Android 10

Kotlin

编程通俗演义

掌握Kotlin编程

提高Android应用开发效率

牛 搞 著

清華大學出版社

Android 10 Kotlin 编程通俗演义

牛 搞 著

清華大學出版社
北 京

内 容 简 介

Google 已经将 Kotlin 列为 Android 开发第一开发语言。Kotlin 与 Java 无缝兼容，同时 Kotlin 作为一门新语言，其语法极其简洁精练，稍微熟悉之后，开发效率立即会有明显提升。

本书分为 20 章，严格参考 Android 10 官方开发文档，全面讲解利用 Kotlin 开发 Android 应用的各种技术，章节精心安排、循序渐进，内容准确、翔实、全面而又通俗易懂，绝不是术语的罗列，也绝不是不知所云的翻译。

本书既适合 Android 应用开发初学者、转向 Kotlin 编程的 Android 应用开发人员阅读，也适合高等院校和培训学校计算技术相关专业的师生参考。

图书在版编目（CIP）数据

Android 10 Kotlin 编程通俗演义/牛搞著. —北京：清华大学出版社，2020.5
ISBN 978-7-302-55274-1

Ⅰ. ①A… Ⅱ. ①牛… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字（2020）第 050762 号

责任编辑：夏毓彦
封面设计：王　翔
责任校对：闫秀华
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市铭诚印务有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张：**26.75　　**字　　数：**685 千字
版　　次：2020 年 6 月第 1 版　　**印　　次：**2020 年 6 月第 1 次印刷
定　　价：89.00 元

产品编号：084558-01

前　　言

写作背景

2020 年了，Android 开发的热度怎么样了？学习它，对就业和薪资提升帮助大吗？我想这是大多数人最关心的问题。

一门技术在职场中的需求热度，通过大型求职招聘网站可以很容易分析出结论。大体可以这样说，移动端开发作为软件生态的一部分，从来都有很强的需求。在 2017 年之前，Android 原生开发曾一度进入低谷，因为很多团队都选择基于 JavaScript 的跨平台开发框架。但是，这些框架也存在一些先天缺陷，主要是由于 Android 与 iOS 的巨大差异造成的（这两大系统不可能统一，为了商业利益，必须互相制造壁垒）。事实已经证明了一点，跨平台开发始终绕不开原生开发。所以，2017 年之后，Android 原生开发重新被重视，甚至有国外公司完全回归了原生开发。当前，跨平台开发依然在迅速发展，但是始终绕不过原生开发，而且有些功能只能用原生开发实现。所以，要进行移动开发，必须学习原生开发!

本书作者有 15 年以上软件开发实战经验、5 年以上 IT 实训教学经验，深入了解各种技术、架构、设计模式，对 IT 教育有丰富的体验和深入的思考，对各种技术善于以通俗易懂的语言进行透彻讲解。

本书导读

本书是《Android 9 编程通俗演义》的姊妹篇，作者在其基础上修正部分错误，改进多处设计，将开发语言由 Java 升级为 Kotlin，紧紧追赶 Google 的步伐。

“我有一个梦想，让天下没有难学的技术！”本书与《Android 9 编程通俗演义》一书的写作风格一致：通俗易懂，具体直观，注重实践，以为读者节省脑细胞作为终极目标。

我一直希望能写出一本让读者轻轻松松学编程的书，如果能把学习当作一种休闲方式，那该是多么美好的事情！当然了，众口难调，一本书的风格不可能满足所有人的口味。在本书创作中，作者已尽量做到照顾更多的人，尤其照顾基础差的人，并且尽量少说黑话，努力使它成为一部不那么“反人类”的作品，相信大部分人都很容易接受这种风格。因为从上一本书的读者反馈看来，效果很不错！

本书应该怎么去阅读？答案就一句话：“看就行了！”

如果你是一个勤快人，可以边看边跟着做；如果是一个懒人，那么仅仅停留在“看”上。你可以躺着看、坐着看、趴着看，最好不要走路看，因为对眼睛不好。

本书翔实地讲述一个 Android App 的实现过程，并对很多基础知识进行了专门补齐。实现 App 的每一步都有截图，你不用写代码，也能看到结果。所以，阅读体验是很轻松的。

本书从头至尾讲了一个故事：开发一个 Android 版高仿 QQ App 的故事。本书的内容结构是这样的：

- 第 1 章：Kotlin 语言快速入门。
- 第 2~4 章：Android 开发准备与初步体验。
- 第 5～14 章：Andorid 基本功能与界面开发。
- 第 15、16 章：实现仿 QQ App 单机版。
- 第 17～19 章：Android 多线程、网络开发。
- 第 20 章：实现仿 QQ App 网络聊天版。

示例源码下载

第 14 章之前讲解基础知识，示例项目为 FirstCotlinApp，其 Git 仓库地址是 https://gitee.com/nnn/FirstCotlinApp.git。

