Learning VB.Net

Tutorial 06 – variables again, group box, list box

Hello everyone... welcome to vb.net tutorials. These are going to be very basic tutorials about using the language to create simple applications, hope you enjoy it. If you have any notes about it, please send them to <u>notes@mka-soft.com</u> I will be happy to receive them. Finally if you find these tutorials are useful, it would be nice from you to send a small donation via PayPal to <u>donation@mka-soft.com</u>.

tutorial posted on 2009-June-22.

Hello everyone. Today's tutorial is about using Group boxes, and little about variables and list box. We will create a small application to find the area of squares, rectangles, and triangles. So let us proceed by creating a new windows application project, name it "find area" and save it.

Next set the form title to "Calculate Area", I assume now you already know how to do that. Next place a two labels, two text boxes, and a button on the form to look something like this:



Now change the first label to be "length:", second label to be "area:", and the button text to be "compute". (to do that just click the control and change the text property".

You should get something like this:

🔡 Calcu	late Area		
length:		compute	
area:			

The user enters the length of the square in the first text box, and presses the compute button, and he/she should get the area in the second text box. The second text box should not be modified by the user. To do so, change the ReadOnly property of the second text box to be "true".

🖶 Calculate Area		
length:	compute	
area:		

To write the code that gets the length, calculate the area, and display it, first check the name of the text boxes because you will need these. Next double click the compute button to write its event.

Dim L As Double Dim A As Double	;	define the length variable define the area variable
L = TextBox1.Text A = 4 * L	;	get the length from the window. calculate the area
TextBox2.Text = A	1	display the area in the second text box

Keep in mind that TextBox1, and TextBox2 are the names of the controls used to get the data and display the result. And also remember you can change them if you want to. Now run the application, enter a value of 10 and press compute.



Chang the length and press compute again and see the new results. Next we will do something similar for the rectangle. Place three labels, three text boxes, and a command button to be something like this:



Note that control names here are just the way I got them, you might get the order differently, or you might want to rename the controls. Next double click the compute button for the rectangle and write the following code.

Dim W As Double Dim h As Double Dim A As Double	 define the width variable define the height variable define the area variable
W = TextBox4.Text h = TextBox5.Text A = W * h	get the width get the height compute the area
TextBox3.Text = A	' dsplay the area

Now run the application, and test it.

🔡 Calcu	ilate Area	
length:		compute
area:		
width:	10	compute
height:	20	
area:	200	

The last part is the triangle. Repeat what you did with the rectangle and you should get something similar to this:



And next set the code to compute the area to be:

Dim base As Double Dim Height As Double Dim Area As Double		define the base define the height define the area
base = TextBox8.Text Height = TextBox6.Text	;	get the base get the height
Area = 0.5 * base * Height	1	compute the area $(1/2 \times base \times height)$
TextBox7.Text = Area	•	display the result.

and check the code. It works. Now look at our window.

🔜 Calculate Area		
length:	compute	
area:		
width:	compute	
height:		
area:		
base:	compute	
height:		
area:		

It does not look good actually and a little bit confusing because there are three compute buttons, and the fields are mixed. The solution to this is to use GroupBox control. Now search with the controls and add a GroupBox control on the form.

🔡 Calcu	ilate Area
length:	compute
area:	
width:	compute
height:	
area:	
base:	compute
height:	
area:	
GroupE	dox1

Next select the labels, textboxes and the button that is used for the square, and drag them so that they fit inside the group box. Click then on the group box and set its text property to be square. You should get something similar to this:

💀 Calculate Area	X
Square length: compute	
area:	
width: compute	
height:	
area:	
base: compute	
height:	
area:	
	_

Notice that you might need to move the controls a little bit to make them fit. The group box can be moved using the arrows symbol that appears when you click inside it.

	······	
length:		compute
area:		Ī
Ö		Ó

Repeat the same thing for the rectangle and triangle to get something like this:

🛃 Calculate Area	
Square	
length:	compute
area:	
Rectangle	
width:	compute
height:	
area:	
Triangle	
base:	compute
height:	
area:	

As you can see group box allows easy movement of controls on the form at design time and also gives a better view of your application. Now we start working with list box. It is used to view a list (which is no surprise). What we want to do here is displaying each area being computed on the right side in such a way the previous results and computations are displayed as well. To do so, add a group box to the right of your window and put inside it a listbox control. You should have something like this:

