

02 Android系统基础

本章大纲

1. Andoid 平台发展史
2. Andoid 平台架构
3. Andoid Studio环境搭建
4. 程序结构分析

Android是什么？

- Android是Google开发的基于Linux平台的、开源的、智能手机操作系统。Android包括操作系统、中间件、应用程序框架和应用程序，由于源代码开放，Android系统可被移植到不同的硬件平台上。
- android 一词最早出现于法国作家利尔亚当在1886年发表的科幻小说《未来夏娃》中。他将外表像人的机器起名为“安德罗丁”(android)
- Android之父：Andy Rubby

Android平台发展史

2008年9月，谷歌正式发布了Android 1.0系统，全球第一台Android设备（G1）

2010年12月，谷歌发布了Android 2.3

2011年2月，Android设备用户总数达到1.35亿，发布Android 3.0

2011年8月，Android手机占全球智能手机市场48%的份额，跃居全球第一

2013年5月16日，在I/O大会上，谷歌推出新的Android开发环境Android Studio

2013年7月24日，Nexus 7平板二代发布

2013年9月3日，谷歌发布了Android 4.4

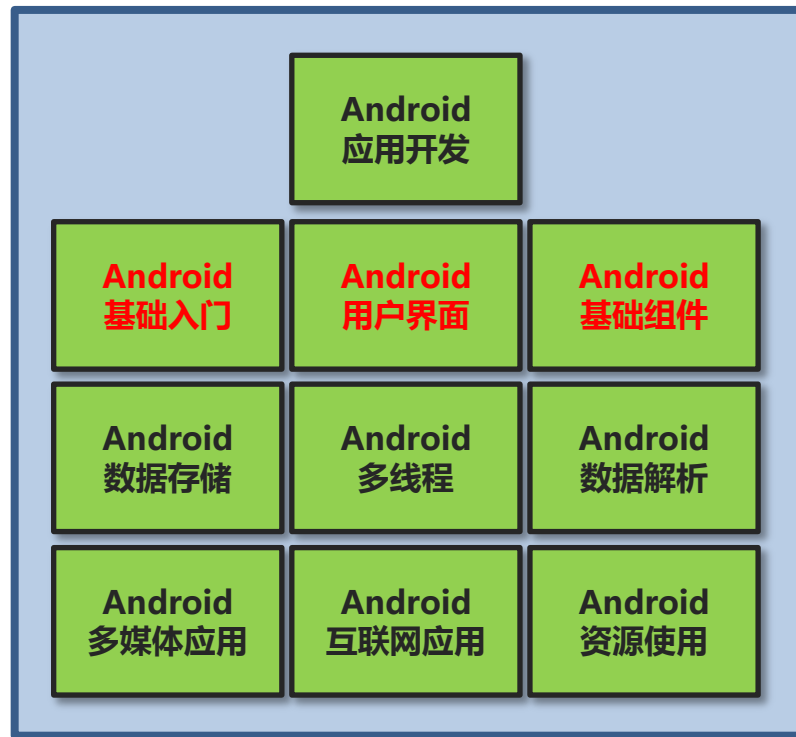
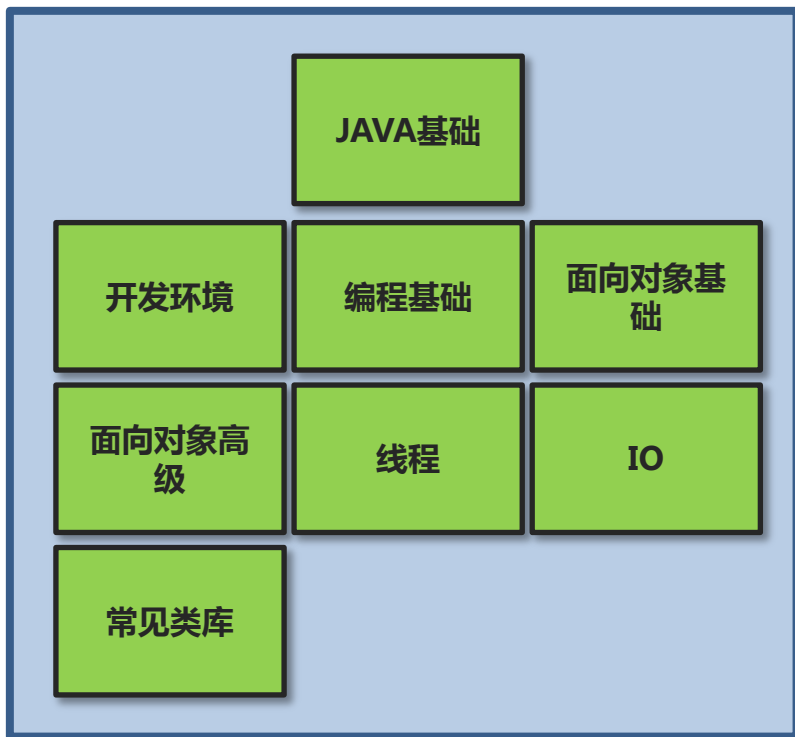
2014年10月15日，发布Android 5.0

