
基于Android 8.0和Android Studio 3.0开发工具
以企业开发常用技术为主线，全面系统地学习Android开发

上海翼成CTO 郭 鹏
《Android群英传》作者 徐宜生
《Android进阶之光》《Android进阶解密》作者 刘望舒
美团高级工程师 Tamic

联合
推荐



Android App 开发

从入门到精通

· 安辉 编著 ·



清华大学出版社



Android App 开发 从入门到精通

· 安辉 编著 ·

清华大学出版社
北京

内 容 简 介

本书使用 Android Studio 3.0 开发环境,同时适配新版的 Android 8.0 操作系统,由浅入深地学习 Android App 的开发。全文共分为 10 章,涵盖 Android Studio 的开发环境搭建、Android 控件的使用、四大组件的使用、Fragment (碎片)、多线程开发、网络编程与数据存储等内容。最后通过项目实战,对所学知识点融会贯通,进一步增强开发能力。

本书内容通俗易懂,案例丰富,不仅适用于 Android 开发的广大从业人员、App 开发的业余爱好者,也可作为大中专院校与培训机构的培训教程。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Android App 开发从入门到精通 / 安辉编著. —北京:清华大学出版社, 2018
ISBN 978-7-302-51358-2

I. ①A… II. ①安… 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 (2018) 第 229155 号

责任编辑:王金柱

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 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:190mm×260mm 印 张:26 字 数:666 千字

版 次:2018 年 12 月第 1 版 印 次:2018 年 12 月第 1 次印刷

定 价:79.00 元

产品编号:072653-01

前 言

我在写书之前一直在 CSDN 上发表文章，同时在“知乎”等网站回答一些 Android 相关的问题，后来有幸收到了清华大学出版社编辑的邀请，于是产生了编写本书的想法。

我最早是从事 Java Web 开发的，出于对 Android 的浓厚兴趣，后来又开始从事 Android 的开发。在开发过程中，我走了很多弯路，阅读了很多 Android 方面的书，从入门类到高级开发类都有，美中不足的是这些书要么篇幅过长要么技术过时，浪费了很多时间。鉴于此，本书将结合我多年的 Android 开发经验，总结企业中常用的开发技术，使用前沿技术兼容最新的 Android 操作系统，使初学者快速加入 Android 开发阵营。即使是中、高级开发者，阅读本书后也能从中获益。

Android 操作系统经过将近 10 年的发展。随着移动 App 的热潮，越来越多的人加入移动开发的大军，企业对 Android 招聘的需求也越来越高。本书内容从基础入门到高级开发，涵盖企业开发中常用的技术点，能让读者对 Android 开发有一个学习框架。最后一章通过模仿商业 App 开发，融会贯通前面的知识点，以提高读者项目开发的实战能力。

本书内容

本书共有 10 章，主要内容如下：

- 第 1 章学习开发工具 Android Studio 的使用，一个好的开发工具可以大大提高开发人员的工作效率。
- 第 2 章讲解 Android 控件相关知识，一个 UI 界面由多个控件组成，只有熟练使用各种控件才能设计出好看的 App，达到 UI 设计师想要的效果。
- 第 3 章学习 Android 中四大组件的使用。在企业的项目开发中，四大组件中的 Activity（活动）、Service（服务）、Broadcast Receiver（广播接收器）使用很频繁，ContentProvider（内容提供者）使用频率相对少一些，只有某些特定需求时才会用到。
- 第 4 章学习 Fragment（碎片）的使用方法，从 Fragment 简单使用到最后的案例开发，一步步深入地学习 Fragment。使用 Fragment 会让 App 模块化，还能解决手机与平板电脑的适配问题。
- 第 5 章学习多线程开发。从多线程的创建，到子线程如何更新 UI，通过阅读源码分析 Handle 的实现原理，最后介绍线程池的使用方法。
- 第 6 章首先学习 Android 的网络编程，通过 Get/Post 方式向服务器发送 HTTP 请求。现在市面上大部分 App 与服务器交互都是返回 Json 数据，所以介绍 Gson 框架，以及 OkHttp 开源项目的使用和封装。最后是数据存储的三种方式。
- 第 7 章学习 Android 的高级应用，主要介绍 Notification 使用、多媒体开发、WebView 使用、定位的三种方式、NDK 和 JNI 开发、Git 管理项目等。

