

零基础 学Kotlin

之Android项目开发实战

郭宝利 编著

深入浅出介绍Kotlin编程入门与实战
以大量实例，助你快速上手，领略Kotlin的魅力

零基础 学Kotlin

之Android项目开发实战

郭宝利 编著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书既不是全面介绍 Kotlin 知识点的图书，也不是全面介绍 Android 开发的图书，而是一本指导读者使用 Kotlin 进行 Android 应用开发的图书，也是一本围绕开发实践讲解所需要的知识体系架构的图书。本书分为 6 章，第 1 章～第 4 章为 Kotlin 知识点，第 5 章和第 6 章是 Android 开发内容，提供了丰富的 Kotlin 代码示例，是第 1 章～第 4 章知识点的运用，它们之间的关系非常紧密。没有任何 Java 基础和 Android 基础的读者，也可以看懂本书。

本书可以作为软件开发人员的学习资料，也可以作为计算机软件专业的教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

零基础学 Kotlin 之 Android 项目开发实战 / 郭宝利编著. —北京：电子工业出版社，2019.4

ISBN 978-7-121-35519-6

I. ①零… II. ①郭… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 252503 号

策划编辑：高洪霞

责任编辑：牛 勇

印 刷：天津画中画印刷有限公司

装 订：天津画中画印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：20.5 字数：551 千字

版 次：2019 年 4 月第 1 版

印 次：2019 年 4 月第 1 次印刷

定 价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

前言

Kotlin 是在 Java 废墟上崛起的！Kotlin 不仅解决了 Java 中存在的很多问题，还引入了很多新的特点，并且完全兼容 Java，因此对 Java 程序员来说具有很大的诱惑力。对于想学习 Java 但还未入门的程序员来说，Kotlin 是一门非常好的语言。截至 2018 年 3 月，Kotlin 在 TIOBE 上的排名已经跃居第 38 位，这种迅猛上升的势头可以说明很多问题。

Java 是使用范围较广的一门开发语言，虽然它有很多优点，但是仍有不少让程序员抓狂的问题。随着这么多年的发展，Java 的各种框架、类库让它越来越庞大，问题的彻底解决变得几乎不太可能。

Kotlin 是一门基于 JVM 的新的编程语言，由著名的 JetBrains 公司开发。简洁、安全、与 Java 互操作、天生带有 IDE 支持，这是官方给出的 Kotlin 的四大特点。正是这四大特点，深深吸引了程序员，快速得到程序员的青睐。

Kotlin 可以开发服务器端程序、网页、桌面应用程序和 Android 应用，本书只介绍 Kotlin 的 Android 应用开发，因为 Android 应用开发有巨大的市场，需要更多软件开发人员在此领域有所作为。

因为 Kotlin 是基于 JVM 的（至少目前是这样），并且完全兼容 Java，所以笔者写作时稍不注意就会让读者在阅读时遇到生疏的 Java 知识点，尤其对于不熟悉 Java 的读者更是如此。笔者在写作时，做了严苛的限制，不出现分毫对 Java 知识点的依赖。这种限制，对于不熟悉 Java 的读者来说，阅读起来会更加顺畅，不存在任何因为生疏的 Java 知识点引起的阅读障碍；而对于熟悉 Java 的读者来说，理解起来会更加透彻，学习速度也会更快。

进行 Android 开发同样需要很多知识储备，它不同于常见的 Windows 操作系统。同样，笔者在这部分内容的写作上也做了严苛的限制，不会出现分毫对 Android 知识点的依赖。

上面提到的两处写作限制，对没有 Java、Android 开发经验的读者来说，阅读起来确实会顺畅很多，但是对于有经验的读者来说岂不是稀释了书中的有效知识点？正如：对于同样的武功，高手和初学者都可以练习，但他们会因自身基础的不同而有不同的感悟。有基础的读者，在阅读第 5 章和第 6 章时，应该将注意力集中在对 Kotlin 知识的运用上，以快速提高编程能力。为了照顾有基础的读者，书中提供了大量示例代码，越是有基础越应该自己动手编写示例。

本书的写作目的是让读者快速掌握使用 Kotlin 进行 Android 应用开发的能力。所以，本书既不是全面介绍 Kotlin 知识点的图书，也不是全面介绍 Android 开发的图书，而是一本指导读

