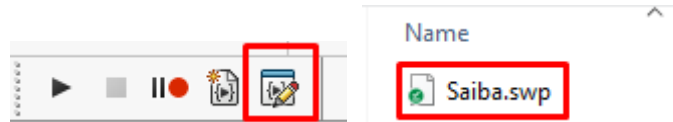


5. Din bara de meniuri (în partea superioară a ecranului) se alege comanda **Rebuild**, pentru actualizarea proprietăților modelului, după care se salvează modelul tridimensional într-un folder de pe hard.
6. De la comanda **Stop Macro**, din bara de instrumente **Macro** se oprește înregistrarea operațiilor. În momentul în care se oprește înregistrarea, se deschide automat caseta de dialog pentru salvarea fișierului care conține operațiile de modelare a piesei, convertite în cod aferent VBA.

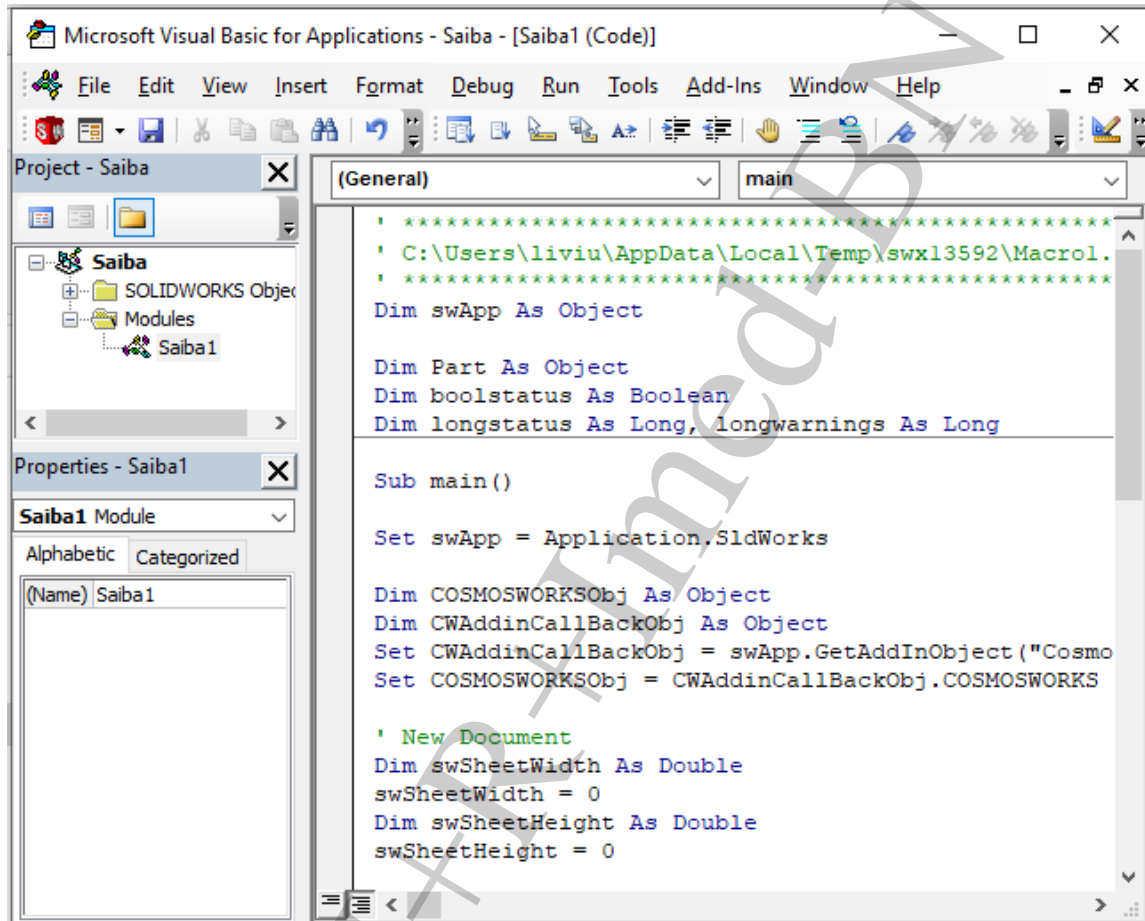


7. Deschiderea fișierului *Saiba.swp* se face prin intermediul comenzii **Edit Macro**, de unde se

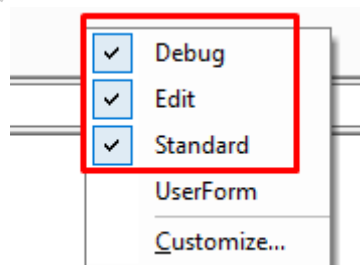
deschide caseta de dialog cu folderul care conține fișierul, apoi se alege fișierul.



8. Interfața aplicației de editare conține, zona de lucru, bara cu instrumente standard, bara de meniuri, bara de editare, etc.

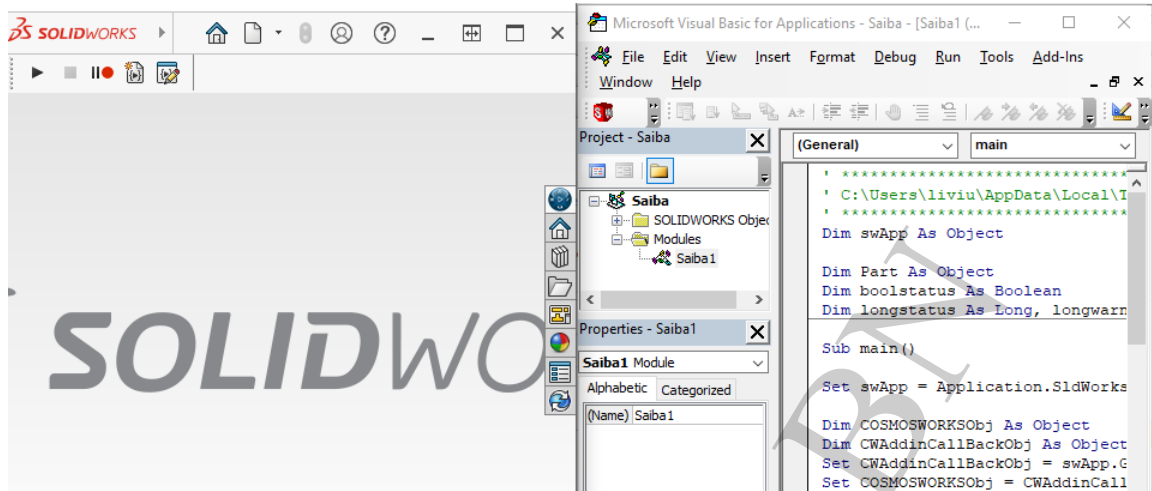