- 第 8 章学习 Android 中各大版本的更新，让我们的 App 解决版本适配问题，完美兼容 5.0 以上的各个版本。
- 第 9 章学习常用功能模板的使用。这些功能是企业开发中可能会碰到的需求，通过模板的学习，知道如何对一个 App 进行功能划分以及如何封装模块。
- 第 10 章通过模仿一个商业 App，从零开始搭建项目，使用前面 9 章所学的内容，将所学知识融会贯通，并进一步熟练掌握。有了项目开发的经验，你在今后的企业开发中就能快速成为一名合格的开发人员。

本书特色

本书定位为**基础类图书**，对每一个知识点的讲解都很详细，从基础入门逐步进入高级应用，让读者能系统全面地学习 Android 开发，更深入地了解 Android 开发体系。本书的内容是我多年 Android 开发经验的总结，也是一个合格的 Android 开发者必须掌握的内容，简单来说，就是企业开发中经常用到的技术。

致谢

首先感谢我的好朋友王帅和芮成兵，他们协助我完成了本书的审阅工作，给本书的修订提供了宝贵的意见；然后感谢翼成的伙伴们，在写书的这段时间里他们给予我很大的帮助。

书中案例源码下载

https://github.com/anssen666/book_source_code

资源与勘误

由于技术水平有限，书中难免会有疏漏，欢迎大家通过 androidcoder666@163.com 邮箱提供反馈意见。另外，还可以通过以下方式进行交流：

- CSDN 博客：https://blog.csdn.net/lowprofile_coding
- QQ 群：202928390
- 微信公众号：Android 开发 666

编者
2018 年 10 月

目 录

第 1 章	Android Studio 的介绍以及使用	1
1.1	探索 Android Studio	1
1.1.1	项目结构	2
1.1.2	Android Studio 主窗口	3
1.1.3	工具窗口	4
1.1.4	代码自动完成	4
1.1.5	样式和格式化	5
1.1.6	版本控制基础知识	5
1.1.7	Gradle 构建系统	6
1.1.8	Debug 调试	6
1.1.9	性能监视器	7
1.1.10	分配跟踪器	7
1.1.11	数据文件访问	7
1.1.12	代码检查	7
1.1.13	日志消息	8
1.2	下载与安装 Android Studio	8
1.2.1	下载 Android Studio	8
1.2.2	开始安装	9
1.3	Android Studio 使用	13
1.3.1	项目结构	13
1.3.2	创建项目	16
1.3.3	Android Studio 自带模拟器运行项目	20
1.3.4	使用 Genymotion 模拟器运行	22
1.3.5	真机运行	25
1.4	调试项目	27
1.4.1	Debug 断点调试	27
1.4.2	日志调试	29
1.5	Eclipse 项目迁移至 Android Studio	30
1.5.1	Eclipse 项目迁移条件	30
1.5.2	将 Eclipse 项目导入 Android Studio	30

1.5.3	验证导入是否成功	32
1.6	创建 Android 库	32
1.6.1	创建库模块	33
1.6.2	将库模块导入到项目中	33
1.6.3	将应用模块转换为库模块	33
1.6.4	开发库模块的注意事项	34
1.6.5	AAR 文件详解	34
1.7	项目依赖库	35
1.7.1	依赖本地库	35
1.7.2	在线依赖库	36
1.8	应用清单文件	36
1.8.1	清单文件结构	37
1.8.2	文件约定	38
1.9	常用快捷键	39
1.10	应用签名	40
1.10.1	证书和密钥库	40
1.10.2	调试项目时签名	41
1.10.3	正式签名	41
1.11	多渠道打包	43
1.11.1	代码实现	43
1.11.2	测试	44
1.12	ADB 详解	45
1.12.1	Mac 下 adb 加入环境变量 (Windows 电脑自行搜索)	45
1.12.2	adb 常用命令	46
1.13	Android Studio 3.0 新特性	46
1.13.1	Android Gradle 插件 3.0.0	47
1.13.2	手动更新 Gradle 版本	47
1.13.3	Kotlin 支持	47
1.13.4	Java 8 支持	48
1.13.5	Android Profiler	48
1.13.6	CPU Profiler	49
1.13.7	Memory Profiler	49
1.13.8	Network Profiler	50
1.13.9	APK profiling	50
1.13.10	Device File Explorer	51
1.13.11	Adaptive Icons wizard	51
1.13.12	Google 的 Maven 存储库	52
1.14	本章小结	52

