

第 1 章

Android 应用开发概述及 开发环境

随着 3G 智能手机时代的到来，人们对 Android 应用开发的需求日趋增多。Android 作为智能手机的操作系统，是新一代基于 Linux 的开源手机操作系统。手机应用软件的开发方式和环境与传统的 Windows 应用程序或者 Web 程序都有很大的不同。本章学习要点如下：

- 掌握 Android 系统的软件架构；
- 掌握搭建 Android 应用开发环境的方法；
- 掌握 Android SDK Manager 的作用；
- 掌握 Android 模拟器的使用；
- 掌握部署 Android 应用到 Android 设备中运行的方法。

1.1 移动开发与智能手机

1.1.1 移动开发概述

Android 一词最早出现于法国作家利尔亚当于 1886 年发表的科幻小说《未来夏娃》中，将外表像人的机器命名为 Android，它是一个全身绿色的机器人。

作为手机操作系统的 Android，是由安卓之父安迪·罗宾 (Andy Rubin) 研发完成的。Google 公司于 2005 年收购了原 Android 公司，并于 2007 年 11 月发布了基于 Linux 的开源手机平台。

最早的手机使用模拟信号，主要作用是移动电话。后来的手机使用数字信号，不仅具有移动电话功能，还可以发送短信。

1.1.2 Android 智能手机的使用特点

Android 智能手机，除了具备模拟手机打电话、发短信、蓝牙、上网等基本功能外，还具有用户定制操作系统的功能，可以像普通的计算机一样，安装或卸载应用程序。

智能手机本质上也是一台计算机，但与普通计算机有一定的差别。普通计算机的键盘、鼠标对应于较多的操作（如翻页、双击等），而手机支持各种手势对应的事件（如长

按等)。智能手机与计算机的差别如下:

- 手机只有用于返回桌面的 Home 键和退出主界面或返回到上一级界面的返回键;
- 手机的用户操作可分为按键和触屏两种。触屏事件(如滑屏、长按等)是 Android 所特有的;
- 手机进入文本编辑时,使用的是软键盘(不同于普通计算机);
- 手机系统集成了众多的硬件,如摄像头、录音机、GPS 芯片、蓝牙芯片、WiFi 网卡等。

手机的存储系统分为运行内存、手机内存和扩展存储三部分。其中,手机内存主要指系统区(包括最底层的 Linux 系统、自带的应用程序和用户应用程序)。此外,手机厂商通常会从手机内存中划分一部分存储用户数据(如照片、音乐等),即标准 SD 卡。

Android 手机的软件系统包括操作系统、中间件和一些主要应用,是基于 Java 系统,运行在 Linux 2.6 内核上的。此外,Android 手机还具有如下特点:

- Android SDK 提供多种开发所必需的工具与 API,如提供访问硬件的 API 函数,简化了摄像头、GPS 等硬件的访问过程;
- 具有自己的运行环境和虚拟机 Dalvik;
- 提供丰富的界面控件功能,加快用户界面的开发速度,保证 Android 平台上程序界面的一致性;
- 提供轻量级的进程间通信机制 Intent,使跨进程组件通信和发送系统级广播成为可能,提供了 Service 作为无用户界面、长时间后台运行的组件;
- 支持高效、快速的数据存储方式。

1.1.3 手机智能操作系统及分类

早期的手机没有操作系统 OS,内部所有的软件都是由生产商在设计时定制的,手机在设计完成后基本没有扩展功能。

为了提高手机的可扩展性,很多手机都使用了专为移动设备开发的操作系统 OS,使用者可根据需要安装不同类型的软件。

智能手机制造商所使用的半导体芯片并不都是相同的,不同的手机所采用的操作系统也可能不同。目前,主流的手机操作系统有如下几种。

- iPhone OS: 由苹果公司开发的手机操作系统。
 - Android: 由 Google 公司发布的基于 Linux 的开源手机平台。
 - Symbian: 由 Symbian 公司开发和维护,后被诺基亚公司收购。该操作系统不开源。
- 注意: Android 手机应用开发是移动开发的一种。

1.2 Android 系统架构

Android 是基于 Linux 内核的软件平台和操作系统,采用了软件堆栈架构。该架构分

为四层，即应用程序层、应用程序框架层、Android 运行时环境层和 Linux 内核层，如图 1.2.1 所示。

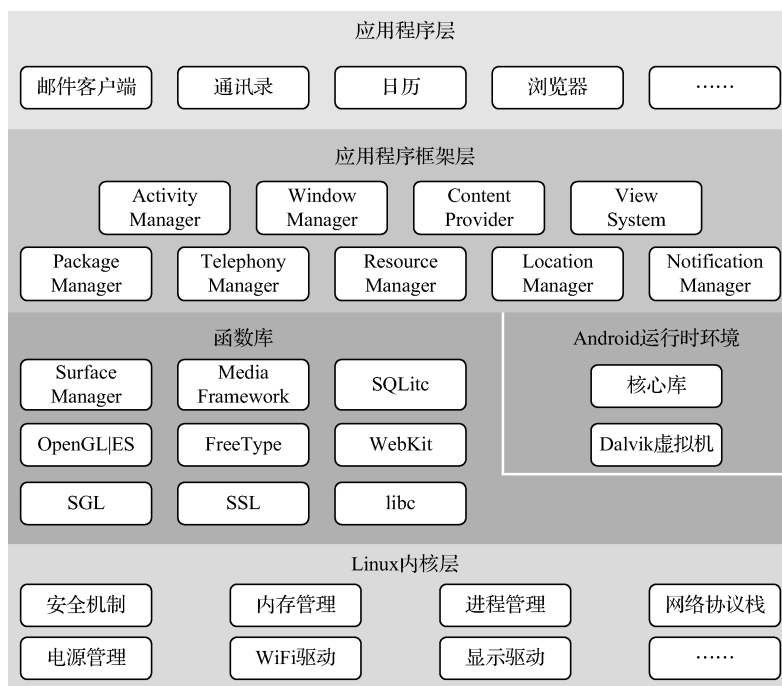


图 1.2.1 Android 系统软件架构

1.2.1 Linux 内核层

Linux 内核是硬件和其他软件堆层之间的一个抽象隔离层，提供由操作系统内核管理的底层基础功能，主要有安全机制、内存管理、进程管理、网络协议栈和驱动程序等。

Android 内核挂载/nfsroot/Androidfs 之后，根据 init.rc 和 init.goldfish.rc 进行初始化并装载系统库、程序等操作直到开机完成。init.rc 包括了文件系统初始化、装载的许多过程。init.rc 的主要工作如下：