第 15 章和第 16 章的项目为无网络通信的仿 QQApp，项目名为 QQApp，其 Git 仓库地址为 https://gitee.com/nnn/QQAppCotlin.git。

第 20 章的项目为带网络通信功能的仿 QQApp，是从 QQAppCotlin 改进而来的，因此项目名和包名皆与 QQAppCotlin 相同，其 Git 仓库地址为 https://gitee.com/nnn/QQAppCotlinHttp.git。

另外，为了模仿 QQApp 中的树状显示效果，作者还创建了一个开源项目 RecyclerListTreeView，托管于 GitHub，现已被多人用于商业项目。在本书中亦有对其用法的详细介绍，地址为 https://github.com/niugao/RecyclerListTreeView。

对本书内容或各项目有任何疑问，可在 gitee 或 GitHub 中的项目仓库页面直接留言，也可在作者的 CSDN 博客 https://blog.csdn.net/niu_gao/中留言。

读者对象

- 了解 Java 语言，想学习 Kotlin 语言和 Android 开发的初学者
- 想快速了解 Android 开发模式的资深开发人士
- 有一定 Android 开发基础，想进一步提升实战能力的开发人员
- 需要工程教育实践案例的高校教师

致谢

首先感谢各位读者，你们的肯定给予我笔耕不辍的信心和动力！其次要感谢清华大学出版社夏毓彦编辑的大力支持和指导，让我可以专注于内容，充分体验作为作者的乐趣。再次感谢我的家人和朋友，是你们的鼓励与支持给了我动力。最后感谢我自己，耐住寂寞，坚持不辍，能为世人留下一两部作品，真的感觉人生没有虚度。

牛　搞

2020 年 3 月

目　　录

第 1 章

◂Kotlin快速入门▸

Java 和 Kotlin 都是 Android 的官方开发语言，但是 Kotlin 已上升为第一开发语言，Java 屈居第二。

Kotlin 的官网地址是 https://kotlinlang.org。

Kotlin 在底层与 Java 完全兼容，而且 Kotlin 是强类型语言，编译产物是 Java 的 class 文件，要基于虚拟机运行，所以 Kotlin 与 Java 可以说是一体两面、无缝结合。但是，Kotlin 比 Java 更进一步，它编写的程序可以做到不依赖于虚拟机运行，这被称为 Native（原生）方式，就像 C 程序的运行方式，当然比虚拟机快多了，这种运行方式对于移动设备来说意义重大！

如果说 Kotlin 代表了未来开发语言的方向也不算夸张，因为它很新，站在了前人的肩膀上。如果去研究一下各种新出现的语言（比如 Apple 的 Swift），会发现它们的语法规则几乎完全一样。

当前的 Java 使用者大都还停留在第 8 版（JDK 1.8），因为很多库、框架或系统都最高支持到 Java 8。写作此书时，Java 13 就要出世了，Java 8 之后的语法改进有很多。这些改进都体现了新式语法，但是很多人对新式语法不熟悉，甚至看到后感到别扭，然而新式语法思想是每个软件开发者都应该理解和掌握的。

其实要掌握新式语法并不困难，还可以说是一件很轻松的事。万变不离其宗，只要掌握了一门语言，再学另一门就很快，当然要有一个条件：有一本好的、适合的指引手册。本书就是为 Java 开发者提供的一本 Kotlin 快速学习手册。

1.1 开发环境配置

开发环境的配置仅需两步：安装 JDK；安装 IDE。

1.1.1 安装 JDK

在地址“https://www.oracle.com/technetwork/java/javase/downloads/index.html”中选择要下载的 JDK，见图 1-1。

