

Type of base pointer

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-10/msg00754.html>

- *From:* "dotNiemand" <dotNiemand@xxxxxxxxxx>
 - *Date:* 23 Oct 2006 07:40:15 -0700
-

Take a look on this code:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace BasePtr
{
class Program
{
class A
{
public virtual void Test()
{
Console.WriteLine("Method of base class");
}
}

class B:A
{
public override void Test()
{
Console.WriteLine("This is a real virtual method!");
}

public void OldTest()
{
base.Test();
}

public void ICanntUnderstand()
{
Console.WriteLine(base.GetType().ToString()); // WHY
IS THE OUTPUT "B?! base class for B is A not A"
}
}

static void Main(string[] args)
```

Type of base pointer

```
{
A a = new A(); // new A instance
a.Test(); // "Method of base class"
B b = new B(); // new B instance
b.Test(); // "This is a real virtual method!"
a = b;
a.Test(); // "This is a real virtual method!"
b.OldTest(); // "Method of base class"

b.ICanntUnderstand(); // WHY IS THE OUTPUT "B?! base class
for B is A not A"

Console.Read();
}
}
}
```

This code returns:

```
Method of base class
This is a real virtual method!
This is a real virtual method!
Method of base class
BasePtr.Program+B
```

I have a virtual function called Test() in class A. Also i have class B – it's derived from A and has overridden method Test() as well. I can easily access method A.Test() from class B – as written in function B.OldTest(). As well base object doesn't have method OldTest(). You can see it in this error description:

```
Error 1 'BasePtr.Program.A' does not contain a definition for
'OldTest' C:\Documents and Settings\Vitaliy\My Documents\Visual Studio
2005\Projects\BasePtr\BasePtr\Program.cs 27 22 BasePtr
```

base class for B is A and error above shows that base is BasePtr.Program.A. But code "base.GetType()" in any method of class B returns value BasePtr.Program.B.

WHY?!

.