- 设置环境变量；
- 创建 system、sdcard、data、cache 等目录；
- 把文件系统 mount 到目录，如 mount tmpfs tmpfs /sqlite_stmt_journals；
- 设置文件的用户群组、权限；
- 设置线程参数；
- 设置 TCP 缓存大小。

Android 源码编译后可得到 system.img、ramdisk.img 和 userdata.img 映像文件。其中，ramdisk.img 是 emulator 的文件系统，system.img 包括主要的包、库等文件，userdata.img 包括一些用户数据。emulator 加载这 3 个映像文件后，会把 system 和 userdata 分别加载到 ramdisk 文件系统中的 system 和 userdata 目录中。

注意：

(1) Android 手机内部存储的文件系统分区格式与 Linux 系统相同，而外部存储 SD 卡则采用 FAT。

(2) 每个 Linux 文件都具有 4 种访问权限：可读 (r)、可写 (w)、可执行 (x) 和无权限 (-)。

(3) 目录或文件权限共有 10 位，第 1 位表示文件、目录和超链接（分别用 -、d 和小写字母 l 表示）；第 2~4 位表示文件所有者的权限；第 5~7 位表示文件所有者所属组成员的权限；第 8~10 位表示所有者所属组之外的用户权限。

1.2.2 函数库和 Android 运行时环境层

函数库和 Android 运行时环境是第二层，位于 Linux 内核之上，也称中间层，由函数库和 Android 运行时环境构成。

由于 Linux 操作系统的内核使用及其组件使用 C 语言编写（少部分使用汇编语言），因此，开发人员可以通过应用程序框架调用一组基于 C/C++ 的函数库，主要包括以下几个。

- **Surface Manager**：支持显示子系统的访问，为多个应用程序提供 2D、3D 图像层的平滑连接。
- **Media Framework**：基于 OpenCORE 的多媒体框架，实现音频、视频的播放与录制功能。
- **SQLite**：关系型数据库引擎。
- **OpenGL | ES**：基于硬件的 3D 图像加速。
- **FreeType**：位图与矢量字体渲染。
- **WebKit**：Web 浏览器引擎。
- **SSL**：数据加密与安全传输的函数库。
- **libc**：标准 C 运行库。它是 Linux 系统中底层的应用程序开发接口。

Android 运行时环境由核心库和 Dalvik 虚拟机构成。核心库为开发人员提供了 Android 系统的特有函数功能和 Java 语言的基本函数功能；Dalvik 虚拟机采用适合内存和处理器受限的专用格式。

1.2.3 应用程序框架层

应用程序框架层提供了 Android 平台的管理功能和组件重用机制，包括活动管理器 (Activity Manager)、窗口管理器 (Window Manager)、内容提供者 (Content Provider)、视图系统 (View System)、包管理器 (Package Manager)、通信管理器 (Telephony Manager)、资源管理器 (Resource Manager)、位置管理器 (Location Manager) 和通知管理器 (Notification Manager)。Android 的三大核心功能如下。

- (1) **View System**：提供绘制图形，处理触摸、按键事件等功能。
- (2) **Activity Manager**：提供管理所有应用程序的 Activity 功能。
- (3) **Window Manager**：提供为所有应用程序分配窗口，并管理这些窗口的功能。

1.2.4 应用程序层

应用程序层提供了一系列核心应用程序，如浏览器、通讯录、相册、地图和电子市场等。

1.3 Android Studio 开发环境

1.3.1 Android Studio 概述

Android Studio 是一项全新的基于 IntelliJ IDEA 的 Android 集成开发和调试环境，与 Eclipse Android 环境相比，具有如下优点：

- IntelliJ IDEA / Android Studio 的智能提示很强大；
- Android Studio 内置终端，方便以命令行方式操作；
- 布局代码与效果的实时（同步）预览；
- 软件版本的联机更新；
- 项目基于 Gradle 的构建支持；
- 不仅提供了大量的组合键，还有众多快捷的设计工具、选择卡。

使用 Android Studio 开发，推荐的计算机硬件配置要求如下：

- Intel i5 以上处理器；
- 8G 及以上内存；
- 128G 及以上固态硬盘（具有较快的启动速度）。

1.3.2 下载和安装 Android Studio 3.1.2

访问 Android Studio 中文社区 <http://www.android-studio.org>，可以找到 Android Studio 3.1.2 的下载链接。Android Studio 3.1.2 安装分为两个阶段，首先安装 IDE（Android Studio），然后再安装 Android SDK。当 IDE 安装完成后，出现的对话框如图 1.3.1 所示。