单击图 1-1 箭头所指图标，进入新页面，在最下面能看到图 1-2 所示的内容。

图 1-1

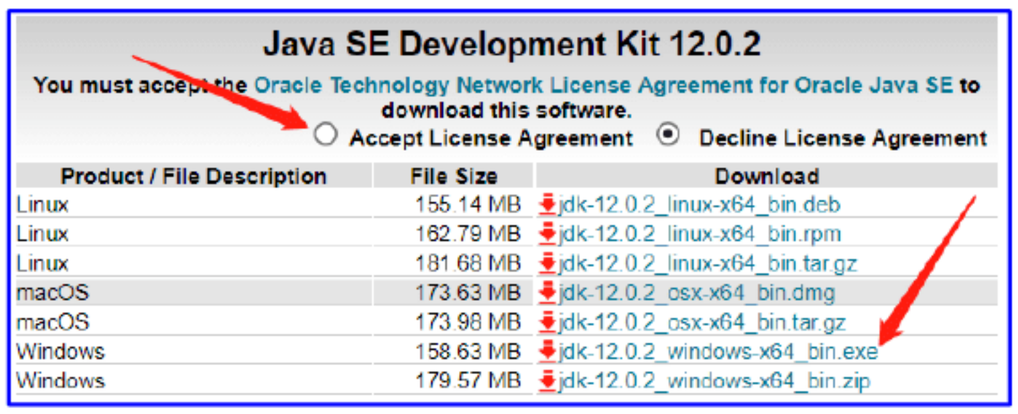

图 1-2

注意一定要选择“Accept License Agreement（同意许可协议）”，才可以用鼠标单击下面的文件链接。

如果是 Windows，建议选择可执行文件（它是安装包，可以自动设置很多配置），并且在安装过程中不要改变默认安装路径，这样不会引起不必要的麻烦。

需要注意的是，JDK 安装到的路径中不能有中文，否则会引起莫名其妙的问题。默认安装位置一般是“系统盘：\Program files”。

1.1.2 安装 IDE

仅有 JDK 虽然可以开发软件，但是要手动维护一切，可以借助开发工具来提高编程效率。Kotlin 是 JetBrains 开发出来的，而 JetBrains 的主业为开发工具，所以选择 JetBrains 家的 IDEA。

IDEA 可以说是 Java 编程的首选 IDE，当然也是 Kotlin 的首选，它的官网地址是“https://www.jetbrains.com/idea/”。下载 IDEA 很简单，在页面中单击图 1-3 所指的“DOWNLOAD”图标即可。进入下载页面，如图 1-4 所示。

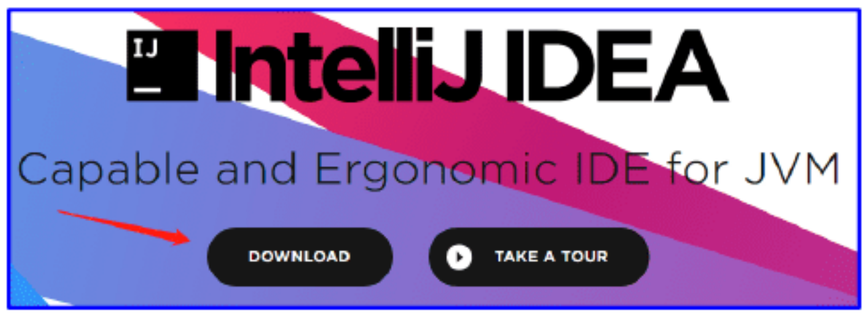

图 1-3

图 1-4

有两个版本可供选择：Ultimate（旗舰版）和 Community（社区版）。旗舰版功能强大，但是要收费；社区版功能少，是免费的。选择社区版来学习 Kotlin 就足够了。

下载的文件是一个安装包，运行它即可安装 IDEA。安装过程最好保持默认设置，安装完就可以用。

跟 JDK 一样，注意 IDEA 所安装到的路径中不能有中文，否则亦会引起莫名其妙的问题，它的默认安装位置一般是“系统盘：\Program files”。

1.1.3　创建第一个 Kotlin 工程

其实 Android 的 IDE Android Studio 就是 IDEA。Google 为 IDEA 开发了 Android 插件，把它和 IDEA 绑定在一起（取名为 Android Studio）供我们下载，所以 Android Studio 的界面与 IDEA 的界面是一样的。

第一次运行 IDEA，会出现如图 1-5 所示的页面。

图 1-5

IDEA 的功能项有“创建新工程”（Create New Project）“引入工程”（Import Project）“打开已有工程”（Open）“从版本控制系统导入工程”（Check out from Version Control）。要创建新工程，选择第一项之后出现创建工程向导，如图 1-6 所示。