9. Cu click stânga mouse pe bara de instrumente se deschide un submeniu din care se selectează afișarea barelor de intrumente **Debug**, **Edit** și **Standard**, necesare operațiilor următoare.



10. Se dorește rularea pas cu pas a codului generat în urma înregistrării operațiilor de modelare a piesei. Pentru observarea poziției cursorului pe liniile de cod și simultan poziția de modelare a piesei, ferestrele aplicațiilor sunt poziționate vertical, după cum se poate observa în

imaginea de mai jos

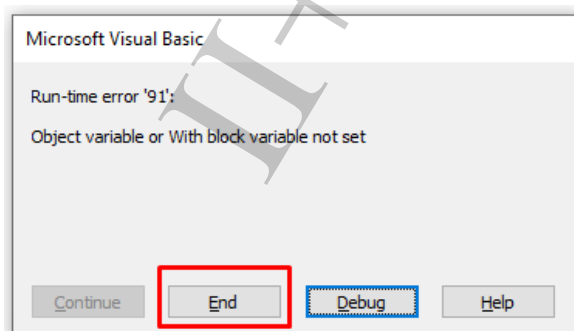


11. Pentru rularea pas cu pas a codului generat se folosește comanda **Step Into**, din bara de instrumente **Debug**, sau scurtătura aferentă acestei comenzi, tasta **F8**. Codul se va rula pas cu pas analizând simultan și crearea modelului tridimensional al șaibe. Pe parcursul rulării codului se observa ca exista linii din cod care nu sunt necesare, cum ar fi liniile de cod care definesc pozitia de rotatie a piesei cu mouse-ul de către user.