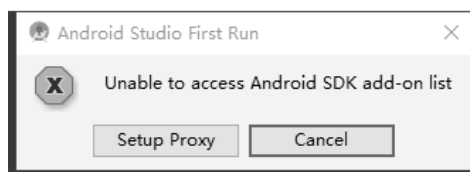


图 1.3.1 Android Studio 首次运行

注意：

(1) Android Studio 相当于 Java 或 Java Web 开发中的 eclipse。

(2) Android SDK 提供了 Android 开发的软件包。

单击 Cancel 按钮，将出现 Android SDK 的安装向导，如图 1.3.2 所示。

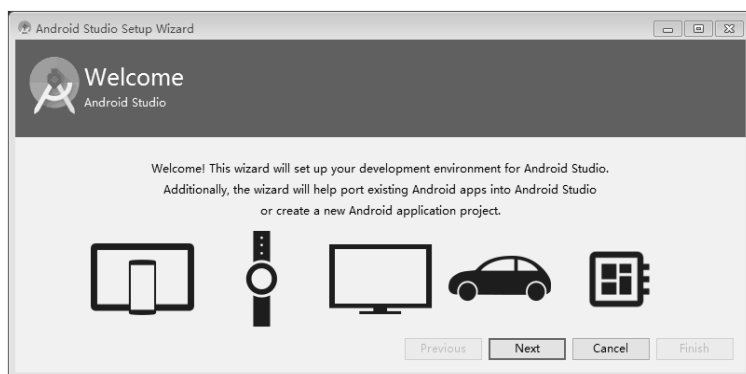


图 1.3.2 Android Studio SDK 安装向导

Android Studio 安装完成后，首次打开时呈现的主菜单，如图 1.3.3 所示。

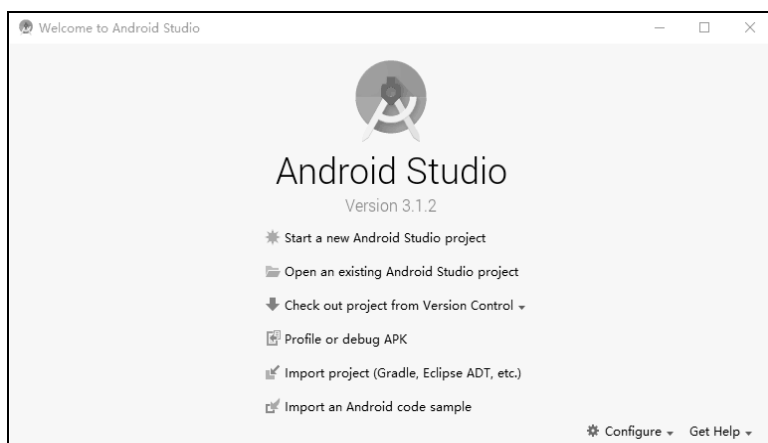


图 1.3.3 Android Studio 主菜单

注意：打开 Android Studio 时，将会自动打开最后一次创建的项目。

1.3.3 Android Studio 相关文件夹

Android Studio 安装完成后，自动创建的几个主要文件夹如下。

- AS 安装位置 C:\Program Files\Android\Android Studio。
- AS 自带 JRE(1.8) C:\Program Files\Android\Android Studio\jre。
- Android SDK C:\Users\×××\AppData\Local\Android\sdk。
- 新建 Android 项目的保存位置 C:\Users\×××\AndroidStudioProjects，其中，×××为 Windows 安装时设定的用户名。

1.3.4 Android SDK 与 Android API

1. Android SDK

Android SDK 是 Android 软件开发包(Software Development Kit)，它提供了在 Windows/

Linux/MAC 平台上开发 Android 应用的组件和各种工具集。工具集不仅包括了 Android 模拟器，还有用来调试、打包和在 Android 设备上安装应用的工具。

在 Android Studio 主菜单中，单击 **Configure**→**Project Defaults**→**Project Structure**，出现的对话框包含了 Android SDK 的位置和内置 JDK 的位置信息，如图 1.3.4 所示。

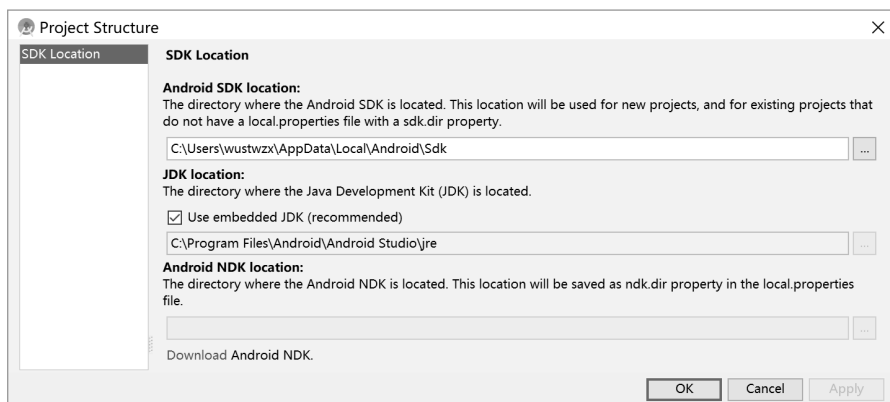


图 1.3.4 Android Studio 配置信息 (1)

在 Android Studio 主菜单中，单击 **Configure**→**SDK Manager**，出现管理 SDK 平台及工具更新（主要是下载和卸载）的对话框，如图 1.3.5 所示。

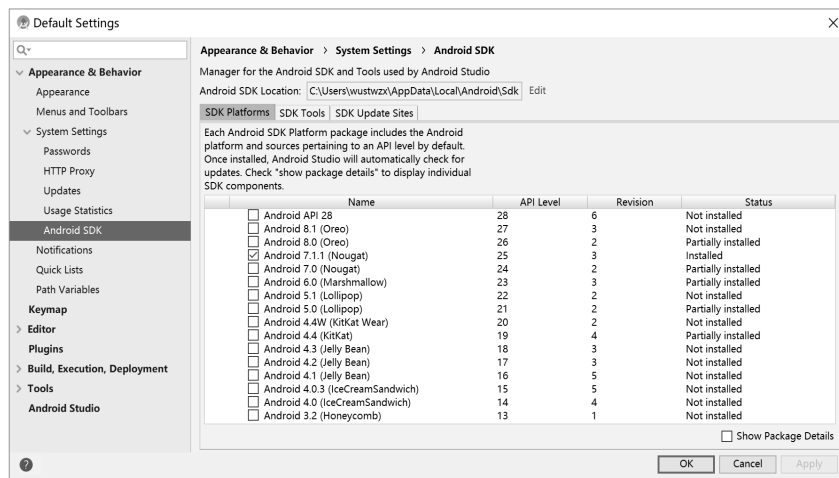



图 1.3.5 Android Studio 配置信息 (2)

注意：

(1) 在 Android Studio 中打开某个项目后，其工具栏包含了 SDK Manager 工具 。

(2) 在 Android Studio 中如果不关联 Android SDK，则无法开发 Android 应用程序。

在 Android SDK 文件夹里，文件夹 **platforms** 是主体，它是各版本开发组件的集合，包括 **android.jar**、字体、**res** 资源、模板等内容。其中，**android.jar** 文件提供了用于开发 Android 应用程序的编程接口 (API)，如图 1.3.6 所示。