第 2 章 Android 控件	53
2.1 View 介绍	53
2.1.1 自定义 View	54
2.1.2 自定义属性	58
2.2 ViewGroup 介绍	59
2.3 几种常用的布局	67
2.3.1 LinearLayout (线性布局)	67
2.3.2 RelativeLayout (相对布局)	70
2.3.3 FrameLayout (框架布局)	72
2.3.4 三大布局嵌套以及动态添加 View	73
2.4 初级控件的使用	75
2.4.1 TextView (文本视图)	76
2.4.2 Button (按钮)	79
2.4.3 EditText (文本编辑框)	83
2.4.4 ImageView (图像视图)	86
2.4.5 RadioButton (单选按钮)	87
2.4.6 Checkbox (复选框)	89
2.4.7 ProgressBar (进度条)	91
2.4.8 ProgressDialog (进度对话框)	93
2.4.9 AlertDialog (简单对话框)	94
2.4.10 PopupWindow (弹出式窗口)	96
2.4.11 DialogFragment	99
2.5 Android 高级控件的使用	102
2.5.1 ListView (列表视图)	102
2.5.2 GridView (网格视图)	109
2.5.3 RecyclerView (循环视图)	113
2.5.4 SwipeRefreshLayout (下拉刷新)	127
2.5.5 ViewPager (翻页视图)	131
2.6 通过 xml 文件修饰 View	134
2.6.1 shapes (设置圆角、边框、填充色、渐变色)	134
2.6.2 selector (设置点击、选中点击效果)	136
2.6.3 layer-list (把 item 按照顺序层叠显示)	137
2.7 本章小结	138
第 3 章 Android 四大组件	139
3.1 Activity (活动)	139
3.1.1 Activity 的生命周期	139
3.1.2 启动 Activity 的两种方式	142

3.1.3	在 Activity 中使用 Toast	143
3.1.4	Activity 启动与退出动画	147
3.1.5	Activity 销毁	156
3.1.6	Activity 与 Activity 之间传递数据	158
3.1.7	Activity 的软键盘弹出方式	160
3.1.8	Activity 任务栈	161
3.1.9	Activity 四种启动模式	162
3.2	Service (服务)	164
3.2.1	Activity 中启动 Service 以及销毁 Service	164
3.2.2	Activity 与 Service 通信	167
3.3	Broadcast Receiver (广播接收器)	171
3.3.1	动态注册广播	171
3.3.2	静态注册广播	172
3.3.3	广播基本总结	174
3.3.4	应用内广播 LocalBroadcastManager	174
3.4	ContentProvider (内容提供者)	175
3.5	本章小结	176
第 4 章	Fragment 探索	177
4.1	Fragment 简介	177
4.2	Fragment 生命周期	177
4.3	FragmentManager 与 FragmentTransaction 的使用	182
4.3.1	FragmentManager (Fragment 管理类) 的使用	182
4.3.2	FragmentTransaction (Fragment 事务) 的使用	183
4.4	Activity 动态操作 Fragment	183
4.5	Fragment 与 Activity 交互数据	186
4.6	Fragment 案例——实现底部导航栏	187
4.6.1	分析需求	188
4.6.2	代码实现	188
4.7	本章小结	195
第 5 章	Android 多线程开发	196
5.1	多线程的创建	196
5.2	子线程中更新 UI 的四种方法	197
5.2.1	用 Activity 对象的 runOnUiThread 方法	197
5.2.2	View.post 的使用	199
5.3	Handler 的使用	199
5.3.1	为什么要用 Handler	199
5.3.2	使用 Handler	200

