

Kotlin---类型判断与强转

None_Ling [关注](#)

0.245 2018.11.22 20:07:23 字数 279 阅读 16,214

准备工作

准备两个类 `Person` 与 `Child`

`Person`类如下：

```
1 | open class Person(name: String, age: Int) {
2 |     var mName = name
3 |     var mAge = age
4 |
5 |     open fun printInfo(): Int = Log.e("TAG", "$mName....$mAge")
6 | }
```

`Child`类如下：

```
1 | class Child(name: String, age: Int, canWalk: Boolean) : Person(name, age) {
2 |     var mCanWalk = canWalk
3 |
4 |     override fun printInfo(): Int = Log.e("Child", "Child...$mName...$mAge...$mCanWalk")
5 |
6 |     fun printCanWalk() = Log.e("TAG", "Child Can Walk....$mCanWalk")
7 | }
```

类型判断

在java中使用 `instanceof` 来判断类型。而在Kotlin中，则是使用 `is` 来判断：

```
1 | var person = Person("Bob", 30)
2 | if (person is Child) {
3 |     person.printCanWalk()
4 | } else {
5 |     person.printInfo()
6 | }
```

可以看到在判断完类型后，可以直接调用该类型的函数。这部分是编译器替我们完成的，并且在条件表达式中同样可以使用

```
1 | if (person is Child && person.mCanWalk) {
2 |     person.printCanWalk()
3 | }
```

类型强转

当有确定类型时候，就可以使用 `as` 进行强制转换了：

```
1 | var person = Person("Bob", 30)
2 | var child = person as Child
```

但是如果在强转时候，发现类型不匹配，则会报Exception。而如果是 `null` 类型的对象强转时，可以通过 `?` 来允许可空数据进行判断

```
1 | var person : Any = Person("Bob", 30)
2 | var child: Child? = person as? Child
3 | child?.printCanWalk()
```

这样的话，当类型判断为空时，则会返回null，而在执行 `child?.printCanWalk()` 语句时，会判断 `child` 是否为空，如果为空的话，则不会执行，如果不为空的话，就会执行。

而将person标为 `Any`，可以相当于Java中的Object，为了去除编译器的警告。