👼 FindArea - Microsoft Visual Basic 20	2008 Express Edition	
File Edit View Project Build Debug	j Data Format Tools Window Help	
👔 🗃 • 🖬 🥔 👗 🖻 🐘	글 일 🤊 • 연 • 🚚 • 🖳 🕨 💷 🖼 💭 😁 📸 :	چ 🗖 💰 🛠
Toolbox - 4 ×	Start Page Form1.vb* Form1.vb [Design]*	
📑 ComboBox		
ContextMenuStrip	🖷 Calculate Area	
🔁 DataGridView		
P DataSet	Square	- Bo
T DateTimePicker	length: Compute ListBox1	
🛐 DirectoryEntry	area:	
🔍 DirectorySearcher	Pertande	
DomainUpDown	width: compute	
ErrorProvider	bainbh	
EventLog		
🚑 FileSystemWatcher	area:	
FlowLayoutPanel	Triangle	
FolderBrowserDialog	base: compute	
FontDialog	beight:	
GroupBox		
[F1] HelpProvider	area:	
AN HScrollBar		
imageList		
A Label		
A LinkLabel		
ISCHOX		

The ListBox1 that appears here does not appear while the application is running. What you see now is the name of the listbox control. Now we modify the code of the square so that we add the area computed into the list box. Double click on the compute button of the square, and modify its code to be like this:

Dim L As Double Dim A As Double	;	define the length variable define the area variable
L = TextBox1.Text A = 4 * L	;	get the length from the window. calculate the area
TextBox2.Text = A	1	display the area in the second text box
ListBox1.Items.Add('tł	ne area of the square is:" & A) ' add the item

ListBox1 is used to communicate with the list, **.Items** is used to access the list of items that it is displaying (which is at the beginning of execution is empty), **.Add** is used to add the information or message into the Items of the list box. "the area of the square is:" & A is evaluated first, and then the result is added into the list box items.

For the rectangle, modify its compute button to be something like this:

```
Dim W As Double ' define the width variable
Dim h As Double ' define the height variable
Dim A As Double ' define the area variable
W = TextBox4.Text ' get the width
h = TextBox5.Text ' get the height
A = W * h ' compute the area
TextBox3.Text = A ' dsplay the area
Dim S1 As String
Dim S2 As String
S1 = "The area of rectangle is:" ' the message
S2 = S1 & A ' write the message then the area, and
' store the new message in S2
ListBox1.Items.Add(S2) ' add the message to the list box
```

The same method to add item to the list box is called, but the way we generate the message is a little bit different. Finally the rectangle code should be:

Dim base As Double Dim Height As Double Dim Area As Double	-	define the base define the height define the area
base = TextBox8.Text Height = TextBox6.Text	;	get the base get the height
Area = 0.5 * base * Height	1	compute the area $(1/2 \text{ x base x height})$
TextBox7.Text = Area	1	display the result.
<pre>Dim Str As String Str = "The area of triangle is: ListBox1.Items.Add(Str)</pre>		& Area ' generate the whole message directly ' add the message to the list

Now you see also that the message is generated in a different way again. So you can choose the way that best suits you. Now run the application and calculate few areas.

🔡 Calculate Area			
Square length: 3 area: 12 Rectangle width: 20 height: 30 area: 600	compute	History the area of the square is:40 The area of rectangle is:600 The area of triangle is: 1000 the area of the square is:40 the area of the square is:12	
Triangle base: 40 height: 50 area: 1000	compute		

You should get something similar to this at run time. You can compute more than once and see that the list box keeps adding new results. Next we see how to make a clear button to clear the

content on the form (for the listbox and the textboxes). Simply add a button on the form, you should get something similar to this:

📕 Calculate Area	
Square	History
area:	ListBox1
Rectangle	ă l
width: compute	
height:	
area:	
Triangle	
base: compute	
height:	
area:	clear

And add the following code in the clear button's event:

```
TextBox1.Text = "" ' clear all the text boxes
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
```

As you can see text boxes are cleared by setting their text property to empty string "", while listboxes you just give a Clear command for them. Run the application, compute a number of areas, and finally hit clear and see how the listbox and all textboxes are cleared from the text.

So basically we worked just a little bit with variables, saw how to work with groupbox, and saw the basic operation of listboxes. More details will be given into listboxes, but for now we need to focus more on the programming aspect of the language.

If you need the code you can download it from the web site, and also you can check the videos. If you have notes about the tutorial I will be happy to hear from you, just email me at notes@mka-soft.com.