Aceste linii se transformă în comentarii prin selecția textului și alegerea comenzii **Comment Block**, din bara de instrumente **Edit**, introdusă în interfață la punctul precedent.

12. La rularea pas cu pas pot apărea unele situații în care unele variabile nu sunt definite automat. Caseta de dialog care apare în acest caz poate termina operația de compilare selectând comanda **End**, poate da posibilitatea depanării codului selectând comanda **Debug** sau pune la dispoziție utilizarea **Help**-ului. În cazul în care se constată că nu sunt necesare acele linii de cod la compilarea corectă a codului, acestea se transformă în comentarii, după cum se poate vedea în imagine de mai jos.



```
' Save As
longstatus = Part.SaveAs3("C:\Users\liviu\AppData\Local\Temp\91\StudyManagerObj = Nothing
'StudyManagerObj = Nothing
'ActiveDocObj = Nothing
'Set CWAddinCallBackObj = Nothing
'Set COSMOSWORKSObj = Nothing
End Sub
```

După rularea pas cu pas a codului și transformarea liniilor de cod în blocuri de tip comentariu, se mai rulează încă o dată codul, cu scopul de a observa dacă nu apar erori la generarea modelului.

13. Dacă nu apare nici o eroare, liniile de cod care sunt în plus (transformate în comentarii) se vor șterge, iar codul se organizează în blocuri cu comentarii pentru fiecare comandă. Codul organizat este prezentat în imaginea de mai jos.

```

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long

```

```

Sub main()
Set swApp = Application.SldWorks
' Crearea unui nou document
Set Part = swApp.NewDocument("C:\ProgramData\SolidWorks\SOLIDWOR
Dim swPart As PartDoc
Set swPart = Part
swApp.ActivateDoc2 "Part1", False, longstatus
Set Part = swApp.ActiveDoc
Dim myModelView As Object
Set myModelView = Part.ActiveView
myModelView.FrameState = swWindowState_e.swWindowMaximized
boolstatus = Part.Extension.SelectByID2("Annotations", "DCABINET
'selectie plan Right
boolstatus = Part.Extension.SelectByID2("Right Plane", "PLAN
'inserare schita
Part.SketchManager.InsertSketch True
Part.ClearSelection2 True
'desenarea cercului - diametrul exterior al
Dim skSegment As Object
Set skSegment = Part.SketchManager.CreateCir
'desenarea cercului - diametrul interior
Part.ClearSelection2 True
Set skSegment = Part.SketchManager.Creat
' Extrudarea schitei-crearea modelul
Dim myFeature As Object
Set myFeature = Part.FeatureManager.
'reconstructia modelului
Part.SelectionManager.EnableContourSelection = False
boolstatus = Part.EditRebuild3()
Part.ClearSelection2 True
' Salvarea modelului de saiba pe hardul computerului
longstatus = Part.SaveAs3("C:\Users\Liviu\My Drive\000000.DRIVE

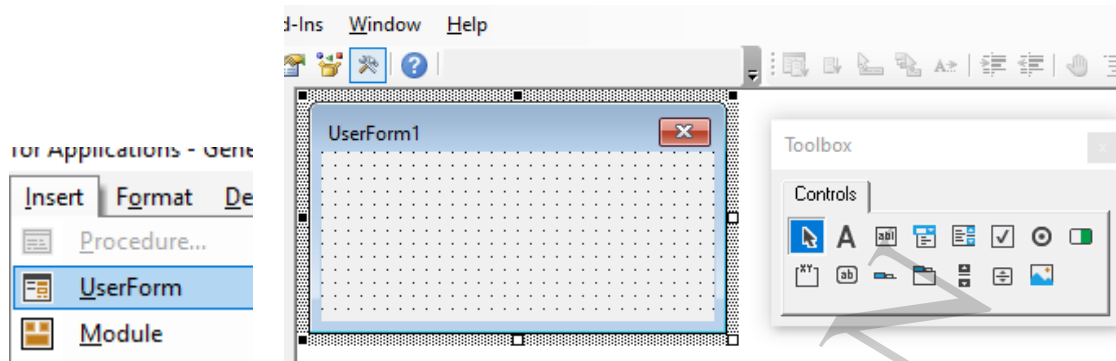
```

```

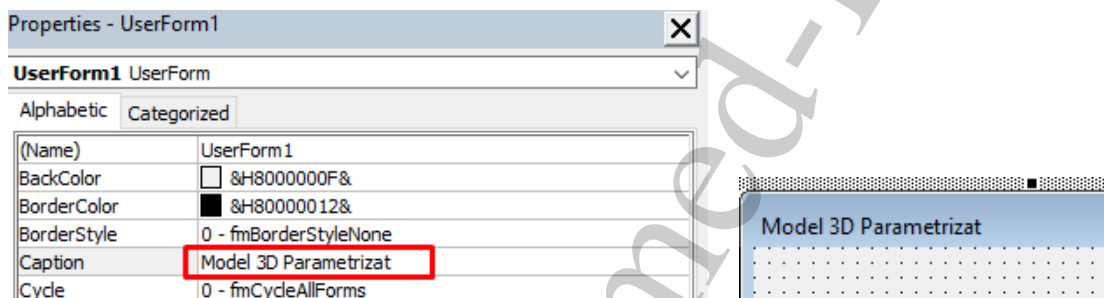
End Sub