者使用 Kotlin 进行 Android 应用开发的图书，也是一本围绕开发实践讲解所需要的知识体系架构的图书，这是程序员学习编程语言的最终目标。笔者在写作时，时刻围绕这个目标，心无旁骛地给读者做介绍，虽然有很多精彩的 Kotlin 知识点时时吸引笔者去写，但笔者最终还是忍住了。虽然书中某些知识点的介绍貌似有些偏离这个目标，如类的设计、PD 软件等，但是这些是软件开发实践中重要的知识点。一个只会写代码而不了解设计的程序员，就像一个只顾低头走路而不看方向的人。

本书的结构简洁：第 1 章和第 2 章属于了解性内容，介绍了 Kotlin 的特点，为使用 Kotlin 进行 Android 开发做准备；第 3 章和第 4 章介绍 Kotlin 相关的知识，包括 Kotlin 的基本语法、接口、类和对象；第 5 章和第 6 章是 Android 开发的重点，介绍 Android 开发的主要知识体系架构，以及数据操作的各种方式，这两章也是前面介绍的 Kotlin 知识点的具体应用。

虽然笔者编著了多部计算机开发类图书并已出版，但每次编著图书都感觉是全新的开始，前面的写作经验不敢过多依赖，唯一能够依赖的是笔者多年的计算机软件开发经验和学习开发语言时的思路，相信这些对读者的阅读和学习会有有一定的益处。笔者对内容的表述不敢有丝毫懈怠，唯恐让读者误解，加上笔者水平有限，总怀惴惴之心，唯恐误人子弟。好在，代码的执行是严格的，错误的代码无论如何也不会有运行结果。所以，推敲内容的文字表述，严格调试每个示例代码，介绍开发经验和学习思路，这是笔者在写作过程中始终努力去做到的。

但愿本书能够让读者的阅读有轻松的感觉，能够让读者真正掌握一门语言去做开发，笔者已经在写作本书的过程中尽最大努力了，当然还需要读者在学习过程中努力，才能真正达到目标。学习可以充实自己，努力可以提升自己，低头做事，抬头看路，坚持正确的方向总会有收获，这本书就是笔者的收获，希望每位读者也能有自己的收获。

郭宝利 于北京

2018.11.9

轻松注册成为博文视点社区用户 (www.broadview.com.cn)，扫码直达本书页面。

- **下载资源：**本书如提供示例代码及资源文件，均可在 [下载资源](#) 处下载。
- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/35519>



目录

第 1 章 了解 Kotlin	1
1.1 Kotlin 是什么	1
1.2 Kotlin 的特点	2
1.2.1 简洁	2
1.2.2 安全	2
1.2.3 与 Java 互操作	3
1.2.4 自带 IDE 支持	3
1.3 用 Kotlin 能开发什么	3
1.3.1 服务器端程序	4
1.3.2 网页应用	4
1.3.3 本地执行程序	5
1.3.4 Android 应用	5
1.4 为什么选择 Kotlin	5
1.4.1 Java 程序员的选择	5
1.4.2 新手的选择	6
1.5 发展趋势展望	7
第 2 章 准备工作	8
2.1 开发工具介绍	8
2.1.1 IntelliJ IDEA	8
2.1.2 Android Studio	21
2.1.3 Eclipse	28
2.2 构建工具	30
2.2.1 Gradle	30
2.2.2 Maven	34
2.3 编程规范	39
2.3.1 软件 UI 规范和人机交互规范	39
2.3.2 编码风格	44
2.4 本书阅读指导	53

