

Hyrje në VBA

Çfarë është Visual Basic?

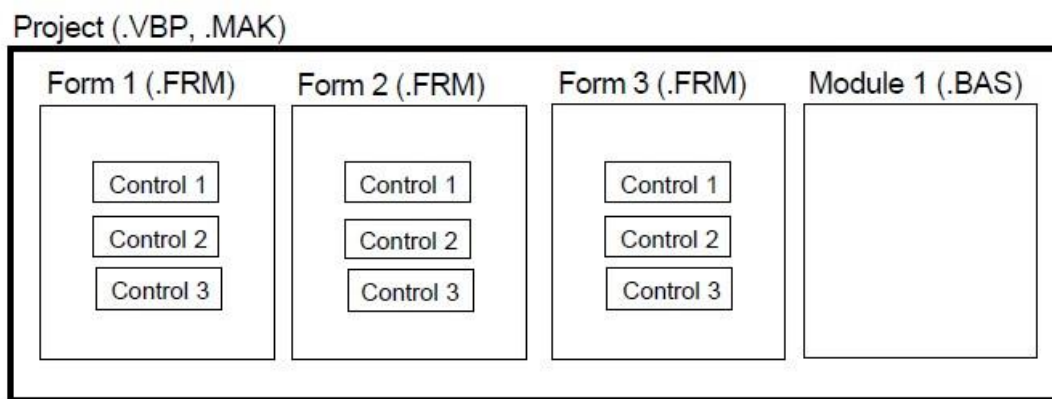
Visual Basic është një mjet që përdoret për të programuar aplikacione në Windows. Disa nga cilësitë e Visual Basic janë si më poshtë:

1. përmban një gamë të gjerë objektsh, që ndihmojnë në 'vizatimin' e një aplikacioni e më pas implementimin funksional të tij;
2. përmban shumë ikona dhe piktura, të cilat mund të përdoren në aplikacione të ndryshme;
3. jep mundësinë në kapjen e ngjarjeve të mousit dhe tastierës;
4. jep mundësinë në kapjen e eventeve të printerit;
5. ofron vektorë matematikë, funksione grafike dhe trajtime të teksteve të ndryshme, që në Visual Basic emërtohen si String;
6. ofron suport sekuencial dhe të rastësishëm të skedarëve të ndryshëm;
7. ofron disa mekanizma që quhen debug dhe trajtim gabimesh, të cilat janë mjete shumë të rëndësishme për programuesit për të gjetur gabimet dhe rregulluar problemet e ndryshme të një aplikacioni;
8. ofron dhe paketa, të cilat bëjnë shpërndarjen e aplikacioneve të thjeshta;

Janë cilësitë e mësipërme që i japin Visual Basic karakteristikat që të konsiderohet si një nga mjetet më të preferuara nga programuesit për të ndërtuar aplikacionet e tyre me natyrë Desktop, dhe që përgjithësisht, janë të orientuar drejt platformës Windows. Në vazhdim, do t'i referohemi Visual Basic me emrin gjuhë programimi, duke qenë se është i tillë, dhe shpesh herë do t'i referohemi edhe me shkurtimin VB.

Struktura Bazë e një Aplikacioni në Visual Basic (VB)

Struktura e një aplikacioni në visual studio ka pak a shumë pamjen e treguar në figurën më poshtë:



Sikurse shikohet edhe nga figura, dallohen cilat janë pjesët kryesore të një aplikimi (projekti) në VB. Po i listojmë shkurtimisht më poshtë këto elemente, për tu ndalur më vonë më gjerë në shpjegimin e tyre.

Format – përfaqësojnë dritaret që ndërtohen dhe që i referohemi si ndërfaqe përdoruesi.

Kontrollet – janë elemente grafike, të cilët vendosen në forma për të ofruar interaktivitet me përdoruesin. (P.sh., text boxe, label, scroll bar, butonat etj. Vini re që Format dhe Kontrollet janë objekte në vetvete).

Karakteristikat – çdo element i një forme apo kontrolli specifikohet (Properties) nga një karakteristikë, e cila përcaktohet në mënyrë shumë të detajuar dhe nga një meny e veçantë në VB, për të cilën do flasim më vonë. VB shpesh herë i atribuon aplikacioneve veti të paracaktuara, të cilat mund të ndryshohen gjatë kohës së dizenjimit, ose gjatë kohës së ekzekutimit.

Metodat – janë procedura built-in që thirren për të kryer një ose disa veprime mbi një objekt të caktuar. (Procedura built-in përfaqësojnë procedura standarde që janë para programuara në VB, janë standarde dhe mund të përdoren në aplikacionet që ndërtohen në VB).

Procedurat e Ngjarjeve – përfaqëson një kod që lidhet me disa objekte. E veçanta e këtij kodi është se ai ekzekutohet vetëm kur ndodh një ngjarje (event) e caktuar.

Procedurat e përgjithshme – përfaqëson një kod që nuk lidhet me objekte. Ky kod thirret nga aplikacioni.