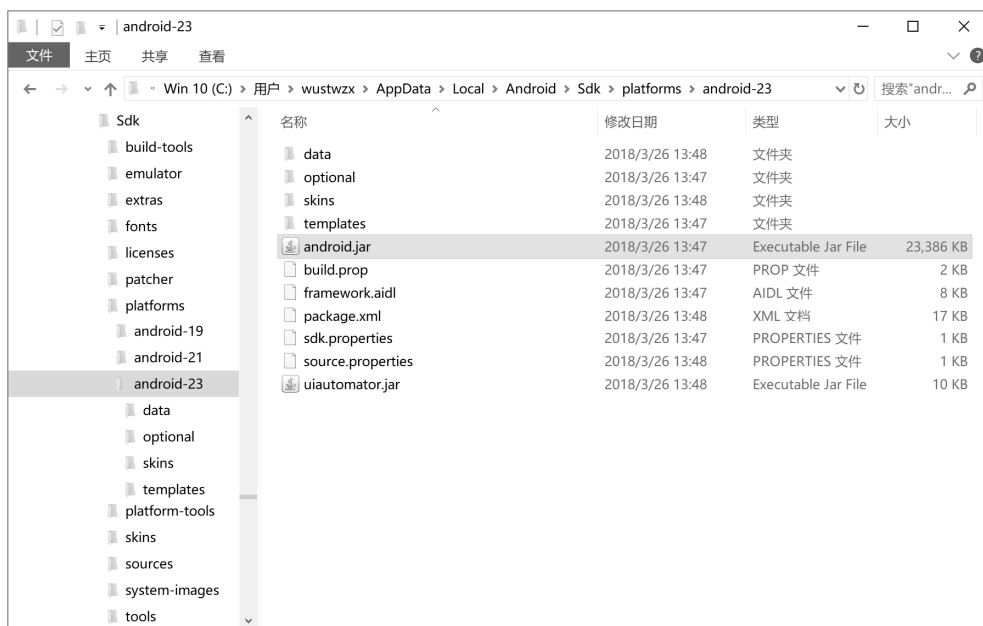


图 1.3.6 Android SDK 目录结构

除了 `platforms` 文件夹以外，SDK 文件夹还包含以下子文件夹。

- `add-ons`: 存放 Android 的扩展库，如用于地图开发的 Google Maps。
- `build-tools`: 包含各版本的 SDK 编译工具，如将 `.class` 字节码文件转换成 Android 字节码 `.dex` 文件的批处理程序 `dx.bat`、生成 Android 设备进程间通信代码的应用程序 `aidl.exe` 等。
- `extras`: 扩展开发包，如 HAXM 加速。
- `platform-tools`: 包含各版本的 SDK 通用工具，如用于将 Android 手机连接到 PC 端的 Android 调试桥（Android Debug Bridge, ADB）程序 `adb.exe`。又如数据库工具 SQLite。
- `system-images`: AVD 模拟器映像文件。
- `sources`: 存放 Android API 的源码。
- `tools`: 包含 `avdmanager.bat`、`sdkmanager.bat` 和 `emulator.exe` 等重要工具。

注意：

(1) 只有下载 Android 扩展库后，才会生成文件夹 `add-ons`。

(2) 设置 Android 应用的图标时，可以使用系统自带的图标库，这些图标文件就位于文件夹 `sdk\platforms\android-19\data\res\drawable-hdpi` 里。

(3) 为了方便在命令行方式下使用 `adb` 命令，一般应将文件夹 `platform-tools` 的路径添加到系统环境变量 `path` 里。

2. Android API 核心包

标准的 Android API 包含在许多软件包里，而这些软件包又包含在文件 `android.jar` 里。下面介绍 Android 开发中常用的软件包。

- `android.util`: 包含一些辅助类，如时间、日期的操作。
- `android.text`: 包含文本处理类。
- `android.text.method`: 提供为各种控件输入文本的类。
- `android.os`: 提供基本的操作服务、消息传递和进程间通信，提供了 `Binder`、`Handler`、`FileObserver`、`Looper` 和 `PowerManager` 等类。
- `android.app`: 实现 Android 的应用程序模型，主要包含 `Activity` 和 `Service` 组件，另外还有对话框和通知等重要类。
- `android.view`: 提供基础的用户界面接口框架，是 Android 的核心框架，包含类 `Menu`、`View`、`ViewGroup` 及一系列监听器和回调函数。
- `android.widget`: 包含在应用程序屏幕中使用的各种 UI 元素，通常派生自 `View` 类，包括 `TextView`、`EditText`、`ImageView`、`ListView` 和 `Button` 等控件。
- `android.webkit`: 默认浏览器操作接口，包含表示 Web 浏览器的类，主要有 `WebView`、`CacheManager` 和 `CookieManager`。
- `android.content`: 包含 `ContentProvider` 组件，还有 `Context` 和 `Intent` 等重要类。
- `android.content.pm`: 实现与包管理器相关的类。包管理器包含各种权限、安装包、安装程序、安装服务、安装组件（如 `Activity`）和安装应用程序。
- `android.content.res`: 用于访问结构化和非结构化资源文件。主要的类包括 `AssetManager`（用于结构化资源）和 `Resources`。
- `android.database`: 实现抽象数据库的理念，提供了 `Cursor` 接口。
- `android.database.sqlite`: 将 SQLite 用于物理数据库，主要包括 `SQLiteOpenHelper`、`SQLiteDatabase` 等类。
- `android.provider`: 提供一些类，访问 Android 的 `ContentProvider`，如 `Contacts`、`MediaStore`、`Browser` 和 `Settings` 等。
- `android.media`: 提供一些类，管理多种音频、视频的媒体接口，包含 `MediaPlayer`、`MediaRecorder`、`Ringtone`、`AudioManager` 和 `FaceDetector`。
- `android.hardware`: 实现与物理照相机相关的类。`android.graphics.Camera` 表示一种图形概念，与物理照相机完全无关。
- `android.bluetooth`: 提供一些类来处理蓝牙功能。主要的类包括 `BluetoothAdapter`、`BluetoothDevice`、`BluetoothSocket`、`BluetoothServerSocket` 和 `BluetoothClass`。
- `android.net`: 提供帮助网络访问的类，实现基本的套接字级网络 API。
- `android.net.wifi`: 管理 WiFi 连接。
- `android.telephony`: 提供手机设备的通话接口，包含类 `CellLocation`、`PhoneNumberUtils`

和 TelephonyManager。

- android.telephony.gsm: 可用于根据基站收集手机位置, 还包含负责处理 SMS 消息的类。
- android.location: 定位相关类。
- com.google.android.maps: 包含类 MapView 等 Google 地图所需类。
- android.gesture: 包含处理用户定义的手势所需的所有类和接口。
- android.graphics: 底层的图形库, 包含画布、颜色过滤、点、矩形等。
- android.graphics.drawable: 实现绘制协议和背景图像, 支持可绘制对象动画。
- android.graphics.drawable.shapes: 实现各种形状。
- android.view.animation: 提供对补间动画的支持。
- android.opengl: 提供 OpenGL 的工具, 可 3D 加速。