第 3 章 基本语法55

3.1 基本类型55

3.1.1 数值55

3.1.2 字符58

3.1.3 布尔58

3.1.4 数组58

3.1.5 字符串62

3.2 集合65

3.2.1 Set65

3.2.2 List66

3.2.3 Map67

3.2.4 集合常用函数68

3.2.5 扩展阅读75

3.3 变量75

3.3.1 变量声明75

3.3.2 Any 类型和 null76

3.3.3 智能转换和 is 运算符77

3.3.4 可空变量和不可空变量77

3.4 运算符78

3.4.1 赋值运算符78

3.4.2 算术运算符79

3.4.3 关系运算符79

3.4.4 区间运算符80

3.4.5 逻辑运算符81

3.4.6 位运算符81

3.4.7 其他运算符81

3.4.8 优先级84

3.4.9 运算符重载84

3.5 表达式84

3.5.1 if 表达式84

3.5.2 when 表达式86

3.5.3 throw、try...catch 表达式87

3.5.4 正则表达式88

3.5.5 return 表达式91

3.6 语句91

3.6.1 异常处理91

3.6.2 判断语句93

3.6.3 循环语句95

3.7 自定义函数98

3.7.1 函数声明98

3.7.2 函数调用101

3.7.3	函数的范围	102
3.7.4	函数分类	105
3.8	程序组织及可见性修饰符	112
3.8.1	包	112
3.8.2	可见性修饰符	114
第4章	接口、类和对象	116
4.1	综述	116
4.1.1	定义接口	117
4.1.2	定义类	117
4.1.3	关于对象	118
4.1.4	编程示例	119
4.2	类	119
4.2.1	类的声明	119
4.2.2	构造函数	120
4.2.3	类的继承	123
4.2.4	类的实例化	126
4.3	类的成员	126
4.3.1	属性	126
4.3.2	函数	131
4.3.3	嵌套类	132
4.3.4	内部类	132
4.3.5	匿名内部类	133
4.3.6	初始化模块	133
4.3.7	对象声明	134
4.4	特殊的类	137
4.4.1	抽象类	137
4.4.2	枚举类	138
4.4.3	密封类	140
4.4.4	数据类	142
4.5	类的扩展	145
4.5.1	扩展函数	145
4.5.2	扩展属性	148
4.5.3	扩展伴生对象	148
4.6	类委托	148
4.7	对象	149
4.7.1	对象声明	150
4.7.2	匿名对象	151
4.8	类的设计	153
4.8.1	设计原则	153
4.8.2	常用图标	158

4.8.3	PowerDesigner 设计类图	160
4.9	标准类库	162
第 5 章	Kotlin 开发 Android 应用	164
5.1	再次开发 HelloWorld 程序	164
5.1.1	配置 Android 开发环境	164
5.1.2	开发 HelloWorld	165
5.1.3	Android 开发步骤	169
5.2	了解 Android 开发	172
5.2.1	Android 系统的技术架构	172
5.2.2	Android 应用程序的构成	174
5.3	Activity 和 Intent	177
5.3.1	Toast 提示	177
5.3.2	打开多个 Activity	179
5.3.3	关闭 Activity	186
5.3.4	Activity 生命周期	189
5.4	关于控件	191
5.4.1	概述	191
5.4.2	关于布局	194
5.4.3	控件大全	194
5.5	布局	198
5.5.1	线性布局	198
5.5.2	相对布局	206
5.5.3	帧布局	208
5.5.4	表格布局	208
5.5.5	网格布局	210
5.6	常用控件	212
5.6.1	TextView	212
5.6.2	EditText	220
5.6.3	Button	229
5.6.4	ImageView	230
5.6.5	选择用途的控件	235
5.7	关于事件	243
5.7.1	事件机制	243
5.7.2	监听事件机制	243
5.7.3	回调事件机制	247
5.8	对话框	252
5.8.1	简单对话框	252
5.8.2	单选对话框	253
5.8.3	多选对话框	254
5.8.4	进度对话框	254

5.8.5	自定义对话框	257
5.9	菜单	259
5.9.1	选项菜单	260
5.9.2	上下文菜单	262
5.9.3	弹出菜单	265
第 6 章	数据操作	267
6.1	综述	267
6.1.1	SharedPreferences	267
6.1.2	文件	268
6.1.3	数据库和数据库框架 (SQLite 数据库)	268
6.1.4	ContentProvider	269
6.1.5	网络	269
6.2	SharedPreferences	269
6.2.1	获取 SharedPreferences 对象	270
6.2.2	读取信息	271
6.2.3	写入信息	271
6.2.4	其他知识点	273
6.3	文件	274
6.3.1	资源文件读取	274
6.3.2	内部数据文件的读写	277
6.3.3	外部数据文件的读写	281
6.4	SQLite 数据库	284
6.4.1	概述	284
6.4.2	使用 SQLiteOpenHelper 类	286
6.4.3	使用 SQLiteDatabase 类	291
6.5	ContentProvider	298
6.5.1	综述	299
6.5.2	开发 ContentProvider	303
6.5.3	开发 ConvertResolver	308
6.5.4	监听 ContentProvider 数据改变	310
6.6	网络	311
6.6.1	开发步骤	312
6.6.2	代码示例	314
后记		317