Modulet – përfaqësojnë një koleksion procedurash të përgjithshme, deklarime ndryshoresh dhe konstantesh që përdoren nga aplikacioni. (do ndalemi më vonë në shpjegimin e secilit prej elementeve të një moduli) Çfarëdolloj aplikacioni të jeni duke ndërtuar, ai patjetër do të ketë në përbërjen e tij elementet e mësipërm.

Mjedisi i Punës i VB

Cilat janë hapat që duhet të ndjekim për të programuar apo ndërtuar një aplikacion në VB?

Ekzistojnë tre hapa kryesore që duhen zbatuar për të ndërtuar një aplikacion në VB:

1. vizatimi i ndërfaqes së përdoruesit;
2. bashkëngjitja e propertive (karakteristikave) të kontrolleve që do ketë ky aplikacion;
3. lidhja e kontrolleve me kodin përkatës që do t'i menaxhojë ato;

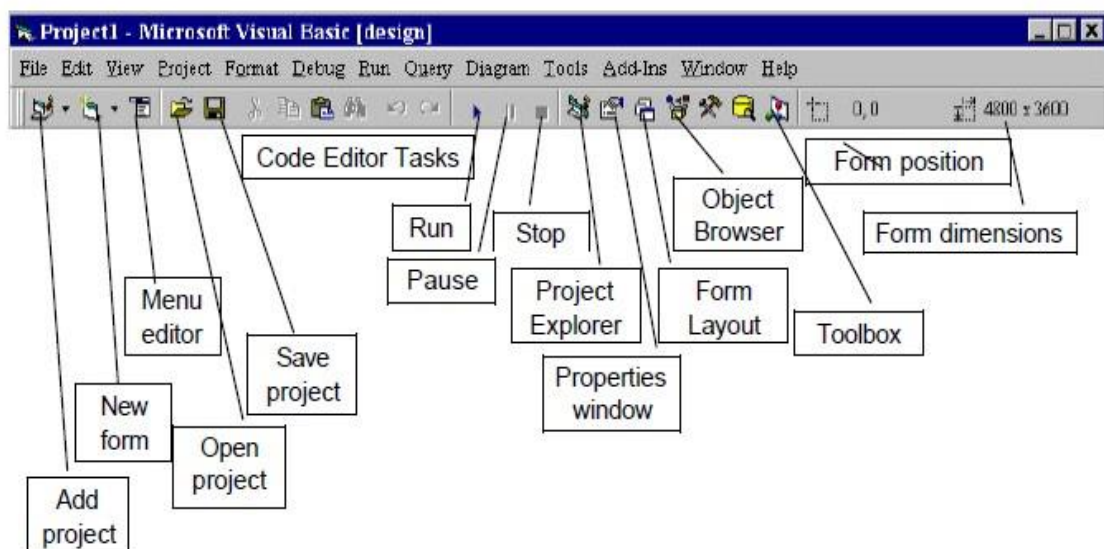
Përpara se të ndalemi në secilin nga hapat e mësipërm, duhet të dini që Visual Basic operon në tre gjendje, ose mënyra:

Design mode – që është një gjendje/mënyrë që përdoret për të bërë aplikacionin. Kjo është gjendja që ne e përdorim Visual Basic për të ndërtuar aplikacionin tonë.

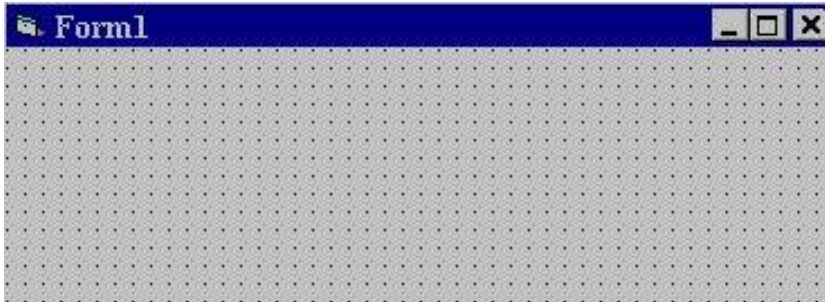
Run mode – që është një gjendje/mënyrë që përdoret për të ekzekutuar aplikacionin

Break mode – që është një gjendje/mënyrë që përdoret për të debuguar aplikacionin. Në momentin që startohet Visual Basic shfaqen gjashtë dritare:

