1. **为什么使用虚方法？**

在里式转换原则中的第一条，当子类赋值给父类的时候，父类就不能调用子类的方法了，但是我们能给父类定义为虚方法：virtual，给子类定义为重写方法：override，这样就能通过父类调用子类中的方法了

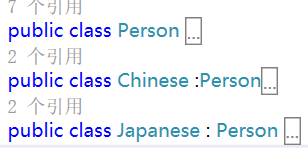
1. **使用虚方法重写的条件：**
2. 必须具有继承关系，并且子类赋值了给父类
3. 父类方法与子类方法同名
4. 父类方法需通过关键字：virtual定义成虚方法，子类方法需通过关键字：override定义成重写方法
5. **父类中定义了一个虚方法，那么子类中不重写也没关系**
6. 我们给父类赋值了xx对象，那么就会使用xx对象的重写方法

如：Student类、Teacher类、还有Person父类

当我们：Person per = new Student()；的时候就会调用Student类

当我们：Person per = new Teacher()；的时候就会调用 Teacher类

1. **虚方法的使用：**
2. 三个类需要存在继承关系



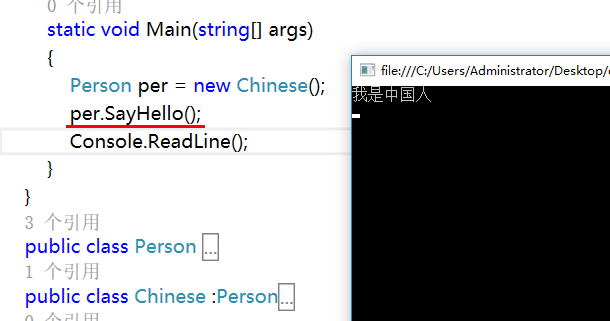
1. 我们需要在父类的方法中通过关键字：virtual定义成虚方法



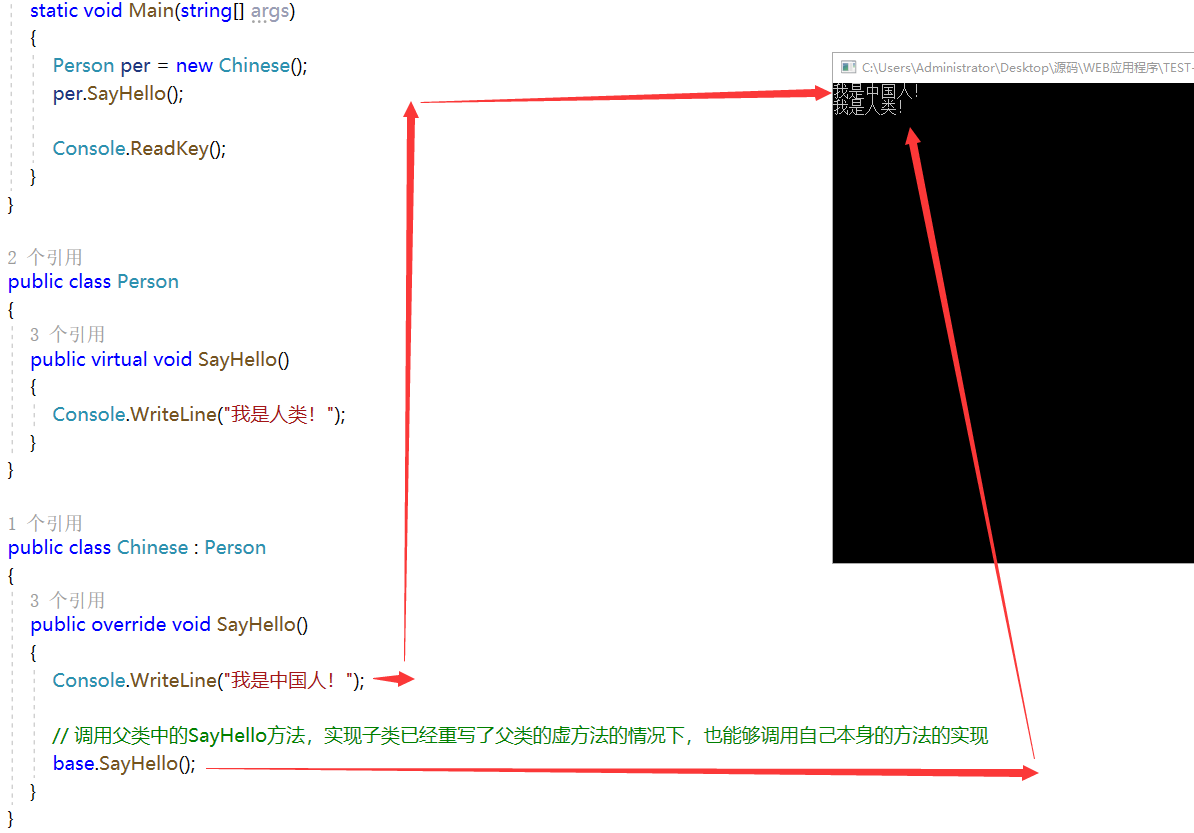
1. 我们需要在子类中通过关键字：override 定义成重写方法



1. 我们把子类Chinese赋值给了父类，并且进行了输出，这时候的输出结果在不写虚方法的时候应该是Person类的打招呼，这时候输出结果变成了Chinese类的打招呼



1. 当我们在子类Chinese所重写父类Person的SayHello虚方法中，回过头来使用关键字[base]去调用父类本身的虚方法，这样就能够实现子类已经重写了父类的虚方法且子类已经赋值给父类的情况下，父类的实例也能够调用自己本身的方法的实现



1. **虚方法的详解**

通过虚方法，我们就能在子类赋值给父类的情况下，父类也能调用子类中的方法了，但是，这时候其实是使用Person类的，因为重写方法的缘故才会子类才会重写方法

我们还能通过写一个父类类型的数组，把子类中的成员装进去（里式转换原则第一条），并且写了虚方法，当存在多个成员的情况下，我们就能通过循环把不同类的SayHello的方法显示出来了，并且数组里面装的是什么类，那么就使用那个类的重写方法，通过虚方法我们就不需要再另外if判断，解决了代码写太多的问题