第1章

了解 Kotlin

很高兴有机会向读者介绍 Kotlin。Kotlin 是一门非常有特点的计算机编程语言、一门能够开发多种应用的编程语言、一门让程序员兴奋的编程语言、一门发展前景和趋势特别让人期待的编程语言。

1.1 Kotlin 是什么

按照百度百科中的解释，Kotlin 是一门基于 JVM 的新的编程语言，由 JetBrains 公司开发。Kotlin 可以编译成 Java 字节码，也可以编译成 JavaScript，方便在没有 JVM 的设备上运行。

Kotlin 是一种新兴的计算机编程语言，在 2017 年快速进入程序员的视野。下面是几个标志性的的重要事件。

- Google 公司选定 Kotlin 作为 Android 第一开发语言，这个待遇连 Google 公司自己的 Go 语言都没有。
- 2017 年 6 月，Kotlin 在 TIOBE 上的排名进入了前 50 名（第 43 名），呈现非常快速的发展势头。
- Flipboard 中国的 Android 项目确定正式将 Kotlin 作为项目开发语言。
- Pinterest 已经成功地将 Kotlin 引入它们的应用程序中，每个月有 1 亿 5 千万人使用。
- Basecamp 的 Android 应用程序是 100% 的 Kotlin 代码。
- Keepsafe 的 App Lock 应用程序已转换为 100% 的 Kotlin 代码，使源代码行数减少了 30%、方法数减少了 10%。

JetBrains 公司是捷克的一家软件开发公司，成立于 2000 年。该公司大名鼎鼎的 Java IDE 产品 IntelliJ 是最优秀的 Java IDE 之一，广受程序员欢迎。下面是 Kotlin 的发展历程。

- 2010 年立项。
- 2011 年 6 月对外公开。
- 2012 年 2 月开源。

- 2013 年 8 月支持 Android Studio。
- 2014 年 6 月全新的开源 Web 站点和域名 KotlinLang.org 上线。
- 2016 年 2 月发布 1.0 正式版。
- 2016 年 9 月发布 1.0.4 版本。

1.2 Kotlin 的特点

简洁、安全、与 Java 互操作、自带 IDE 支持，这是官方给出的 Kotlin 的四大特点。正是因为这四大特点，Kotlin 深深吸引了程序员，快速获得程序员的青睐。

1.2.1 简洁

简洁，意味着代码书写工作量少，代码具有更好的可读性，具有更强的表达能力，更符合程序员的思维习惯。

官方给出了示例代码，用来展示 Kotlin 代码的简洁。

- 仅用一行代码就可以创建 POJO，具备了 getters、setters、equals()、hashCode()、toString() 和 copy() 功能，代码如下：

```
data class Customer(val name: String, val email: String, val company: String)
```

POJO 是 Plain Ordinary Java Object 的缩写，指没有使用 Entity Beans 的普通 Java 对象，可以理解为简单的实体类。POJO 类的作用是方便程序员使用数据库中的数据表，程序员可以很方便地将 POJO 类当作对象来使用，当然也可以方便地调用其 get 和 set 方法。

- 使用一行代码，就可以过滤 List 中大于 0 的项目，代码如下：

```
val positiveNumbers = list.filter { it > 0 }
```

上面的代码中使用了 Lambda 表达式。

- 使用单例模式，创建一个对象即可：

```
object ThisIsASingleton {  
    val companyName: String = "JetBrains"  
}
```

简洁，确实非常简洁。

1.2.2 安全

使用 Kotlin 可以摆脱让人讨厌的空指针异常（NullPointerExceptions）。在 Kotlin 中，用以下代码赋空值给变量，在编译时无法通过。

```
var output: String  
output = null // Compilation error
```

以下代码在编译时自动进行空值检查，也是无法通过的。

```
val name: String? = null // Nullable type  
println(name.length()) // Compilation error
```

1.2.3 与 Java 互操作

可以使用运行在 JVM 上的任何现有库，Kotlin 对 Java 100%兼容，包括对 SAM 的支持（Simple API for XML）。这个特性不仅对 Java 程序员有吸引力，对没有 Java 经验的程序员也是一个好消息，毕竟 Java 语言发展了很多年，积累了强大的资源生态。因为 100%兼容，意味着 Kotlin 可以站在巨人的肩膀上发展，起步就是高的，有大量可用的资源能够借用。

```
import io.reactivex.Flowable
import io.reactivex.schedulers.Schedulers
```

```
Flowable
    .fromCallable {
        Thread.sleep(1000) // imitate expensive computation
        "Done"
    }
    .subscribeOn(Schedulers.io())
    .observeOn(Schedulers.single())
    .subscribe(::println, Throwable::printStackTrace)
```

