

《Android高级进阶》

图书基本信息

书名：《Android高级进阶》

13位ISBN编号：9787121298457

出版时间：2016-10

作者：顾浩鑫

页数：536

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《Android高级进阶》

内容概要

《Android高级进阶》是Android的进阶学习指南，主要为Android初中级开发者进阶所需的知识，高级开发者也可以从本书中发现很多共鸣点。《Android高级进阶》从8个方面对50个知识点进行分类讲解，包括基础篇、系统架构篇、经验总结篇、新技术篇、性能优化篇、移动安全篇、工具篇、测试篇。熟练掌握这些知识点后，应该就能够应付实际项目开发中的绝大部分问题了。

《Android高级进阶》的主要目的在于给读者一个完整的Android中高级开发者知识图谱。笔者希望通过《Android高级进阶》的系统讲解，能够帮助读者在面试和工作中收获自己满意的成绩。

书籍目录

第1篇 基础篇

第1章 Android触摸事件传递机制 2

- 1.1 触摸事件的类型 2
- 1.2 事件传递的三个阶段 3
- 1.3 View的事件传递机制 4
- 1.4 ViewGroup的事件传递机制 10

第2章 Android View的绘制流程 16

- 2.1 绘制的整体流程 17
- 2.2 MeasureSpec 17
- 2.3 Measure 19
- 2.4 Layout 22
- 2.5 Draw 22

第3章 Android 动画机制 25

- 3.1 逐帧动画 (Frame Animation) 25
 - 3.1.1 XML 资源文件方式 25
 - 3.1.2 代码方式 26
- 3.2 补间动画 (Tween Animation) 27
 - 3.2.1 插值器 Interpolator 27
 - 3.2.2 AlphaAnimation 29
 - 3.2.3 ScaleAnimation 30
 - 3.2.4 TranslateAnimation 31
 - 3.2.5 RotateAnimation 32
 - 3.2.6 自定义补间动画 34
- 3.3 属性动画 (Property Animation) 34
 - 3.3.1 Evaluator 35
 - 3.3.2 AnimatorSet 36
 - 3.3.3 ValueAnimator 36
 - 3.3.4 ObjectAnimator 38
- 3.4 过渡动画 (Transition Animation) 40

第4章 Support Annotation Library 使用详解 46

- 4.1 Nullness 注解 47
- 4.2 资源类型注解 48
- 4.3 类型定义注解 50
- 4.4 线程注解 52
- 4.5 RGB 颜色值注解 52
- 4.6 值范围注解 53
- 4.7 权限注解 53
- 4.8 重写函数注解 54
- 4.9 返回值注解 55
- 4.10 @VisibleForTesting 55
- 4.11 @Keep 55

第5章 Percent Support Library使用详解 57

第6章 Design Support Library使用详解 62

- 6.1 Snackbar 62
- 6.2 TextInputLayout 63
- 6.3 TabLayout 64
- 6.4 NavigationView 65

