What do you think of using properties as object initializers in C#;

Asked 16 years, 3 months ago Modified 12 years, 4 months ago Viewed 494 times

3

I was wondering what people thought of using properties as object initializers in C#. For some reason it seems to break the fundamentals of what constructors are used for.

An example...

```
public class Person
{
    string firstName;
    string lastName;

    public string FirstName
    {
      get { return firstName; }
      set { firstName = value; }
    }

    public string LastName
    {
      get { return lastName; }
      set { lastName= value; }
    }
}
```

Then doing object intialization with.....

```
Person p = new Person{ FirstName = "Joe", LastName = "Smith" };
Person p = new Person{ FirstName = "Joe" };
```

c#

Share

Improve this question

Follow

edited Sep 18, 2008 at 0:11

JamesSugrue

15k • 10 • 62 • 93

asked Sep 17, 2008 at 23:57

James

65 • 5

12 Answers

Sorted by: Highest score (default)

10

What you see here is some syntatic sugar provided by the compiler. Under the hood what it really does is something like:

Person p = new Person(FirstName = "Joe", LastName = "Smith");

```
Person _p$1 = new Person();
_p$1.FirstName = "Joe";
_p$1.LastName = "Smith";
Person p = _p$1;
```

So IMHO you are not really breaking any constructor fundamentals but using a nice language artifact in order to ease readability and maintainability.

Share
Improve this answer
Follow

edited Sep 18, 2008 at 0:08

answered Sep 18, 2008 at 0:01

Jorge Ferreira
97.8k ● 25 ● 126 ● 134

6

Object initializers does in no way replace constructors. The constructor defines the contract that you have to adhere to in order to create a instance of a class.

The main motivation for object initializers in the C# language is to support Anonymous Types.

```
var v = new { Foo = 1, Bar = "Hi" };
Console.WriteLine(v.Bar);
```

Share Improve this answer Follow

answered Sep 18, 2008 at 0:11

Hallgrim
15.5k ● 11 ● 48 ● 54

4

IMHO its sweet. Most objects are newed up with the default constructor, and must have some properties set before they are ready to run; so the object initializers make it easier to code against most objects out there.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:00

user1228

Constructors should only really have arguments that are *required* to construct the object. Object initialisers are just a convenient way to assign values to properties. I

2

use object initialisers whenever I can as I think it's a tidier syntax.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:15

Ty.

2,220 ● 1 ● 15 ● 14

2

Since you're already using the new C# syntax, might as well use automatic properties as well, just to sweeten up your code a drop more:

instead of this:

```
string firstName;

public string FirstName
{
  get { return firstName; }
  set { firstName = value; }
}
```

use this:

```
public string FirstName { get; set; }
```

Share

Improve this answer

Follow

edited Jul 30, 2012 at 14:31

phoog

43k ● 6 ● 83 ● 124

answered Sep 18, 2008 at 0:15

BFree

104k ● 21 ● 162 ● 204

1

I think overall it is useful, especially when used with automatic properties. It can be confusing when properties are doing more than get/set. Hopefully this will lead to more methods, and reduce the abuse of properties.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:04

Nescio

28.4k ● 10 ● 55 ● 76

1

Not your original question, but still...

Your class declaration can be written as:

```
public class Person
{
    public string FirstName { get; set; }
    public string LastName {get; set; }
}
```

and if it were my code, I'd probably have an object for Name with fields First and Last.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:40

Jay Bazuzi

46.4k ● 16 ● 115 ● 170

0

It's also quite necessary for projected classes returned from a language integrated query (linq)

```
var qry = from something in listofsomething
          select new {
                        Firstname = something.FirstName,
                        Lastname = something.Surname
                      }
```

Share Improve this answer Follow

answered Sep 18, 2008 at 0:05

Tim Jarvis

18.8k ● 10 ● 59 ● 95

1 You probably mean for lists of anonymous objects (in lieu of "projected classes"). :) – Jon Limjap Sep 18, 2008 at 0:12

0

Adding to Nescio's thoughts - I'd suggest in code reviews actively hunting down expensive transparent operations in property accessors e.g. DB round tripping.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:14

stephbu

5,082 ● 28 ● 42

0

Object Initializers help to reduce coding complexity in that you don't need to create a half dozen different constructors in order to provide initial values for properties. Anything that reduces redundant code is a positive, in my book.

I believe the primary reason the feature was added to the language is to support anonymous types for LINQ.

Share

Improve this answer

Follow

edited Sep 18, 2008 at 0:17

answered Sep 18, 2008 at 0:12

Jeromy Irvine

11.8k ● 3 ● 41 ● 53

0

If you want to enforce the use of a constructor, you could set your object's default parameterless constructor to private, and leave public only some enforced constructors:

```
public class SomeObject
{
    private SomeObject()
    {}

    public SomeObject(string someString) //enforced constructor
    {}

    public string MyProperty { get; set; }
 }
```

Using the above definition, this throws an error:

```
var myObject = new SomeObject { MyProperty = "foo" } //no method accepts zero arguments for constructor
```

Of course this can't be done for all cases. Serialization, for example, requires that you have a non-private default constructor.

Share Improve this answer Follow

answered Sep 18, 2008 at 0:27

Jon Limjap

95.3k ● 15 ● 103 ● 153

0

I for one am not happy with them. I don't think they have a place in the constructor, or MS should got back and refactor them to allow you to use them in a private fasion. If I construct an object I want to pass in some PRIVATE data. I want it set from the outside world once and that's it. With Object Initializers you allow the values passed into the constructor to be modifiable.

Maybe in the future they will change this.

Share Improve this answer Follow

answered Sep 19, 2008 at 20:52