```

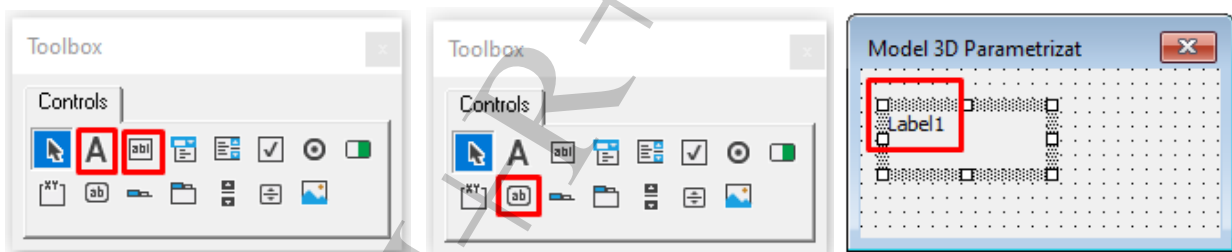
14. Pentru crearea unei interfețe de comunicare între utilizator și aplicație este necesară crearea unei casete de dialog (UserForm). Acest form se creează din meniul **Insert**, Comanda **UserForm**. Se deschide pagina pe care se vor putea adăuga câmpurile de introducere a datelor, denumiri și butoane. Aferent acestei pagini apare și caseta de dialog care conține instrumentele necesare introducerii comenzilor în pagina aferentă formului.



