<Kotlin in Action> Review

Chapter1

Simple Introduction etc

Chapter2-kotlin基础

1. 关键字fun用来声明函数
2. 参数和变量的类型写在名称后面
3. 函数可以定义在文件的最外层不用放在类中
4. 分号省略
5. kotlin中，if else是有结果值的表达式，与java相反，赋值是语句而非表达式
6. 若一个函数的函数体由单个表达式组成，可以简写

e.g fun max(a: Int, b: Int) = if (a>b) a else b

1. 可变变量var，不可变变量val，val声明的变量仅能有一次初始化
2. val引用自身不可变，但是指向的对象可变
3. var可以改变自己的值，但是无法改变类型
4. 字符串转义，“\”，反斜杠。

X = 1

Println(“\$x”) console output: x$

Println(“$x”) console output: 1

1. println(“expression=${expression}”)可以直接在String中打印出expression的值，编译后的代码实际上new了一个StringBuilder，因此效率上ok
2. kotlin中，public是默认属性
3. class Person（val name: String）直接声明了一个具有name作为属性的Person类，且具有public的访问属性，不可修改，创建该类实例时，需要一个String类型的参数
4. 枚举类声明

enum class Color{

RED, ORANGE, GREEN, YELLOW

}

1. 带属性的枚举类

enum class Color(val r: Int, val g: Int, val b: Int){

RED(255, 0 0), ORANGE(255, 165, 0);//注意有分号

Fun rgb() = (r \* 256 + g) \* 256 + b

}

1. when表达式

when(statement){

condition1 -> expression1

condition2, condition3 -> expression2

else -> expression3

}

when{

Boolean condition1 -> expression1

Boolean condition2 -> expression2

}

when 也是表达式

if&when最后一个表达式就是结果

1. kotlin is : Java instance of
2. kotlin显式转换 as

null不能显示转换成其他类型的值

1. “代码块中最后的表达式就是结果”,在所有使用代码块并期望得到一个结果的地方成立
2. while (condition){

//do sth

}

do{

//do sth

}while(condition)

1. kotlin的区间是闭区间 1..3 = [1,2,3]
2. 根据键来访问和更新map

kotlin map[key] = value

java map.put(key, value)

1. for 迭代集合时使用下标

for ((index, element) in list.withIndex()){

//do sth

}

1. in 关键字

1 in 1..10 true

‘c’in ‘a’.. ‘f’true

不在区间内 !in

1. try catch finally 基本同java

Chapter03(函数的定义与调用)

1. 创建集合

val set = hashSetOf(1, 2, 3)

val list = arrayListOf(1, 2, 3)

val map = hashMapOf(1 to “one”, 2 to “two”)

1. kotlin的javaClass等价于Java的getClass
2. 声明参数带默认值的函数

fun max(a:Int = 1,b: Int) = …

使用命名参数，可以省略中间参数，可以用任意顺序给(没有默认值的参数要按照顺序给)

e.g val maxNumber = max(2, a = 1)

1. 顶层函数和属性
2. kotlin允许使用导入类一样的语法来导入函数，可以用as来修改导入的类和函数
3. java调用kotlin扩展函数

in kotlin file

StringUtil.kt

fun String.lastChar(): Char = get(length - 1)

in java file use kotlin function

StringUtil.lastChar(“abc”)

1. 扩展函数不能被子类重写
2. 可变参数声明 vararg
3. 中缀调用to
4. to(“one”) 等价于

1 to one

允许中缀符号调用函数 infix修饰符

1. 解构声明

infix fun Any.to(other: Any) = Pair(this, other)

val (number, name) = 1 to “one”

1. 在kotlin中，可以使用扩展函数toRegex将字符转转换成正则表达式

“abc.abc-abc”.split(“[\\.|-](file://.|-)“.toRegex)

“abc.abc-abc”.split(“.”，”-”)

1. 在一个写在三重引号的字符串中，不需要对任何字符进行转义

Chapter04 类、对象和接口

1. kotlin的类默认是final和public的
2. kotlin用冒号来代替implement和extends
3. 一个类可以实现任意个接口，但是只能继承一个类
4. kotlin接口允许有方法的默认实现
5. 继承的多个接口有同样名称的方法，在子类中调用的时候

super<interfaceOne>.functionName()

super<InterfaceTwo>.functionName()

1. kotlin的嵌套类不能访问外部类的实例
2. sealed修饰符：

隐含这个类是open的

限制继承结构

1. constructor init
2. kotlin接口可以包含抽象属性声明
3. 类委托
4. 伴生对象
5. 对象表达式

Chapter05 Lambda编程