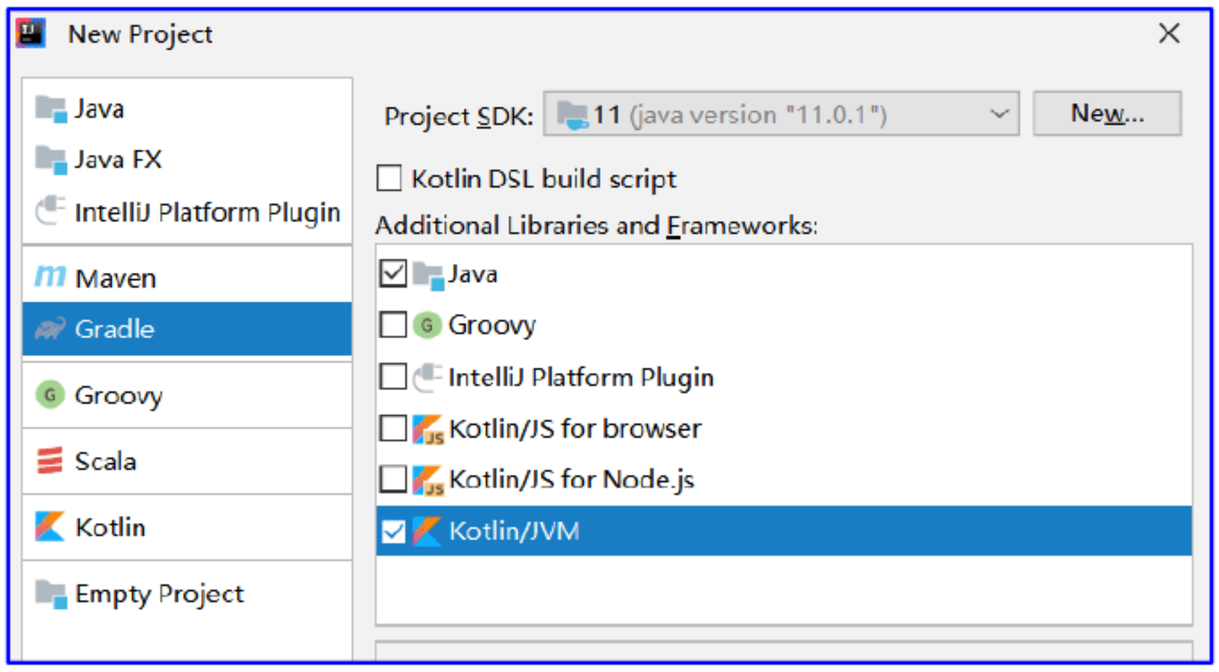

图 1-6

可以创建多种类型的工程，仅 Kotlin 工程就有多种形式，那么选择哪一种呢？推荐创建基于 Gradle 的 Kotlin/JVM 工程。Gradle 是当前如日中天的工程管理软件之一（另一个与它齐名的是 Maven），主要用于管理 Java 工程，但是 Kotlin 与 Java 是同一“种族”，所以也适合使用 Gradle 管理。使用 Kotlin 可以开发多种程序：

- Kotlin/JS 表示用 Kotlin 开发 JavaScript 程序，其实是把 Kotlin 代码翻译成 JavaScript 代码，然后才能在浏览器或 Node.js 环境中执行。
- Kotlin/JVM 表示用 Kotlin 开发基于 JVM 的程序，也就是基于 Java 虚拟机运行的程序，其实就是以 Kotlin 代替 Java 编写代码，编译出的就是 class 文件。

建议选择 Kotlin/JVM 类型的工程，因为 Kotlin 对此类型支持得最好，在此类型的工程中可以使用 Kotlin 的所有特性。选好后单击 Next（下一步）按钮，出现如图 1-7 所示的内容。

在图 1-7 中可以设置程序的名字，在 GroupId（组名）中一般填入的是颠倒的域名，这里主要用于区分不同组织发布的程序，因为域名肯定是唯一的，所以都填写域名。本例中填的是“com.niuedu”，随手写的，只是为了演示一下。

ArtifactId（产品名）指的是程序名，默认与工程同名，所以最好不要用中文，中间也不能用空格等非常规字符，比如取名“HelloKotlin”就合乎规则和习惯。填完后，单击 Next 按钮，进入图 1-8 所示的页面。

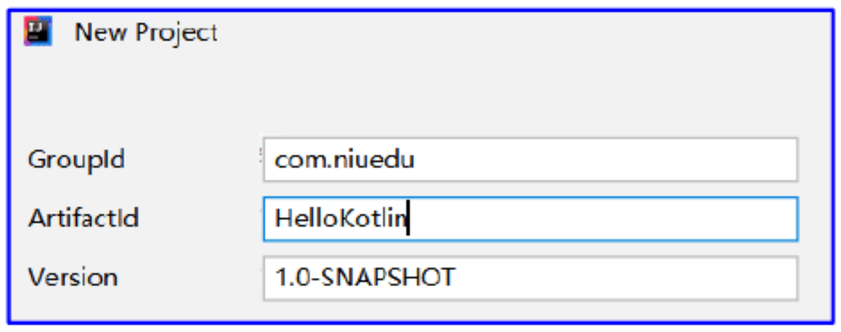

图 1-7

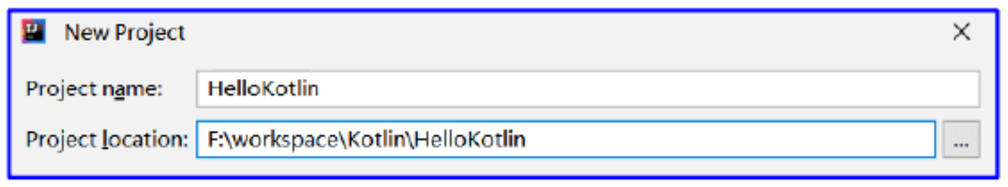

图 1-8

可以修改工程的名字和工程所保存到的路径，这里就用默认的路径，单击 Finish（完成）按钮，Gradle 会根据配置自动生成一个工程，同时 IDEA 会进入编辑模式，如图 1-9 所示。