5.3.3	Handler、Looper 与 MessageQueue 三者的关系	202
5.4	使用 AsyncTask 创建后台线程	207
5.5	线程池的使用	208
第 6 章	Android 网络编程与数据存储	216
6.1	基于 Android 平台的 HTTP 通信	216
6.1.1	使用 Get 方式向服务器提交数据	217
6.1.2	使用 Post 方式向服务器提交数据	218
6.1.3	使用 GSON 解析 JSON 格式的数据	219
6.1.4	OkHttp 开源项目的使用	221
6.2	数据存储	243
6.2.1	SharedPreferences	243
6.2.2	SQLite 数据库	244
6.2.3	文件存储	247
6.3	本章小结	247
第 7 章	Android 高级应用	248
7.1	Notification (通知) 使用	248
7.1.1	创建通知	249
7.1.2	通知优先级	250
7.1.3	更新通知	250
7.1.4	删除通知	251
7.1.5	自定义通知布局	251
7.2	多媒体开发	252
7.2.1	播放音频	252
7.2.2	播放视频的三种方式	257
7.3	调用浏览器打开网页	262
7.3.1	启动 Android 默认浏览器	262
7.3.2	启动指定浏览器打开	262
7.3.3	优先使用	262
7.4	WebView 的使用	263
7.4.1	WebView 加载网页的四种方式	263
7.4.2	WebViewClient 与 WebChromeClient 的区别	263
7.4.3	WebView 的简单使用	264
7.5	复制和粘贴	268
7.5.1	复制文本	268
7.5.2	粘贴文本	268
7.6	定位的使用	268
7.6.1	定位的三种方式	269

7.6.2	定位的相关类	270
7.6.3	GPS 获取经纬度	271
7.6.4	根据经纬度反向编码获取地址	273
7.7	NDK 与 JNI 开发	275
7.7.1	什么是 NDK	275
7.7.2	NDK 下载	276
7.7.3	在 Mac 下加入 NDK 环境变量	277
7.7.4	什么是 JNI	278
7.7.5	NDK 与 JNI 的简单使用	278
7.8	使用 SourceTree 上传项目到 GitHub	281
7.8.1	什么是 Git	281
7.8.2	什么是 GitHub	282
7.8.3	什么是 SourceTree	283
7.8.4	使用 SourceTree 操作 GitHub	283
7.9	将项目发布到 JCenter	292
第 8 章	Android 5.X、6.X、7.X、8.X 各版本特性	299
8.1	Android 5.X 版本新特性	299
8.1.1	悬挂式 Notification	299
8.1.2	利用 DrawerLayout 和 NavigationView 实现侧边栏	301
8.1.3	TabLayout 和 ViewPager 结合使用	304
8.1.4	CoordinatorLayout、FloatingActionButton 和 Snackbar 的使用	306
8.2	Android 6.X 版本新特性	307
8.3	Android 7.X 版本新特性	310
8.3.1	多窗口支持	311
8.3.2	FileProvider 解决 FileUriExposedException	312
8.4	Android 8.X 版本新特性	314
第 9 章	常用功能模板	319
9.1	启动页与首次启动的引导页	319
9.1.1	需求分析	321
9.1.2	代码实现	321
9.1.3	启动页	321
9.1.4	引导页	323
9.2	检查更新并下载安装	327
9.3	Banner 广告轮播图	332
9.3.1	运行效果图	332
9.3.2	代码实现	333
9.4	微信登录、分享与支付	337

9.4.1	代码实现	337
9.4.2	微信登录	342
9.4.3	微信分享	344
9.4.4	微信支付	344
9.4.5	签名	346
9.4.6	微信开放平台官网的后台配置	347
9.4.7	运行软件	347
9.4.8	微信官方开发文档	348
9.5	百度地图	349
9.5.1	百度定位 SDK	350
9.5.2	百度地图 SDK	357
第 10 章	实现开发者头条	364
10.1	启动页实现	364
10.1.1	启动页的目标效果	364
10.1.2	代码实现	365
10.2	使用 DrawerLayout 控件实现侧滑菜单栏	370
10.2.1	侧滑菜单的目标效果	370
10.2.2	代码实现	370
10.3	开发者头条首页实现	377
10.3.1	源代码的实现	378
10.3.2	精选 Fragment	382
10.4	开发者头条首页优化	387
10.4.1	需要在线依赖	388
10.4.2	标题栏和三个切换选项卡	388
10.4.3	分析 TabLayout 切换源代码	391
10.4.4	精选文章列表控件从 ListView 替换成 RecyclerView	393
10.5	RecyclerView 实现下拉刷新和上拉加载更多	395
10.5.1	实现步骤	396
10.5.2	实现详解	396

