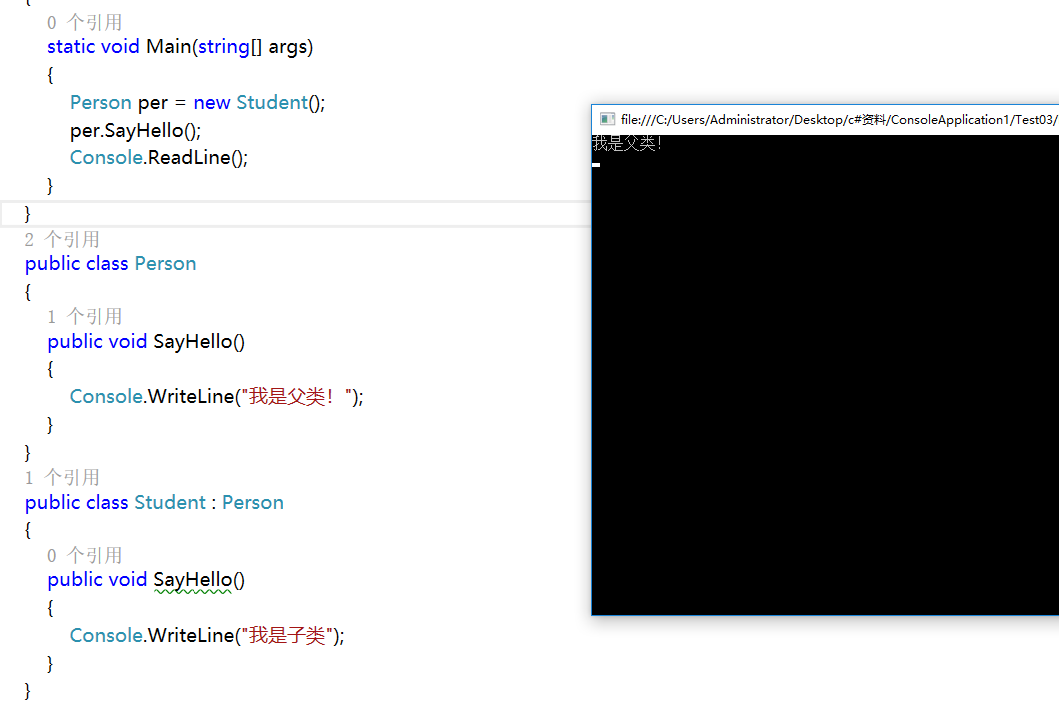
**重写：override**

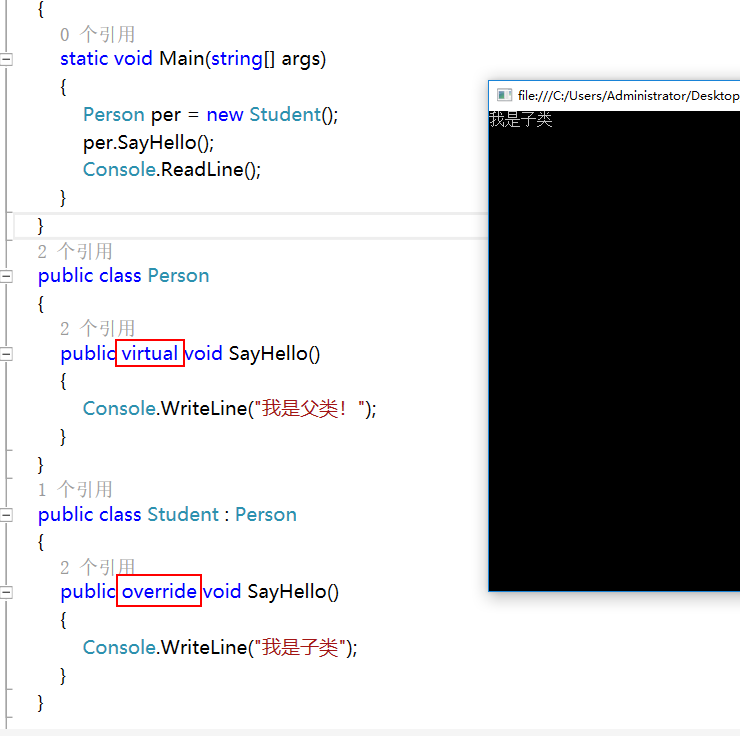
1. 重写最前提条件要有继承关系
2. 只有在父类中存在虚方法/抽象方法的时候子类才能重写



1. 重写只能够针对该类
2. **在子类赋值给父类的情况下，父类就只能够调用自己类中的方法，不能够调用子类中的方法，当我们重写了之后，并且完成了子类赋值给了父类，我们就能够在父类中调用子类的方法了，实现了一个类型多种形态**
3. 子类赋值给父类，并且子类并没有重写父类中的方法，那么父类也就只能够调用父类中的方法



1. 子类赋值了给父类，并且子类重写了父类中的方法，那么父类就能成功调用子类中的方法了，实现了多态



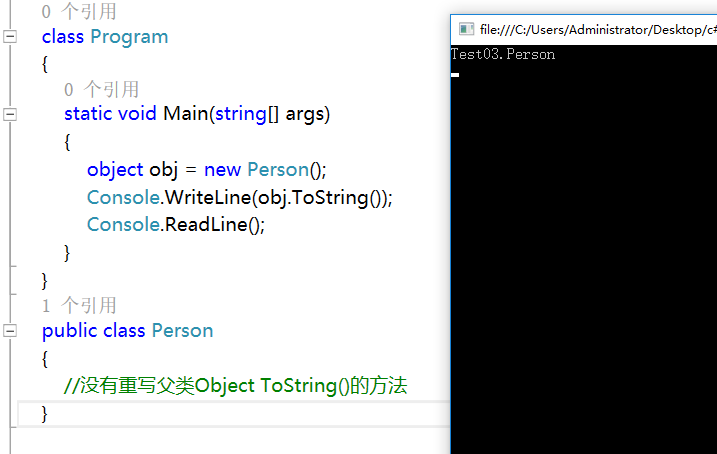
**重写ToString()：**

ToSting()属于object类，是所有类型的基类，并且ToString()这个方法能够把一切对象转化为字符串，那么我们能不能重写呢？

1. 先判断ToSting()这个方法是否为虚方法/抽象方法

Object类中ToString()这个方法为虚方法

1. 在虚方法中，子类可以不进行重写，不重写也不会报错
2. 在抽象类中，子类必须重写父类的抽象方法，不重写的话程序就会报错
3. 按照我们平常使用ToStding()这个方法，我们都没经过重写，并且程序都没报错，所以该方法为虚方法
4. 因为object是所有类型的父类，并且里面的ToSting()；方法是虚方法，如果没有重写，当我们把一个子类赋值给父类的时候，就只能够调用父类的ToString()的方法



1. 那么我们现在重写父类Object中ToString()的方法！就能够在把子类赋值给父类的情况下也能够调用子类中的方法了