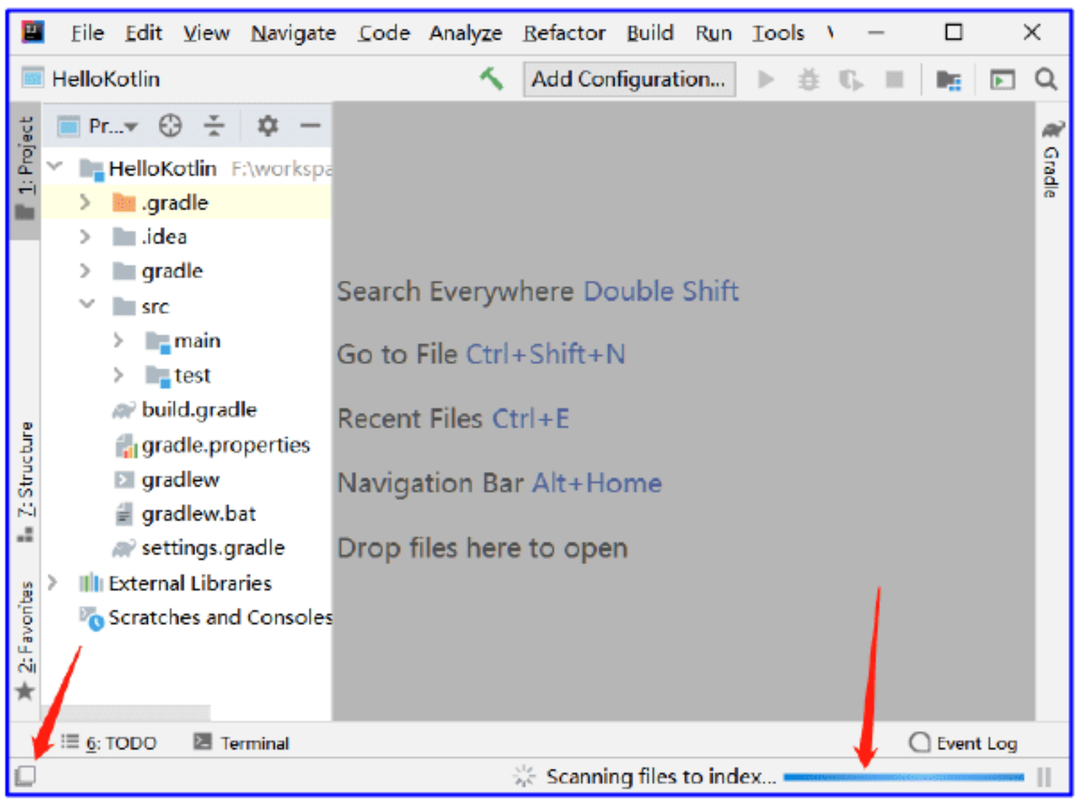

图 1-9

整个工作区的构成非常主流化，左边是工程目录，右边是内容编辑区（现在没有打开任何文件）。注意下面的状态栏，左边这个小图标是切换左右竖向工具条的开关，初用者如果不小心点，就会找不着某些窗口。右边是进度条，要十分注意这个地方，如果此处有进度条或文字出现，则表示 IDEA 正在忙着做什么事，此时最好不要动工程中的文件，等 IDEA 忙完了，再编辑文件。尤其是第一次创建工程，可能需要很长时间，因为 Gradle 工程会严重依赖网络，自动下载很多文件，包括管理工程的插件以及工程所依赖的库，如果网速慢（这是很可能的，因为文件仓库服务器在国外），就可能需要漫长的等待。如果工程创建不成功，十有八九是因为网络问题导致某些文件没有下载成功，这时就需要重试（IDEA 会提示重试）下载。

