

“互联网 + 教育”新形态一体化系列教材

Android 应用开发案例与实战

主 编 马鸿雁 迟永芳 吴艳红
副主编 杨 昆 刘文林 王海军 陈 孚



合肥工业大学出版社
HEFEI UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

Android应用开发案例与实战 / 马鸿雁, 迟永芳, 吴艳红主编. —合肥: 合肥工业大学出版社, 2023.5

ISBN 978-7-5650-6268-1

I. ①A… II. ①马… ②迟… ③吴… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国国家版本馆CIP数据核字(2023)第028731号

Android应用开发案例与实战

ANDROID YINGYONG KAIFA ANLI YU SHIZHAN

马鸿雁 迟永芳 吴艳红 主编

责任编辑 孙南洋
出版发行 合肥工业大学出版社
地 址 (230009)合肥市屯溪路193号
网 址 www.hfutpress.com.cn
电 话 人文社科出版中心: 0551-62903200
营销与储运管理中心: 0551-62903198
规 格 787毫米×1092毫米 1/16
印 张 18
字 数 460千字
版 次 2023年5月第1版
印 次 2023年5月第1次印刷
印 刷 廊坊市广阳区九洲印刷厂
书 号 ISBN 978-7-5650-6268-1
定 价 58.00元

如果有影响阅读的印装质量问题, 请与出版社营销与储运管理中心联系调换



随着手机设备的飞速发展和日益普及，手机的操作系统也备受人们的关注。在众多的操作系统中，Android以漂亮的UI、广泛的连接性、良好的存储、多点触控、多任务运行、多媒体支持等特点脱颖而出，成为最受欢迎的系统之一。

本书以Android为基础，紧紧围绕最新的Android技术精髓展开深入讲解，以清晰的思路、精炼的实例和任务使读者快速入门，并逐步掌握Android编程的知识。本书注重基础理论与实用开发相结合，突出应用编程思想与开发方法的介绍，所选实例和任务都具有较强的概括性和实际应用价值。

本书是作者根据从事多年的Android开发工作和讲授计算机专业相关课程的教学实践，在已编多部讲义和教材的基础上编写而成的；内容充实，循序渐进，选材上注重系统性、先进性和实用性；编写时注重实践性，精选的所有例题已在Android Studio上调试通过，可直接引用，读者也可按照书中提示步骤自己动手完成。本书分为12章。

第1章为Android开发概述，主要介绍了Android的开发环境搭建、如何在搭建的环境中创建Android项目以及如何调试程序等内容。

第2章为开发工具介绍，主要介绍了如何使用DDMS和Android调试桥等内容。

第3章为Android UI编程，主要介绍了常见的Widget组件以及列表、菜单和布局等内容。

第4章为Android活动简介，主要介绍了Activity的创建与注册、生命周期以及如何使用Intent等内容。

第5章为Android多媒体，主要介绍了音频处理、使用系统相机和视频处理等内容。

第6章为Android传感器，主要介绍了如何使用传感器、传感器坐标系统和常见的8个传感器等内容。

第7章为Android服务简介，主要介绍了Service的创建和注册、启动和停止以及生命周期等内容。

第8章为Android广播简介，主要介绍了如何发送和接收广播等内容。

第9章为Android的数据持久化，主要介绍了使用SharedPreferences、使用文件存储、使用



SQLite 数据库、使用 ContentProvider 共享数据等内容。

第 10 章为 Android 网络编程，主要介绍了使用 HttpURLConnection 和使用 WebView 等内容。

第 11 章为 Android 管理器与地图服务，主要介绍了 2 个管理器和地图服务等内容。

第 12 章为足迹生成器，主要介绍了如何创建一个足迹生成器的大项目。

由于编者水平有限，加之编写时间仓促，书中难免存在错误和疏漏之处，希望广大读者批评指正。

编 者

2023 年 2 月



第1章 Android开发概述1

知识入门 2

循序渐进 4

1.1 搭建Android开发环境 4

1.1.1 下载安装JDK 4

1.1.2 配置环境变量 5

1.1.3 下载安装Android Studio 6

1.2 创建第一个项目 7

1.2.1 首次启动Android Studio 7

1.2.2 创建项目 10

1.2.3 了解项目 11

1.2.4 模拟器上运行程序 14

1.2.5 真机上运行程序 17

1.3 调试程序 18

1.3.1 增加断点 18

1.3.2 开始调试 18

1.3.3 单步调试 19

知识拓展 20

本章习题 22

第2章 开发工具介绍 25

知识入门 26

循序渐进 27

2.1 使用DDMS 27

2.1.1 管理进程 27

2.1.2 使用文件浏览器 29

2.1.3 使用模拟器控制 30

2.1.4 使用日志 32

2.1.5 使用Screen Capture捕捉设备
屏幕 33

2.2 使用Android调试桥 33

2.2.1 使用ADB 33

2.2.2 显示连接到计算机的设备 34

2.2.3 针对特定设备操作 34

2.2.4 启动和停止adb 34

2.2.5 使用adb操作文件和apk 35

2.2.6 使用adb shell 37

知识拓展 38

本章习题 40

第3章 Android UI编程 41

知识入门 42

循序渐进 42

3.1 常见的Widget组件 42

3.1.1 使用可滚动的文本视图——
TextView 43

3.1.2 使用文本框——EditText 44

3.1.3 使用可滚动的视图——
ScrollView 46

3.1.4 使用按钮——Button 48

3.1.5 使用图片按钮——
ImageButton 49

3.1.6 使用复选框——CheckBox 50

3.1.7 使用单选框——RadioGroup
..... 52



- 3.1.8 使用下拉列表框——Spinner 55
- 3.1.9 使用进度条——ProgressBar 58
- 3.1.10 使用拖动条——SeekBar 59
- 3.1.11 使用图片视图——
ImageView 59
- 3.1.12 使用网格视图——GridView 60
- 3.1.13 使用消息提醒——Toast 64
- 3.2 使用列表——ListView 68
- 3.3 使用菜单——Menu 72
- 3.4 布局 75
 - 3.4.1 创建布局 75
 - 3.4.2 使用布局类 77
- 知识拓展 87
- 本章习题 87

第 4 章 Android 活动简介 89

- 知识入门 90
- 循序渐进 90
- 4.1 Activity 的创建与注册 90
 - 4.1.1 创建 Activity 90
 - 4.1.2 注册 Activity 92
- 4.2 Activity 的生命周期 93
- 4.3 使用 Intent 95
- 知识拓展 101
- 本章习题 102

第 5 章 Android 多媒体 105

- 知识入门 106
- 循序渐进 109
- 5.1 音频处理 109

- 5.1.1 使用 MediaRecorder 录制
音频 109
- 5.1.2 使用 MediaPlayer 播放
音频 111
- 5.2 使用系统相机 115
- 5.3 视频处理 120
 - 5.3.1 录制视频 120
 - 5.3.2 播放视频 123
- 知识拓展 132
- 本章习题 132

第 6 章 Android 传感器 133

- 知识入门 134
- 循序渐进 135
- 6.1 使用传感器 135
- 6.2 传感器坐标系统 139
- 6.3 常见传感器介绍 139
 - 6.3.1 加速传感器 139
 - 6.3.2 重力传感器 139
 - 6.3.3 陀螺仪传感器 140
 - 6.3.4 线性加速传感器 140
 - 6.3.5 方向传感器 141
 - 6.3.6 磁场传感器 141
 - 6.3.7 距离传感器 141
 - 6.3.8 光线传感器 142
- 知识拓展 144
- 本章习题 148

第 7 章 Android 服务简介 149

- 知识入门 150
- 循序渐进 150
- 7.1 Service 的创建和注册 150
 - 7.1.1 创建 Service 150
 - 7.1.2 注册 Service 151



