示了一个简单的类层次结构。在这之后，我们将不会在图中显示 object 了，因为所有的类最终都派生自它。

```
class SomeClass
{ ... }

class OtherClass: SomeClass
{ ... }

class MyNewClass: OtherClass
{
     ...
}
```

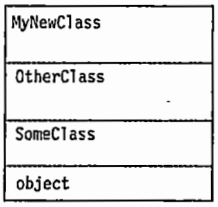

图 8-3 类层次结构

8.4 屏蔽基类的成员

虽然派生类不能删除它继承的任何成员，但可以用与基类成员名称相同的成员来屏蔽（mask）基类成员。这是继承的主要功能之一，非常实用。

例如，我们要继承包含某个特殊方法的基类。该方法虽然适合声明它的类，但不一定适合派生类。在这种情况下，我们希望在派生类中声明新成员以屏蔽基类中的方法。在派生类中屏蔽基类成员的一些要点如下。

- 要屏蔽一个继承的数据成员，需要声明一个新的相同类型的成员，并使用相同的名称。
- 通过在派生类中声明新的带有相同签名的函数成员，可以屏蔽继承的函数成员。请记住，签名由名称和参数列表组成，不包括返回类型。
- 要让编译器知道你在故意屏蔽继承的成员，可使用 new 修饰符。否则，程序可以成功编译，但编译器会警告你隐藏了一个继承的成员。
- 也可以屏蔽静态成员。

下面的代码声明了一个基类和一个派生类，它们都有一个名为 Field1 的 string 成员。使用 new 关键字以显式地告诉编译器屏蔽基类成员。图 8-4 展示了每个类的实例。

```
class SomeClass                          //基类
{
   public string Field1;
   ...
}

class OtherClass : SomeClass             //派生类
{
   new public string Field1;             //用同样的名称屏蔽基类成员
    ↑
 关键字
```

图 8-4　屏蔽基类成员

在下面的代码中，OtherClass 派生自 SomeClass，但隐藏了两个继承的成员。注意 new 修饰符的使用。图 8-5 阐明了这段代码。

```
class SomeClass                                          //基类
{
   public string Field1 = "SomeClass Field1";
   public void   Method1(string value)
       { Console.WriteLine($"SomeClass.Method1:  { value }"); }
}


class OtherClass : SomeClass                             //派生类
{ 关键字
    ↓
   new public string Field1 = "OtherClass Field1";  //屏蔽基类成员
   new public void   Method1(string value)          //屏蔽基类成员
    ↑ { Console.WriteLine($"OtherClass.Method1:  { value }"); }
} 关键字


class Program
{
   static void Main()
   {
      OtherClass oc = new OtherClass();       //使用屏蔽成员
      oc.Method1(oc.Field1);                  //使用屏蔽成员
   }
}
```

该代码产生以下输出：

```
OtherClass.Method1:  OtherClass Field1
```

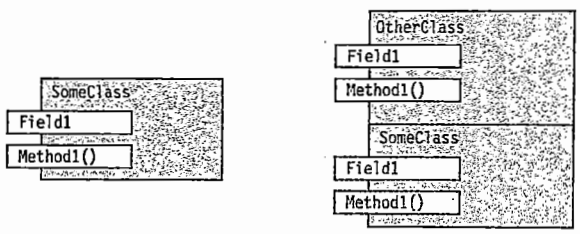

图 8-5　隐藏基类的字段和方法