第 1 章

Android Studio 的介绍以及使用

孔子云：“工欲善其事，必先利其器”。一个好的开发工具可以让开发人员的工作效率有大幅度的提高，本章主要学习 Android Studio 的使用。以前都是用 Eclipse 开发 Android 程序，自从 2013 年 Google 官方发布了 Android Studio，现在很少有人使用 Eclipse 开发 Android 程序了。本书不对 Eclipse 多做介绍。

Android Studio 是 Google 于 2013 年 I/O 大会针对 Android 开发推出的新开发工具，是基于 IntelliJ IDEA 开发的，IntelliJ 在业界被公认为最好的 Java 开发工具之一。尤其是在智能代码助手、代码自动提示、重构、J2EE 支持、各类版本工具（Git、SVN、GitHub 等）、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能，可以说是超常的。IDEA 是 JetBrains 公司的产品，它的旗舰版本还支持 HTML、CSS、PHP、MySQL、Python 等，免费版本只支持 Java 等少数语言。

1.1 探索 Android Studio

Android Studio 是基于 IntelliJ IDEA 的官方 Android 应用集成开发环境（IDE）。除了 IntelliJ 强大的代码编辑器和开发者工具，Android Studio 提供了更多可提高 Android 应用构建效率的功能，例如：

- 基于 Gradle 的灵活构建系统。
- 快速且功能丰富的模拟器。
- 可针对所有 Android 设备进行开发的统一环境。
- Instant Run，可将变更推送到正在运行的应用，无须构建新的 APK。
- 帮助构建应用程序和导入示例代码以及 GitHub 集成。
- 丰富的测试工具和框架。
- 可捕捉性能、易用性、版本兼容性以及其他问题的 Lint 工具。

- C++和NDK支持。
- 内置对 Google 云端平台的支持，可轻松集成 Google Cloud Messaging 和 App 引擎。

1.1.1 项目结构

Android Studio 中的每个项目包含一个或多个含有源代码文件和资源文件的模块。模块类型包括：

- Android 应用模块。
- 库模块。
- Google App 引擎模块。

默认情况下，Android Studio 会在 Android 项目视图中显示项目文件，如图 1-1 所示。该视图按模块组织结构，便于快速访问项目的关键源文件。

所有构建文件在项目层次结构顶层 Gradle Scripts 下显示，并且每个应用模块都包含以下文件夹：

- manifests: 包含 AndroidManifest.xml 文件。
- java: 包含 Java 源代码文件，包括 JUnit 测试代码。
- res: 包含所有非代码资源，例如 XML 布局、UI 字符串和位图图像。

磁盘上的 Android 项目结构与此扁平项目结构有所不同。要查看实际的项目文件结构，可以从 Project 下拉菜单（在图 1-1 中显示为 Android）选择 Project。

用户还可以自定义项目文件的视图，重点显示应用开发的特定方面。例如，选择项目的 Problems 视图会显示指向包含任何已识别编码和语法错误（如布局文件中缺少一个 XML 元素结束标记）的源文件链接，如图 1-2 所示。