7.2 启动和停止 Service	152	9.3.1 创建和删除数据库	192
7.2.1 启动 Service	152	9.3.2 创建和删除表	193
7.2.2 停止 Service	153	9.3.3 操作记录	194
7.3 Service 的生命周期	156	9.3.4 查询记录	196
知识拓展	166	9.4 使用 ContentProvider 共享数据	202
本章习题	166	9.4.1 使用 ContentProvider	202
第 8 章 Android 广播简介	167	9.4.2 使用 ContentResolver	206
知识入门	168	知识拓展	213
循序渐进	168	本章习题	217
8.1 发送广播	168	第 10 章 Android 网络编程	219
8.1.1 发送普通广播	169	知识入门	220
8.1.2 发送有序广播	171	循序渐进	221
8.2 接收广播	171	10.1 使用 HttpURLConnection	221
8.2.1 创建广播接收器	172	10.1.1 使用 GET 方法	221
8.2.2 注册广播接收器	173	10.1.2 使用 POST 方法	222
8.2.3 新建 Intent 过滤器	173	10.2 使用 WebView	227
8.2.4 注销广播接收器	174	10.2.1 直接加载网页	227
知识拓展	177	10.2.2 加载 HTML 代码	229
本章习题	180	10.2.3 JavaScript 支持	230
第 9 章 Android 的数据持久化	181	知识拓展	237
知识入门	182	本章习题	238
循序渐进	183	第 11 章 Android 管理器与地图服务	239
9.1 使用 SharedPreferences	183	知识入门	240
9.1.1 使用 SharedPreferences 保存数据	184	循序渐进	242
9.1.2 使用 SharedPreferences 读取数据	185	11.1 管理器	242
9.2 使用文件存储	188	11.1.1 电话管理器	242
9.2.1 在程序默认位置创建和写入文件	188	11.1.2 短信管理器	242
9.2.2 在程序默认位置读取文件	189	11.2 地图服务	245
9.3 使用 SQLite 数据库	192	11.2.1 准备工作	245
		11.2.2 百度地图显示	250
		11.2.3 对地图进行设置	251
		11.2.4 定位	251



知识拓展 260

本章习题 261

第 12 章 足迹生成器 263

12.1 准备工作 264

12.2 界面 UI 实现 265

12.2.1 界面规划 265

12.2.2 主界面 265

12.2.3 新建足迹界面 266

12.2.4 显示足迹界面 268

12.2 数据库实现 269

12.2.1 设计表结构 269

12.2.2 实现 DatabaseHelper 270

12.3 功能实现 271

12.3.1 实现 MainActivity 271

12.3.2 实现 NewFootPrintActivity
..... 273

12.3.3 实现 ShowFootPrintActivity
..... 275

12.4 运行程序 277

参考文献 279



1. Android 版本体系

Android 是一种基于 Linux 内核和其他开源软件的修改版本的移动/桌面操作系统，主要为智能手机和平板电脑等触摸屏移动设备设计，于 2007 年 11 月亮相。第一款商用 Android 设备 HTC Dream 于 2008 年 9 月推出。目前最新版是 2022 年 5 月 12 日发布的 Android 13。表 1-1 中列出了 Android 的各种版本。

表 1-1 Android 各个版本

版本号	代号	发布时间
Android 1.1		2008 年 9 月
Android 1.5	Cupcake	2009 年 4 月
Android 1.6	Donut	2009 年 9 月
Android 2.0	Éclair	2009 年 10 月 26 日
Android 2.1		2009 年 10 月 26 日
Android 2.2	Froyo	2010 年 5 月 20 日
Android 2.3	Gingerbread	2010 年 12 月 7 日
Android 3.0	Honeycomb	2011 年 2 月 3 日
Android 3.2		2011 年 7 月 13 日
Android 4.0	IceCreamSandwich	2011 年 10 月 19 日
Android 4.1	Jelly Bean	2012 年 6 月 28 日
Android 4.2		2012 年 10 月 30 日
Android 4.3		2013 年 7 月 25 日
Android 4.4	KitKat	2013 年 11 月 1 日
Android 5.0	Lollipop	2014 年 11 月 13 日
Android 6.0	Marshmallow	2015 年 9 月 30 日
Android 7.0	Nougat	2016 年 8 月 22 日
Android 7.1		2016 年 12 月 5 日
Android 8.0	Oreo	2017 年 8 月 22 日
Android 9.0	Pie	2018 年 8 月 7 日
Android 10.0	AndroidQ (内部叫 Quince Tart)	2019 年 9 月 4 日



续表

版本号	代号	发布时间
Android 11	内部叫 Red Velvet Cake	2020 年 9 月 9 日
Android 12	Snow Cone	2021 年 10 月 5 日
Android 13	Tiramisn	2022 年 5 月 12 日

2. Android的特性

这里介绍 Android 几种最为重要的特性。

- (1)漂亮的 UI: Android 操作系统的基本屏幕提供了漂亮又直观的用户界面。
- (2)连接性: 可以连接 GSM/EDGE、IDEN、CDMA、EV-DO、UMTS、Bluetooth、Wi-Fi、LTE、NFC 和 WiMAX。
- (3)存储: 用于数据存储的轻量级关系型数据库 SQLite。
- (4)媒体支持: 支持的媒体包括 H.263、H.264、MPEG-4 SP、AMR、AMR-WB、AAC、HE-AAC、AAC 5.1、MP3、MIDI、Ogg Vorbis、WAV、JPEG、PNG、GIF 和 BMP。
- (5)消息: SMS 和 MMS。
- (6)Web 浏览器: 基于开源的 WebKit 布局引擎, 再加上支持 HTML5 和 CSS3 Chrome 的 V8 JavaScript 引擎。
- (7)多点触控: Android 原生支持多点触控, 从最初的手持设备开始便有。
- (8)多任务: 用户可以从一个任务跳到另一个任务, 并且可以同时运行各种应用。
- (9)GCM: 谷歌云消息 (GCM) 是一种服务, 可以使开发人员向 Android 设备的用户发送短消息数据, 而无须专有的同步解决方案。

3. API 级别

API 级别是一个用于唯一标识 API 框架版本的整数, 由某个版本的 Android 平台提供, 如 Android 5.1 的 API 级别为 22、Android 5.0 的 API 级别为 21 等。

4. Android Studio

Android Studio 是谷歌推出的一个 Android 集成开发工具, 用于开发和调试, 类似于 Eclipse ADT。在 JetBrains IntelliJ IDEA 的基础上, Android Studio 提供了以下构架。

- (1)基于 Gradle 的构建支持。
- (2)Android 专属的重构和快速修复。
- (3)提示工具以捕获性能、可用性、版本兼容性问题。
- (4)支持 ProGuard 和应用签名。
- (5)基于模板的向导来生成常用的 Android 应用设计和组件。
- (6)功能强大的布局编辑器, 可以让 UI 控件进行效果预览。

5. R.java 文件

R.java 文件自动生成, 用来定义 Android 程序中所有不同类型的资源的索引。它是只读的, 开发人员无法对其进行修改。



1.1

搭建 Android 开发环境

本节将讲解对 Android 开发环境的搭建，其中包含下载安装 JDK 和下载安装 Android Studio 这两部分内容。

1.1.1 下载安装 JDK

JDK 是 JAVA 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的 JAVA 应用程序。JDK 是整个 JAVA 开发的核心，它包含了 JAVA 的运行环境 (JVM+JAVA 系统类库) 和 JAVA 工具。本小节将介绍如何下载和安装 JDK。

1. 下载 JDK

以下是下载 JDK 的具体操作步骤。

(1) 在浏览器中打开 JDK 的下载网页 (<https://www.oracle.com/java/technologies/javase-downloads.html>)，如图 1-1 所示。此页面会展示最新版本的 JDK。