- 6.4.1 导航菜单 66
- 6.4.2 导航头部 67
- 6.5 FloatingActionButton 70
 - 6.5.1 使用浮动操作按钮 70
 - 6.5.2 其他选项 71
 - 6.5.3 点击事件 71
- 6.6 CoordinatorLayout 72
- 6.7 CollapsingToolbarLayout 73
- 6.8 BottomSheetBehavior 75
- 第7章 Android Studio中的NDK开发 77
 - 7.1 ABI的基本概念 77
 - 7.2 引入预编译的二进制 C/C++ 函数库 79
 - 7.3 直接从 C/C++ 源码编译 79
 - 7.3.1 配置 ndk.dir 变量 79
 - 7.3.2 在 Gradle 中配置 NDK 模块 79
 - 7.3.3 添加 C/C++ 文件到指定的目录 81
 - 7.4 使用 .so 文件的注意事项 81
 - 7.4.1 使用高平台版本编译的 .so 文件运行在低版本的设备上 81
 - 7.4.2 混合使用不同的C++ 运行时编译的 .so 文件 82
 - 7.4.3 没有为每个支持的 CPU 架构提供对应的 .so 文件 82
 - 7.4.4 将 .so 文件放在错误的地方 82
 - 7.4.5 只提供 armeabi 架构的 .so 文件而忽略其他 ABIs 的 83
- 第8章 Gradle 必知必会 85
 - 8.1 共享变量的定义 85
 - 8.2 通用配置 87
 - 8.3 aar 函数库的引用 88
 - 8.4 签名和混淆的配置 90
- 第9章 通过Gradle打包发布函数库到JCenter和Maven Central 92
 - 9.1 Maven Central 和 JCenter 92
 - 9.1.1 Maven Central 93
 - 9.1.2 JCenter 93
 - 9.2 Android Studio 获取函数库的原理 94
 - 9.3 上传函数库到 JCenter 96
 - 9.3.1 步骤一：在 Bintray 网站上注册一个账号 96
 - 9.3.2 步骤二：创建一个 Sonatype 账号 96
 - 9.3.3 步骤三：在 Bintray 网站使能自动签名 97
 - 9.3.4 步骤四：生成 POM 相关的信息 100
 - 9.3.5 步骤五：上传函数库到 Bintray 104
 - 9.3.6 步骤六：发布 Bintray 用户仓库到 JCenter 107
 - 9.3.7 步骤七：同步函数库到 Maven Central 109
- 第10章 Builder模式详解 110
 - 10.1 经典的 Builder 模式 110
 - 10.2 Builder 模式的变种 113
 - 10.3 变种 Builder 模式的自动化生成 119
 - 10.4 开源函数库的例子 122
- 第11章 注解在 Android 中的应用 124
 - 11.1 注解的定义 124
 - 11.2 标准注解 125
 - 11.2.1 编译相关注解 125

- 11.2.2 资源相关注解 125
- 11.2.3 元注解 125
- 11.3 运行时注解 127
- 11.4 编译时注解 127
 - 11.4.1 定义注解处理器 127
 - 11.4.2 注册注解处理器 131
 - 11.4.3 android-apt插件 132
- 第12章 ANR产生的原因及其定位分析 134
 - 12.1 ANR产生的原因 135
 - 12.2 典型的ANR问题场景 135
 - 12.3 ANR的定位和分析 136
 - 12.3.1 Logcat日志信息 136
 - 12.3.2 traces.txt日志信息 138
 - 12.4 ANR的避免和检测 141
 - 12.4.1 StrictMode 141
 - 12.4.2 BlockCanary 142
- 第13章 Android异步处理技术 144
 - 13.1 Thread 144
 - 13.2 HandlerThread 146
 - 13.3 AsyncQueryHandler 149
 - 13.4 IntentService 150
 - 13.5 Executor Framework 153
 - 13.6 AsyncTask 155
 - 13.7 Loader 156
 - 13.8 总结 159
- 第14章 Android数据序列化方案研究 160
 - 14.1 Serializable 160
 - 14.2 Parcelable 166
 - 14.3 SQLiteDatabase 169
 - 14.4 SharedPreferences 170
 - 14.5 JSON 171
 - 14.6 Protocol Buffers及Nano-Proto-Buffers 171
 - 14.7 FlatBuffers 171
- 第15章 Android WebView Java 和 JavaScript 交互详解 173
 - 15.1 Java 调用 JavaScript 173
 - 15.2 JavaScript 调用 Java 174
- 第2篇 系统架构篇
- 第16章 MVP模式及其在Android中的实践 180
 - 16.1 MVP的基本概念 180
 - 16.2 MVP与MVC的区别 181
 - 16.3 MVP的开源实现 182
 - 16.3.1 Android-Architecture 182
 - 16.3.2 TODO-MVP 182
 - 16.3.3 TODO-MVP-Loaders 183
 - 16.3.4 TODO-MVP-Clean 183
 - 16.3.5 TODO-Databinding 184
 - 16.3.6 其他开源参考实现 184
 - 16.4 MVP的好处 185
 - 16.5 MVP存在的问题 185