2015年5月29日，在谷歌I/O开发者大会上，谷歌发布Android Studio 1.3版

2015年5月28日，“Marshmallow（棉花糖）”的安卓6.0系统正式推出

2017年8月22日，发布了Android 8.0的正式版，正式名称为：Android Oreo（奥希奥）

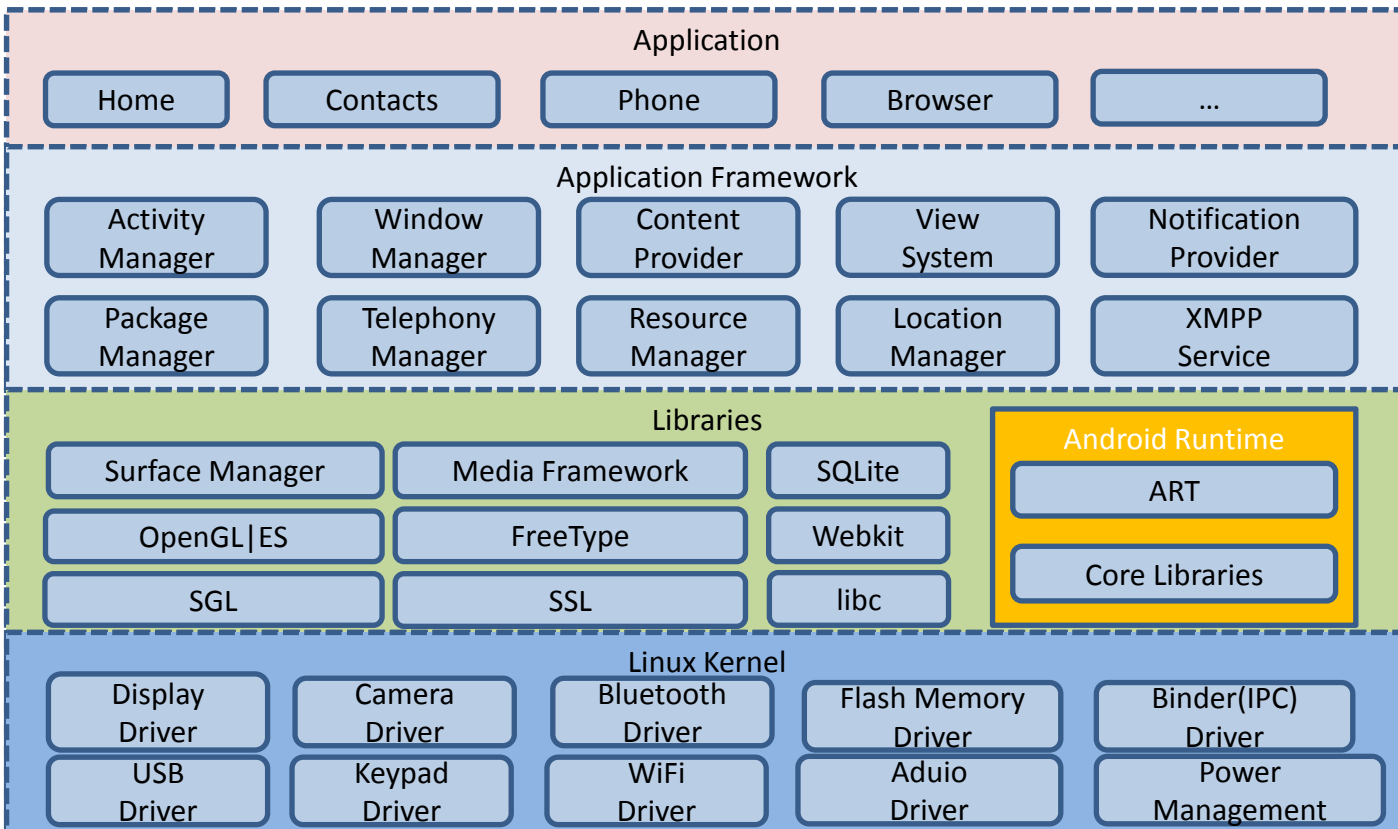
Android开发体系



本章大纲

1. Andoid 平台发展史
2. Andoid 平台架构
3. Andoid Studio环境搭建
4. 程序结构分析

Android系统框架

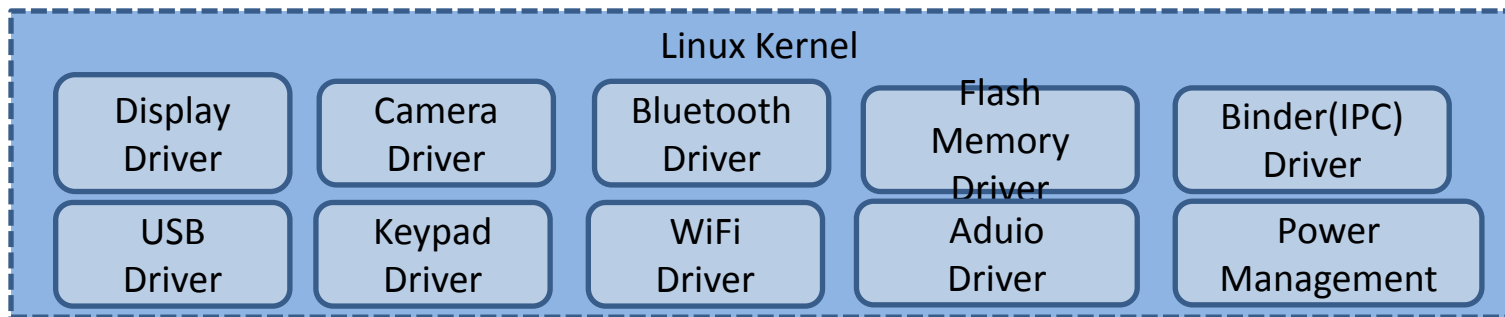


Android系统框架

- Android的软件结构的几个层次
 - Linux内核层相关驱动，C
 - 核心类库（Libaries）和运行时环境（Runtime），C\C++
 - 应用程序框架（Application Framework），Java
