注意这里的代码变化，首先 Student 类的后面没有显式地定义主构造函数，同时又因为定义了次构造函数，所以现在 Student 类是没有主构造函数的。那么既然没有主构造函数，继承 Person 类的时候也就不需要再加上括号了。其实原因就是这么简单，只是很多人在刚开始学习 Kotlin 的时候没能理解这对括号的意义和规则，因此总感觉继承的写法有时候要加上括号，有时候又不要加，搞得晕头转向的，而在你真正理解了规则之后，就会发现其实还是很好懂的。

另外，由于没有主构造函数，次构造函数只能直接调用父类的构造函数，上述代码也是将 this 关键字换成了 super 关键字，这部分就很好理解了，因为和 Java 比较像，我也就不再多说了。

这一小节我们对 Kotlin 的继承和构造函数的问题探究得比较深，同时这也是很多人新上手 Kotlin 时比较难理解的部分，希望你能好好掌握这部分内容。

2.5.3 接口

上一小节的内容比较长，也偏复杂一些，可能学起来有些辛苦。本小节的内容就简单多了，因为 Kotlin 中的接口部分和 Java 几乎是完全一致的。

接口是用于实现多态编程的重要组成部分。我们都知道，Java 是单继承结构的语言，任何一个类最多只能继承一个父类，但是却可以实现任意多个接口，Kotlin 也是如此。

我们可以在接口中定义一系列的抽象行为，然后由具体的类去实现。下面还是通过具体的代码来学习一下，首先创建一个 Study 接口，并在其中定义几个学习行为。右击 com.example.helloworld 包→New→Kotlin File/Class，在弹出的对话框中输入“Study”，创建类型选择“Interface”。

然后在 Study 接口中添加几个学习相关的函数，注意接口中的函数不要求有函数体，代码如下所示：

```
interface Study {
    fun readBooks()
    fun doHomework()
}
```

接下来就可以让 Student 类去实现 Study 接口了，这里我将 Student 类原有的代码调整了一下，以突出继承父类和实现接口的区别：

```
class Student(name: String, age: Int) : Person(name, age), Study {
    override fun readBooks() {
        println(name + " is reading.")
    }

    override fun doHomework() {
        println(name + " is doing homework.")
    }
}
```