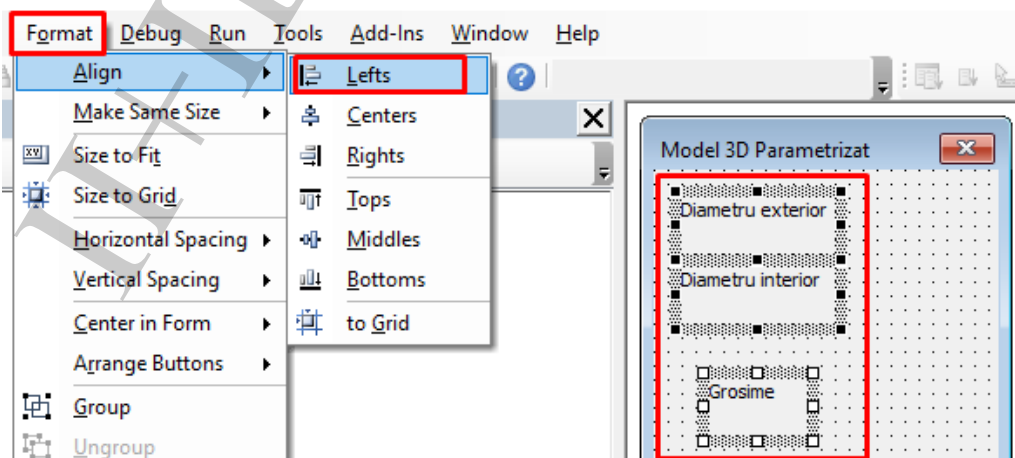
15. Caseta de dialog de tip form se denumește *Model 3D Parametrizat*, din caseta de dialog poziționată în partea stângă a ecranului.



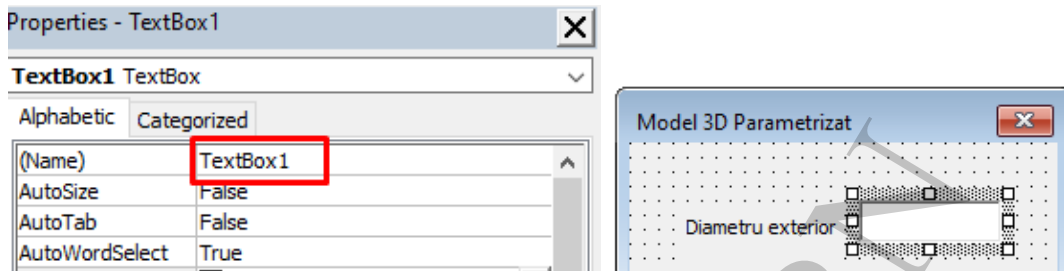
16. Introducerea *Denumirilor (etichetelor)*, a căsuțelor text și a butoanelor de comandă se face din caseta **Toolbox**.



17. Se introduc etichete pentru fiecare variabilă, căsuțe text și butonul de comandă. Etichetele introduse se denumesc în funcție de variabilele care definesc parametrii șabei.

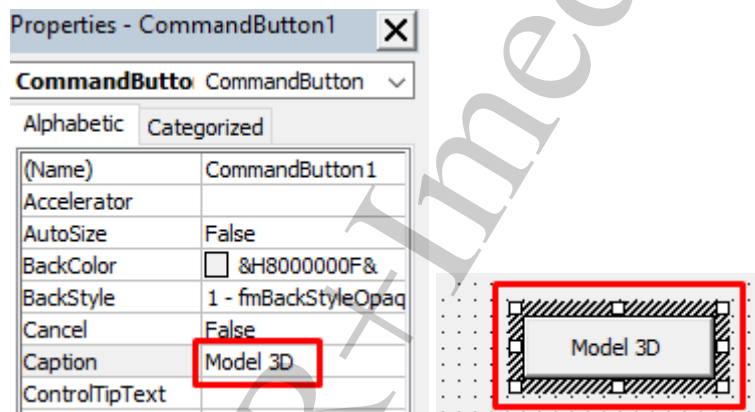


Denumirile introduse în pagină se pot alinia din meniul **Format, Align, Lefts**. Variabilele care se introduc de la tastatură de către utilizator sunt: *Diametrul exterior*, *Diametrul interior* și *Grosimea șaibe*.

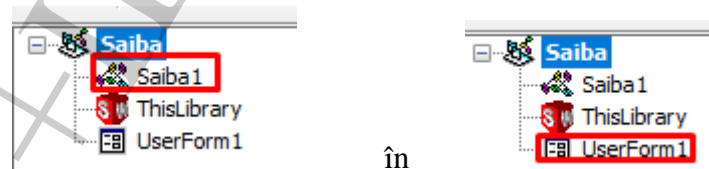


Se are în vedere denumirea fiecărui **TextBox**, deoarece în etapele următoare, denumirea lor va fi introdusă și apelată la compilarea codului.