1.2.4 自带 IDE 支持

有好的 IDE 支持，会极大地提高代码书写的效率。如图 1-1 所示是 IDE 对 Kotlin 代码自动提醒的支持。

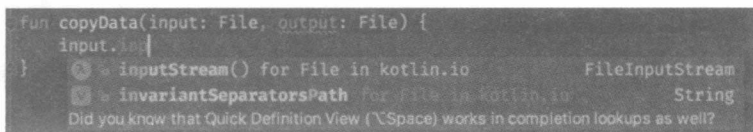


图 1-1 代码提醒

代码提醒功能对于对 Kotlin 代码不熟悉的程序员来说非常有利，能够自动提醒，也非常有利于程序员提高代码书写效率，在提醒清单中选择需要的代码，按 Enter 键即可完成录入。

如图 1-2 所示是代码智能检查功能。

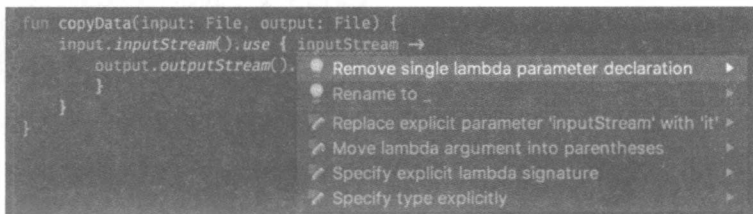


图 1-2 代码智能检查

1.3 用 Kotlin 能开发什么

移动端的应用开发是当前 IT 开发的主流，但是 PC 端的开发仍然有很大的市场，这是一个 IT 多彩的时代。能够开发哪种应用，是影响用户是否选择这门开发语言的一个关键要素。令人

欣慰的是，常规的这些应用开发 Kotlin 都支持。

1.3.1 服务器端程序

服务器端应用程序是指运行在服务器上，根据前端发送的请求来执行，或者调用数据库数据，或者调用其他服务，最终将结果反馈给请求者。根据应用目的的不同，可以简单分成如下三大类。

- 服务类应用程序，包括基于 SOAP 协议的 WebService 和基于 HTTP 协议的 Restful。
- 数据库类应用程序，包括各种业务逻辑开发、实体类的开发。
- 连接类应用程序，包括负载处理、传输处理等。

以上应用程序都可以使用 Kotlin 进行开发。下面是用于服务器端开发的框架，读者在这里稍作了解，以后可根据工作需要重点关注。

- Ktor 是由 JetBrains 公司构建的 Kotlin 原生 Web 框架，具有高可伸缩性，并提供易于使用、符合习惯用法的 API。
- Spring 利用 Kotlin 的语言功能，提供更简洁的 API，从 5.0 版本开始，在线项目生成器允许用 Kotlin 快速生成一个新项目。
- Vert.x 是在 JVM 上构建响应式 Web 应用程序的框架，为 Kotlin 提供了专门支持，包括完整的文档。
- kotlinox.html 是在 Web 应用程序中构建 HTML 的 DSL，可以作为传统模板系统（如 JSP 和 FreeMarker）的替代品。
- 持久化的可用选项包括直接 JDBC 访问、JPA 和通过 Java 驱动程序使用 NoSQL 数据库。对于 JPA，kotlin-jpa 编译器插件使 Kotlin 编译的类符合框架的要求。

1.3.2 网页应用

网页应用是指运行在浏览器环境中，靠浏览器解释执行的应用。Kotlin 可以编译成 JavaScript 代码，并且是 100%兼容的。目前的实现目标是 ECMAScript 5.1，也有最终目标为 ECMAScript 2015 的计划。

当用户选择 JavaScript 目标时，作为项目一部分的任何 Kotlin 代码及 Kotlin 附带的标准库都会转换为 JavaScript。但是，这不包括使用的 JDK 和任何 JVM、Java 框架或库。任何不是 Kotlin 的文件都会在编译期间被忽略掉，编译时会做如下转换。

