Kotlin基础 — object

kotlin的延迟加载主要有两种：lateinit和lazy

# 1.lateinit

lateinit是kotlin中的一个关键字，使用方法如下

// 声明一个string变量

lateinit var a1: String

private fun test() {

// 初始化

a1 = "test1"

}

在var之前添加lateinit，然后选择你想要的时候，初始化，但是有以下需要注意的地方：

1. lateinit只能修饰变量var，不能修饰常量val
2. lateinit不能对可空类型使用
3. lateinit不能对java基本类型使用，例如：Double、Int、Long等

4）在调用lateinit修饰的变量时，如果变量还没有初始化，则会抛出未初始化异常，报错

补充

Kotlin相信当开发者显式使用lateinit var 关键字的时候，他一定也会在后面某个合理的时机将该属性对象初始化的(然而，谁知道呢，也许他用完才想起还没初始化)。

# 2.lazy

lazy也是kotlin中常用的一种延迟加载方式，使用方法如下：

val a2:String by lazy{

println("开始初始化")

// 初始化的值

"sss"

}

使用时，在类型后面加by lazy{}即可，{}中的最后一行代码，需要返回初始化的结果，

上述代码中，"sss"即为最后初始化的值。下面是lazy的一些注意点：

1. lazy只能对常量val使用，不能修饰变量var
2. lazy的加载时机为第一次调用常量的时候，且只会加载一次（毕竟是个常量，只能赋值一次）

原文链接：https://blog.csdn.net/liyi1009365545/java/article/details/84236433