18. Butonul introdus în interfață se va redenumi în **Model 3D**, prin selecția acestuia și completarea denumirii în caseta **Properties**.



19. Pentru o mai bună gestionare a codului, acesta se copiază în totalitate din fișierul *Saibal* (CTRL+A, selecție, iar CTRL+C, copiere) în fișierul *UserForm1* (se selectează cu dublu click mouse caseta *UserForm1*). Din fișierul deschis se șterg toate liniile de cod și se lipește codul din fișierul *Saibal* (CTRL+V).



20. În fișierul *Saibă1*, rămas gol se scrie secvența de cod de mai jos, care face posibilă afișarea casetei de dialog de introducere a parametrilor șaibe de către utilizator.

```
Sub main ()
    UserForm1.Show
End Sub
```


21. Pentru a face legătura dintre datele introduse de utilizator și comenzile de creare a schițelor cercurilor și a comenzii de extrudare, în cod se fac următoarele modificări:

- pentru desenarea diametrului exterior al șabei în poziția alocată valorii numerice a razei se introduce numele **TextBox**-ului alocat introducerii de la tastatură a diametrului. Având în vedere faptul că unitatea de măsură este în sistem internațional (metri), iar campul de introducere este câmpul alocat valorii numerice a razei cercului, diametrul introdus de la tastatură se împarte la 2, iar apoi ca să fie în milimetri se împarte la 1000,

```
'desenarea cercului - diametrul exterior al saibei
Dim skSegment As Object
Set skSegment = Part.SketchManager.CreateCircle(0#, 0#, 0#, 0, (TextBox1 / 1000) / 2, 0#)
```

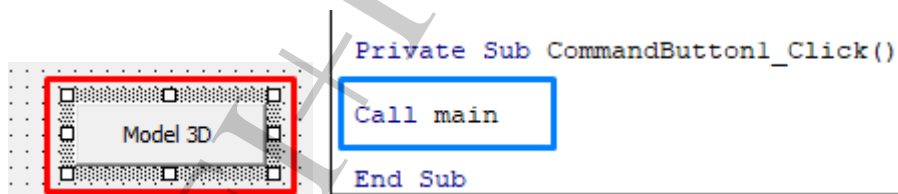
- modificarea în linia de cod aferentă diametrului interior este prezentată în figura de mai jos,

```
'desenarea cercului - diametrul interior al saibei
Part.ClearSelection2 True
Set skSegment = Part.SketchManager.CreateCircle(0#, 0#, 0#, 0, (TextBox2 / 1000) / 2, 0#)
```

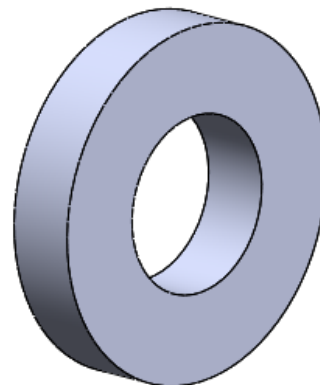
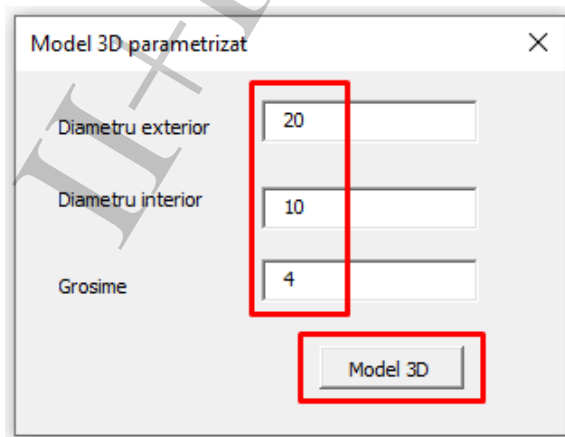
- pentru realizarea legăturii între valoare grosimi introduse prin interfață de la tastatură și cod se realizează prin completarea textului încadrat în secvența de cod de mai jos.

```
' Extrudarea schitei-crearea modelului saibei
Dim myFeature As Object
Set myFeature = Part.FeatureManager.FeatureExtrusion2(True, False, False, 0, 0, TextBox3 / 1000, 0.01,
```