图 1-1 Android 视图中的项目文件

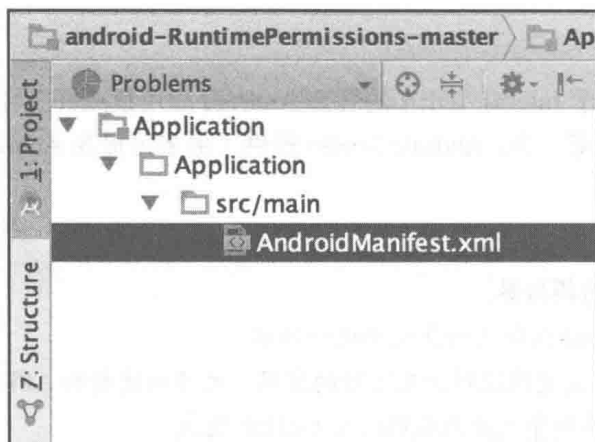


图 1-2 项目的 Problems 视图

1.1.2 Android Studio 主窗口

Android Studio 主窗口由如图 1-3 所示的几个逻辑区域组成。

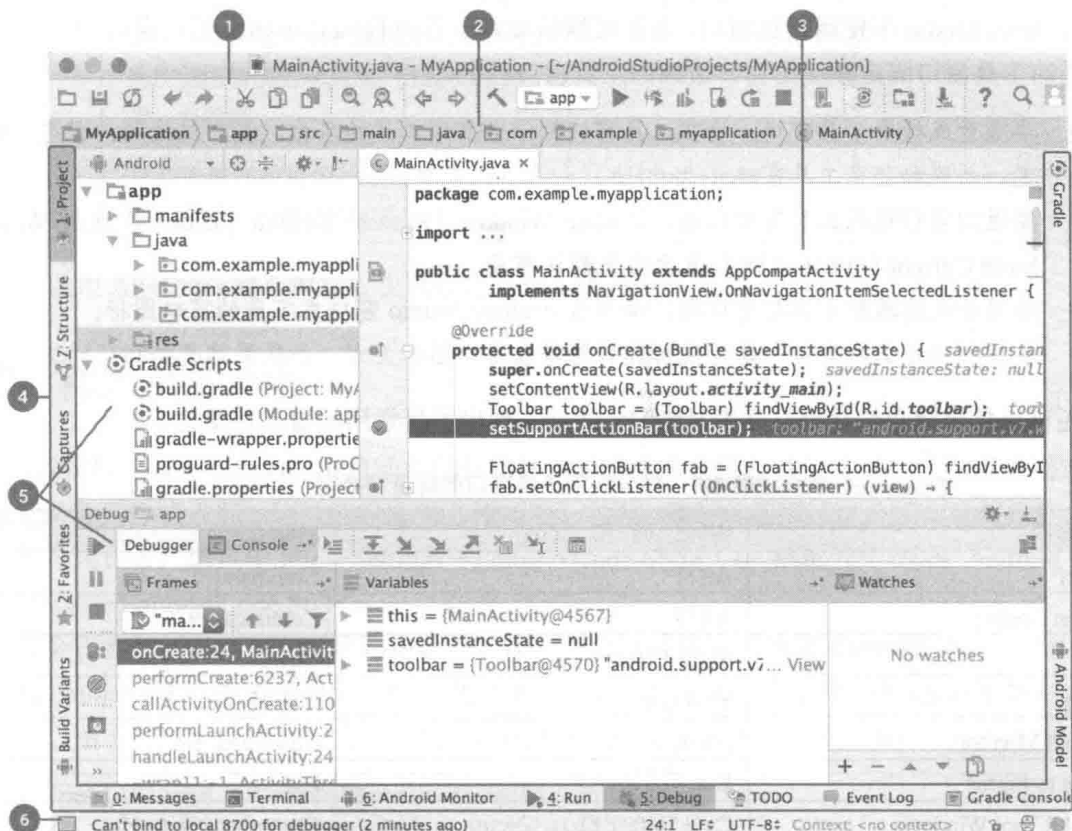


图 1-3 Android Studio 主窗口

① 工具栏，提供执行各种操作的工具，包括运行应用和启动 Android 工具。

② 导航栏，可以帮助在项目中导航，以及打开文件进行编辑。此区域提供 Project 窗口所示结构的精简视图。

③ 编辑器窗口，是创建和修改代码的区域。编辑器可能因当前文件类型的不同而有所差异。例如，在查看布局文件时，编辑器显示布局编辑器。

④ 工具窗口栏，在 IDE 窗口外部运行，并且包含可用于展开或折叠各个工具窗口的按钮。

⑤ 工具窗口，提供对特定任务的访问，例如项目管理、搜索和版本控制等。可以展开和折叠这些窗口。

⑥ 状态栏，显示项目和 IDE 本身的状态以及任何警告或消息。

用户可以通过隐藏或移动工具栏和工具窗口调整主窗口，以便留出更多屏幕空间，还可以使用键盘快捷键访问大多数 IDE 功能。

可以随时通过按两下 Shift 键或点击 Android Studio 窗口右上角的放大镜搜索源代码、数据库、操作和用户界面的元素等。此功能非常实用，例如在忘记如何触发特定 IDE 操作时，可以利用此功能进行查找。