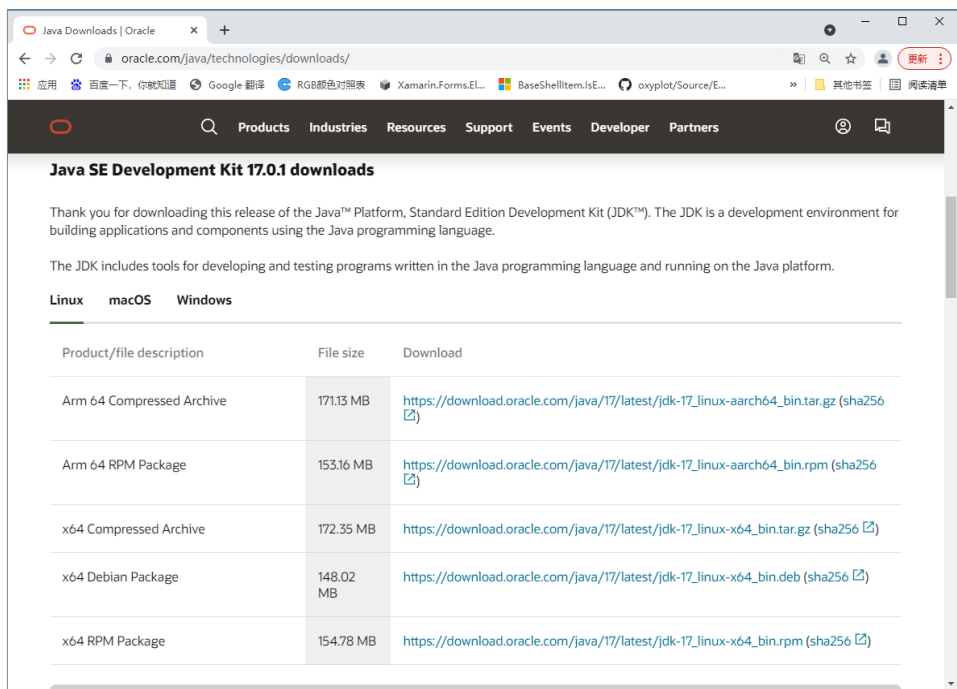


图 1-1 JDK 的下载网页



注意：在此网页中提供了3个平台的JDK版本，分别为Linux、macOS及Windows系统。开发者可以根据自己的系统进行选择下载。

(2) 点击对应版本后面的下载链接就可以实现下载。本书下载的是Windows平台最新版本

2. 安装JDK

下载JDK后，就可以进行安装了，以下是在Windows操作系统下安装jdk-*.exe的操作步骤。

(1) 双击jdk-*.exe文件，弹出“Java(TM) SE Development Kit *(64-bit)-安装程序”对话框。

(2) 点击“下一步”按钮，弹出“Java(TM) SE Development Kit *(64-bit)-目标文件夹”对话框。

(3) 点击“下一步”按钮，弹出“Java(TM) SE Development Kit *(64-bit)-进度”对话框。

(4) 一段时间后，会弹出“Java(TM) SE Development Kit *(64-bit)-完成”对话框。点击“关闭”按钮，此时JDK就安装完成了。

1.1.2 配置环境变量

环境变量相当于给系统或用户应用程序设置的一些参数，具体起什么作用与具体的环境变量相关。例如，path变量告诉系统，当要求系统运行一个程序而没有告诉它程序所在的完整路径时，系统除了在当前目录下面寻找此程序外，还应到哪些目录下去寻找。当JDK正确安装后，就可以对环境变量进行配置了。具体的操作步骤如下。

(1) 右击Windows 10左下角的“开始”按钮，在弹出的菜单中选择“系统”命令，弹出“设置”对话框。

(2) 在对话框右侧，点击“高级系统设置”按钮，弹出“系统”对话框。

(3) 点击“环境变量(N)...”按钮，弹出“环境变量”对话框。

(4) 点击“系统变量”标签下的“新建”按钮，弹出“新建系统变量”对话框。

(5) 将“变量名”设置为“JAVA_HOME”，“变量值”设置为“C:\Program Files\Java\jdk-***”（该路径是安装的JDK的文件位置），如图1-2所示。

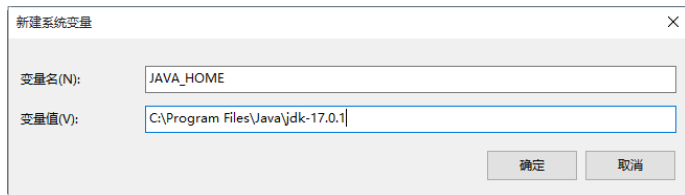


图 1-2 “新建系统变量”对话框创建 JAVA_HOME 变量

(6) 填写完毕后，点击“确定”按钮，完成JAVA_HOME变量创建，回到“环境变量”对话框。

(7) 再次点击“新建”按钮，新建CLASSPATH变量，如图1-3所示。“变量名”为“CLASSPATH”，“变量值”为“.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;”。

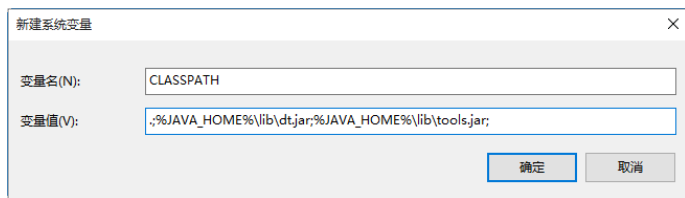


图 1-3 “新建系统变量”对话框创建 CLASSPATH 变量

(8) 在“环境变量”对话框中，选中 Path 变量后。点击“编辑”按钮，弹出“编辑环境变量”对话框。

(9) 点击“新建”按钮，在其中输入 JDK 的路径，如“C:\Program Files\Java\jdk-***\bin”。输入完毕后，点击“确定”按钮，回到“环境变量”对话框。此时环境变量就配置好了。

在完成了 JDK 的安装和环境变量的配置以后，需要进行验证。

(1) 右击 Windows 标志按钮，在弹出的菜单中，点击“运行”命令，弹出“运行”对话框。

(2) 在文本框中填入 cmd，点击“确定”按钮，弹出“命令提示符”窗口。

(3) 在该窗口中，执行命令“java -version”，得到如图 1-4 所示的结果，这表示正确安装了 JDK。其他的环境变量配置还需要在应用时进行确认。

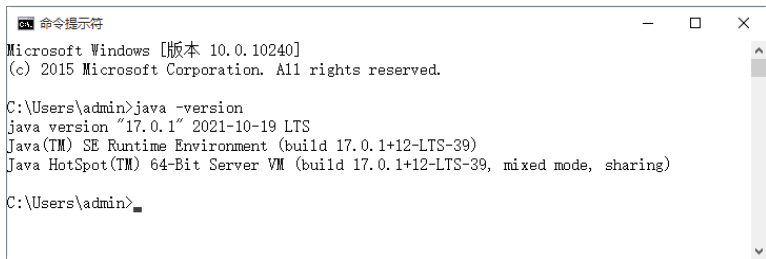


图 1-4 “命令提示符”窗口

1.1.3 下载安装 Android Studio

本小节将讲解如何下载和安装 Android Studio。

1. 下载 Android Studio

下载 Android Studio 工具的具体操作步骤如下。

(1) 在浏览器中输入“<https://developer.android.google.cn/studio/>”网址，打开 Android Studio 的官方网站。

(2) 点击“Download Android Studio”按钮，弹出“Download Android Studio”对话框。

(3) 选择“I have read and agree with the above terms and conditions”按钮，“Download Android Studio for Windows”按钮被激活。点击该按钮，Android Studio 就会被下载。

2. 安装 Android Studio

下载 Android Studio 安装包后，就可以对其进行安装了。具体的操作步骤如下。

(1) 双击 Android Studio 安装包，弹出“Welcome to Android Studio Setup”对话框。

(2) 点击“Next”按钮，弹出“Choose Components”对话框。开发者可以根据自己的需求选



择要安装的组件，一般这些组件需要全部选择。

注意：最新版的 Android Studio 在这一步并没有 SDK，需要稍后再进行安装。

(3) 点击“Next”按钮，弹出“Configuration Settings”对话框。在此对话框中需要设置 Android Studio 的安装位置，默认为“C:\Program Files\Android\Android Studio”。

(4) 点击“Next”按钮，弹出“Choose Start Menu Folder”对话框。

注意：如果开发者想要创建快捷方式，可以选中“Do not create shortcuts”复选框。

(5) 点击“Install”按钮，弹出“Installing”对话框，实现对 Android Studio 的安装。

(6) 安装完成后，弹出“Installation Complete”对话框。

(7) 点击“Next”按钮，弹出“Completing Android Studio Setup”对话框，如图 1-5 所示。此时 Android Studio 就安装完成了。

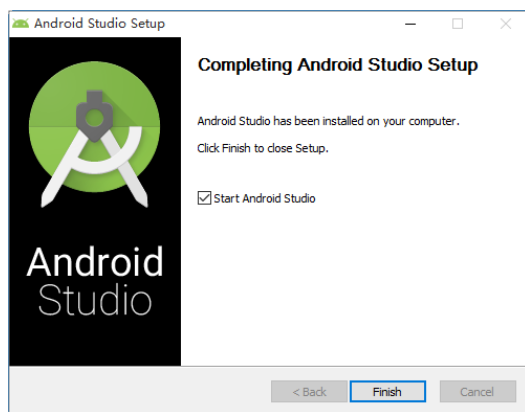


图 1-5 “Completing Android Studio Setup”对话框

1.2

创建第一个项目

本节将讲解如何启动 Android Studio、创建项目，在读者对项目有一个大致了解后，继续讲解如何在模拟器上运行程序、在真机上运行程序等。

1.2.1 首次启动 Android Studio

安装 Android Studio 后就可以使用该工具了。如果是首次使用该工具则需要对其进行一些设置，之后就不用再进行设置了。以下是首次启动 Android Studio 的具体操作步骤。

(1) 双击“Android Studio”或是在“Completing Android Studio Setup”对话框中选中“Start Android Studio”后，点击“Finish”按钮，会弹出“Import Android Studio Settings”对话框，如图 1-6 所示。

