

Forum:[Hack Forums](#)

Thread: [\[C#\] Inheritance and virtual methods](#)

killerdoob420 - August 13, 2010, 11:40 am

This tutorial will explain inheritance and virtual methods and how to use them in C#(im assuming you have a basic knowledge of oop concepts). Inheritance is when a class uses another class has a base and acquires all its variables and methods, take this class for example:

Code class Derp

```
{
    public int var;
    public void Write(string text)
    {
        Console.Write(text);
    }
```

}this next class uses that one as a base(it inherits from it)

Code class Herp : Derp

```
{
    public void WriteLine(string text)
    {
        Write(text + "\n");
    }
```

}

As you can see the class Herp contains the method and variable from its base class as well as the other method we added to it.

Virtual methods are a way of changing the behavior of inherited methods:

Code class Derp

```
{
    public virtual void DoWork()
    {
        Console.WriteLine("Doing base class work");
    }
}
```

Code class Herp : Derp

```
{
    public override void DoWork()
    {
        Console.WriteLine("Doing inherited class work");
        base.DoWork();
    }
```

}As you can see the inherited class is overriding the virtual method DoWork, the line 'base.DoWork();' calls the original DoWork method from the base class so the output of DoWork from the class Herp would be:

Doing inherited class work
Doing base class work

please leave a comment if you liked this tutorial or if you have any problems or questions about it

laptopguy - August 13, 2010, 11:41 am

Very nice tutorial. Not the biggest but it's straight to the point and informative. Nice one, and rated 5 stars ?

killerdoob420 - August 13, 2010, 11:53 am

(Aug 13, 2010 11:41 AM)kRaeTwin Wrote: [\[CITE\]](#) Very nice tutorial. Not the biggest but it's straight to the point and informative. Nice one, and rated 5 stars ?
thanks, i tried to be as clear as possible

enacje - August 13, 2010, 12:01 pm

the problem with this kind of subjects is that it is kinda hard to explain when to use it. I'm working on a polymorphism tutorial but I can't think of some good examples when to use it on a way people understand it :P

Same for you you didn't mention when and why you should use it

killerdoob420 - August 13, 2010, 12:09 pm

(Aug 13, 2010 12:01 PM)enacje Wrote: [\[CITE\]](#) the problem with this kind of subjects is that it is kinda hard to explain when to use it. I'm working on a polymorphism tutorial but I can't think of some good examples when to use it on a way people understand it :P

Same for you you didn't mention when and why you should use it I use this alot for games, but polymorphism is a tricky one ?

laptopguy - August 13, 2010, 12:10 pm

(Aug 13, 2010 12:09 PM)killerdoob420 Wrote: [\[CITE\]](#) I use this alot for games, but polymorphism is a tricky one ?

Could you give an example in how to use this in a standard Windows Application ? Like what is it's values ?

killerdoob420 - August 13, 2010, 12:29 pm

(Aug 13, 2010 12:10 PM)kRaeTwin Wrote: [\[CITE\]](#) Could you give an example in how to use this in a standard Windows Application ? Like what is it's values ? well i guess you could use it for user profiles or something, i can code and actual example if you like

laptopguy - August 13, 2010, 12:32 pm

(Aug 13, 2010 12:29 PM)killerdoob420 Wrote: [\[CITE\]](#) well i guess you could use it for user profiles or something, i can code and actual example if you like

Sure, that would be good mate.

b0tn3rd - August 13, 2010, 1:06 pm

Well the typical example of the usecase is in multiple implementations of things that act alike:

e.g. 5 different types of buttons...

e.g. 5 different commands called from a single parser.

Make a base class `IrcCommand`.

Make a derived class `IrcCommandScan`

Make a derived class `IrcCommandDownload`

Make a string map to `IrcCommand` reference and register your implementations there. Makes it easy to have your commands extendable without having a huge if then else clause...

killerdoob420 - *September 7, 2010, 5:31 am*
bringing some of my old tuts up

killerdoob420 - *October 22, 2010, 5:04 pm*

eliteh4xer - *July 4, 2011, 10:43 pm*
Constructor Can't Be Inherited Why