- 第17章 MVVM模式及Android DataBinding实战 186
 - 17.1 Data Binding 表达式 187
 - 17.2 数据对象 188
 - 17.3 数据绑定 188
 - 17.4 事件绑定 189
- 第18章 观察者模式的拓展：事件总线 191
 - 18.1 为何要使用 191
 - 18.2 原理 192
 - 18.3 开源实现 193
 - 18.3.1 EventBus 193
 - 18.3.2 otto 194
 - 18.4 与观察者模式及 Android 广播的区别 196
- 第19章 书写简洁规范的代码 197
 - 19.1 Java 编码规范 197
 - 19.1.1 源代码文件的定义 197
 - 19.1.2 源代码文件的结构 197
 - 19.1.3 遵循的格式 198
 - 19.1.4 命名约定 200
 - 19.1.5 Javadoc 200
 - 19.2 Android 命名规范 200
 - 19.2.1 布局文件的命名 200
 - 19.2.2 资源文件的命名 201
 - 19.2.3 类的命名 201
 - 19.3 CheckStyle 的使用 202
- 第20章 基于开源项目搭建属于自己的技术堆栈 203
 - 20.1 APP 的整体架构 203
 - 20.2 技术选型的考量点 205
 - 20.3 日志记录能力 205
 - 20.4 JSON 解析能力 207
 - 20.4.1 gson 207
 - 20.4.2 jackson 207
 - 20.4.3 Fastjson 208
 - 20.4.4 LoganSquare 208
 - 20.5 数据库操作能力 210
 - 20.5.1 ActiveAndroid 210
 - 20.5.2 ormlite 211
 - 20.5.3 greenDAO 211
 - 20.5.4 Realm 212
 - 20.6 网络通信能力 213
 - 20.6.1 android-async-http 213
 - 20.6.2 OkHttp 215
 - 20.6.3 Volley 216
 - 20.6.4 Retrofit 217
 - 20.7 图片缓存和显示能力 217
 - 20.7.1 BitmapFun 218
 - 20.7.2 Picasso 218
 - 20.7.3 Glide 218
 - 20.7.4 Fresco 219
 - 20.7.5 Android-Universal-Image-Loader 219

