

第42章 仮想メソッド



派生クラスで、書き換えられることを前提にしたメソッドを作ることができます。

派生クラスでは、継承されたメソッドと同じシグニチャ(名前とパラメータリスト)と 戻り値の型を持ったメソッドを定義することができます。これをメソッドのオーバーライドといいます。基本クラスでは、オーバーライドされるメソッドにvirtual修飾子をつけます。派生クラスでは、オーバーライドしたメソッドにoverride修飾子をつけます。オーバーライドされる基本クラスのメソッドを仮想メソッドといいます。

派生クラスでは、仮想メソッドをオーバーライドしなくてもかまいません。この場合 基本クラスのメソッドがそのまま使われることになります。

また、staticなメソッドは仮想メソッドにすることはできません。

さて、派生クラスの参照を、基本クラスの 参照変数に代入すると、基本クラスの隠蔽されたメンバを呼び出すことができました。

これは、オーバーライドされたメソッドについても当てはまるのでしょうか。

実験してみましょう。

```
// virtualmethod01.cs

using System;

class MyClass1
{
    public virtual void show()
    {
        Console.WriteLine("Class1のshow()");
    }
    public void xshow()
    {
        Console.WriteLine("MyClass1のxshow()");
    }
}

class MyClass2 : MyClass1
{
    public override void show()
    {
        Console.WriteLine("MyClass2のshow()");
    }

    new public void xshow()
    {
        Console.WriteLine("MyClass2のxshow()");
    }
}

class virtualmethod01
{
    public static void Main()
    {
        MyClass2 mc2 = new MyClass2();
        mc2.show();
        mc2.xshow();

        Console.WriteLine("-----");
        MyClass1 mc1;
        mc1 = mc2;
        mc1.show();
        mc1.xshow();
    }
}
```

MyClass1には、仮想メソッドのshowメソッドがあります。

また、単なるインスタンスメソッドのxshowメソッドがあります。

MyClass2は、MyClass1クラスから派生してきました。

MyClass2では、showメソッドをオーバーライドしています。

また、xshowメソッドはnewキーワードで基本クラスのxshowメソッドを隠蔽しています。

Mainメソッドでは、まずMyClass2のインスタンス(mc2)を生成しています。

```
mc2.show();
```

では、当然オーバーライドされたshowメソッドが呼び出されますね。

```
mc2.xshow();
```

これは、newで基本クラスのxshowメソッドが隠蔽されているので書きかえられたxshowメソッドが呼び出されますね。

ここまでは、当たり前の結果が予想されます。

さて、次ですが、MyClass1の参照変数mc1にMyClass2の参照を代入しています。

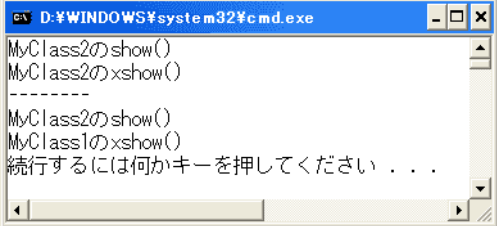
```
mc1.show();
```

は、何が呼ばれるのでしょうか。今までの経験ではMyClass1のshowメソッドが呼ばれそうですね。でも、本当でしょうか。

```
mc1.xshow();
```

これは、すでにやったようにMyClass1のxshowメソッドが呼ばれますね。

では、実行結果を見てみましょう。



ありや、mc1.show();ではMyClass2のshowメソッドが呼び出されていますね。
mc1.xshow();では、当然MyClass1のshowメソッドが呼び出されています。

つまり、基本クラスの参照変数に代入された派生クラスの参照に応じたバージョンのメソッドが呼び出されます。

これが、メソッドの名前の隠蔽とは全く異なる点です。どのメソッドが呼び出されるかは実行時に決定されます。

次の例をみるとよりはっきりすると思います。

```
// virtualmethod02.cs
```

```
using System;
```

```
class MyClass1
{
    public virtual void show()
    {
        Console.WriteLine("MyClass1");
    }
}

class MyClass2 : MyClass1
{
    public override void show()
    {
        Console.WriteLine("MyClass2");
    }
}

class MyClass3 : MyClass2
{
    public override void show()
    {
        Console.WriteLine("MyClass3");
    }
}

class virtualmethod02
{
    public static void Main()
    {
        MyClass3 mc3 = new MyClass3();
        MyClass2 mc2 = new MyClass2();
        MyClass1 mc1 = new MyClass1();

        MyClass1 mcx;

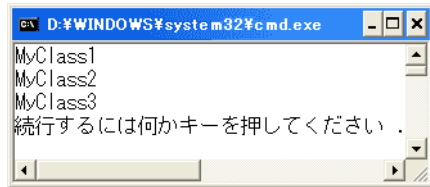
        mcx = mc1;
        mcx.show();
    }
}
```

```
        mcx = mc2;  
        mcx.show();  
  
        mcx = mc3;  
        mcx.show();  
    }  
}
```

Mainメソッドではmcx.show();を3回実行しています。

しかし、それぞれmcxに代入されている参照が異なります。

実行結果は次のようになります。



[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 17/Sep/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。