下面了解一下工程的组织结构。

1.1.4　工程组织结构

当工程创建成功后，可以看到图 1-10 这样的目录结构。

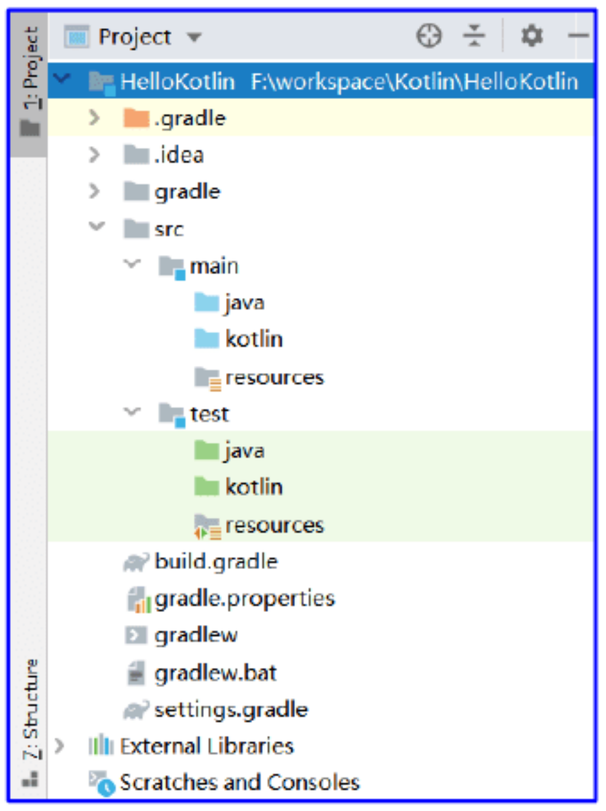

图 1-10

稍微解释一下：.gradle、.idea、gradle 这三个文件夹是 IDEA 自己产生的，用于工程管理，我们不用它们。

src 下是工程源码和非源码文件的保存地，但是不能随便放这些文件，src/main 下存放的是源码，java 下存放的是 Java 源码，kotlin 下存放的是 kotlin 源码，resources 下存放的是非源码文件，比如图片、配置文件等。test 与 main 的目录结构相同，test 起什么作用呢？它下面存放的是单元测试代码。

注意！这种目录结构是固定的，不要试想通过一些配置来改变目录的名字或作用，这种理念名曰“约定大于配置”。

根目录下的 build.gradle 文件是整个工程管理的核心文件，因为它是工程的配置文件，当前它的内容是这样的：

```
plugins {
    id 'java'
    id 'org.jetbrains.kotlin.jvm' version '1.3.41'
}

group 'com.niuedu'
version '1.0-SNAPSHOT'

sourceCompatibility = 1.8

repositories {
    mavenCentral()
}

dependencies {
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk8"
    testCompile group: 'junit', name: 'junit', version: '4.12'
}

compileKotlin {
    kotlinOptions.jvmTarget = "1.8"
}
compileTestKotlin {
    kotlinOptions.jvmTarget = "1.8"
}
```

这些代码是用 Groovy 语言编写的，配置了工程的一些工具或参数，比如工程管理所需插件（Plugins）、源码兼容性（Source Compatibility）、仓库（Repositories）、依赖的库（Dependencies）等。改动比较多的是依赖库。

至于根目录下的其他文件，都是与 Gradle 相关的配置文件或脚本工具，也是自动生成的，我们不需要关心。

虽然工程下有这么多文件，但是这个工程是空工程，因为没有实质的程序代码，下面就来添加代码完成第一个程序。

1.1.5 添加代码

与 Java 相似，先创建一个包，在包下再创建文件，与 Java 不同，main 函数不用写在类中，直接作为全局函数即可。

首先创建一个包，见图 1-11。

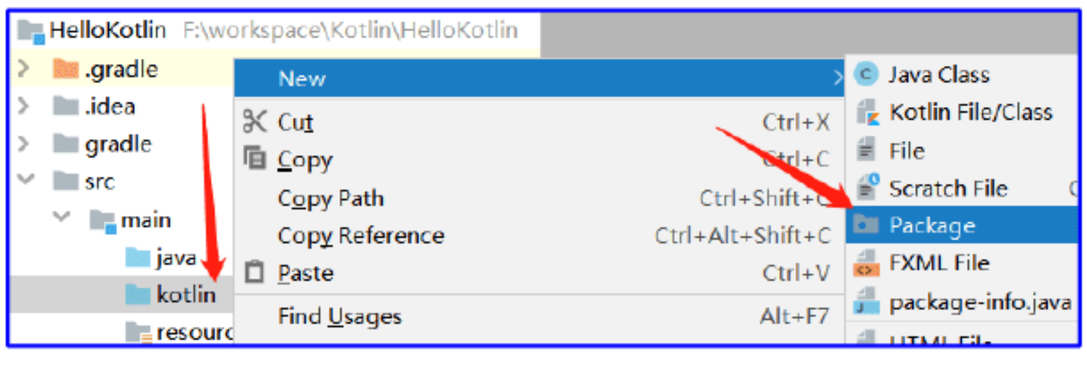

图 1-11

用鼠标右击“kotlin”目录，在弹出的快捷菜单中选择“New”→“Package”，出现图 1-12 所示的界面。在这里填上包名，单击“OK”按钮，会在 main/java 下创建 com.niuedu 包，在包上右击，弹出如图 1-13 所示的快捷菜单。