第3篇 经验总结篇

第21章 64K方法数限制原理与解决方案 222

21.1 64K 限制的原因 222

21.2 使用 MultiDex 解决 64K 限制的问题 223

21.2.1 Android 5.0 之前的版本 223

21.2.2 Android 5.0 及之后的版本 223

21.3 如何避免出现 64K 限制 223

21.4 配置 MultiDex 224

21.5 MultiDex Support Library 的局限性 226

21.6 在开发阶段优化 MultiDex 的构建 227

第22章 Android 插件框架机制研究与实践 230

22.1 基本概念 231

22.1.1 宿主和插件 231

22.1.2 ClassLoader 机制 231

22.2 开源框架 231

22.2.1 android-pluginmgr 232

22.2.2 dynamic-load-apk 232

22.2.3 DynamicAPK 232

22.2.4 DroidPlugin 233

22.2.5 Small 234

第23章 推送机制实现原理详解 235

23.1 推送的开源实现方案 236

23.1.1 基于 XMPP 协议 236

23.1.2 基于 MQTT 协议 236

23.2 推送的第三方平台 236

23.3 自己实现推送功能 237

23.3.1 长连接的建立 (TCPConnectThread) 237

23.3.2 数据的发送 (TCPSendThread) 237

23.3.3 数据的接收 (TCPReceiveThread) 238

23.3.4 心跳包的实现 (TCPHeartBeatThread) 240

第24章 APP 瘦身经验总结 241

24.1 APP 为什么变胖了 241

24.2 从 APK 文件的结构说起 242

24.3 优化图片资源占用的空间 245

24.3.1 无损压缩 [ImageOptim] 246

24.3.2 有损压缩 [ImageAlpha] 246

24.3.3 有损压缩 [TinyPNG] 246

24.3.4 PNG/JPEG 转换为 WebP 246

24.3.5 尽量使用 NinePatch 格式的 PNG 图 247

24.4 使用 Lint 删除无用资源 248

24.5 利用 Android Gradle 配置 248

24.5.1 minifyEnable 248

24.5.2 shrinkResources 249

24.5.3 resConfigs 249

24.5.4 ndk.abiFilters 250

24.6 重构和优化代码 250

24.7 资源混淆 251

24.8 插件化 251

第25章 Android Crash 日志收集原理与实践 252

- 25.1 Java 层 Crash 捕获机制 253
 - 25.1.1 基本原理 253
 - 25.1.2 线程信息 254
 - 25.1.3 SharedPreferences 信息 255
 - 25.1.4 系统设置 257
 - 25.1.5 Logcat 中的日志记录 261
 - 25.1.6 自定义 Log 文件中的内容 264
 - 25.1.7 MemInfo 信息 266
- 25.2 Native 层 Crash 捕获机制 267
- 25.3 Crash 的上报 269
- 第4篇 新技术篇
- 第26章 函数式编程思想及其在Android中的应用 272
 - 26.1 代码的简化 274
 - 26.2 Operators 简介 275
- 第27章 依赖注入及其在Android中的应用 277
 - 27.1 基本概念 277
 - 27.1.1 构造函数注入 278
 - 27.1.2 Setter 函数注入 279
 - 27.1.3 接口注入 279
 - 27.2 为何需要框架 280
 - 27.3 开源框架的选择 280
 - 27.3.1 ButterKnife 280
 - 27.3.2 RoboGuice 282
 - 27.3.3 Dagger 285
 - 27.3.4 Dagger2 288
 - 27.3.5 框架的对比 289
- 第28章 Android世界的Swift : Kotlin在Android中的应用 290
 - 28.1 选择 Kotlin 的原因 290
 - 28.2 Kotlin 的安装和配置 291
 - 28.3 Kotlin 语言的特性 292
 - 28.3.1 可表达性 292
 - 28.3.2 空类型安全 294
 - 28.3.3 扩展函数 295
 - 28.4 Kotlin 的 Gradle 配置 296
 - 28.5 将 Java 类转换成 Kotlin 类 299
 - 28.6 相关资料 302
- 第29章 React Native For Android入门指南 304
 - 29.1 环境配置 304
 - 29.1.1 Homebrew 304
 - 29.1.2 nvm 305
 - 29.1.3 Node.js 305
 - 29.1.4 watchman 306
 - 29.1.5 flow 306
 - 29.2 Android 开发环境的要求 306
 - 29.3 React Native 工程配置 307
 - 29.3.1 安装react-native 307
 - 29.3.2 生成工程 307
 - 29.4 Android Studio 工程概览 308
 - 29.5 React Native 依赖库修改为本地 314

