

Forum:[Hack Forums](#)

Thread: [Adding Custom objects into a Controls/Listboxes \[OOP\]](#)

+Haze - December 30, 2014, 3:05 am

My second tutorial here. This tutorial is more about the object orientation methods that can be used to tackle more complex programming solutions.

So in this tutorial I will explain and show you how you can add custom objects into controls. In general you can use this example with any windows control which you can add objects to. Like the ComboBox. TreeNode. This is just no exclusive to VB, It works on C# too, and I'm sure it would work on any object orientated language.

So Let's begin by designing our form first.

Now that's done, we can now create our class Persion() where we will add our Properties and Methods.

After we done that, we now subscribe to the Clicked Event for our add button. Inside that sub routine we instantiate our persons class. When we Instantiate we use the New Keyword.

In this context, what we're doing is creating a New copy of the Person class. So in simpler terms. Each time we Instanciate. We create a new stickman.

And we define this new stickman, by giving him a Firstname, Lastname, Age and his own HF UID.

So each time we click the add button, a new stick man will be created and his properties(Firstname) will be the value of the txtFirstname textbox and so forth. And then that person is now added to then using the listbox using the ListBox1.Items.Add() method

Finally, this is the most important bit of our tutorial. This is how we get back these objects that we have added into our listbox.

The direct cast allows us to convert the SelectedItem object to the Person object/class we have inserted. And this way we can get back the object we had previously added and view all its unique properties.

Here is a short clip of the end result of this tut.

[Video: https://www.youtube.com/watch?v=sngiayUgJ_4&feature=youtu.be]

so you can see the uses of this type of programming technique. Allowing us to use objects to accomplish complex tasks. And thus making programming easier when you're faced with scenarios where this method could be useful.

Here are the vb.net code on pastebin with syntax highlighting.

<http://pastebin.com/wvPU8BbS>

and here's the .Zip of the source code.

<https://www.mediafire.com/?do58dtcuyu46pd3>

enjoy.

Some feedback would be nice too

RandomUser123 - December 30, 2014, 3:19 am

Great! That can be useful in a lot of cases!

Thanks for this clear tutorial!

\$lick - December 30, 2014, 7:05 am

Why are you using string.format on something with one variable? And that variable being on the end of the string.

+Haze - December 30, 2014, 7:56 am

Fair enough good point, but do remember I constructed this thread whilst I had no sleep. So yaa..

Dangerous Beef - December 30, 2014, 12:32 pm

I thought you used arrays :(Oh well thanks the possible the core of an future inventory system I'm making.

+Haze - December 30, 2014, 12:39 pm

Nope, using arrays would be really messy when you have to store more than 1 property. And you'd have to constantly keep resizing your array, unless you use a List. But this method is the best to my knowledge on doing something like this. I just wish more would see this.

Mad Hatter - January 3, 2015, 5:49 pm

Thanks for the help Haze, I appreciate it