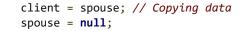
Why isnt this object null after a shallow copy?

Ask Question



Below is my copy of spouse to client (both are same object type). Spouse is then set to null.

2





I then pause (using a breakpoint on a different line) and check the memory of client and spouse. spouse is null, however client isn't.

Shouldn't client be null because its memory is a result of a *shallow copy*?

Cheers

c# shallow-copy

edited Jan 14 '16 at 4:18

Rob ◆
23.8k 12 57 77

asked Jan 14 '16 at 4:15



Robert 48 6

1 It is not doing a shallow copy. If client and spouse are reference types (i.e., if they're an instance of a class), you are changing *pointers*, not copying data. And even if it *did* do a shallow copy, one would expect client not to be null, as it is

```
'Normal C#' would be for ref parameters. – Rob ♦ Jan 14 '16 at 4:17
```

Correct, they are reference types. Thanks Rob. I would mark this as the answer but its a comment. — Robert Jan 14 '16 at 4:19

3 Answers



This doesn't happen because you are changing the value of the *pointers*, not the object itself.

3

Let's imagine this scenario:



```
var spouse = new Person(); //Let's say the memory address
```



Now, we have this:

answered Jan 14 '16 at 4:25



Rob ◆
23.8k 12 57 77



Your spouse lives at 123 Sesame Street.



Now you write down on another piece of paper: CLIENT:. Then you copy whatever it says after SPOUSE on the first piece of paper.

Now you have two pieces of paper. One says "SPOUSE: 123 Sesame Street". The other says "CLIENT: 123 Sesame Street".

Now you erase the address on the page that says SPOUSE.

What does the page that says CLIENT now say?

Your confusion is apparent in your choice of jargon.

Do not say "makes a shallow copy". Say "copies a reference", because that's what you're doing. "Shallow" is relative without saying relative to what. Say what is really happening: the value is being copied and *the value is a reference*.

Do not say "this object is null". That's like saying "the car in my driveway that is not there"; it's nonsensical. A *variable* can *contain* a null reference. A *reference* can be a *null reference*; it is the reference that refers to no object. But it is not an object; it is the absence of an object.

When you make your language precise then these sorts of confusions start to drop away rapidly.

edited Jan 14 '16 at 4:31

answered Jan 14 '16 at 4:24



Eric Lippert 552k 149 1075

Home

PUBLIC



Tags

Users

Jobs

Teams Q&A for work



The way you are using to Shallow copy is not the proper way in C#. C# provide us a function we can use to perform a shallow copy. the function is MemberwiseClone().



Try this

Client = Spouse.MemberwiseClone();

Please tell me if it doesn't work for you. cheers

answered Jan 14 '16 at 4:41



Ali Mohsin

69

you can also refer <u>msdn.microsoft.com/en-us/library/...</u> – Ali Mohsin Jan 14 '16 at 4:42

MemberwiseClone is a protected method, this will only work if written inside the class which Spouse is an instance of. – Rob ♦ Jan 14 '16 at 4:48