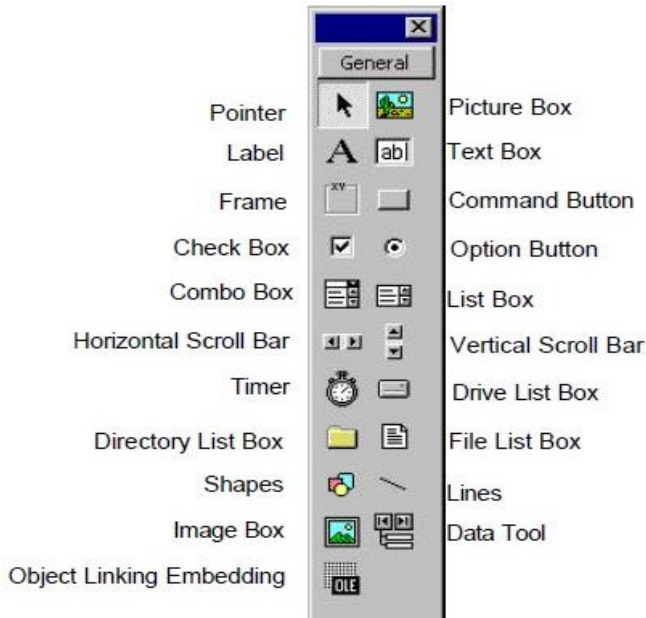
Dritarja Kryesore – e cila konsiston në titullin, meny-të dhe toolbar-i. Tek pjesa e titullit paraqitet emri i projektit dhe jep informacion edhe në lidhje me mënyrën (mode) që është duke punuar visual studio. Në vazhdim do të trajtojmë mënyrën/gjendjen e dizenjimit të visual studios.(design mode ang.). Pjesa e menyve përmban menyte kryesore në formën e një drop down, Ato kanë pamje të ngjashme si menyte e Word-it, Excel-it... etj. Me anë të këtyre menyve mund të kontrollohen veprimet kryesore që mund të realizohen në mjedisin e Visual Basic. Përsa i përket pjesës së toolbar, kjo ofron butonat, që përfaqësojnë shkurtore të disa nga opsioneve të menyve. Një shpjegim më në detaje të kësaj, e realizon edhe figura më poshtë.



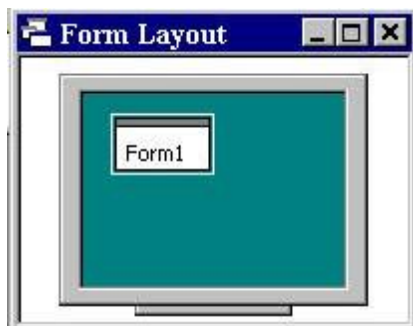
Dritarja e formës – përfaqëson pjesën kryesore, në të cilën zhvillohen aplikacionet. Është pikërisht në këtë dritare që ndërtohet aplikacioni.



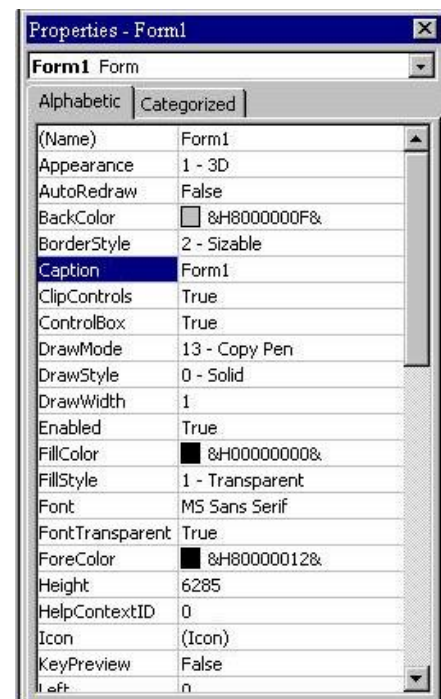
Toolbox – përfaqëson një meny zgjedhje për kontrollet që mund të përdoren në aplikacion



Dritarja e form layout – tregon ku do të shfaqet forma në ekranin apo monitorin e kompjuterit gjatë ekzekutimit të programit.



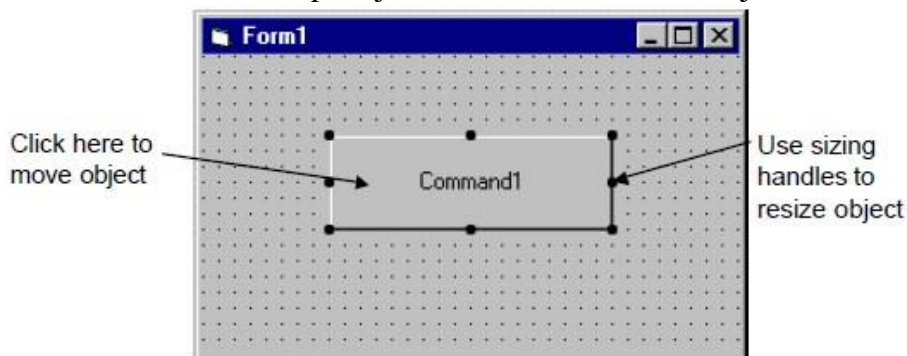
Dritarja e karakteristikave (propertive) – përdoret për të vendosur një vlerë fillestare për properties/karakteristikat për objektet që janë përdorur në formë. Drop-down, në fillim të dritares, liston të gjithë objektet e formës aktuale. Janë të vlefshme dy tipa pamjesh: **alfabetike** dhe e **kategorizuar**. Në këtë dritare ofrohen karakteristikat/properties kryesore për objektin e zgjedhur.