(2)选中“Do not import settings”，点击“OK”按钮，弹出 Android Studio 的 Logo，如图 1-7 所示。

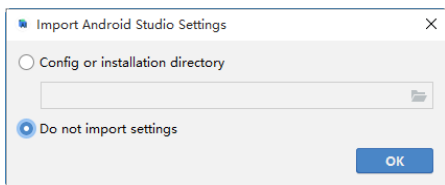


图 1-6 “Import Android Studio Settings”对话框

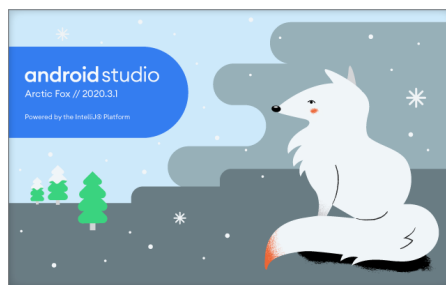


图 1-7 Logo

(3)一段时间后，会弹出“Data Sharing”对话框。

(4)点击“Don't send”按钮，弹出“Android Studio First Run”对话框。

注意：出现此对话框的原因是缺少 Android SDK。

(5)点击“Cancel”按钮，弹出“Finding Available SDK Components”对话框，如图 1-8 所示。

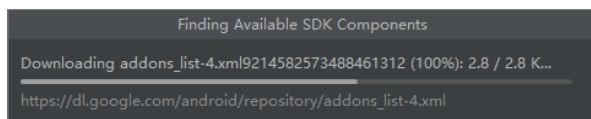


图 1-8 “Finding Available SDK Components”对话框

(6)一段时间后，弹出“Welcome”对话框。

(7)点击“Next”按钮，弹出“Install Type”对话框，如图 1-9 所示。

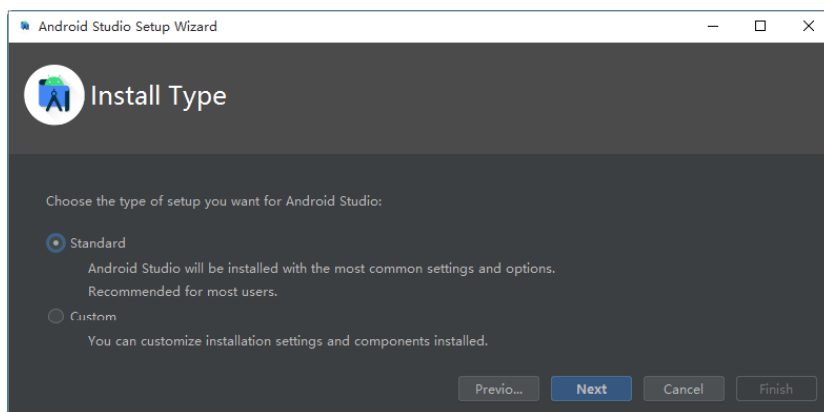


图 1-9 “Install Type”对话框

(8)选中“Standard”复选框，点击“Next”按钮，弹出“Select UI Theme”对话框，如图 1-10 所示。

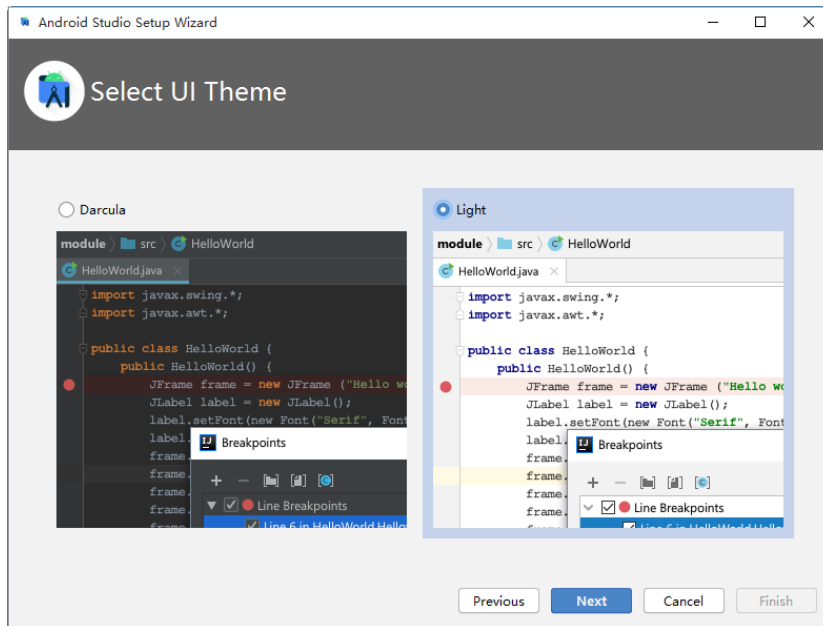


图 1-10 “Select UI Theme”对话框

(9) 开发者可以根据个人喜好选中主题类型。选择完毕后，点击“Next”按钮，弹出“Verify Settings”对话框。在此对话框中会显示相关的配置信息。

(10) 点击“Finish”按钮，弹出“Downloading Components”对话框，如图 1-11 所示。此时需要联网下载相关组件。

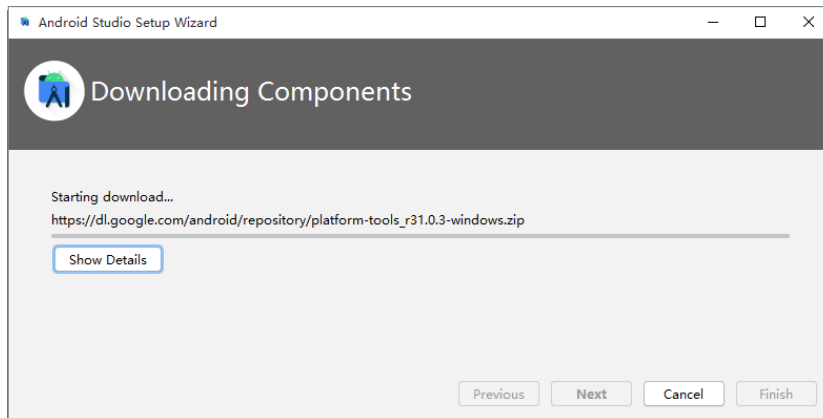


图 1-11 “Downloading Components”对话框

(11) 安装完毕后，“Downloading Components”对话框中的“Finish”按钮就会被激活。

(12) 点击“Finish”按钮，弹出“Welcome to Android Studio”对话框，如图 1-12 所示。

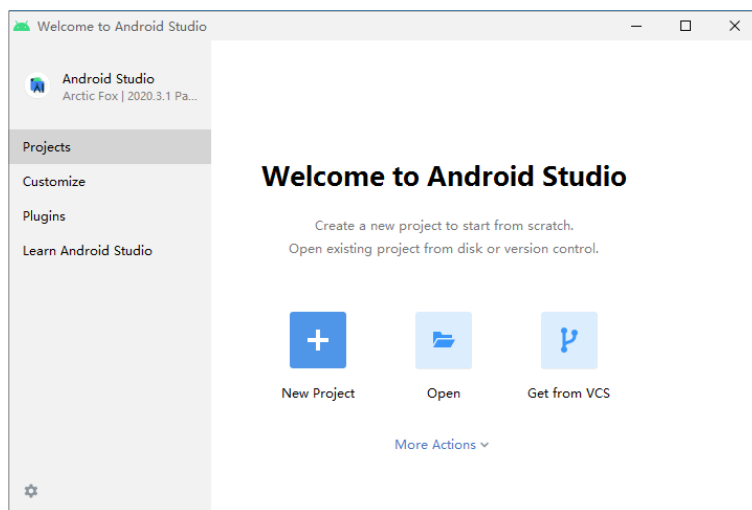


图 1-12 “Welcome to Android Studio” 对话框

注意：之后再启动 Android Studio，就是双击 Android Studio，弹出 Android Studio 的 Logo 一段时间后就会进入“Welcome to Android Studio”对话框。

1.2.2 创建项目

创建 Android 项目的具体操作步骤如下。

(1) 在“Welcome to Android Studio”对话框中，点击“New Project”按钮，弹出“New Project”对话框的“Phone and Tablet”面板，如图 1-13 所示，在此面板中可以选择合适的模板。

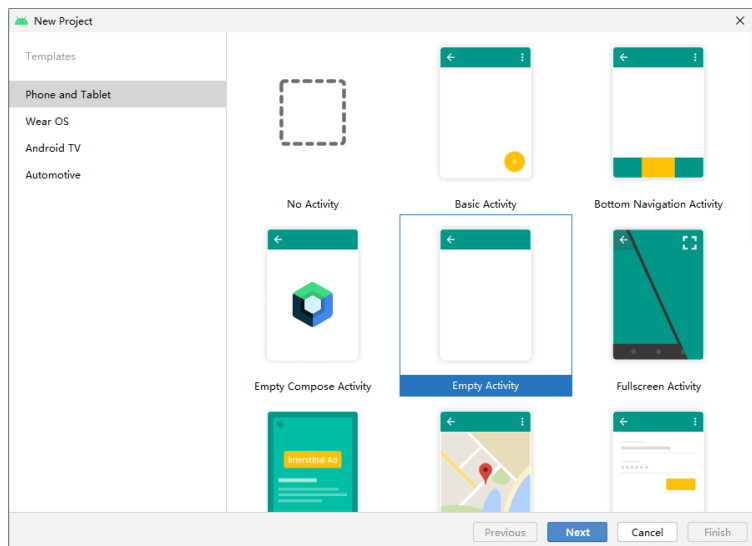


图 1-13 “New Project”对话框的“Phone and Tablet”面板

(2) 选择“Empty Activity”模板，点击“Next”按钮，弹出“New Project”对话框的“Empty Activity”面板，如图 1-14 所示，在此面板中对 Empty Activity 进行设置。