22. Realizarea legăturii dintre datele introduse și funcția de generare a modelului se face apelând funcția “**main**”, în zona destinată butonului **Model 3D**. Calea spre locul introducerii sintaxei de apelarea a funcției se face cu dublu click stânga pe suprafața butonului).

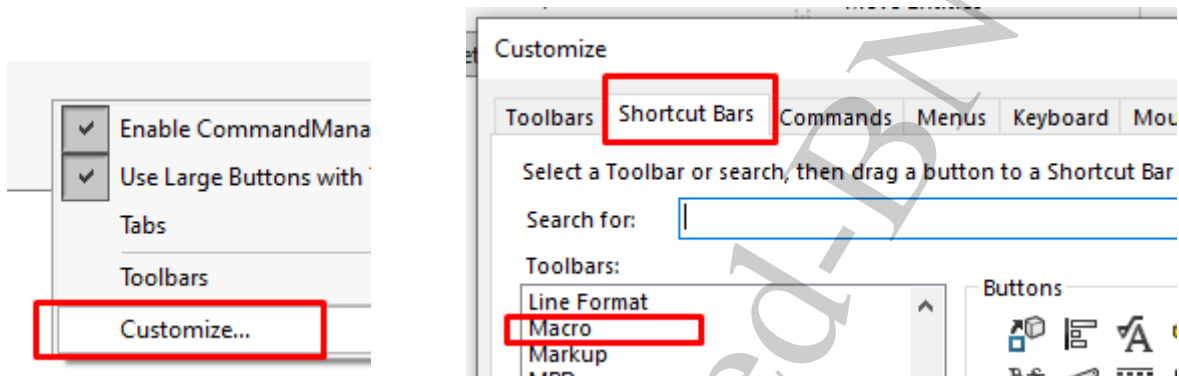


23. Din bara de instrumente se rulează codul, alegând comanda **Run**.

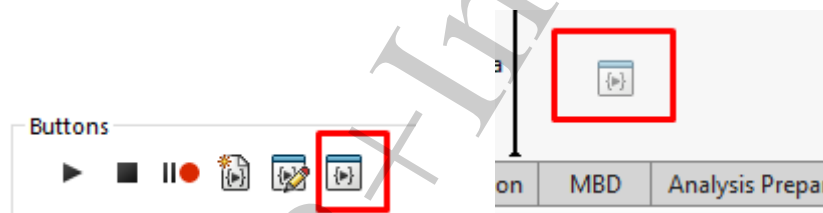


Se introduc dimensiuni de testare, generându-se modelul tridimensional, de asemenea modelul se salvează pe hard, la adresa menționată în cod.

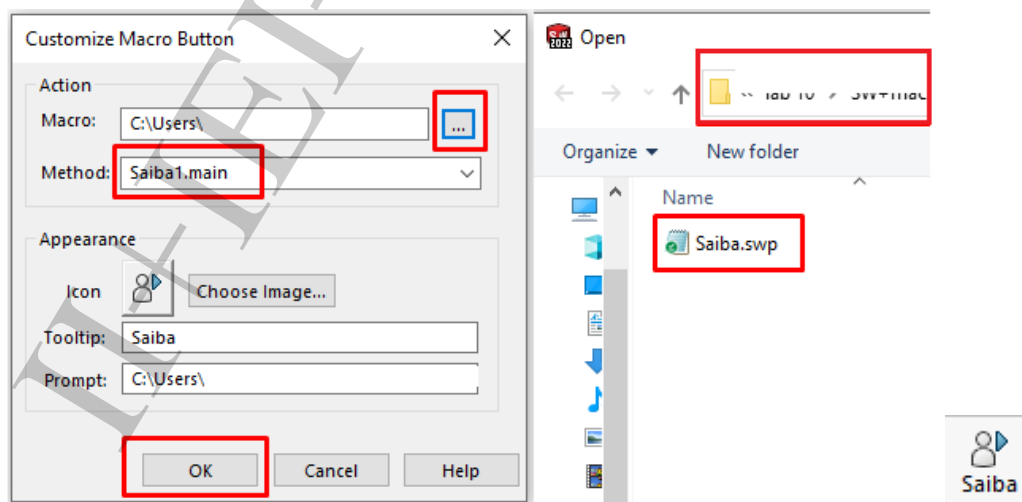
24. Pentru a mări versatilitatea utilizării aplicației, aceasta se poate introduce în bara de instrumente standarde a softului, de la comanda **Customize...** (click dreapta mouse pe bara de instrumente din SoliWorks)