Dritarja e projektit – shfaq listën e të gjitha formave dhe moduleve që përbëjnë aplikacionin. Gjithashtu, mund të bëhet një zgjedhje, nëse duam të shikojmë një pamje të projektit në **Form** apo **Kod**.

Sikurse u përmend më parë, ndërfaqja e përdoruesit ‘vizatohet’ në dritaren e formës. Ka dy mënyra për të vendosur kontrollin në një formë.

- Duke bërë 2-klik në toolbox, krijohet një ndërfaqe me përmasë të paracaktuar (default) në formë. Ne mund ta heqim apo ri-dimensionojmë (resize) këtë ndërfaqe në formë. (Resize do të thotë të ndryshosh përmasat)
- Duke klikuar në menyën tool ne toolbox, dhe pastaj, me anë të mouse-it, të shënjosësh mbi dritaren e formës.



Variablat

Qëllimi kryesor i VB është të manipulojë të dhënat, të cilat i ruan në memorien e kompjuterit. Të dhënat që ruhen, mund të jenë flete pune/worksheet, range ose objekte. VB është një gjuhë programimi, e cila për një formë të dizenuar, e gjeneron vetë kodin e procedurave të ngjarjeve të objekteve të formës. E vetmja gjë që i ngelet përdoruesit, është t’i bashkëngjisë këto procedura në varësi të kërkesës që duhet të përmbushet. Para se të realizojmë shtimin e kodit, është e nevojshme të diskutojmë mbi **variablat**.

Një *variabël* nuk është gjë tjetër veçse emërtimi i një vendi, që ruhet në memorien/kujtesën e kompjuterit. Përdoruesit i ofrohen një sërë fleksibilitetesh në momentin që emërton këtë variabël. Është mirë që emri i variablit të jetë sa më përshkrues të jetë e mundur, dhe jo shumë i gjatë në karaktere. Më poshtë, po paraqesim disa shembuj të përdorimit të variablave në VB:

```
x = 1
InterestRate = 0.075
LoanPayoffAmount = 243089
DataEntered = False
x = x + 1
UserName = "Bob Johnson"
DateStarted = #3/14/2010#
MyNum = YourNum * 1.25
```

VB zbaton disa rregulla në lidhje me emrat e variablave:

- ✓ Ju mund të përdorni shkronja, numra ose karaktere të shenjave të pikësimit, por është e detyrueshme që elementi i parë i variablit të jetë patjetër një karakter i tipit shkronjë.
- ✓ VB s’bën asnjë dallim midis shkronjave të mëdha apo shkronjave të vogla.
- ✓ Ju nuk mund të përdorni karaktere të tilla brenda emrit të variablit tuaj : #, \$, %, &, ose !.
- ✓ Gjithashtu, emrat e variablave mund të jenë të gjatë, por jo më shumë se 255 karaktere. Nëse përdorni variabla kaq të gjatë, atëherë me siguri që do hasni probleme me kodin e programit tuaj VB.

Këshillohet që për t’i bërë më të lexueshme, apo më të kuptueshme variablat, të përdorni një kombinim midis shkronjave të vogla dhe shkronjave të mëdha si p.sh.: për të deklaruar Interest Rate është mirë të përdoret variabli InterestRate, ose duke përdorur një nënvijë p.sh. interest_rate. VB ka disa fjale kyçe, ose fjalë me kuptim, të cilat kanë një domethënie të veçante, dhe të cilat kursesi nuk mund t’i përdorni për të emërtuar variablat tuaja. Fjalët e tilla janë: *Sub, Dim, With, End, Next* dhe *For*. Nëse përpiqeni për të përdorur një nga

këto fjalë si emra variablash, atëherë do merrni një mesazh gabimi gjatë kompilimit të kodit tuaj. Kështu që kur të ndërtoni procedurën apo funksionin tuaj në VB, kini parasysh të gjitha rregullat e mësipërme.

Tipet e të dhënave në VBA

Data Type	Bytes Used	Range of Values
Boolean	2	True or False
Integer	2	-32,768 to 32,767
Long	4	-2,147,483,648 to 2,147,483,647
Single	4	-3.402823E38 to 1.401298E45
Double (negative)	8	-1.79769313486232E308 to -4.94065645841247E-324
Double (positive)	8	4.94065645841247E-324 to 1.79769313486232E308
Currency	8	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Date	8	1/1/100 to 12/31/9999
String	1 per char	Varies
Object	4	Any defined object
Variant	Varies	Any data type
User defined	Varies	Varies