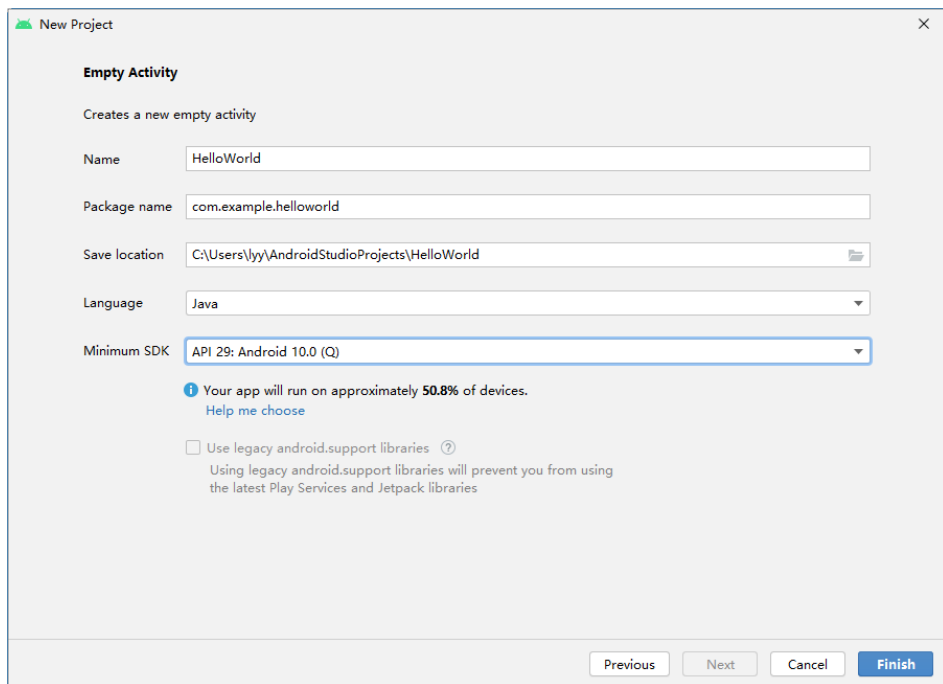


图 1-14 “New Project”对话框的“Empty Activity”面板

(3) 修改项目名称(默认为“My Application”),将“Language”设置为“Java”,点击“Finish”按钮,此时会打开创建的项目,如图 1-15 所示。开发者可以在此项目中编写程序及进行布局。

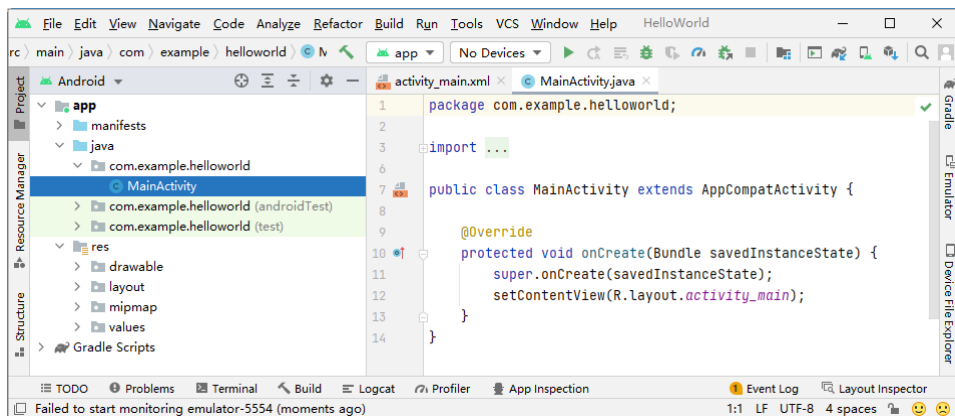


图 1-15 项目界面

1.2.3 了解项目

在创建的项目中,App 下方会有 3 个文件夹,下面依次介绍这 3 个文件夹中的内容。

1. manifests 文件夹

此文件夹中包含一个 AndroidManifest.xml 文件,此文件中是 App 的配置信息。双击此文件可以看到其中的内容,如图 1-16 所示。



```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.helloworld">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="HelloWorld"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/Theme.HelloWorld">
12        <activity
13            android:name=".MainActivity"
14            android:exported="true">
15            <intent-filter>
16                <action android:name="android.intent.action.MAIN" />
17
18                <category android:name="android.intent.category.LAUNCHER" />
19            </intent-filter>
20        </activity>
21    </application>
22
23 </manifest>
```

图 1-16 AndroidManifest.xml文件

让我们从第 2 行开始介绍该文件。

第 2 行: manifest 节点, 其属性包括 package, 指注册了 com.wes.helloworld 这个完整包名字。

第 5 行: application 节点, 其属性包括以下 6 个。

(1) allowBackup: 是否允许用户通过 adb backup 和 adb restore 来进行对应数据的备份和恢复。

(2) icon: 指定应用程序的普通图标。

(3) label: 指定应用的名称。

(4) roundIcon: 指定应用程序的圆形图标。

(5) supportsRtl: 是否支持 right-to-left 布局。

(6) theme: android 应用的风格。

第 12 行: activity 节点, 其属性包括以下 2 个。

(1) name: 指定活动的名称。

(2) exported: 表示该组件是否可以被其他应用程序访问或和其交互。

第 15 行到第 19 行: Intent 过滤器。要理解本行代码我们首先要了解每个组件都是由 Intent 启动的, 意图 (Intent) 的相关知识会在之后的章节中讲解。而 Intent 过滤器就是为了 Intent 准备的, 现在读者只须要知道该过滤器指定了该 Activity 是本应用程序的主入口。

2. java 文件夹

此文件夹中主要放置了源代码和测试代码。其中, 源代码可以说是项目中最主要的部分, 双击 MainActivity 文件, 我们就可以看到其中的代码了, 如图 1-17 所示。

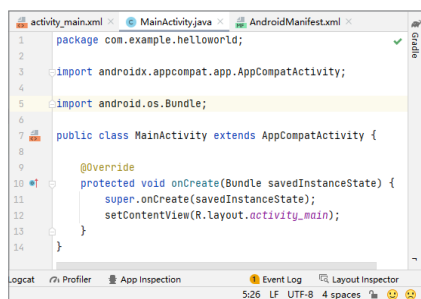


图 1-17 MainActivity文件

从图中我们知道该文件中的代码总共 14 行，去掉其中的空行、导入声明和花括号，真正的代码只有 4 行。下面将对这 4 行代码进行简单的介绍。

第 1 行：声明该类属于 com.example.helloworld 包。

第 3 行和第 5 行：声明导入的类，分别导入了 androidx.appcompat.app.AppCompatActivity 类和 android.os.Bundle 类。其中 AppCompatActivity 类可以被称为一个活动，从某种意义上说，Android 所有应用都是活动。Bundle 类是捆绑的意思，用来保存一些重要的数据。

第 7 行创建声明 public class 并声明其继承自 AppCompatActivity。

第 10 行创建 onCreate() 方法，它是 AppCompatActivity 开始的入口。

第 11 行创建调用父类的 onCreate() 方法，以创建 AppCompatActivity。

第 12 行创建设置界面，方法是将页面与资源文件绑定。setContentView() 方法的参数 R.layout.activity_main 就是 R 文件中布局文件 activity_main.xml 的引用了。