1.1.3 工具窗口

Android Studio 不使用默认窗口，而是根据情境在工作时自动显示相关工具窗口。默认情况下，最常用的工具窗口固定在应用窗口边缘的工具窗口栏上。

- 要展开或折叠工具窗口，请在工具窗口栏中点击该工具的名称，还可以拖动、固定、取消固定、关联和分离工具窗口。
- 要返回当前默认工具窗口布局，请点击 Window→Restore Default Layout 或点击 Window→Store Current Layout as Default 自定义默认布局。
- 要显示或隐藏整个工具窗口栏，请点击 Android Studio 窗口左下角的窗口图标。
- 要找到特定工具窗口，请将鼠标指针悬停在窗口图标上方，并从菜单选择相应的工具窗口。

还可以使用键盘快捷键打开工具窗口。表 1-1 列出了最常用的窗口快捷键。

表 1-1 部分实用工具窗口的键盘快捷键

工具窗口	Windows 和 Linux	Mac
Project	Alt+1	Command+1
Version Control	Alt+9	Command+9
Run	Shift+F10	Ctrl+R
Debug	Shift+F9	Ctrl+D
Android Monitor	Alt+6	Command+6
Return to Editor	Esc	Esc
Hide All Tool Windows	Ctrl+Shift+F12	Command+Shift+F12

如果想要隐藏所有工具栏、工具窗口和编辑器选项卡，请点击 View→Enter Distraction Free Mode。此操作可启用无干扰模式。要退出“无干扰模式”，请点击 View→Exit Distraction Free Mode。

用户可以使用快速搜索在 Android Studio 中的大多数工具窗口中执行搜索和筛选。要使用快速搜索，请选择工具窗口，然后输入搜索查询。

1.1.4 代码自动完成

Android Studio 有三种自动补全代码快捷键，如表 1-2 所示。

表 1-2 代码自动完成的键盘快捷键

类型	说明	Windows 和 Linux	Mac
基本自动完成	显示对变量、类型、方法和表达式等的基本建议。如果连续两次调用基本自动完成，将显示更多结果，包括私有成员和非导入静态成员	Ctrl+空格	Control+空格

(续表)

类型	说明	Windows 和 Linux	Mac
智能自动完成	根据上下文显示相关选项。智能自动完成可识别预期类型和数据流。如果连续两次调用智能自动完成, 将显示更多结果, 包括链	Ctrl+Shift+空格	Control+Shift+空格
语句自动完成	自动完成当前语句, 添加缺失的圆括号、大括号、花括号和格式化等	Ctrl+Shift+Enter	Shift+Command+Enter

还可以按 Alt+Enter 组合键执行快速修复并显示建议的操作。

1.1.5 样式和格式化

在编辑时, Android Studio 将自动应用代码样式设置中指定的格式设置和样式。可以通过编程语言自定义代码样式设置, 包括指定选项卡和缩进、空格、换行、花括号以及空白行的约定。要自定义代码样式设置, 请点击 File→Settings→Editor→Code Style (在 Mac 上, 点击 Android Studio→Preferences→Editor→Code Style)。

IDE 会在你写代码时自动对代码进行格式化, 也可以通过按快捷键 Ctrl+Alt+L (在 Mac 上, 按 Opt+Command+L) 格式化代码、按快捷键 Ctrl+Alt+I (在 Mac 上, 按 Alt+Option+I*) 自动缩进所有行。图 1-4 (a) 是格式化之前的代码, 图 1-4 (b) 是格式化之后的代码。

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
}
```

(a)

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
}
// Get reference to the drawer layout and set event listener
```

Formatted 7 lines
Show reformat dialog: ⌘⇧L

(b)

图 1-4-2 格式化前后的代码

1.1.6 版本控制基础知识

Android Studio 支持多个版本控制系统 (VCS), 包括 Git、GitHub、CVS、Mercurial、Subversion