- 创建面向客户端 JavaScript 的 Kotlin 代码。
 - 与 DOM 元素交互。Kotlin 提供了一系列静态类型的接口与文档对象模型（Document Object Model，DOM）交互，允许创建和更新 DOM 元素。
 - 与图形（如 WebGL）交互。用户可以使用 Kotlin 在网页上用 WebGL 创建图形元素。
- 创建面向服务器端 JavaScript 的 Kotlin 代码。
 - 使用服务器端技术。用户可以使用 Kotlin 与服务器端 JavaScript（如 node.js）进行交互。

1.3.3 本地执行程序

这是早期的、最传统的应用程序，它可以是 C/S 模式的程序，也可以是单机程序。这类应用依赖操作系统的环境，需要 JVM 的支持。

1.3.4 Android 应用

Android 应用是 Kotlin 开发的重点，也是目前 Kotlin 最成熟的应用。Kotlin 与 JDK 6 完全兼容，保证 Kotlin 应用程序可以在较旧的 Android 设备上运行而无任何问题。Kotlin 工具在 Android Studio 中完全支持，并且兼容 Android 构建系统。允许在 Kotlin 应用程序中使用所有现有的 Android 库，包括注解处理。Kotlin 团队为 Android 开发提供了一套超越标准语言功能的工具。

- Kotlin Android 扩展是一个编译器扩展，可以让用户摆脱代码中的 `findViewById()` 调用，并将其替换为合成的编译器生成的属性。
- Anko 是一个围绕 Android API 的、Kotlin 友好的提供包装器的库，以及一个可以用 Kotlin 代码替换布局.xml 文件的 DSL。

1.4 为什么选择 Kotlin

正在从事 Java 编程工作的程序员深知 Java 存在诸多问题，Kotlin 除解决这些问题外，还提供了 Java 不具备的一些特性，并且 100% 兼容 Java，支持无成本迁移，这为 Java 程序员提供了一个非常好的选择。

1.4.1 Java 程序员的选择

下面先来看看 Java 存在的问题。

- 空引用 (Null references): 连空引用的发明者都承认这是一个“十亿美元 (billion-dollar) 的错误”。不论怎样都无法避免它，因为 Java 的类型系统就是不安全的。
- 原始类型 (Raw types): 开发应用程序时总会为了保持兼容性而卡在泛型原始类型的问题上，我们都知道要努力避免 Raw types 的警告，但是它们毕竟是在语言层面上存在的，这必定会造成误解并形成不安全因素。
- 协变数组 (Covariant arrays): 可以创建一个 string 类型的数组和一个 object 类型的数组，然后把 string 数组分配给 object 数组。这样的代码虽然可以通过编译，但是一旦尝试在运行时分配一个数给 object 数组，就会在运行时抛出异常。
- Java 8 存在高阶方法 (Higher-order functions)，但是它们是通过 SAM 类型实现的。SAM 是一个单个抽象方法，每个函数类型都需要一个对应的接口。当想要创建一个并不存在的 Lambda 表达式或不存在对应关系的函数类型时，就要自己创建函数类型作为接口。
- 泛型中的通配符: 泛型总是难以操作、阅读、书写和理解。对编译器而言，异常检查也变得很困难。

对于上面列出的这些缺陷，Kotlin 通常直接移除了那些特性，同时添加了一些新的特性。

- Lambda 表达式。

- 数据类 (Data classes)。
- 函数字面量和内联函数 (Function literals & inline functions)。
- 函数扩展 (Extension functions)。
- 空安全 (Null safety)。
- 智能转换 (Smart casts)。
- 字符串模板 (String templates)。
- 主构造函数 (Primary constructors)。
- 类委托 (Class delegation)。
- 类型推断 (Type inference)。
- 单例 (Singletons)。
- 声明点变量 (Declaration-site variance)。
- 区间表达式 (Range expressions)。

Kotlin 之所以能跟随 JVM 的生态系统不断地进步，是因为它没有任何限制。Kotlin 编译出来的正是 JVM 字节码。在 JVM 看来，Kotlin 跟其他语言是一样的。事实上，在 IntelliJ 或 Android Studio 上用 Kotlin 的插件时，会自带一个字节码查看器，可以显示每个方法生成的 JVM 字节码。