3. res 文件夹

此文件夹中存放的是所有的项目资源。下面是对其中 2 个子文件夹的介绍。

(1) drawable 文件夹：存放了一些 xml 文件，-*dpi 表示存储分辨率的图片，用于适配不同的屏幕。

(2) layout 文件夹：该文件夹中存放的是布局文件 activity_main.xml，双击可以打开该文件，如图 1-18 所示。

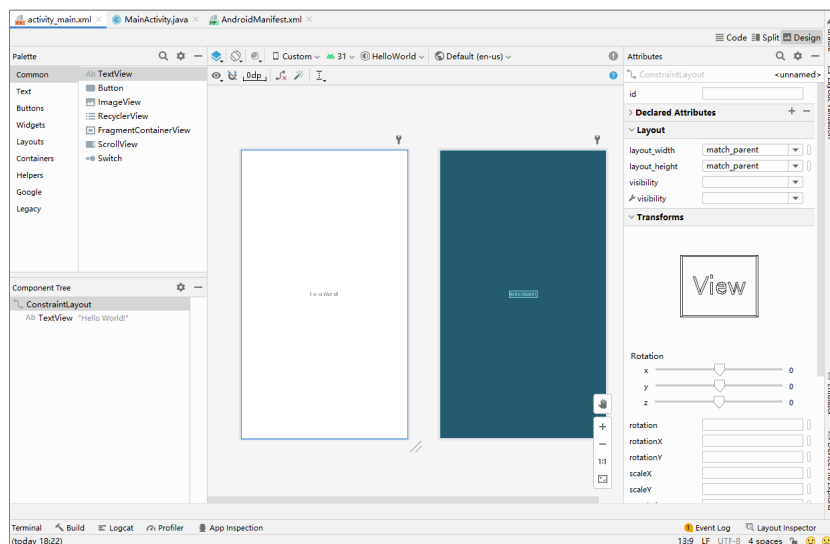


图 1-18 activity_main.xml 文件

从图 1-18 中我们看到的是一个可视化的设计界面，可以通过拖动的方式添加控件，点击某一控件后，在右边的“Attributes”面板中会出现对应的属性，开发者可以在此面板中对点击的组件的属性进行设置。如果不习惯使用这种方式进行布局，还可以使用代码实现，点击“Code”按钮，就会进入代码界面，如图 1-19 所示。

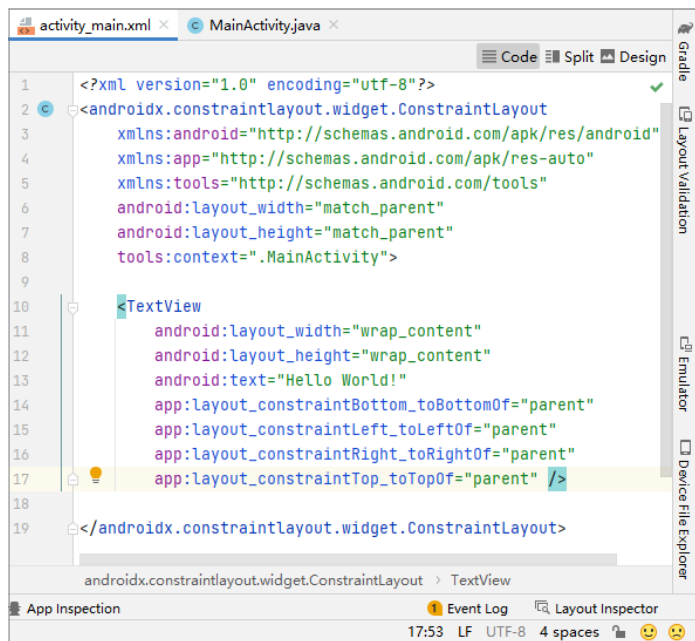


图 1-19 代码界面

在图中可以看到 activity_main.xml 文件总共有 18 行代码，接着我们就简单认识一下布局文件的代码。

- 第 1 行: xml 文件头，每个 xml 文件必需的声明，其中包括版本和编码方式。
- 第 2 行至第 8 行: 约束布局的开始节点，其中包括了若干属性，如宽度、高度等。
- 第 10 行至第 17 行: 完整的文本视图节点，其中包括的属性有宽度、高度和显示内容。
- 第 19 行: 约束布局的结束节点。

1.2.4 模拟器上运行程序

本小节将讲解如何创建模拟器及如何在模拟器上运行程序。

1. 创建模拟器

- (1) 在菜单栏中选择“Tools|AVD Manager”命令，弹出“Android Virtual Device Manager”界面。
- (2) 点击“+Create Virtual Device...”按钮，弹出“Virtual Device Configure”对话框的“Select Hardware”面板，如图 1-20 所示。

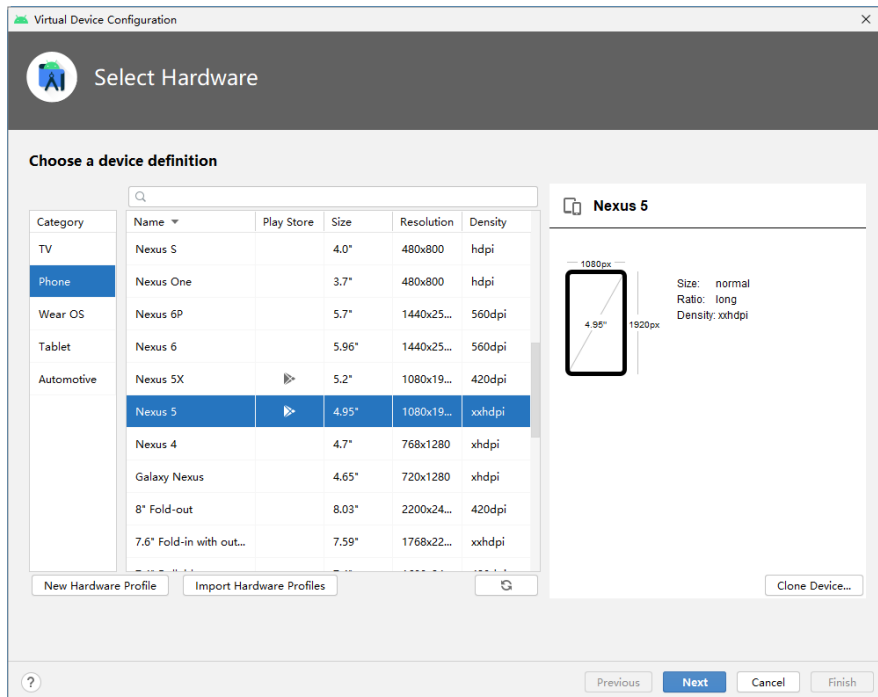


图 1-20 “Virtual Device Configure”对话框的“Select Hardware”面板

(3) 选择对应的设备，点击“Next”按钮，弹出“Virtual Device Configure”对话框的“System Image”面板，选择一个镜像，如图 1-21 所示。

