

## Re: Is this proper in OO theory?

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-07/1614.html>

---

**From:** Marina (*someone\_at\_nospam.com*)

**Date:** 07/07/04

Date: Wed, 7 Jul 2004 11:25:40 -0400

I don't feel like I'm getting a clear example of the issue from this particular code sample.

In general, there is nothing wrong with putting anything in the base class that will be useful to all its children, even if it requires some parameters.

"Daniel Billingsley" <dbillingsley@NO\_durcon\_SPAAMM.com> wrote in message news:uKwNqTDZEHA.2972@TK2MSFTNGP12.phx.gbl...

> *The base class obviously can't know anything about a class which inherits from it, but is it acceptable to have a base class method which will be given information from an inherited class?*

>

> *For (a stupid) example:*

>

> -----

>

> *public abstract class BaseClass*

> {

> *private int \_someInteger=0;*

>

> *protected int AddNumbers(int anotherNumber)*

> {

> *return anotherNumber + \_someInteger;*

> }

>

> }

>

> *public class Class2 : BaseClass*

> {

> *private int \_anotherInteger;*

> *public int GetANumber()*

> {

> *return base.AddNumber(\_anotherInteger);*

> }

> }

>

microsoft.public.dotnet.languages.csharp: Re: Is this proper in OO theory?

> -----

>

> *The underlying issue is that the generic code in the AddNumbers method*  
from

> *example is known to be needed in all the classes inheriting from*  
BaseClass,

> *but needs some additional information based on the specifics of the*  
> *inherited class (it's a data access method in reality).*

>

>