1.3.5 Android Studio 常用组合键

Android Studio 开发时, 为了提高编辑效率, 需要掌握的快捷操作如下。

- Alt+Enter: 在出现红色波浪线的地方使用, 提供了许多问题的解决方案, 如自动导包、自动生成接口方法和 try...catch 块等。
- Ctrl+Alt+O: 优化已导入的包, 清除不必要的包。
- Ctrl+D: 复制光标所在行的代码至下一行。
- Ctrl+Shift+/: 用于代码的注释和取消。
- Ctrl+Y: 删除光标所在的一行。
- 菜单 Code→Reformat Code: 代码格式化 (Ctrl+Alt+L 与 QQ 快捷键冲突)。
- Ctrl+F12: 显示类成员和继承的方法。
- Ctrl+H: 打开类 (或接口) 继承关系图。
- Ctrl+O: 显示所有可以重写的父类方法 (含接口方法)。
- Ctrl+Alt+T: 把选中的一组代码包在一块内, 如 if、for 和 try...catch 等。
- Alt+Insert (MAC 无 Insert 键, 可使用右键菜单→Generate): 生成代码, 如 set/get 方法、构造方法等。

1.4 创建和运行 Android 应用

1.4.1 创建一个 Hello 级 Android Studio 项目

在 Android Studio 主菜单中, 选择 Start a new Android Studio project 后, 出现创建 Android 项目对话框。输入应用名称为 “HelloAndroid” 后的效果, 如图 1.4.1 所示。

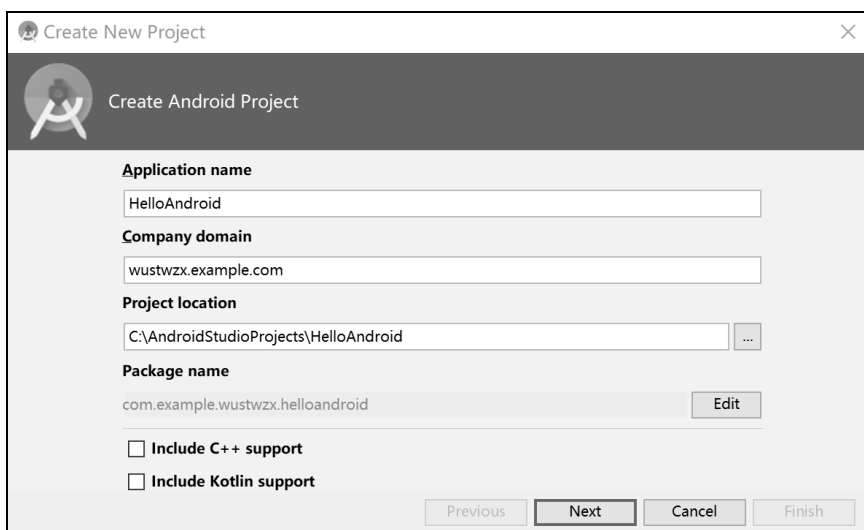


图 1.4.1 新建 Android 项目对话框

进入选择目标设备对话框，一般默认勾选 **Phone and Tablet** 项（表示手机和平板），根据需要指定目标设备运行的最低 Android 版本，如图 1.4.2 所示。

