2.5.1 类与对象

现在我们就按照刚才所学的基本思想来尝试进行面向对象编程。首先创建一个 Person 类。右击 com.example.helloworld 包→New→Kotlin File/Class，在弹出的对话框中输入“Person”。对话框在默认情况下自动选中的是创建一个 File，File 通常是用于编写 Kotlin 顶层函数和扩展函数的，我们可以点击展开下拉列表进行切换，如图 2.17 所示。

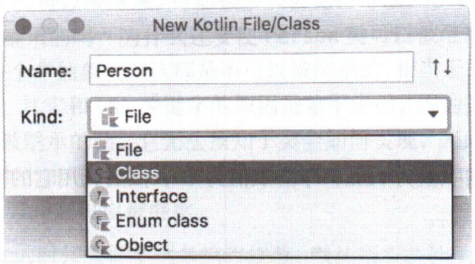

图 2.17 选择创建的类型

这里选中 Class 表示创建一个类，点击“OK”完成创建，会生成如下所示的代码：

```
class Person {
}
```

这是一个空的类实现，可以看到，Kotlin 中也是使用 class 关键字来声明一个类的，这一点和 Java 一致。现在我们可以在这个类中加入字段和函数来丰富它的功能，这里我准备加入 name 和 age 字段，以及一个 eat()函数，因为任何一个人都有名字和年龄，也都需要吃饭。

```
class Person {
    var name = ""
    var age = 0

    fun eat() {
        println(name + " is eating. He is " + age + " years old.")
    }
}
```

简单解释一下，这里使用 var 关键字创建了 name 和 age 这两个字段，这是因为我们需要在创建对象之后再指定具体的姓名和年龄，而如果使用 val 关键字的话，初始化之后就不能再重新赋值了。接下来定义了一个 eat()函数，并在函数中打印了一句话，非常简单。

Person 类已经定义好了，接下来我们看一下如何对这个类进行实例化，代码如下所示：

```
val p = Person()
```

Kotlin 中实例化一个类的方式和 Java 是基本类似的，只是去掉了 new 关键字而已。之所以这么设计，是因为当你调用了某个类的构造函数时，你的意图只可能是对这个类进行实例化，因此