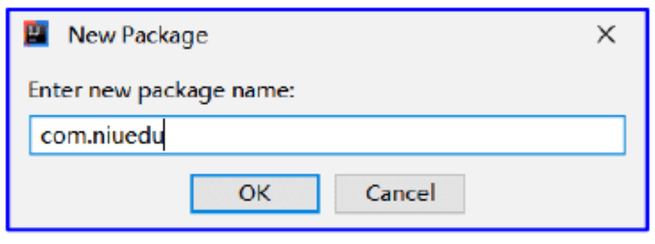

图 1-12

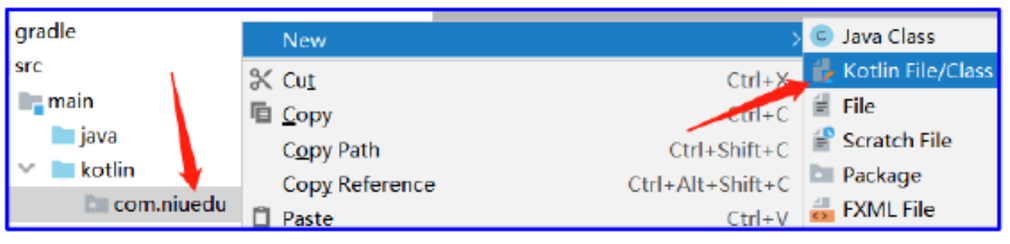

图 1-13

选择“Kotlin File/Class”之后，出现图 1-14 所示的界面。

接下来选择所创建文件的类型，填入文件名“HelloApp”，选择“File”并双击，会在 main/kotlin/com.niuedu 下添加文件 HelloApp.kt。编辑此文件，最终内容如下：