25. Din bara de instrumente Macro care apare în caseta de dialog Customize, se trage cu mouse-ul (click stânga) butonul aferent comenzii New Macro Button pe bara de instrumente.



26. Se deschide automat caseta de dialog de mai jos, iar la **Macro:** se alege fișierul finalizat în interfața VBA. Automat se completează și câmpul de la secțiune **Method** cu funcția care se apelează la deschiderea aplicației. După finalizarea comenzii, în bara de instrumente apare pictograma **Saibă**.

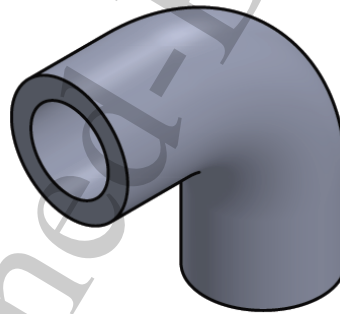
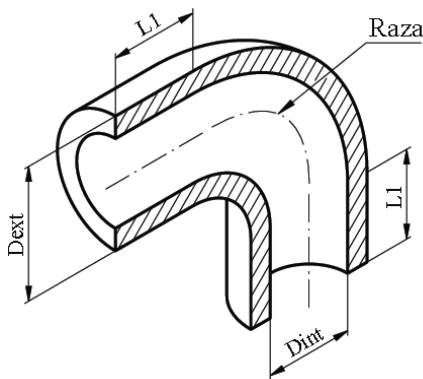


Finalizarea aplicației este completată de încă o rulare, cu scopul verificării modului de lucru al aplicației și a corectitudinii modelului generat.

10.6 Aplicații propuse

1. Utilizând comenzile aferente modului Macro, să se dezvolte o interfață de tip UserForm care să genereze modele de piese corespunzătoare modelului din figura 10.1. La crearea interfeței se va avea în vedere crearea variabilelor de intrare după cum urmează:

- Dext - diametrul exterior al piesei
- Dint - diametrul interior al piesei
- Raza – raza de racordare
- L1 – lungimea țevii (egală la ambele extremități)



a) Definirea datelor de intrare

b) Modelul piesei generat din aplicație

Fig. 10.1 Modelul piesei

2. Să se realizeze o aplicație care să genereze modele tridimensionale având configurația piesei din figura 10.2. Variabilele care definesc forma piesei sunt:

- L – lungimea piesei
- W – lățimea piesei
- H – înălțimea piesei

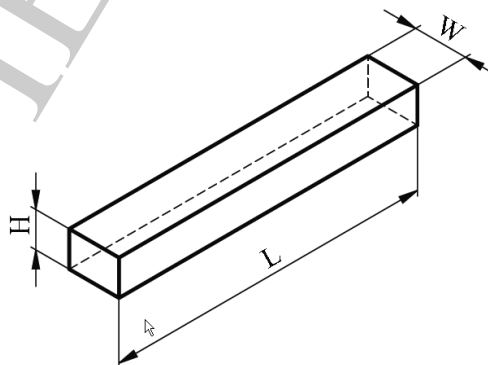


Fig. 10.2 Reprezentarea modelului piesei