1. Kur flitet për tipa të dhënash, i referohemi mënyrës në të cilën një program ruan të dhënat në memorie p.sh.: **numra të plotë**, numra **reale** apo **stringje**. Në gjuhë të ndryshme programimi, përdoren emërtime të ndryshme dhe tipa të ndryshëm për të ruajtur të dhënat. Duhet të kini parasysh që VB jo gjithmonë e kërkon deklarimin e variablove, por për të shmangur problemet, është mirë që të mësoheni me këtë praktikë, domethënë deklarimin gjithmonë të variablove që po përdorni. VB ka një varietet tipa të dhënash, tabela më sipër jep një listë të tipave të të dhënave më të zakonshme që përdoren në të.
2. **Date** - një tjetër tip e dhëne, që përdoret më së shumti në VB. Për të ruajtur këtë tip të dhëne mund të përdoret edhe tipi i të dhënës **String**, por mbani parasysh që për këto raste **nuk do keni mundësi** të kryeni veprime me datat. Ndërkohë, përdorimi i tipit të dhënës Date, ofron mundësinë e realizimit të veprimeve me datën, p.sh. llogaritja e diferencës së ditëve midis dy datave.
Më poshtë tregohen disa shembuj të deklarimit të datave:

```
Dim Today As Date
Dim StartTime As Date
Const FirstDay As Date = #1/1/2010#
Const Noon = #12:00:00#
```

3. **String** është një bashkësi karakteresh (segment teksti). Në variablin e tipit string mund të përfshihen gërmat e mëdha, të vogla, shenjat e pikësimit, numrat, (këto numra nuk mund të përdoren në llogaritje pasi trajtohen si tekste). Ne do të përdorim tipin *string* për të emërtuar objekte, etiketa objektivësh, dhe etiketa të dhënash në faqe libri Excel.
4. **Boolean** është një variabël, vlera e të cilit është ose *True* ose *False*. Ne do të përdorim variablat *Boolean* në deklarimet logjike, deklarimet e tipit *If*, *Then*, *ciklet* (loops).
5. **Range** është një variabël, i cili mund të përdoret me të gjitha vetitë dhe metodat e objektit të tipit Range.
Ne duhet gjithashtu, të përdorim deklarimin **Set për të inicializuar** vlerën e një variabli të tipit *range*.

Për shembull, për të vendosur *MyRange* të barabartë me qelizën *A1* duhet të shtypim si më poshtë:

$$\text{Set MyRange} = \text{Range}(\text{"A1"})$$

Tipi i të dhënave **Range** përdoret për të përcaktuar p.sh. çfarë qelize do të jetë si **StartCell**. Ne përdorim Ranges për ta bërë kodin tonë më dinamik. P.sh. përdorim Input Box për të përcaktuar se ku përdoruesi kërkon të fillojë një tabelë.

Për shembull:

```
Dim text As String,
StartCell As Range
text = InputBox("shkruaj qelizën fillestare për tabelën:")
Set StartCell = Range(text)
```

6. **Worksheets.** Tipi i të dhënave **worksheets** përcakton një objekt flete libri Excel. Ne përdorim këtë tip objekti kur krijojmë cikle dhe/ose funksione që shërbejnë për të kryer metoda që përsëriten, apo për të vendosur veti që përsëriten për shumë fletë pune (worksheets).
7. **Object.** Variabli **object** përdoret për të përcaktuar çdo objekt në Excel. Për shembull një objekt mund të jetë një objekt vizatimi, një rang, një fletë, ose ndonjë lloj tjetër. Secila prej vetive ose metodave të objektit mund të përdoret me variablin objekt që i shoqërohet.

Deklarimi dhe shtrirja e ndryshoreve (variablave)

Sikurse u tha, jo domosdoshmërisht duhet të deklaroni variablat në një program VB. Nëse nuk e bëni një gjë të tillë, atëherë vetë VB do t'i konsiderojë këto variabla si variabla 'variant', të cilat konsiderohen që veprojnë si kameleon, pra që ndryshojnë tipin e tyre në varësi të veprimit që duam të realizojmë me to. P.sh. nëse një variabël është i tipit *variant*, dhe ruan vlera të tipit string p.sh. "143", nëse me këtë variabël duam të realizojmë veprime të tipit numër, atëherë është e mundur të realizohet duke e marrë vlerën "143" në thjesht 143, pra pa thonjëza. Mënyra më e thjeshte për të deklaruar një variabël është shprehja **Dim**. Më poshtë keni disa shembuj variablash të deklaruara:

```
Dim YourName As String
Dim AmountDue As Double
Dim RowNumber As Long
Dim X
```

Sikurse shikohet, tre variablat e para janë përcaktuar saktë se kujt tipi i përkasin, ndërkohë variabli i katërt nuk e ka këtë përcaktim. Rrjedhimisht, ky variabël, konsiderohet si variabël 'variant' për të cilin diskutuam.