- 29.5.1 下载 react-native.aar 314
- 29.5.2 react-native.aar 的文件内容 315
- 29.5.3 Gradle 本地依赖 316
- 29.5.4 将 node_modules 上传到 svn/git 318
- 29.6 React Native 学习建议 319
- 第30章 Android在线热修复方案研究 320
 - 30.1 在线热修复的基本流程 320
 - 30.2 Dexposed 321
 - 30.2.1 如何集成 322
 - 30.2.2 基本用法 323
 - 30.2.3 在线热修复 325
 - 30.2.4 平台的限制 328
 - 30.3 AndFix 329
 - 30.3.1 如何集成 329
 - 30.3.2 补丁包生成工具 331
 - 30.3.3 平台的限制 332
 - 30.4 Nuwa 332
 - 30.4.1 基本原理 332
 - 30.4.2 如何集成 333
 - 30.4.3 补丁生成工具 334
 - 30.4.4 平台的限制 334
 - 30.5 总结 334
- 第31章 面向切面编程及其在Android中的应用 335
 - 31.1 AOP 的基本概念 335
 - 31.2 代码织入的时机 336
 - 31.3 基于 AspectJ 实现 Android 平台的 AOP 337
 - 31.3.1 Hugo 的用法简介 337
 - 31.3.2 Hugo 的实现原理 339
 - 31.4 其他 AOP 开源框架 344
- 第32章 基于Facebook Buck改造Android构建系统 345
 - 32.1 Buck环境配置 346
 - 32.1.1 Homebrew 方式 346
 - 32.1.2 手动构建方式 346
 - 32.1.3 安装 Watchman 348
 - 32.1.4 安装 Android SDK 和 Android NDK 348
 - 32.2 快速创建基于 Buck 构建的 Android 工程 349
 - 32.3 Buck 的基本概念 351
 - 32.3.1 构建规则 (Build Rule) 352
 - 32.3.2 构建目标 (Build Target) 354
 - 32.3.3 构建文件 (Build File) 355
 - 32.3.4 构建目标模式 (Build Target Pattern) 356
 - 32.4 项目改造实战 357
 - 32.4.1 步骤一：手动下载工程依赖的第三方 Jar包或者aar包 357
 - 32.4.2 步骤二：将 R.* 常量修改为非 final 的 357
 - 32.4.3 步骤三：创建 BUCK 文件 358
 - 32.4.4 步骤四：编译 Buck 的 buck-android-support 363
 - 32.4.5 步骤五：Exopackage 的使用 363
 - 32.5 Buck 的自动化改造 366
- 第5篇 性能优化篇

- 第33章 代码优化 368
 - 33.1 数据结构的选择 368
 - 33.2 Handler 和内部类的正确用法 370
 - 33.3 正确地使用 Context 373
 - 33.3.1 Context 的种类 374
 - 33.3.2 错误使用 Context 导致的内存泄漏 374
 - 33.3.3 不同 Context 的对比 376
 - 33.4 掌握 Java 的四种引用方式 376
 - 33.5 其他代码微优化 377
 - 33.5.1 避免创建非必要的对象 377
 - 33.5.2 对常量使用 static final 修饰 378
 - 33.5.3 避免内部的 Getters/Setters 378
 - 33.5.4 代码的重构 378
- 第34章 图片优化 379
 - 34.1 图片的格式 379
 - 34.1.1 JPEG 380
 - 34.1.2 PNG 380
 - 34.1.3 GIF 380
 - 34.1.4 WebP 380
 - 34.2 图片的压缩 380
 - 34.2.1 无损压缩 ImageOptim 381
 - 34.2.2 有损压缩 ImageAlpha 381
 - 34.2.3 有损压缩 TinyPNG 381
 - 34.2.4 PNG/JPEG 转换为 WebP 381
 - 34.2.5 尽量使用 NinePatch 格式的 PNG 图 382
 - 34.3 图片的缓存 382
- 第35章 电量优化 383
 - 35.1 BroadcastReceiver 383
 - 35.2 数据传输 384
 - 35.3 位置服务 384
 - 35.4 AlarmManager 386
 - 35.5 WakeLock 386
- 第36章 布局优化 388
 - 36.1 include 标签共享布局 388
 - 36.2 ViewStub 标签实现延迟加载 389
 - 36.3 merge 标签减少布局层次 391
 - 36.4 尽量使用 CompoundDrawable 392
 - 36.5 使用 Lint 393
- 第37章 网络优化 395
 - 37.1 避免 DNS 解析 395
 - 37.2 合并网络请求 395
 - 37.3 预先获取数据 396
 - 37.4 避免轮询 396
 - 37.5 优化重连机制 396
 - 37.6 离线缓存 396
 - 37.7 压缩数据大小 396
 - 37.8 不同的网络环境使用不同的超时策略 397
 - 37.9 CDN 的使用 397
- 第6篇 移动安全篇

