# 第一部分 Kotlin基础篇

## 一 新建Kotlin项目

## 二 Kotlin中类与对象

### 1 类与继承

kotlin中定义类如下：

class ClassName [constructor][(params:Type,…)] [{ ... }]

[]表示可以省略

类声明由类名、类头（指定其类型参数、主构造函数等）以及由花括号包围的类体构成。**类头与类体**都是可选的； 如果一个类没有类体，可以省略花括号

class Empty

在 Kotlin 中的一个类可以有一个主构造函数以及一个或多个次构造函数。主构造函数是类头的一部分：它跟在类名（与可选的类型参数）后

class Person constructor(firstName: String) { ... }

如果主构造函数没有**任何注解或者可见性修饰符**，可以省略这个 constructor关键字。

class Person(firstName: String) { ... }

主构造函数不能包含任何的代码。初始化的代码可以放到以 init 关键字作为前缀的初始化块（initializer blocks）中。

在实例初始化期间，初始化块按照它们出现在类体中的顺序执行，与属性初始化器交织在一起：

class User(var name:String,var age:Int){

init {

println("${this.javaClass.simpleName} init:$name $age")

}

}

请注意，主构造的参数可以在初始化块中使用。它们也可以在类体内声明的属性初始化器中使用：

class Customer(name: String) {

val customerKey = name.toUpperCase()

}

事实上，声明属性以及从主构造函数初始化属性，Kotlin 有简洁的语法：

class Person(val firstName: String, val lastName: String, var age: Int) { …… }

如果构造函数有注解或可见性修饰符，这个 constructor 关键字是必需的，并且这些修饰符在它前面：

class Customer public @SuppressWarnings constructor() {

}

次构造函数

3 构造函数

4 创建类实例

var user:User = User("heihei",20)

var user2:User2 = User2("美女",18)

var user1:User1 = User1()