<i>Shtrirja e variablave</i>	<i>Si deklarohet variabli</i>
Vetëm në procedurë	Duke përdorur shprehjen Dim ose Static në procedurë në të cilën përdoret variabli
Vetëm në module	Duke përdorur shprehjen Dim ose Private para shprehjes së parë në Funcion ose Sub në modul
Të gjitha procedurat në module	Duke përdorur shprehjen Public para shprehjes së parë në Function ose Sub në modul

Ka tre mënyra për të shkruar një variabël:

- në mënyrë *Default*
- në mënyrë *Implicite*
- në mënyrë *explicite*

Implicite. - Të përcaktosh një variabël në mënyrë **implicite**, duhet përdorur njëri nga prapashtesat më lart për tipin përkatës të të dhënës. P.sh. për të deklaruar një tip variabli string, apo int, atëherë do kemi:

TextValue\$ = "This is a string"

Amount% = 300

Explicite. - Deklarimi i variablave në mënyrë **explicite** është paska më kompleks, pasi paraprakisht duhet përcaktuar shtrirja e variabiles, e pastaj duhet shkruar deklarimi i saj. Ekzistojnë 4 nivele:

- niveli **procedurë**

Brenda procedurës variablat deklarohen: *Dim*

MyInt as Integer

Dim MyDouble as Double

Dim MyString, YourString as String

Këto tipa variablash nuk e ruajnë me vlerën e tyre, pasi procedura ka përfunduar.

- niveli **procedurë, statike**

Nëse duam që variabla ta ruajë vlerën, edhe pasi ka përfunduar procedura atëherë duhet zëvendësuar fjala **Dim** me fjalën **Static** si më poshtë:

Static MyInt as Integer

Static MyDouble as Double

- nivel **forme dhe moduli**

Variablat e këtij tipi e ruajnë vlerën e tuaj për te gjitha procedurat që ndodhen brenda formës ose modulit. Variablat në nivel forme ose moduli deklarohen në pjesën e deklarimit të variablave në dritaren e kodit të formës. Për të deklaruar këto variabla përdoret fjala kyçe **Dim**.

Dim MyInt as Integer

Dim MyDate as Date

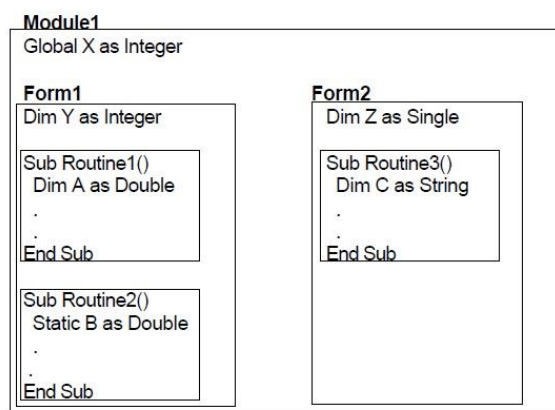
- nivel **gobal**

Në ndryshim nga sa u tha më sipër, variablat e nivelit global, kanë vlerë, domethënë e ruajnë vlerën e tyre për të gjitha procedurat dhe format në aplikacion. Edhe këto tipa variablash deklarohen në pjesën e deklarimit të variablave në dritaren e kodit e të formës ose modulit, por kanë fjalën kyçe Global para emrit të variablit si më poshtë:

Global MyInt as Integer

Global MyDate as Date

Për të përvetësuar më së miri karakteristikat e shtrirjes së variablave, shikojmë shembullin më poshtë:



Sikurse shikohet, procedura *Routine1* ka qasje në variablat X, Y dhe A, ku A e humbet vlerën e saj, pasi mbaron procedura *Routine1*, duke qenë se është variabël lokal për këtë procedurë.

Procedura *Routine2* ka qasje në variablat X, Y, dhe B, por në këtë rast B e mban vlerën e saj duke qenë se është deklaruar me fjalën kyçe Static.

Procedura *Routine3* ka qasje në variablat X, Y, dhe C, ku C si variabël lokal që është, e humbet vlerën e saj me përfundimin e procedurës.

Konstantet

Një variabël mund të mos e ndryshojë vlerën e saj gjatë gjithë procedurës, funksionit apo programit përkatës. Në këtë rast, konsiderohet si konstante. Deklarimi i një konstante realizohet si më poshtë:

```

Const NumQuarters As Integer = 4
Const Rate = .0725, Period = 12
Const ModName As String = "Budget Macros"
Public Const AppName As String = "Budget Application"
  
```

Vini re ndryshimin midis një variabël **statik** dhe një **konstante**. Konstantja përfaqëson një variabël vlera e të cilës është konstante, pra nuk ndryshon, ndërsa variabël statik nënkupton një variabël, vlera të cilit nuk ndryshon, pra mbetet statike brenda një sub procedure, ose funksioni. P.sh.:

```
Sub MySub()
    Static Counter As Integer
    Dim Msg As String
    Counter = Counter + 1
    Msg = "Number of executions: " & Counter
    MsgBox Msg
End Sub
```

Për të gjurmuar vlerat e variablave gjatë zbatimit të programit gjenden dy mundësi:

- Duke përdorur dritaren Watch Window klikoni mbi **View > Watch Window**
Ndriçoni çdo variabël nga dritarja Code Window dhe zhvendoseni në dritaren Watch Window
- Duke vendosur kursorin e mausit mbi emrin e variablit në dritaren **Code Window**, mund të shikoni vlerat e tij.