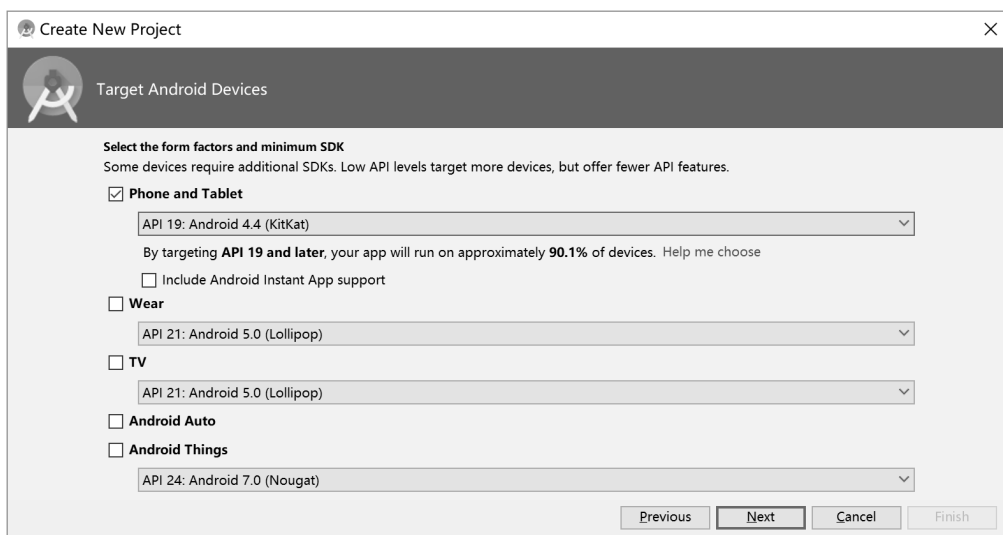


图 1.4.2 选择目标设备对话框

进入选择 Activity 模板对话框，一般默认选择 **Empty Activity** 项，如图 1.4.3 所示。

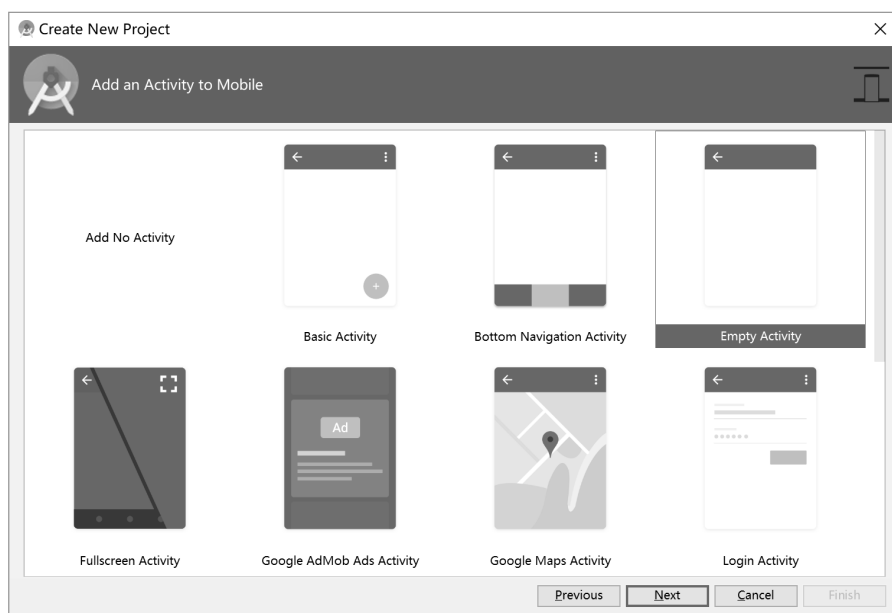


图 1.4.3 选择目标设备对话框

最后一个对话框（Login Activity）是指定应用主界面对应的 Activity 及布局名称，一般选择默认值，如图 1.4.4 所示。

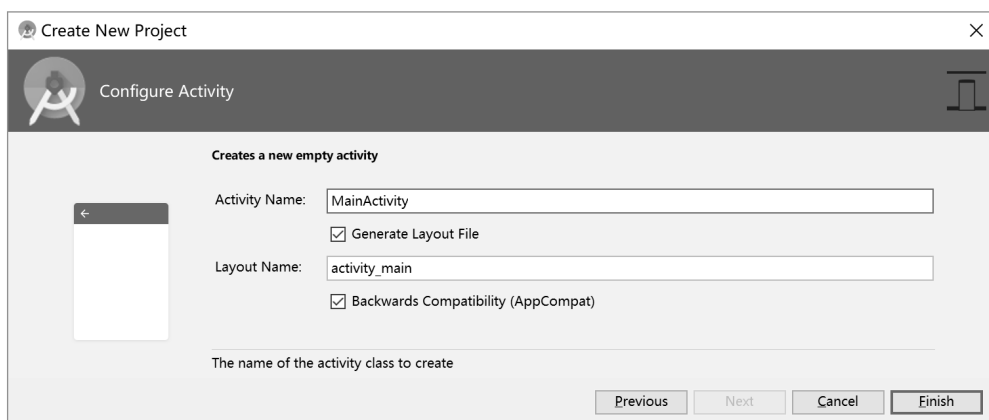


图 1.4.4 设置应用主界面对应的 Activity 及布局名称

单击 Finish 按钮，需要等待系统构建项目完成后，才能进入 Android Studio 集成开发环境，如图 1.4.5 所示。

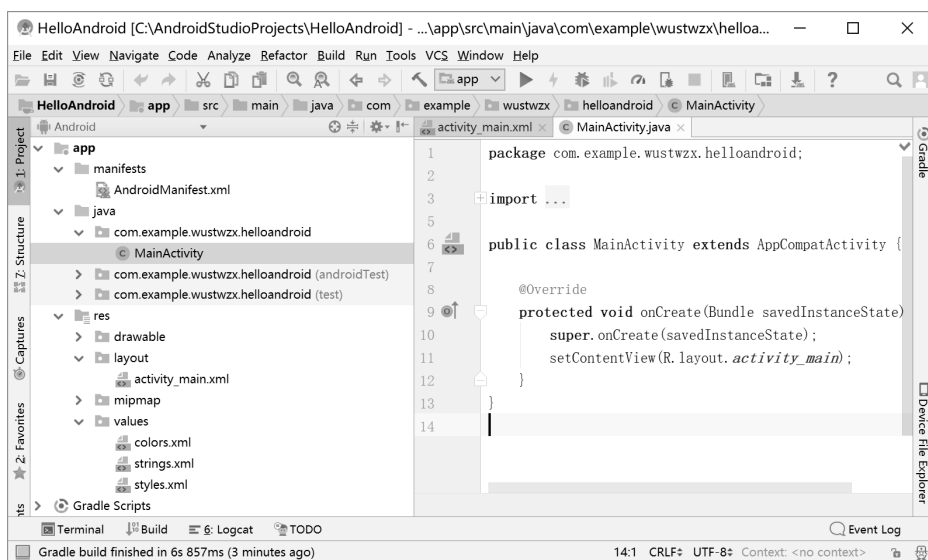


图 1.4.5 Android Studio 集成开发环境



在 Android Studio 集成开发环境里，包含菜单栏、工具栏、项目区、文档编辑区和一些选项卡。

项目结构默认使用 Android 视图，为了获取更多的信息，可以切换至 Project 视图，查看 Android API 和项目添加的依赖包。

窗口底部包含了 Terminal、Build 和 Logcat 等选项卡。其中，Terminal 用于在命令行方式下执行 Android 平台提供的一些命令；Build 用于查看项目的编译和构建信息；Logcat 用于查看 Android 设备运行的日志信息。

例如，在 Terminal 控制台，查看已连接可用的 Android 设备的命令如下：

```
adb devices
```

工具栏  用于项目 Gradle 构建，其结果在  Build 控制台里查看。如果有错误，会输出相应的信息以便修改。只有在项目构建没有错误后，方可安装和运行。

窗口左侧包含了 Project 和 Structure 两个重要的选项，默认为 Project。当编辑窗口打开某个 Java 类文件（含 Android API 的源码）时，选择 Structure 选项，就能查看该类的组成结构图。

如果已经打开某个项目窗口，再通过菜单 File→Open 打开项目，默认会出现如图 1.4.6 所示的对话框。

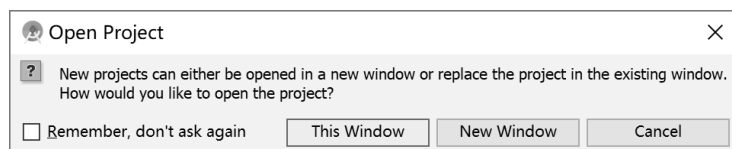


图 1.4.6 Android Studio 默认打开项目方式

一旦选择 This Window 并勾选了左边的任选框，以后再打开项目时就不会出现上述对话框了，而是默认在当前窗口中打开其他项目。

在 Android Studio 开发过程中，有时需要同时打开多个项目窗口。解决办法是使用菜单 File→Settings→Appearance & Behavior→System Settings→Project Opening，选中 Confirm window to open project in 项，如图 1.4.7 所示。