 - 应用程序（Application），Java

Android系统框架

- Linux内核层
 - 硬件和其他软件堆层之间的一个抽象隔离层
 - 提供安全机制、内存管理、进程管理、网络协议堆栈和驱动程序等核心系统服务



Android系统框架

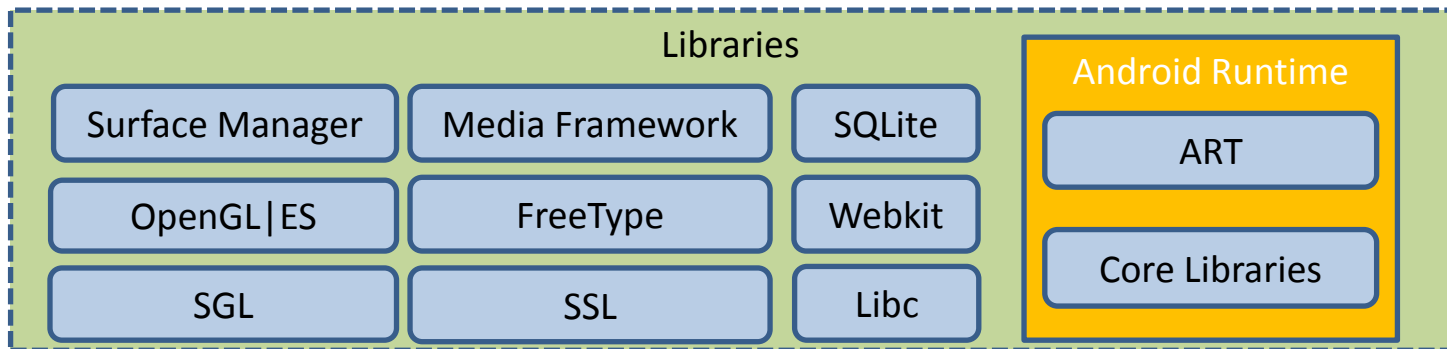
- 核心类库和运行时环境层
 - 由核心运行库和Android运行时环境构成函数库，主要提供一组基于C/C++的函数库
 - Surface Manager，提供应用程序的2D、3D图像层的平滑连接
 - Media Framework，实现音视频的播放和录制功能
 - SQLite，轻量级的关系数据库引擎
 - OpenGL ES，基于3D图像加速
 - FreeType，位图与矢量字体渲染
 - WebKit，Web浏览器引擎
 - SGL，2D图像引擎
 - SSL，数据加密与安全传输的函数库
 - Libc，标准C运行库，Linux系统中底层应用程序开发接口