Përdorimi shprehjeve të vlerë dhënies

Vlerë dhënia në VB nënkupton dhënien e një vlere të caktuar një variabli ose një objekti.

Më poshtë janë paraqitur disa shembuj vlerë dhënie:

```
x = 1
x = x + 1
x = (y * 2) / (z * 2)
HouseCost = 375000
FileOpen = True
Range("TheYear").Value = 2012
```

Vlerë dhënia realizohet me anë të operatorit = , ndërsa në rastet kur duam të kontrollojmë nëse dy vlera janë të njëjta ose jo, atëherë përdoren dy shenja barazimi, d.m.th. = = .

Operatorët aritmetikë në VBA

Operatorët luajnë një rol kryesor në VB, pasi mbi bazën e tyre realizohen veprimet kryesore. Më poshtë është tabela përmbledhëse e tyre:

Funksioni	Veprimi perkates
Mbledhja	+
Zbritja	-
Shumëzimi	*
Pjesëtimi	/
Eksponcenciali	^
Bashkimi i stringjeve	&
Pjesëtimi i integer-ave	\
Moduli aritmetik (mbetja)	mod

Termi “concatenation” në programim, i referohet termit të bashkimit, që do të thotë nëse keni dy stringje që dëshironi t’i bëni “concat”, kjo do të thotë që dëshironi t’i bashkoni ato, duke i përfshirë brenda një stringu të vetëm.

Operatorët logjikë në VBA

Më poshtë paraqiten operatorët logjikë në VBA:

Operatori	Çfarë bën operatori
Not	Realizon një mohim logjik të shprehjes
And	Realizon një lidhje logjike midis dy shprehjeve
Or	Realizon një ndarje logjike midis dy shprehjeve
XoR	Realizon një përjashtim logjik midis dy shprehjeve
Eqv	Realizon një ekuivalencë logjike midis dy shprehjeve
Imp	Realizon një implikim logjik midis dy shprehjeve

Radha e kryerjes së veprimeve në VB është e njëjta si në Excel për veprimet aritmetike, sikurse dhe ato logjike. Kjo radhë ndryshon vetëm në rastet kur ka kllapa, që detyrojnë radhën e veprimeve të realizohet sipas tyre.

```
x = 3
y = 2
z = x + 5 * y
```

Ndërsa në rastin e kllapave do kishim:

```
z = x + (5 * y)
```

Komentet fillojnë ose me fjalën Rem ose një apostrof ('). P.sh.:

Rem ky është nje koment

*'ky është nje koment x = 2 * y 'nëse y=2 ath x=4*

Message Box

Message Boxes lejojnë të printohet diçka për përdoruesin në një kuti të vogël dialogu.

Nëpërmjet një Message Box mund të printoni: tekst, vlera variablash, të dyja njëkohësisht, duke përdorur simbolin & (concatenation sign). P.sh.:

MsgBox “Pershendetje”



MsgBox “Lartësia është" & "h" & vbCrLf & "dhe gjerësia është" & " w " & "."



Ju mund të përdorni funksionin MsgBox për të përcaktuar tipin e butonit dhe llojin e përgjigjes:

MsgBox (prompt, [buttons], [title], [helpfile, context])

Ku: **prompt** është tekst/variabel/ bashkim i të dyjave, i cili do të shfaqet në kutinë e dialogut.

Argumenti **title** lejon që të fusni një titull për kutinë e dialogut.

Argumentet **helpfile** dhe context ju lejojnë të jepni për përdoruesin opsionin e ndihmës.

Argumenti **buttons** merr një vlerë VB Constant, i cili përcakton numrin dhe stilin e butonave të vlefshëm për përdoruesin.

- vbOKOnly afishon/paraqet vetëm një buton OK
- vbOKCancel afishon/paraqet butonat OK dhe Cancel
- vbYesNoCancel afishon/paraqet butonat Yes, No dhe Cancel
- vbYesNo afishon/paraqet butonat Yes dhe No

Ju mund të kapni përgjigjen e përdoruesit tek Message Box me anën e një variabli të VBA të quajtur response.

- vbOKIf përgjigje OK
- vbCancelIf përgjigje Cancel
- vbAbortIf përgjigje Abort
- vbRetryIf përgjigje Retry
- vbIgnoreIf përgjigje Ignore
- vbYesIf përgjigje Yes
- vbNoIf përgjigje No



Input Box

Input Boxes krijojnë mundësinë që përdoruesi të fusë ndonjë vlerë në një kuti të vogël dialogu. Funkzioni InputBox gjithmonë i caktohet **një variabli**. Ky funksion ka sintaksën e përgjithshme si më poshtë:

InputBox (prompt, [title], [default], [xpos], [ypos], [helpfile, context])

Ku: argumentet prompt dhe title të funksionit InputBox janë të njëjtë si tek funksioni MsgBox.

