## 抽象类

抽象类不能实例化,只能被派生.用abstract关键字,创建抽象基类,也可声明抽象方法如

```
abstract class Shape
{
    //强制所有子类来定义如何被呈现
    public abstract void Draw();
    ...
}
```

抽象方法只可以定义在抽象类中.派生类如果没有实现抽象方法,也必须是抽象的.

## 成员遮蔽

使用new 如果派生类定义的成员和定义在基类中的成员一致,派生类就遮蔽了父类的版本.

例如:

```
class threedcircle : circle
{
    //隐藏任何在我之上的PetName 属性
    public new string PetName {get;set;}

    //隐藏任何在我之上的Drow()实现
    public new void Drow()
    {
    }
}
```

## 基类和派生类转换规则

检测兼容性有2种方式: as 和 is

as 关键字

```
Hexagon hex2 = frank as Hexagon ;
if(hex2 == null)
    //提示frank 无法转换成Hesagon 证明 2个类型不兼容
```

is 关键字

```
ClassA
{
....
}
Object o=new Object();
Boolean b1=(o is Object); //b1为true.
Boolean b2=(o is ClassA) ; b2为false.
```

如果对象引用是null,is操作符总是返回false,因为没有可检查的其类型的对象。