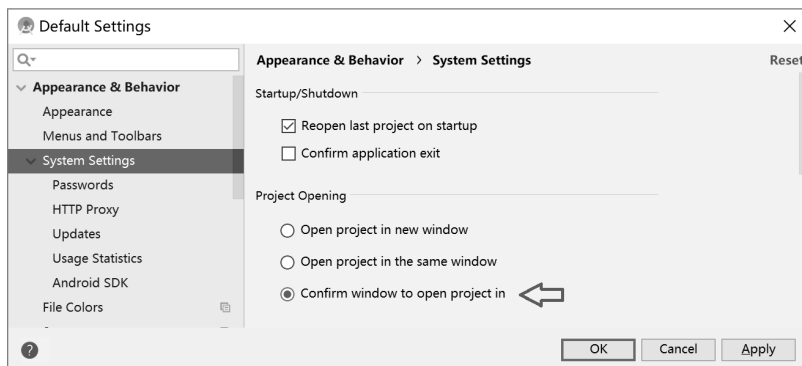


图 1.4.7 Android Studio 设置允许打开多个项目窗口


注意：

- (1) 不同 Android Studio 版本的界面布局有一定的差异，本书使用的是 3.1.2 版本。
- (2) 项目创建时指定的目标设备最低 Android 版本信息并未保存在项目清单文件 AndroidManifest.xml 里。
- (3) 打开 Java 类文件时，通过【Ctrl+Click】组合键方式可查看 Android API 及其源码。
- (4) 在编辑某个项目的同时，使用系统菜单 File 也可以新建 Android 项目。

1.4.2 手机模拟器与 AVD Manager

1. 使用 Android Studio 内置模拟器

AVD (Android Virtual Device) 是 Android SDK 提供的最重要工具之一，它使开发人员在没有物理设备的情况下，可以在计算机上对 Android 程序进行开发、调试和仿真。

Android Studio 工具栏上的 AVD Manager 工具 ，用于创建、编辑和运行模拟器，如图 1.4.8 所示。

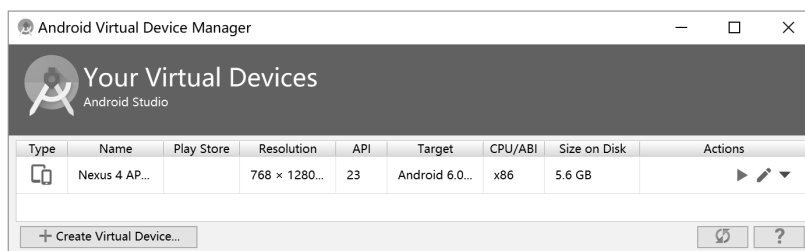


图 1.4.8 AVD Manager


在 Android Studio 中，可以创建和启动多个模拟器，每个模拟器对应一个 ID，它就是模拟器的电话号码。其中，第一个启动的模拟器 ID 为 5554，第二个启动的模拟器 ID 为 5556，Android 系统允许手机（或模拟器）向自己发送短信息。

注意：Android Studio 已经不再支持在模拟器控制 (Emulator Control) 中操作模拟器，改为在 Android 模拟器上进行操作。单击模拟器右侧下方的三个小点，在弹出的 Extended controls 对话框中进行操作，如在 Phone 选项卡里，可模拟打电话和发短信操作。

2. 使用第三方模拟器

Android 内置模拟器启动速度较慢，操作也不够流畅。使用 Android 手机进行测试，可以弥补上述不足，但却连接不便。因此，业界也开始广泛使用第三方模拟器。下面介绍一款对硬件要求较低、基于 Android 4.4 的夜神安卓模拟器。

访问课程网站 <http://www.wustwzx.com/android/index.html>，可下载夜神安卓模拟器安装包。为了部署 Android 应用到该模拟器，在启动该模拟器后，进入 Android Studio 的 Terminal 终端，执行桥接命令 adb，如图 1.4.9 所示。



```
Terminal
+ Microsoft Windows [版本 10.0.17134.167]
X (c) 2018 Microsoft Corporation. 保留所有权利。

C:\AndroidStudioProjects\HelloAndroid>adb connect 127.0.0.1:62001
connected to 127.0.0.1:62001

C:\AndroidStudioProjects\HelloAndroid>
```

图 1.4.9 创建 Android Studio 与夜神 AVD 的连接

3. 设备文件浏览器

在 Android Studio 3.1.2 里，使用菜单 View→Tool Windows→Device File Explorer，可浏览 Android 设备里的文件，实现文件管理工作，包括文件的打开、删除、导入与导出等，如图 1.4.10 所示。

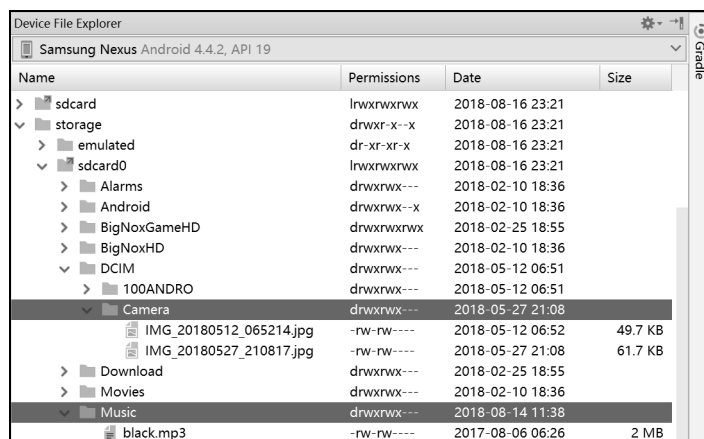


图 1.4.10 Android Studio 3.1.2 的设备文件浏览窗口

注意：

(1) Android Studio 界面右下方，提供了快速进入设备文件浏览器的工具。

(2) 对于 Android 6.0 及以上版本的模拟器或手机，将照片和音乐等文件存放在路径/storage/emulated/0 下。

(3) 对于 Android 6.0 以下版本的模拟器，照片和音乐等文件存放在路径/storage/sdcard0 下。

(4) 对于 Android 6.0 及以上版本的手机，如果手机没有 root 权限，则无法浏览路径/storage/emulated/0，但在路径/sdcard 下可以找到照片和音乐等文件。


(5) 对于没有 root 权限的 Android 手机，无法访问某些位于手机内部存储的系统文件夹（如位于/data/app/packname 里的.apk 文件）；而模拟器里的所有文件均可被访问。

1.4.3 安装和运行 Android 应用

安装 Android 应用到 Android 手机上运行之前，通常需要将手机与计算机相连接。在物理上将手机通过手机数据线与计算机连接前，应打开手机的 USB 调试开关，其方法是运行手机的设置→开发人员选项→USB 调试。