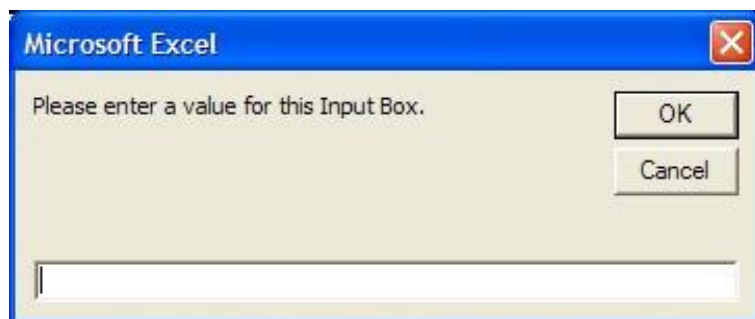
Argumenti **default** ju lejon të fusni një vlerë që shfaqet si vlerë default në Input Box.

Argumentet **xpos** dhe **ypos** ju lejojnë të poziciononi dritaren Input Box, në mënyrë relative në lidhje me qoshet majtas dhe lart të ekranit.

Argumentet **helpfile** dhe **context** janë njëllorj si tek funksioni MsgBox.

P.sh.: Ju i lejoni përdoruesit të fusë një numër, dhe pastaj, vlera e futur nga përdoruesi, ruhet në variablin me emër UserNumber.

UserNumber = InputBox (prompt, [title], [default], [xpos], [ypos], [helpfile, context])



Ne mundemi të përdorim funksione matematike të VBA pa pasur nevojë të krijojmë formula në spreadsheet. Në VBA gjenden:

1. Funksione baze matematike
2. Funksione trigonometrike

- **6 - funksione bazë matematike:**

Abs - Funksioni Abs llogarit vlerën absolute të variablilit. Ky funksion përdoret me tipa të dhënash të tipit integer dhe të tipit double (si edhe me variabla të tjera numerike). Vlera që kthehet nga ky funksion është e të njëjtit tip si vlera e përdorur në funksion. P.sh.: $\text{Abs}(-10) = 10$

Sqr - Funksioni Sqr llogarit rrënjën katrore të një numri. Ky funksion përdoret me çdo tip të dhëne numerike (duhet të jetë më e madhe se 0). Gjithmonë kthen një vlerë të tipit double. P.sh.: $\text{Sqr}(100) = 10$

Int - Funksioni Int heq pjesën dhjetore të një variabli të tipit double dhe kthen pjesën e plotë (integer part). Rezultati i kthyer do të jetë i tipit integer. Për numrat pozitive funksioni Int kthen numrin e rrumbullakosur nga poshtë. Për numrat negative, funksioni Int do të kthejë numrin e parë negativ më të vogël ose të barabartë me vlerën origjinale.

P.sh.: $\text{Int}(5.6) = 5$ ose $\text{Int}(-4.3) = -5$

Rnd - Funksioni Rnd gjeneron një numër rasti. Ky funksion do të kthejë një vlerë të tipit double midis 0 dhe 1. Për të krijuar numra rasti integer në një segment të caktuar, përdorni formulën:

$$\text{Int}((\text{upperbound} - \text{lowerbound} + 1) * \text{Rnd} + \text{lowerbound})$$

Exp - Vlera e kthyer nga funksioni Exp do të jetë një vlerë e tipit double. P.sh.: $\text{Exp}(2) = e^2 = 7.389056099$

Log - Funksioni Log llogarit logaritmin natyror. Rezultati që kthehet është një vlerë e tipit double. Ju mund të llogarisni logaritmin me bazë n të një numri të dhënë duke pjesëtuar logaritmin natyror të këtij numri me logaritmin natyror të numrit n. P.sh.: për të llogaritur log e numrit 15 me bazë 10 ($\text{Log}_{10}15$), ju duhet të shtypni: $\text{Log}(15) / \text{Log}(10) = 1.176091259$

- **Funksionet kryesore trigonometrike**

që do të përshkruajmë janë: Sin, Cos, Tan dhe Atn. Të gjithë këto funksione marrin këndin si parametër dhe madje vlera e këndit duhet futur në radian. Për të konvertuar nga gradë në radian, shumëzoni degrees me $\pi/180$. Për të përdorur variablin pi, në përcaktojmë një konstante si më poshtë: $\text{Const pi} = 3.14$

Funksionet Sin, Cos, dhe Tan. Vlera e kthyer është e tipit double. Rezultati i Sin dhe Cos do të jetë midis -1 dhe 1. Psh:

$$\text{Sin}(\pi/4) = 0.7071067812$$

$$\text{Cos}(\pi/3) = 0.5$$

$$\text{Tan}(\pi/6) = 0.5773502692$$

Funksioni Atn llogarit arctangentin. Rezultati është një kënd në radian, i cili do të jetë i tipit double. Rezultati i Atn do të jetë midis $-\pi/2$ dhe $\pi/2$ radians. Për të konvertuar nga radian në gradë, shumëzoni radians me $180/\pi$. P.sh.: $\text{Atn}(0.5773502692) = \pi/6$