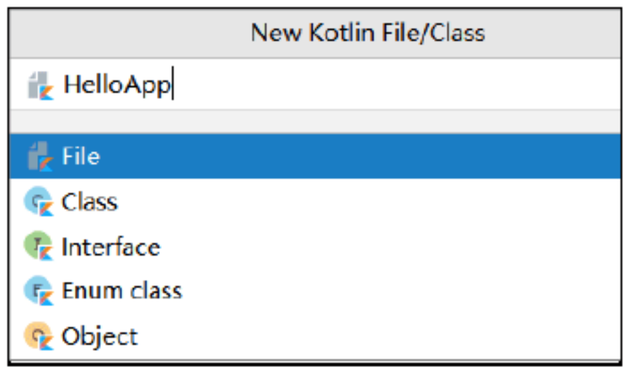

图 1-14

```
package com.niuedu

fun main() {
    println("Hello world!")
}
```

Main()函数很简单，其内容是打印一条文本。代码有了，如何运行呢？请看下节讲解。

1.1.6 运行程序

与 Java 工程一样，需要先配置运行方式。单击图 1-15 所示的位置，进入配置页面。

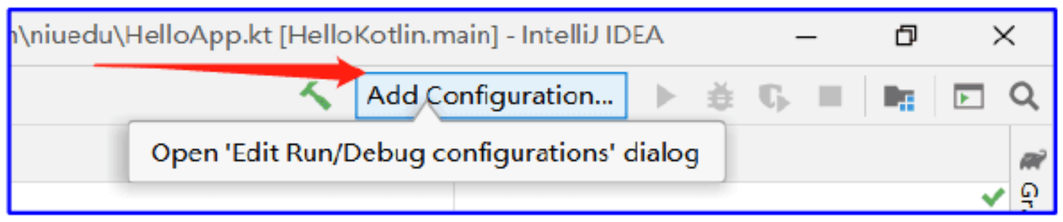

图 1-15

图 1-16 是运行方式的配置页面。

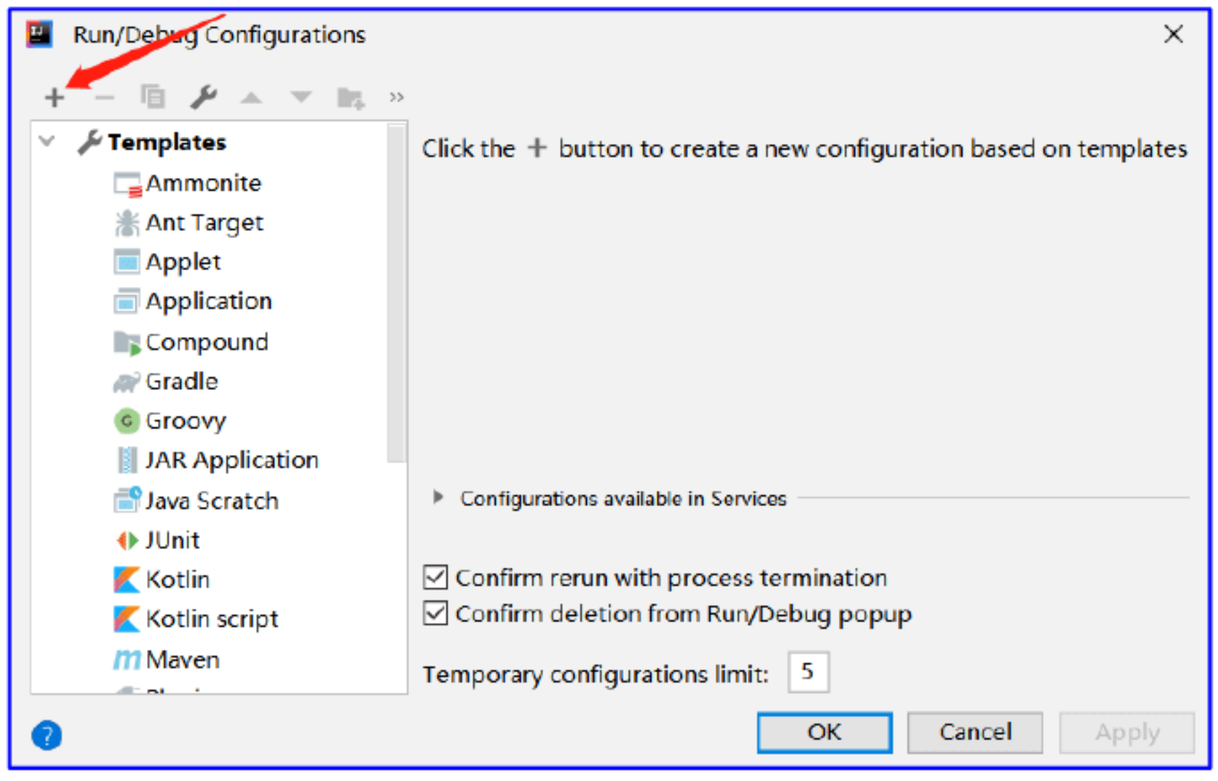

图 1-16

单击左上角的“+”图标，添加一条运行方式。选择正确的方式，这里应选“Kotlin”，如图 1-17 所示。

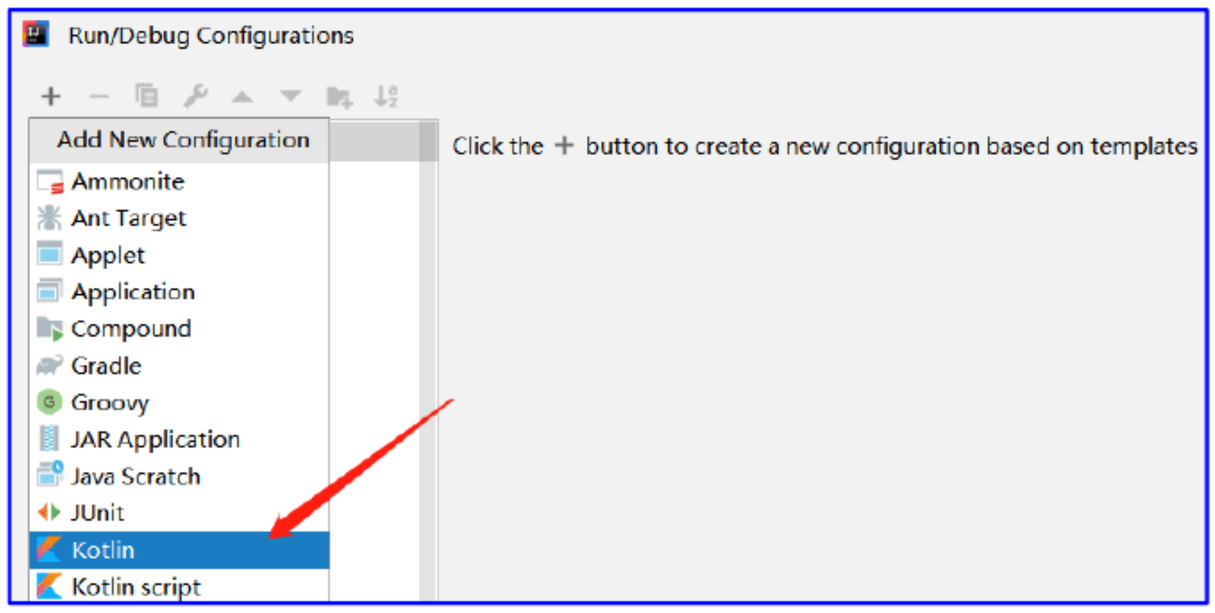

图 1-17

选择后则会出现如图 1-18 所示的界面。

将运行方式取名为“app”。在“Main class”字段填入的类名对应着 HelloApp.kt 文件，虽然 main 函数是全局函数，但是因为 Kotlin 代码最终要转成 class，所以它必然会有一个主类，从这里的类名可以看出 Kotlin 文件是如何与 Java 类对应的。

最后要注意的是，“Use classpath of module”字段需选择“HelloKotlin.main”，否则找不到 HelloAppKt 类。完成后单击“OK”按钮，回到主页面，此时可以看到运行方式旁边的图标变成绿色了，如图 1-19 所示。