父类引用 指向了  子类实体。BaseClass bc = new SubClass( );

（1）父类引用.方法

方法没被子类重写，调用的是父类方法

方法被子类重写，调用的是子类重写的方法

被new的普通方法，调用的是基类的重名方法

方法不能是子类新增的方法

（2）父类引用.变量，变量不具有多态性，调用的一定是父类的变量。如果子类新增的变量和继承的变量同名，调用的仍旧是父类变量。

需要使用子类新增方法和变量，先把引用变量向下转型，获得转型后的引用，

1. 转型引用.方法：调用的是子类新增方法和子类重写的方法和从父类那里继承来但没重写的方法和New的子类的普通方法
2. 转型引用.变量：使用的是子类的新增变量和从父类那里继承来的且不和子类新增变量重名的变量，如果调用从父类继承来的重名变量，使用base。

派生类可以通过base关键字调用基类被隐藏或重写的方法和变量。

普通函数的隐藏

派生类只能重写基类的虚函数，不能重写基类的普通方法。要想修改基类的普通方法，需要用new关键字隐藏基类的普通方法。但是用基类的引用调用的仍是基类的普通方法。

namespace ConsoleApplication4

{

class Base

{

public void baseStr()

{

Console.WriteLine("我是基类的普通方法");

}

public virtual void baseInt()

{

Console.WriteLine("我是基类的可以重写的方法");

}

}

class PaiSheng : Base

{

public override void baseInt()

{

Console.WriteLine("我是派生类重写的基类方法");

}

public new void baseStr()

{

Console.WriteLine("我是派生类的与基类的普通方法重名的方法");

}

}

class Program

{

static void Main(string[] args)

{

Base b = new PaiSheng();

b.baseInt();

b.baseStr();

PaiSheng p = (PaiSheng)b;

p.baseInt();

p.baseStr();

}

}

}

方法重写的原则

两同两小一大原则：方法名相同，形参列表相同，子类方法返回的类型比父类的更小或相等，子类方法抛出的异常比父类更小或相等，子类方法的访问权限应比父类的更大或相等。

重写的方法要么都是类方法，要么都是实例方法。

在子类中，通过base调用父类被重写的方法。

父类被private修饰的方法不被继承，子类中定义同名方法不是重写，而是创建了一个新方法。

重写不会删除父类中的方法，而是对子类的实例隐藏，暂时不使用, 被覆盖的方法只能在子类中只能通过base调用.