- 第38章 Android混淆机制详解 400
 - 38.1 Java 代码的混淆 400
 - 38.1.1 Proguard 的特性 401
 - 38.1.2 Proguard 的使能和配置 401
 - 38.1.3 proguard-rules.pro 文件的编写 404
 - 38.1.4 Proguard 生成的文件 407
 - 38.1.5 Proguard 混淆规则汇总 409
 - 38.2 Native (C/C++) 代码的混淆 409
 - 38.3 资源文件的混淆 409
- 第39章 Android反编译机制详解 411
 - 39.1 资源文件的反编译 411
 - 39.1.1 ApkTool 的安装 411
 - 39.1.2 ApkTool 的使用 412
 - 39.2 Java 代码的反编译 413
- 第40章 客户端敏感信息隐藏技术研究 414
 - 40.1 敏感信息嵌套在 strings.xml 中 415
 - 40.2 敏感信息隐藏在 Java 源代码中 415
 - 40.3 敏感信息隐藏在 BuildConfig 中 417
 - 40.4 使用 DexGuard 418
 - 40.5 对敏感信息进行伪装或者加密 419
 - 40.6 敏感信息隐藏在原生函数库中 (.so文件) 419
 - 40.7 对APK进行加固处理 419
- 第41章 Android 加固技术研究 421
 - 41.1 爱加密的主要功能 421
 - 41.1.1 漏洞分析 421
 - 41.1.2 加密服务 422
 - 41.1.3 渠道监测 423
 - 41.2 常见 APP 漏洞及风险 423
 - 41.2.1 静态破解 423
 - 41.2.2 二次打包 424
 - 41.2.3 本地储存数据窃取 424
 - 41.2.4 界面截取 424
 - 41.2.5 输入法攻击 424
 - 41.2.6 协议抓取 424
 - 41.3 Android 程序反破解技术 424
 - 41.3.1 对抗反编译 424
 - 41.3.2 对抗静态分析 425
 - 41.3.3 对抗动态调试 425
 - 41.3.4 防止重编译 425
 - 41.4 加固技术研究知识储备 426
 - 41.4.1 掌握常见的破解分析工具 426
 - 41.4.2 掌握 Dalvik 指令集代码 428
 - 41.4.3 掌握 Dex 和 Odex 文件格式 428
 - 41.4.4 掌握 Smali 文件格式 428
 - 41.4.5 掌握基于 Android 的 ARM 汇编语言基础 428
- 第42章 Android安全编码 429
 - 42.1 WebView 远程代码执行 429
 - 42.2 WebView 密码明文保存 430
 - 42.3 Android 本地拒绝服务 431

- 42.3.1 非法序列化对象导致的 ClassNotFoundException 431
- 42.3.2 空 Action 导致的 NullPointerException 432
- 42.3.3 强制类型转换导致的 ClassCastException 433
- 42.3.4 数组越界导致的 IndexOutOfBoundsException 433
- 42.4 SharedPreferences 全局任意读写 434
- 42.5 密钥硬编码 434
- 42.6 AES/DES/RSA 弱加密 434
- 42.7 随机函数使用错误 437
- 42.8 WebView 忽略 SSL 证书 438
- 42.9 HTTPS 证书弱校验 438
 - 42.9.1 自定义 X509TrustManager 未实现安全校验 438
 - 42.9.2 自定义 HostnameVerifier 默认接受所有域名 441
 - 42.9.3 SSLSocketFactory 信任所有证书 442
- 42.10 PendingIntent 使用不当 443
- 第7篇 工具篇
- 第43章 Android调试工具Facebook Stetho 446
 - 43.1 视图布局监视 447
 - 43.2 数据库监视 447
 - 43.3 网络监视 448
 - 43.3.1 网络模块使用的是 HttpURLConnection 449
 - 43.3.2 网络模块使用的是 OkHttp 452
 - 43.4 dumpapp 454
 - 43.4.1 插件的编写 454
 - 43.4.2 插件的集成 456
 - 43.4.3 插件的使用 456
 - 43.5 Javascript 控制台 457
 - 43.6 最佳实践 457
- 第44章 内存泄漏检测函数库 LeakCanary 460
 - 44.1 基本概念 461
 - 44.2 LeakCanary 的集成 461
 - 44.3 LeakCanary 的原理 465
 - 44.4 LeakCanary 的定制 469
 - 44.4.1 RefWatcher 的自定义 469
 - 44.4.2 通知页面样式的自定义 470
 - 44.4.3 内存泄漏堆栈信息保存个数的自定义 471
 - 44.4.4 Watcher 的延时 471
 - 44.4.5 自定义内存泄漏堆栈信息和 heap dump 的处理方式 471
 - 44.4.6 忽略特定的弱引用 472
 - 44.4.7 不监视特定的 Activity 类 472
- 第45章 基于Facebook Redex实现Android APK的压缩和优化 474
 - 45.1 转换的时机 474
 - 45.2 管道的思想 475
 - 45.3 减少字节码的意义 475
 - 45.4 混淆和压缩 475
 - 45.5 使用内联函数 476
 - 45.6 无用代码的消除 477
 - 45.7 Redex 的集成和使用 478
 - 45.7.1 依赖的安装 478
 - 45.7.2 下载, 构建和安装 478

