```
    @Override
    public int hashCode() {
        return brand.hashCode() + (int) price;
    }

    @Override
    public String toString() {
        return "Cellphone(brand=" + brand + ", price=" + price + ")";
    }
}
```

看上去挺复杂的吧？关键是这些代码还是一些没有实际逻辑意义的代码，只是为了让它拥有数据类的功能而已。而同样的功能使用 Kotlin 来实现就会变得极其简单，右击 com.example.helloworld 包→New→Kotlin File/Class，在弹出的对话框中输入“Cellphone”，创建类型选择“Class”。然后在创建的类中编写如下代码：

```
data class Cellphone(val brand: String, val price: Double)
```

你没看错，只需要一行代码就可以实现了！神奇的地方就在于 data 这个关键字，当在一个类前面声明了 data 关键字时，就表明你希望这个类是一个数据类，Kotlin 会根据主构造函数中的参数帮你将 equals()、hashCode()、toString()等固定且无实际逻辑意义的方法自动生成，从而大大减少了开发的工作量。

另外，当一个类中没有任何代码时，还可以将尾部的大括号省略。

下面我们来测试一下这个数据类，在 main()函数中编写如下代码：

```
fun main() {
    val cellphone1 = Cellphone("Samsung", 1299.99)
    val cellphone2 = Cellphone("Samsung", 1299.99)
    println(cellphone1)
    println("cellphone1 equals cellphone2 " + (cellphone1 == cellphone2))
}
```

这里我们创建了两个 Cellphone 对象，首先直接将第一个对象打印出来，然后判断这两个对象是否相等。运行一下程序，结果如图 2.22 所示。

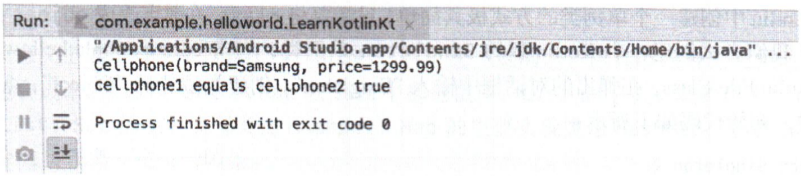

图 2.22 测试 Cellphone 数据类的结果

很明显，Cellphone 数据类已经正常工作了。而如果 Cellphone 类前面没有 data 这个关键字，得到的会是截然不同的结果。如果感兴趣的话，你可以自己动手尝试一下。