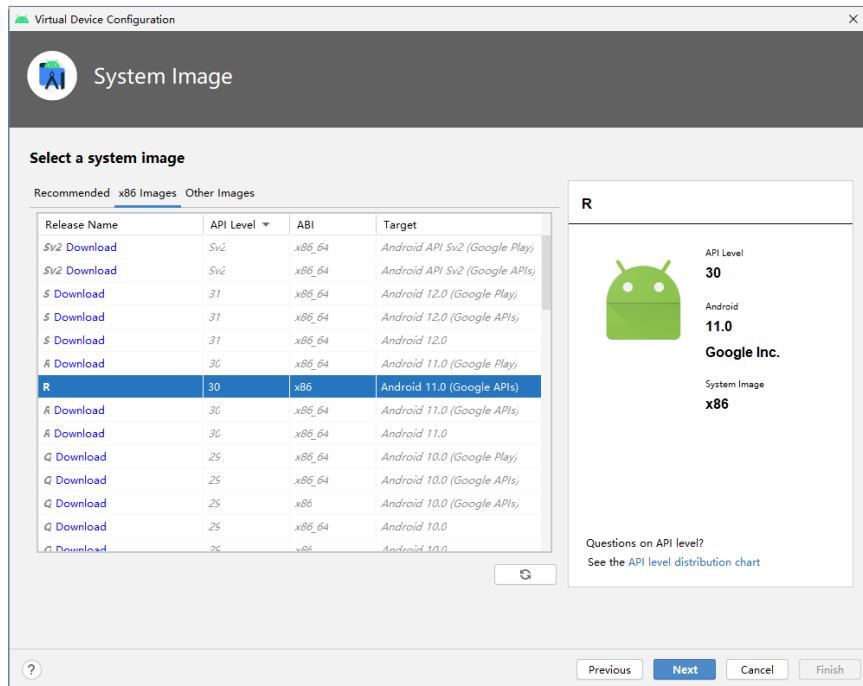


图 1-21 “Virtual Device Configure”对话框的“System Image”面板

(4) 点击“Next”按钮，弹出“Virtual Device Configure”对话框的“Android Virtual Device

(AVD)”面板，在此面板中可以直接使用默认输入及选项，如图 1-22 所示。

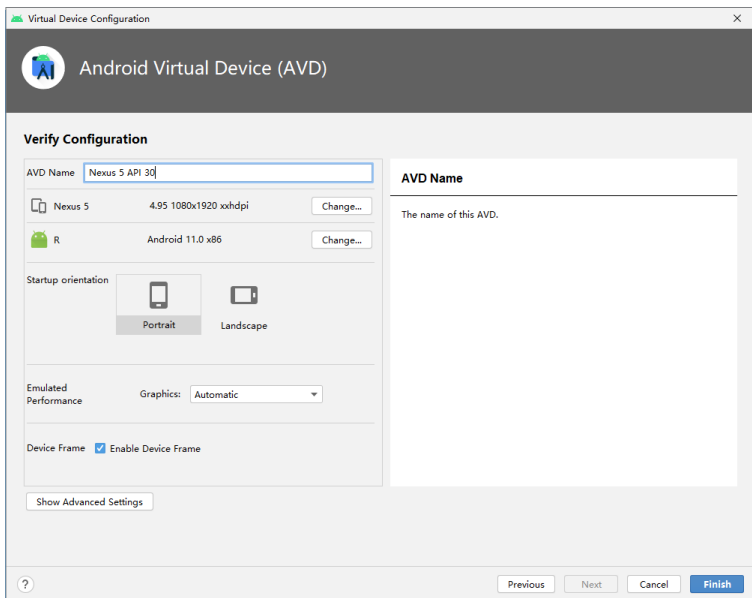


图 1-22 “Virtual Device Configure”对话框的“Android Virtual Device (AVD)”面板

(5) 点击“Finish”按钮后，此时名为“Nexus 5 API 30”的模拟器就创建好了，如图 1-23 所示，它会显示在“Android Virtual Device Manager”界面的“Your Virtual Devices”面板中。

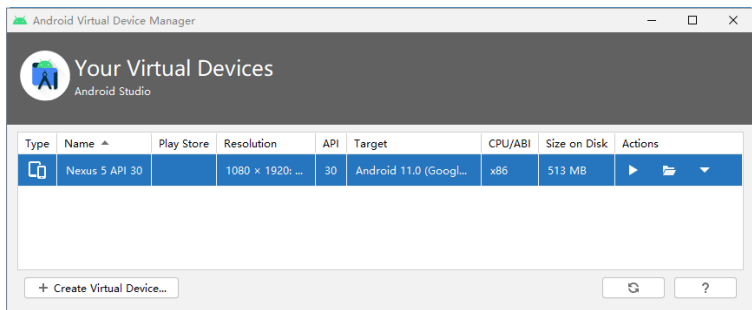


图 1-23 创建好的模拟器

2. 运行程序

创建好模拟器后，就可以在创建的模拟器上运行程序了。此时需要将程序运行设备设置为创建的模拟器，本书中为“Nexus 5 API 30”，如图 1-24 所示。然后点击运行按钮就可以了，运行效果如图 1-25 所示。

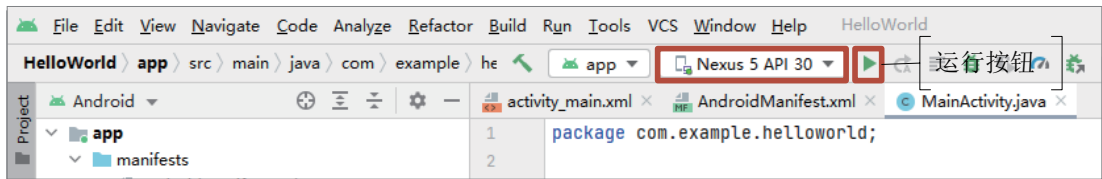


图 1-24 设置程序运行设备

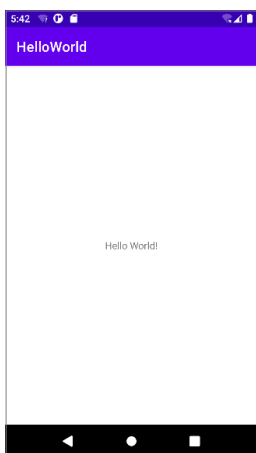


图 1-25 运行效果

1.2.5 真机上运行程序

在真机上运行程序，首先需要启动开发者模式。这里以 Redmi Note 9 为例介绍它的开启步骤。

(1) 在手机“设置”菜单中，点击“我的设备”选项，进入“我的设备”界面。

(2) 向下滚动，点击“全部参数”选项，进入“全部参数”界面。

(3) 找到 MIUI 版本号，并连续点击 MIUI 版本号多次，即可进入开发者模式。

(4) 返回“设置”界面，点击“更多设置”选项，进入“更多设置”界面，找到“开发者选项”。

(5) 点击“开发者选项”，在“开发者选项”界面中，开启“开启开发者选项”，如图 1-26 所示。

(6) 在“开发者选项”界面中下滑开启“USB 调试”，如图 1-27 所示。此时就可以进行真机测试了。

在运行测试时，需要使用 USB 线将手机连接到电脑上，然后将程序运行设备设置为真机，最后点击运行按钮，运行程序，效果如图 1-28 所示。



图 1-26 开启“开发者选项”界面



图 1-27 开启“USB”调试



图 1-28 效果

1.3 调试程序

调试程序是在进行应用开发时必不可少的一个重要环节，本节将讲解一些简单的调试技巧以帮助开发者更高效更快速地开发程序。

1.3.1 增加断点

选中希望程序运行暂停的代码，单击代码的左侧就完成了断点的增加。以 HelloWorld 项目为例，如果希望程序在运行至 `setContentView()` 时暂停，那就在第 12 行代码左侧单击，这时会出现一个红色的小圆点，这个小圆点就是所谓的断点了，如图 1-29 所示。

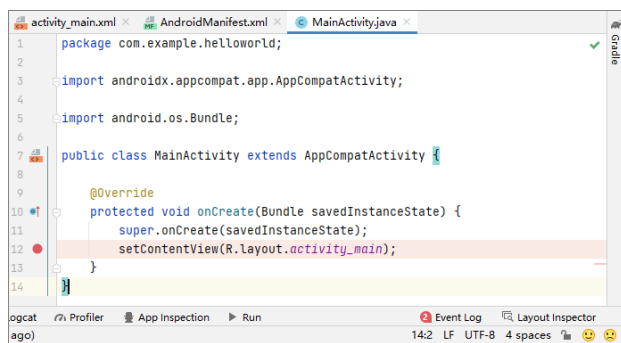


图 1-29 断点

1.3.2 开始调试

添加断点后，就可以开始调试了。选择工具栏中的“Debug”选项（绿色的虫子图标）。此时会以 Debug 模式启动 App。App 启动后，运行至第一处断点处会停下来，同时 IDE 下方出现 Debug 视图，红色的箭头指向的是现在调试程序停留的代码行，如图 1-30 所示。

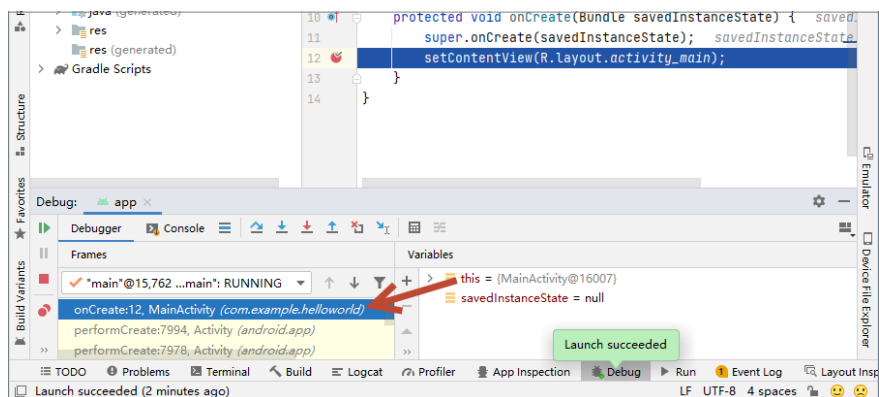


图 1-30 Debug 视图



1.3.3 单步调试

在显示的 Debug 视图中，会看到一些箭头按钮，如图 1-31 所示。这些箭头按钮就是用于实现单步调试的。

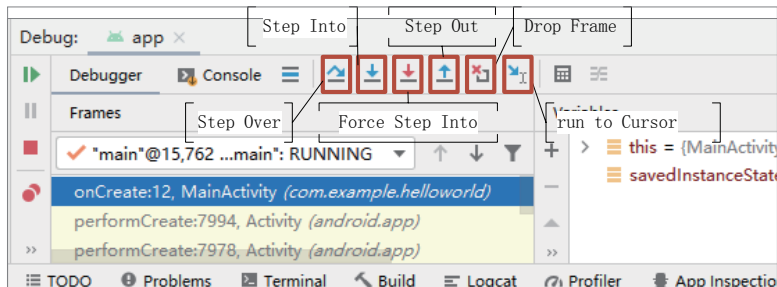


图 1-31 单步调试按钮

箭头按钮的介绍如下。

(1) Step Over: 单步跳过，点击该按钮将导致程序向下执行一行。如果当前行是一个方法调用，则此行调用的方法被执行完毕后再到下一行。

(2) Step Into: 单步跳入，执行该操作将导致程序向下执行一行。如果该行有自定义的方法，则进入该方法内部继续执行。需要注意的是如果是类库中的方法，则不会进入方法内部。

