



对象复制

北京理工大学计算机学院
金旭亮

概述

对象的“复制”，指的是把一个现有的对象“克隆（clone）”一份，得到一个新对象，这个新对象与老对象“一模一样”，但.....

新、老对象是完全独立的！

对象的“浅复制”

```
class MyClass {  
    public int MyClassValue = 100;  
}
```



```
public static MyClass CloneObject( MyClass obj )  
{  
    MyClass newObj = new MyClass();  
    newObj.MyClassValue = obj.MyClassValue;  
    return newObj;  
}
```



使用简单的字段值复制方式复制一个对象，称为对象的“**浅复制 (shallow copy)**”方式。

示例：ObjectCloneDemo

考虑复合对象的情况

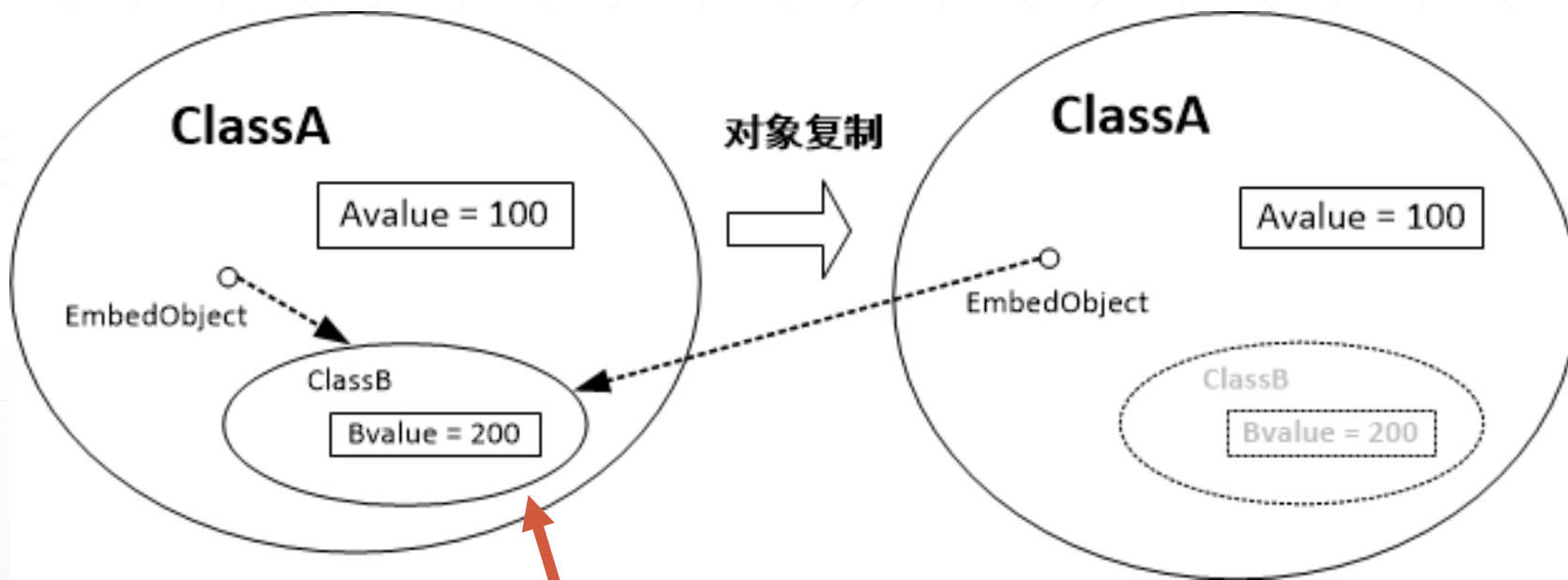


ClassA 包容一个
ClassB 对象

```
1 reference
public static ClassA CloneObject(ClassA obj)
{
    ClassA newObj = new ClassA();
    newObj.AValue = obj.AValue;           //字段复制
    newObj.EmbedObject = obj.EmbedObject; //引用复制
    return newObj;
}
```