Android系统框架

- 核心类库和运行时环境层

- Android运行时环境

- 核心库，提供Android系统特有函数功能和Java语言函数功能
 - Dalvik虚拟机，实现基于Linux内核的线程管理和内存管理
 - Android5.0开始，**ART**取代了Dalvik虚拟机

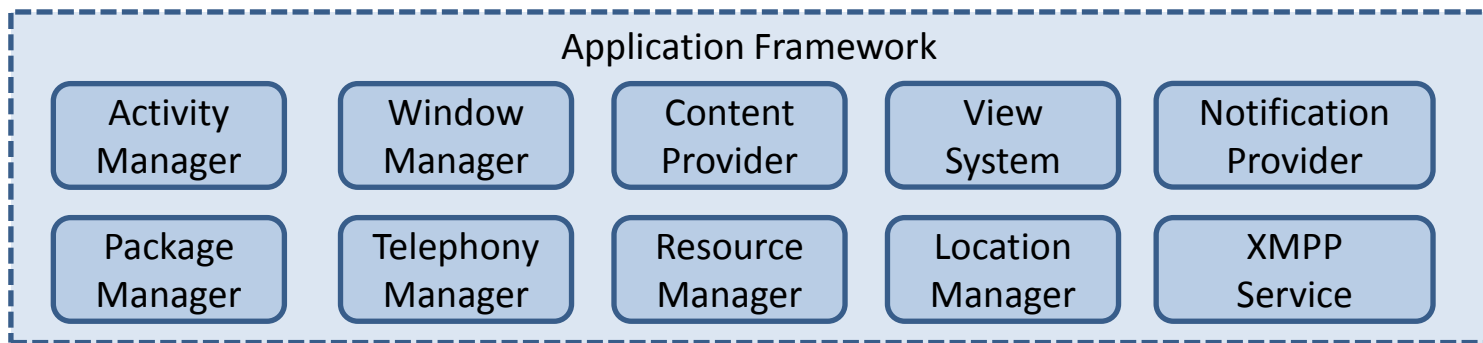


Android系统框架

- 应用程序框架层
 - 提供Android平台基本的管理功能和组件重用机制
 - Activity Manager, 管理应用程序的生命周期
 - Windows Manager, 启动应用程序的窗体
 - Content Provider, 共享私有数据, 实现跨进程的数据访问
 - Package Manager, 管理安装在Android系统内的应用程序
 - Telephony Manager, 管理与拨打和接听电话的相关功能

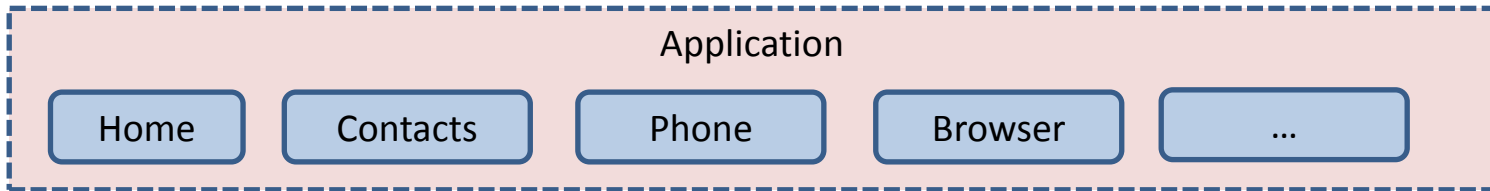
Android系统框架

- 应用程序框架
 - Resource Manager, 允许应用程序使用非代码资源
 - Location Manager, 管理与地图相关的服务功能
 - Notification Manager, 允许应用程序在状态栏中显示提示信息



Android系统框架

- 应用程序层
 - 提供一系列的核心应用程序
 - 包括电子邮件客户端、浏览器、通讯录和日历等



本章大纲

1. Andoid 平台发展史
2. Andoid 平台架构
3. Andoid Studio环境搭建
4. 程序结构分析

Android Studio环境配置

- 安装JDK，配置基本Java环境
- 安装Android Studio3.3.2，绿色解压
- 获得Android SDK，绿色解压
- 在Android Studio中配置Android SDK
- Gradle简介及配置