单击 Android Studio 工具栏上的 ▶ 图标，弹出 Select Deployment Target 对话框，选择对应的手机或虚拟机设备，即可安装和运行。

注意：手机较 AVD 而言，响应速度更快。

如果 Android 设备的版本低于建立 Android 应用项目时指定的最低版本，则无法安装应用。解决办法：单击 Android Studio 工具栏上的  工具，依次单击 App 项和 Flavors 项，更改 Min Sdk Version 选项值，如图 1.4.11 所示。

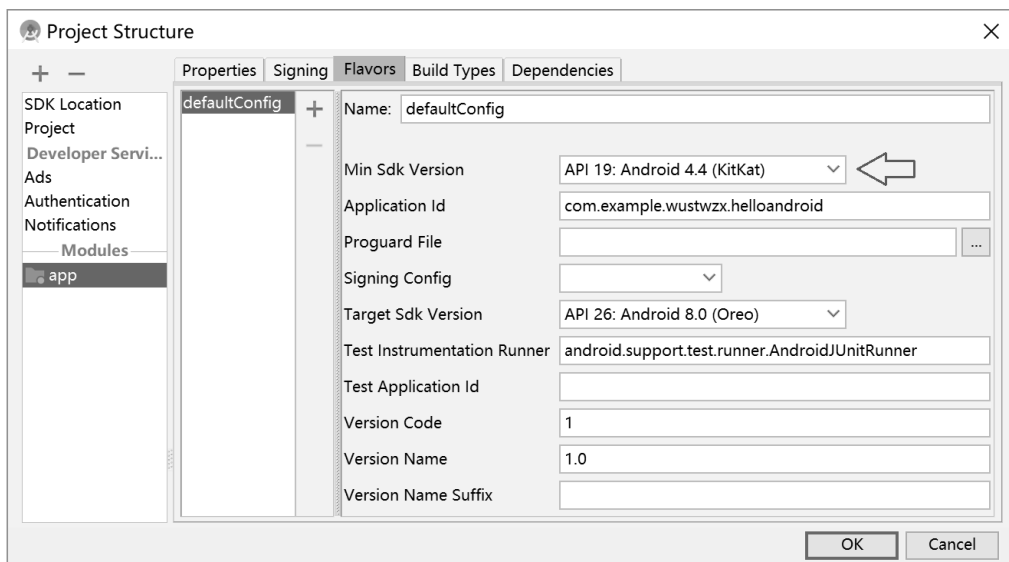


图 1.4.11 查看和修改应用要求目标设备的最低 Android 版本

习 题 1

一、判断题

1. 使用手机自带的文件管理程序，可以浏览 Android 应用的安装目录。
2. 在计算机上安装手机驱动程序之前，必须打开手机的 USB 调试开关。
3. 在安装 Android Studio 之前，必须安装 JDK。
4. 在 Android Studio 中，SDK Manager 的主要作用是 Android 平台的下载管理。
5. 在 Android Studio 创建项目的对话框中，包含了对运行设备最低 Android 版本的指定。
6. AVD 是 Android 虚拟设备的英文缩写。

二、选择题

1. 下列不属于 Android 应用程序框架层的是_____。
A. Activity Manager B. SQLite
C. Location Manager D. Notification Manager
2. Android 移动设备底层（内核）使用_____系统。
A. DOS B. Linux C. Windows D. UNIX
3. 用于 Android 应用开发的 android.jar 文件，位于 sdk 文件夹的_____子文件夹里。
A. add-ons B. platforms
C. system-images D. tools
4. 在 Android Studio 创建 AVD 时，默认选择的目标设备是_____。
A. TV B. Android Auto
C. Phone and Tablet D. Wear
5. 按照缩进风格格式化程序代码，应使用 Android Studio 的_____菜单。
A. Build B. Tools C. View D. Code

三、填空题

1. 通常将 Android 软件系统划分为_____层。
2. 在 Android 系统中，应用程序设计使用_____语言。
3. 在 Android Studio 中，使用组合键_____优化程序所需要的软件包。
4. 在 Android Studio 中，先启动的内置模拟器的电话号码是_____。
5. 使用 Android Studio 3.1.2 提供的设备文件浏览器功能，应选择的菜单是_____。

实 验 1

一、实验目的


- (1) 掌握 Android 集成开发环境的搭建方法。
- (2) 掌握 Android SDK 的作用及 Android SDK Manager 的使用。
- (3) 掌握使用向导创建 Android Hello 项目的方法。
- (4) 掌握 Android 模拟器的创建与使用。
- (5) 掌握 Android 项目的部署及运行方法。

二、实验内容及步骤


1. 快速搭建 Android 集成开发环境

- (1) 访问 Android 中文社区 <http://www.android-studio.org>, 依次选择下载→历史版本下载, 下载 Android Studio 3.1.2 版本后进行安装。
- (2) Android Studio 安装完成后, 再安装 Android SDK。
- (3) 在 Android Studio 中, 使用 Android SDK Manager 工具  下载所需要的 Android API 版本 (如 API 25, 对应于 Android 7.1.1)。

2. 使用向导, 创建 Android Studio 项目

- (1) 在 Android Studio 中, 使用菜单 File→New→New Project。
- (2) 在出现的对话框中, 全部选择默认值。
- (3) 使用 Android Studio 工具栏上的  工具, 查看项目的配置信息。
- (4) 使用【Ctrl+Click】组合键方式, 查看 Android API 及其源码。
- (5) 逐步掌握 Android Studio 快捷键的使用。

3. Android 模拟器的使用

- (1) 单击 Android Studio 工具栏上的模拟器工具 , 进入模拟器管理界面。
- (2) 创建一个与已有 API 版本相对应的模拟器。
- (3) 启动模拟器, 模拟手机的实际操作。
- (4) 单击 Android Studio 界面右侧下方的设备文件管理器工具按钮, 对模拟器文件进行导入/导出操作。

4. 部署 Android 项目并做运行测试、应用程序卸载

- (1) 确保手机开发者选项可用, 打开 USB 调试开关。
- (2) 单击 Android Studio 工具栏上的工具 , 分别部署项目到手机和模拟器中并运行。
- (3) 练习卸载 Android 应用的操作。
- (4) 查看 Android Studio 的各控制台 (如 Build、Run 和 Logcat 等) 的输出信息。

三、实验小结及思考

(由学生填写, 重点写上机中遇到的问题。)