Kotlin 来自业界而不是学术界，它解决了开发者现今面临的很多实际问题，上面列出的只是比较重要的。从 Java 切换到 Kotlin 无须成本，Kotlin 提供了一个高质量的从 Java 转换到 Kotlin 的工具，可以一键将现有 Java 项目一次性转换成 Kotlin 项目，而该项目仍可以正常编译，即使是一个包含上百万行代码的复杂项目。

Kotlin 程序能够使用所有现存的 Java 框架和库，甚至包括那些依赖注解处理的高级框架。它们之间的交互是无缝的，不需要包装或适配层。Kotlin 可以整合 Maven、Gradle 及其他构建系统。

1.4.2 新手的选择

对于新手程序员来说，他们所有的选择都会围绕从业收入及自己的能力来进行。能够带来理想的收入，需要掌握的知识是在自己能力范围内的，这种语言就是新手程序员最容易选择的。

目前，从事移动开发工作是一个不错的选择，多数人使用 Java 语言从事 Android 开发。但是，Kotlin 替代 Java 只是时间问题。不需要太多的原因，解决了 Java 现存的问题、100%兼容 Java，即使没有其他新的特性，仅这两个原因就足够了，况且它确实还引入了很多新特性。除此之外，Google 公司选择 Kotlin 作为 Android 第一开发语言，后面会持续地投入和支持，这为 Kotlin 语言的长远发展奠定了基础。

前面已经介绍了 Kotlin 的很多特点，下面总结一下影响程序员选择 Kotlin 的几点原因。

- Kotlin 十分平易近人，语法精炼直观。
- Kotlin 不苛求特定的编程哲学，例如极度的函数式编程或面向对象编程风格。
- Kotlin 允许用户继续使用工作效率提升工具。IntelliJ 的 IDE 对 Kotlin 的支持十分完善：可以对代码进行重构、搜索、导航，以及使用自动完成功能，而且 IDE 充分支持调试、单元测试和性能分析等。

- 由知名公司为 Kotlin 提供强大的商业支持。JetBrains 公司有一个高度称职的大团队负责该项目，有稳定的商业模式，甚至在其部分旗舰产品中使用 Kotlin，这表明短期内 Kotlin 不会被放弃。尽管 Kotlin 是免费的，但是 JetBrains 公司还是能够通过它增加商业版本 IDE 的销售来赚大钱，这将激励他们根据用户的意愿持续改进它。与此相比，对于那些由不相关产品资助的语言开发者，当用户需求与之前的设计理念冲突时，他们很少会因此做出调整。
- Kotlin 语言十分注重语法的可读性，代码审查不会成为问题，对 Kotlin 不熟悉的团队成员仍然能够完成该工作。

1.5 发展趋势展望

对趋势的预测源于已有的经验和对当前事物的认知。笔者作为一名在 IT 行业从业 20 多年的老程序员，经历过多种语言的兴衰。再优秀的语言，也有崛起、兴盛、衰亡的过程，作为从 Java 废墟中崛起的 Kotlin，它也摆脱不了 Java 的发展轨迹。好在，我们处在一个 Kotlin 崛起的时代，并且预期 Kotlin 有更长久的兴盛期，因为它有更加优秀的特性。

- Google 公司将 Kotlin 确定为 Android 第一开发语言。作为全球性的大型 IT 公司，Google 公司的认可会为 Kotlin 赢得更多支持，吸引更多的公司、程序员参与到 Kotlin 的开发中来。
- Kotlin 完全继承了 Java 的生态。这意味着，Kotlin 会完全继承 Java 的语言生态，不用从零开始建立自己的语言生态。语言生态的建立是需要时间积累的。语言生态包括框架、学习资料、参与的程序员、使用该语言的项目、成熟的解决方案、第三方相关产品、大公司的参与等。一个好的语言生态的建立，绝对不是一朝一夕的事情，即使该语言足够优秀。
- JetBrains 公司把 Kotlin 当作重点项目。尽管 Kotlin 是免费的，但是 JetBrains 公司还是能够通过它增加商业版本 IDE 的销售来赚大钱，这将激励他们根据用户的意愿持续改进它。与此相比，对于那些由不相关产品资助的语言开发者，当用户需求与之前的设计理念冲突时，他们很少会因此做出调整。

基于以上三点，笔者认为，Kotlin 会快速形成良好的语言生态，继而发展自己独有的语言生态，快速进入繁荣期，并且会比市面上常见的语言拥有更长的繁荣期。