(3) Force Step Into: 强制单步跳入，和 Step Into 功能类似，主要区别在于如果当前行有任何方法，则不管该方法是由我们自行定义的还是由类库提供的，都能跳入方法内部继续执行。

(4) Step Out: 如果有断点则走到下一个断点；如果没有断点，而是在一个调用的方法当中，则会跳出这个方法，继续走。

(5) Drop Frame: 中断执行，并返回到方法执行的初始点，在这个过程中该方法对应的栈帧会从栈中移除。换言之，如果该方法是被调用的，则返回到当前方法被调用处，并且所有上下文变量的值也恢复到该方法未执行时的状态。

(6) Run to Cursor: 设置多个断点时，可利用该按钮在两个断点之间跳转。

任务 1-1

创建一个显示“Hello, Android”的界面

任务描述

- (1) 在界面显示一个文本视图。
- (2) 文本视图中的内容为“Hello, Android”。

任务实施

1. 创建项目

创建 Android 项目，项目名为 HelloAndroid。

2. 修改 activity_main.xml 文件的代码

打开 activity_main.xml 文件，修改 TextView 中的 text 属性，将该属性设置为“Hello, Android”，代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
```



创建一个显示“Hello, Android”的界面



```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, Android"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

运行程序，效果如图 1-32 所示。

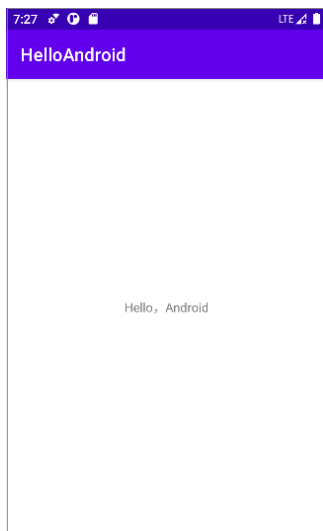


图 1-32 效果



1. Android 体系结构

Android 系统自上而下由 4 层组成，分别为应用层、应用框架层、各种库和运行时环境层及操作系统层，如图 1-33 所示。下面依次介绍这 4 层。

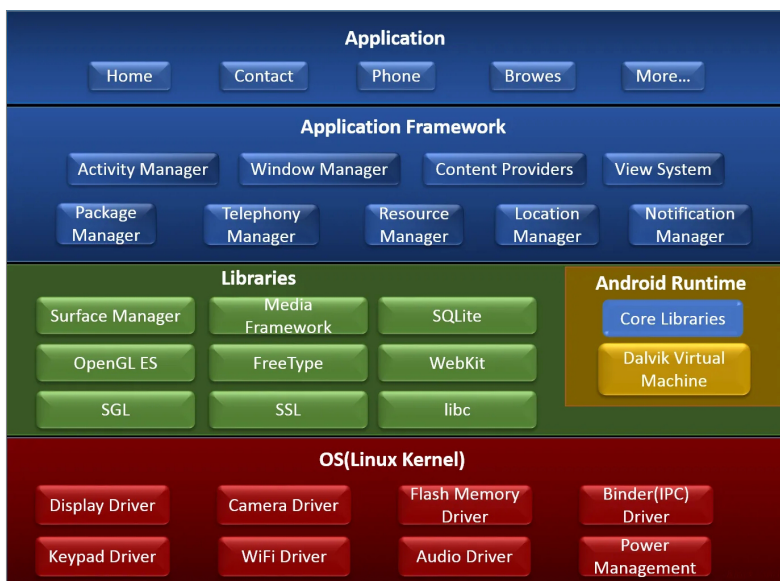


图 1-33 Android 系统结构图

(1)应用层。在应用层中，开发者可以使用JAVA 语言调用自带的各种应用程序，包括联系人、电话、浏览器、电子邮件客户端等各种功能。

(2)应用框架层。该层为系统提供了各种各样的API，它包括以下 10 种。

① Activity Manager: 活动管理器，一个应用程序由至少一个活动 (Activity) 构成，活动管理器负责管理 Activity 的生命周期，并为程序提供退出机制。

② Window Manager: 窗口管理器，管理所有的窗口程序。

③ Content Providers: 内容提供者，负责共享程序的数据，该机制解决了各个应用程序的数据私有和共享的问题 (在第 9 章会有详细讲解)。

④ View System: 视图系统，可以用来构建应用程序，它包括各种可重用的组件: 列表、网格、文本框、按钮等。

⑤ Notification Manager: 消息管理器，它可以帮助开发者在状态栏中显示自定义的提示信息。

⑥ Package Manager: 包管理器，它可以帮助开发人员管理所有的包。

⑦ Telephony Manager: 电话管理器，管理 Android 手机中所有的电话接入和拨出等操作。

⑧ Resource Manager: 资源管理器，提供非代码资源的访问，如本地字符串、图形和布局文件。

⑨ Location Manager: 位置管理器，使用它可以开发 LBS (Location Based Service) 程序。

⑩ XMPP Service: 可扩展通讯和表示协议服务 (The Extensible Messaging and Presence Protocol), XMPP 是一种基于 XML 的协议，具有超强的可扩展性。经过扩展以后的 XMPP 可以通过发送扩展的信息来处理用户的需求。

(3)各种库和运行时环境层。Android 应用框架需要系统底层的一些 C/C++ 库的支持。这些库包括以下 8 种。

① Bionic 系统 C 库: C 语言标准库，系统最底层的库，C 库通过 Linux 系统来调用。



②多媒体库: Android系统多媒体库, 基于PackerVideo OpenCORE, 支持各类音频格式的录制和播放, 包括MPEG4、MP3、AAC、AMR等; 支持各类视频的录制和播放, 包括3GP、MP4等; 支持各类图片格式的处理, 包括JPG、PNG等。

③SGL: 2D图形引擎库。

④SSL: 位于TCP/IP与各种应用层协议之间, 为数据通信提供支持。

⑤OpenGL ES 1.0: 支持3D效果。

⑥SQLite: 关系数据库, 提供数据存储服务。

⑦Webkit: Web浏览器引擎。

⑧FreeType: 提供位图和矢量的支持。

(4)操作系统层。这一层包含操作系统所具有的功能, 这些功能的介绍如下。

①显示驱动(Display Driver): 基于Linux的帧缓冲(Frame Buffer)驱动。

②键盘驱动(KeyBoard Driver): 作为输入设备的键盘驱动。

③USB驱动(USB Driver): 为设备提供USB驱动。

④Flash内存驱动(Flase Memory Driver): 闪存驱动程序。

⑤照相机驱动(Camera Driver): 常用的基于Linux的v412(Video for Linux)的驱动。

⑥音频驱动(Audio Driver): 常用的基于ALSA的高级Linux声音体系驱动。

⑦蓝牙驱动(Bluetooth Driver): 基于IEEE 802.15.1标准的无线传输技术。

⑧WiFi驱动: 基于IEEE 802.11标准的驱动程序。

⑨Binder IPC驱动: Android的一个特殊的驱动程序, 提供进程间通信的功能。

⑩Power Management(电源管理): 管理电池电量。

2. Dalvik虚拟机

在Android操作系统中, 每个Java程序都运行在一个独立的Dalvik虚拟机上。Dalvik被设计为一个设备, 可同时高效地运行多个虚拟系统。每一个Android应用都运行在一个Dalvik虚拟机实例中, 每一个虚拟机实例都是一个独立的进程空间, 它只能执行.dex的可执行文件。也就是说, 当Java程序通过编译后生成的.class文件, 还需要通过SDK中的dx工具转为.dex格式才能正常在虚拟机上执行。



一、填空题

1. 目前Android的最新版本是_____。

2. Android Studio是谷歌推出的一个_____集成开发工具, 用于_____和调试, 类似于_____。

二、选择题

1. 下列提供了数据库服务的选项是()。

A. SSL

B. OpenGL ES 1.0

C. SQLite

D. Webkit



2. 下列是 Android 布局文件的选项是()。

- A. MainActivity 文件
- B. AndroidManifest.xml 文件
- C. activity_main.xml
- D. 其他

三、判断题

- 1. API 级别是一个用于唯一标识 API 框架版本的字符串。 ()
- 2. AndroidManifest.xml 文件中是 App 的配置信息。 ()

四、操作题

显示一个“Hello, TextView”的界面。

