

Re: return object of an unknown type

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-01/msg03796.html>

- *From:* "DeveloperX" <nntpDev@xxxxxxxxxxxxxx>
 - *Date:* 25 Jan 2007 06:23:51 -0800
-

I understand now, no :)

You can overload methods but they have to have different signatures and return type is not part of the signature.

So the following is legal, but a version where getdata takes no parameters in both is not.

I think Allan's ideas are best.

```
public class MyMainClass
{
  /// <summary>
  /// test
  /// </summary>
  public MyMainClass(){ }
  /// <summary>
  /// test2
  /// </summary>
  /// <returns></returns>
  public virtual Control GetData()
  {
    object o = new Form();
    return (Form)o;
  }
  public virtual Form GetData(Form pForm)
  {
    object o = new Form();
    return (Form)o;
  }
}
public class MyClass2 : MyMainClass
{
  public MyClass2(){ }
  public override Control GetData()
  {
    MyMainClass a = new MyMainClass();
    a.GetData();
    object o = base.GetData ();
  }
}
```

Re: return object of an unknown type

```
//will break as base returns a form but its for example purposes
return (TextBox)o;
}

public override Form GetData(Form pForm)
{
return base.GetData (pForm);
}
}
```

On 25 Jan, 13:27, "Alexander Widera"

<a...@xx> wrote:

ok ... i try it:

```
public class MyMainClass
{
protected virtual object o = new TextBox();

public MyMainClass(){ }

public virtual TextBox GetData()
{
// do something..
// only placeholer
// .....

// and return
return (TextBox)this.o;
}

}public class MyClass2 : MyMainClass
{

// something like this
protected override object o = new Form();

public MyClass2(){ }

// and something like this...
public override Form GetData()
{
// do something..
// only placeholer
// .....

// and return
return (Form)this.o;
}
}
```

Re: return object of an unknown type

}the both classes are nearly the same... only the difference, that the Method GetData() should return different types. is there a way to manage this? (and not to write the method completely again?)

alex

"DeveloperX" <nntp...@xxxxxxxxxxxxxxxx> schrieb im
Newsbeitrag news:1169730546.999831.171290@xx

Ok, so you have something like this? I'm using Form and TextBox just as examples.

```
public class MyMainClass
{
public MyMainClass(){ }
public virtual Control GetData()
{
object o = new Form();
return (Form)o;
}
}
public class MyClass2 : MyMainClass
{
public MyClass2(){ }
public override Control GetData()
{
object o = base.GetData ();
//will break as base returns a form but its for example purposes
return (TextBox)o;
}
}
```

If you can do a little example like that which compiles and a comment for the bit you don't understand it should be a little clearer.

On 25 Jan, 12:07, "Alexander Widera"
<a...@xx> wrote:

thanks for your answer, but i think this issn't the result of my
problem
i don't want to make a usual cast ...

Re: return object of an unknown type

in a variable should be the type of an object and a method should return an other object as this type.

perhaps i have to specify the situation:

i have 2 classes ... MyMainClass and MyClass2 ... MyClass2 is derived of MyMainClass

in MyMainClass there is an Method like this

```
public virtual OtherObject GetData()
{
// .. a lot work to do...
return (OtherType)privateObject);
```

}privateObject is a variable of the type "object" and the method returns it casted to "OtherType".

in MyClass2 this Method should not return the privateObject casted to "OtherType" ... it should return it casted to "AnotherType"... because there is much to do in this method (not shown above), i don't want to rewrite the method completely for every child-class of MyMainClass...

has somebody an idea?

Alex

Re: return object of an unknown type

dynamic.

here something in
c#-pseudocode:

```
object myobject = ...  
Type typeofobject = ...
```

```
public typeofobject  
GetTheObject()  
{  
    return (typeofobject)  
    myobject;  
}
```

}Could you help me please?

Thanks in advance,
Alex- Hide quoted text --
Show quoted text -- Hide
quoted text -- Show quoted
text -