- 45.7.3 使用 478
- 第46章 Android Studio你所需要知道的功能 479
 - 46.1 Annotate 479
 - 46.2 .ignore 插件 480
 - 46.3 Live Templates 481
 - 46.4 集成 Bug 管理系统 482
- 第8篇 测试篇
- 第47章 Android单元测试框架简介 486
 - 47.1 Java 单元测试框架JUnit 486
 - 47.2 Android 单元测试框架Robolectric 3.0 488
 - 47.3 Java 模拟测试框架Mockito 490
 - 47.3.1 行为的验证 490
 - 47.3.2 Stub (桩函数) 的使用 491
- 第48章 Android UI自动化测试框架简介 492
 - 48.1 Monkey 492
 - 48.2 MonkeyRunner 493
 - 48.3 UIAutomator 493
 - 48.4 Robotium 494
 - 48.5 Espresso 494
 - 48.6 Appium 494
- 第49章 Android静态代码分析实战 495
 - 49.1 Java代码规范检查工具CheckStyle 495
 - 49.1.1 Gradle方式 495
 - 49.1.2 Android Studio插件方式 497
 - 49.2 Java静态代码分析工具FindBugs 498
 - 49.2.1 Gradle方式 498
 - 49.2.2 Android Studio插件方式 499
 - 49.3 Java静态代码分析工具PMD 500
 - 49.3.1 Gradle方式 500
 - 49.3.2 Android Studio插件方式 501
 - 49.4 Android代码优化工具Lint 501
 - 49.4.1 Gradle方式 501
 - 49.4.2 Android Studio插件方式 502
- 第50章 基于Jenkins + Gradle搭建Android持续集成编译环境 503
 - 50.1 Tomcat的下载和启动 503
 - 50.2 Jenkins的下载和运行 505
 - 50.3 Jenkins插件的安装 506
 - 50.4 Jenkins全局配置 507
 - 50.4.1 配置JDK环境 507
 - 50.4.2 配置Android SDK环境 507
 - 50.4.3 配置Git环境 508
 - 50.4.4 配置SVN环境 508
 - 50.4.5 配置Gradle环境 508
 - 50.5 JOB相关的操作 508
 - 50.5.1 JOB的创建 508
 - 50.5.2 JOB的配置 509
 - 50.5.3 Gradle的配置 510
 - 50.5.4 构建触发器的配置 511
 - 50.5.5 参数化构建 514

50.6 Jenkins预定义的环境变量 514

《Android高级进阶》

精彩短评

- 1、 读书馆借来的，本来打算一个月看完的。结果一天就看完了。内容不是一般的敷衍。
- 2、 how-can-i-detect-copy-pasted-code-using-git.....
- 3、 虽然作者对各种技术细节都没有详细深入的讲解，但是却给读者在安卓实际开发方面提供了大量的信息，可见作者平时一直也在关注各种新技术。这本书买来时60多元，光凭借这些信息，完全值回书价了。安卓方面这类书籍实现是少，大部分是入门级书籍，作者能够填充这块的空白，直接5星！

《Android高级进阶》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com