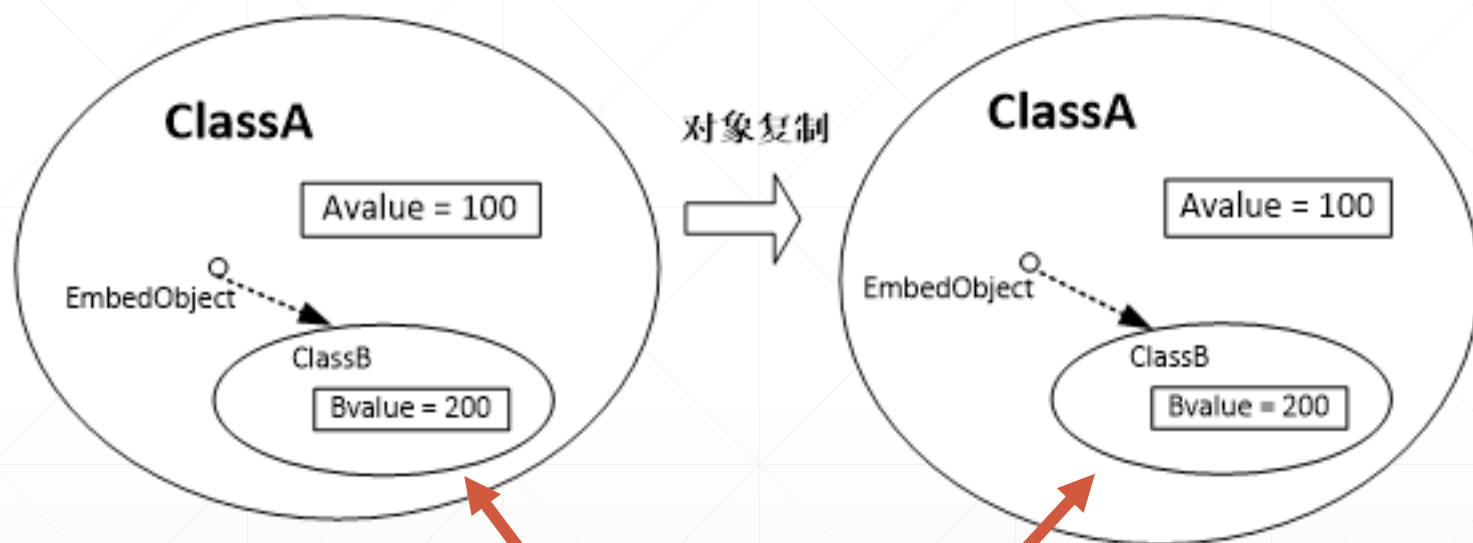
基于字段值复制的
方式克隆组合对象

浅复制的问题



浅复制方式，无法复制出真正独立的新对象。
新老对象共享“一个”内部对象

对象的“深复制”方式



“**深复制 (Deep Copy)**”在复制对象时，会把其所包容的内部对象也复制一份，并且这个过程是“递归”进行的。

支持“深复制”方式的对象

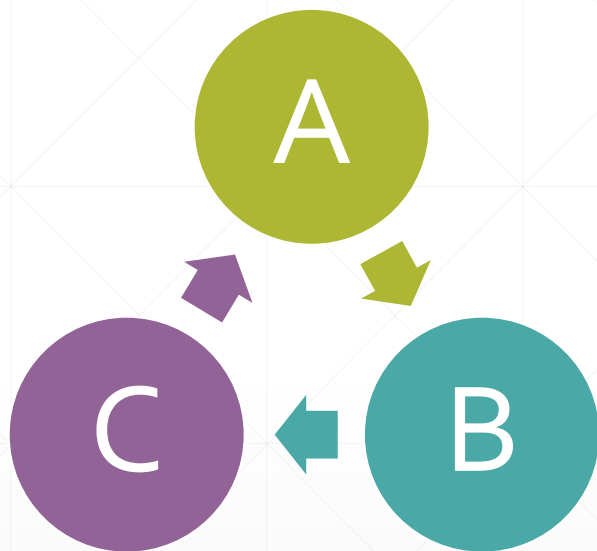
.NET基类库中定义了一个**ICloneable**接口，所有需要实现深复制的对象，应该实现这个接口



示例：DeepCopy

```
6 references
class ClassA : ICloneable
{
    public int AValue = 100;
    public ClassB EmbedObject;    //ClassA包容一个ClassB的对象
    3 references
    public Object Clone()
    {
        ClassA ObjA = new ClassA();
        ObjA.AValue = this.AValue;
        ObjA.EmbedObject =
            (this.EmbedObject as ICloneable).Clone() as ClassB;
        return ObjA;
    }
}
```

如果循环包容.....



请编写代码测试一下复制上述循环包容的对象时，会引发什么样的结果？