

Kotlin 构造函数、继承



SheHuan

1 2019.10.19 19:08:04 字数 1,080 阅读 21,270

Kotlin 的构造函数，以及类的继承，和 Java 相比，在使用上还有些差别的，一些写法并不是很好理解，这里简单的分析记录下。

一、类、对象

在学习 Kotlin 构造函数、继承之前，先简单了解在 Kotlin 中如何定义类、创建对象。

```
1 | class Person {  
2 |     var name: String = ""  
3 |     var age: Int = 0  
4 | }
```

和 Java 一样，在 Kotlin 中也是用 `class` 关键字创建类，默认的可见性修饰符是 `public`。接下来创建一个对象：

```
1 | val p = Person()
```

和 Java 不同的时，Kotlin 中创建对象时省略了 `new` 关键字。

二、构造函数

Kotlin 中构造函数分为主构造函数和次构造函数，每个类默认都会有一个不带参数的主构造函数，例如上边的 `Person` 类。主构造函数的特点是没有函数体，直接定义在类名的后面即可，下边定义 `Person2` 类，并添加带参的主构造函数：

```
1 | class Person2(var name: String, var age: Int) {  
2 |  
3 | }
```

由于主构造函数没有函数体，如果需要在主构造函数中编写代码该怎么做？Kotlin 提供了一个用 `init` 关键字作为前缀的初始化块，可以在这里编写要在主构造函数中完成的业务：

```
1 | class Person2(var name: String, var age: Int) {  
2 |     init {  
3 |         println("main constructor init")  
4 |     }  
5 | }
```

注意主构造函数的参数如果使用了 `var` 或 `val` 修饰符，就相当于在类中声明了对应名称的属性。

一个类可以有多个次构造函数，用 `constructor` 关键字声明。Kotlin 中规定，当一个类既有主构造函数又有次构造函数时，所有次构造函数都必须使用 `this` 关键字直接或间接的调用主构造函数（当类名够加了圆括号，无论有没有参数，就需要遵守这个规定）：

```
1 | class Person2(var name: String, var age: Int) {  
2 |     init {  
3 |         println("main constructor init")  
4 |     }  
5 |  
6 |     constructor(name: String, age: Int, sex: String) : this(name, age) {  
7 | }
```

```
8      }  
9  
10     constructor(name: String, age: Int, sex: String, idCard: String) : this(name, age,  
11  
12     }  
13 }
```

第一个次构造函数直接调用了主构造函数，第二个次构造函数调用了第一个次构造函数，也就是间接的调用了主构造函数。

三、继承

我们上边声明的 `Person`、`Person2` 是不能被直接继承的，因为在 Kotlin 中非抽象类是不能被直接继承的，考虑到抽象类需要有子类继承它才能创建对象，而非抽象类却不需要，所以 Kotlin 默认让非抽象类不能被继承，以防产生不可预知的问题。当然这个默认规则是可以改变的，只需要在非抽象类前加上 `open` 关键字即可：

```
1 open class Person {  
2 }
```

Kotlin 中用冒号 `:` 实现继承，而不是 Java 中的 `extends` 关键字，先看一个简单的：

```
1 class Student : Person() {  
2 }
```

怎么要在被继承的类后加一个括号 `()`？这点又和 Java 的类继承不同。在 Java 的继承特性中有一个规则，子类的构造函数需要调用父类的构造函数，在 Kotlin 中我们同样需要遵循这个规则。这个 `()` 其实就是调用父类的无参构造函数。当 `Student` 类的主构造函数初始化时会调用 `Person` 类的无参数构造函数。

当然，继承一个类时也可以调用父类的有参构造函数：

```
1 class Student(name: String, age: Int) : Person2(name, age) {  
2 }
```

这里继承了 `Person2` 类，并调用了其两个参数的主构造函数，由于 `Person2` 类的主构造函数需要两个参数，我们需要给 `Student` 类定义了包含这两个参数的主构造函数，并将参数传递给 `Person2` 类的构造函数。

如果子类没有主构造函数，有次构造函数，同时父类的构造函数都是带参的，那么继承父类时，如何调用父类的带参构造函数呢？虽然我们一般不会这样定义类来为难自己，但这在 Kotlin 中是支持的。因为子类有次构造函数，所以通过次构造函数，使用 `super` 关键字调用父类的带参构造函数：

```
1 class Student : Person2 {  
2     constructor(name: String, age: Int) : super(name, age) {  
3  
4     }  
5 }
```

由于是通过次构造函数调用父类的带参构造函数，所以 `Person2` 后的括号 `()` 就省略了。