Android SDK的目录结构

- SDK的目录结构如下：
 - add-ons: 附件的包
 - docs: HTML格式的离线文档
 - platforms: SDK核心内容
 - tools: 工具

Android的版本

- Android系统的版本有很多，Google使用API级别来标识具体的Android版本。
- Android程序采用应用程序的向后兼容性。



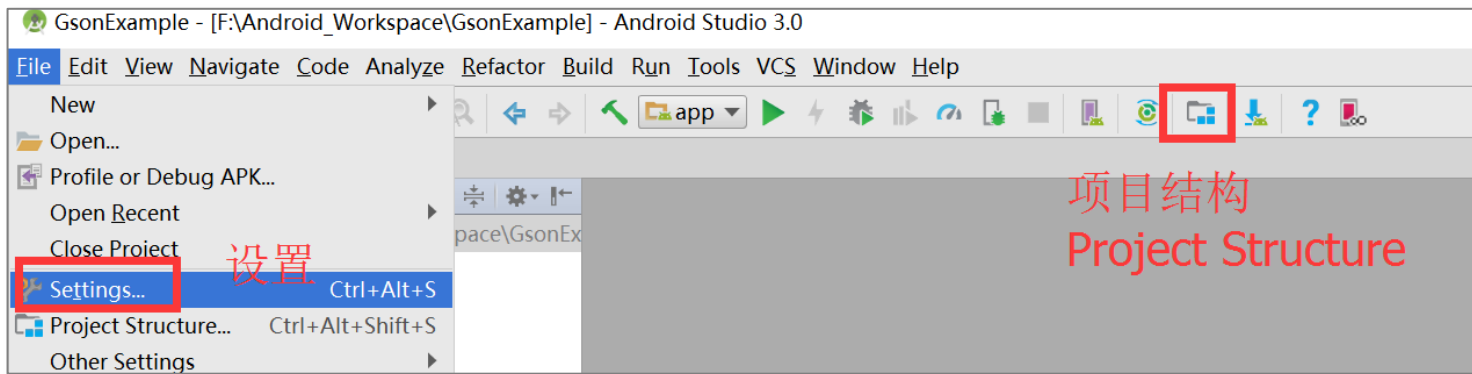
平台版本	API 级别
Android 8.0	26
Android 7.0	24
Android 6.0	23
Android 5.1	22
Android 5.0	21
Android 4.4	19
Android 4.3	18
Android 4.2、4.2.2	17
Android 4.1、4.1.1	16
Android 4.0.3、4.0.4	15
Android 4.0、4.0.1、4.0.2	14

Gradle环境配置

- Gradle是一个基于Apache Ant和Apache Maven概念的项目 **自动化建构工具**。它使用一种基于Groovy的特定领域语言 (DSL) 来声明项目设置，抛弃了基于XML的各种繁琐配置。
- Gradle个版本下载地址：
 - <https://services.gradle.org/distributions/>

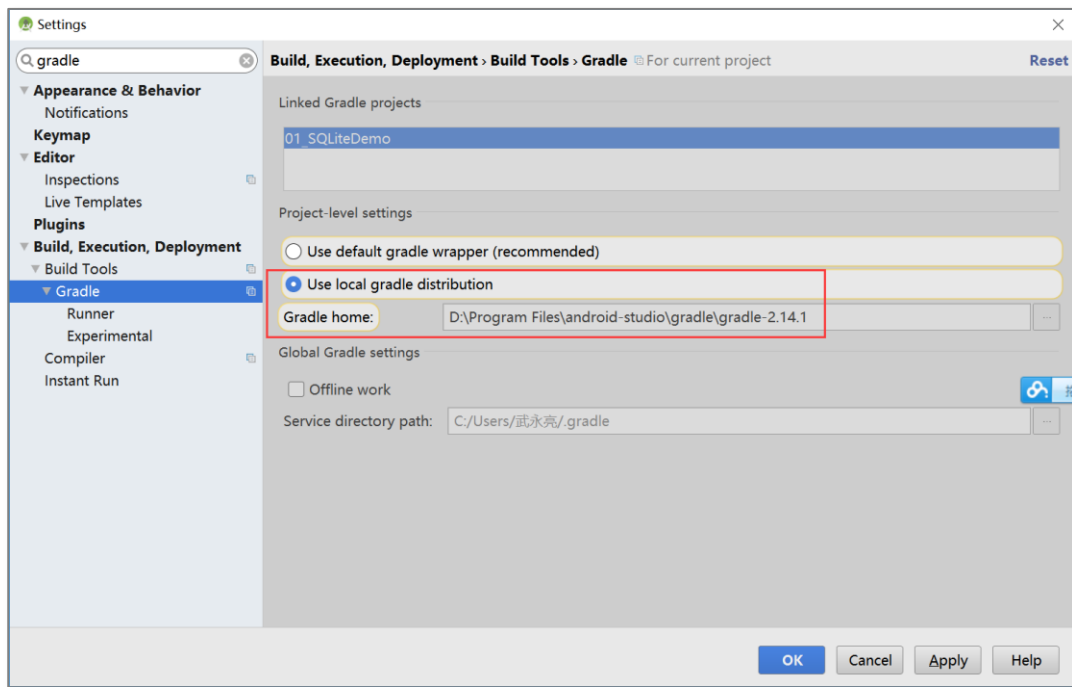
Gradle环境配置

- 在Android studio中配置工程使用Gradle



Gradle环境配置

- 在Android studio中配置工程使用Gradle



Android程序调试环境

1. 真机调试，真实Android设备，打开usb调试选项进行调试
2. Google原生模拟器
3. Genymotion虚拟机安装配置(不支持ARM结构的应用)
4. 夜神模拟器
5. 天天模拟器

夜神模拟器

- <https://www.bignox.com/>

nox_adb.exe connect 127.0.0.1:62001



本章大纲

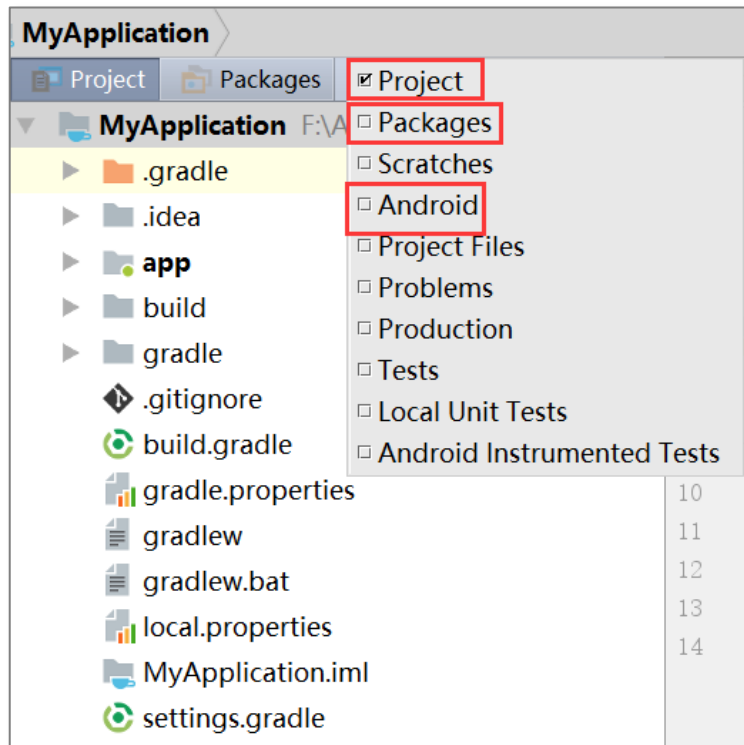
1. Andoid 平台发展史
2. Andoid 平台架构
3. Andoid Studio环境搭建
4. 程序结构分析

Android程序运行过程

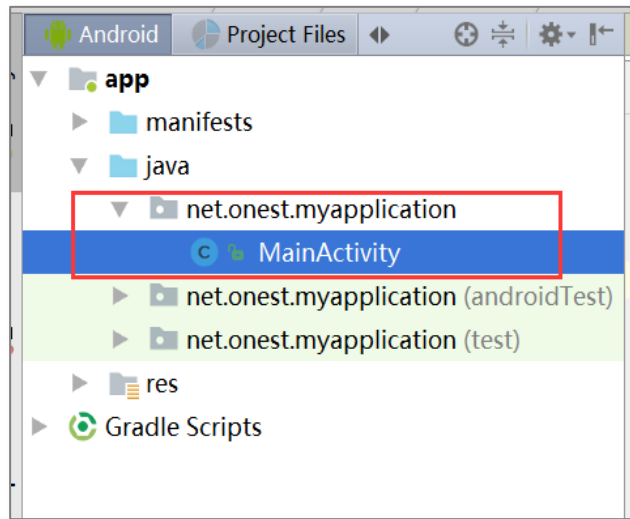


Android Studio视图

- **Android视图:**
 - 列出一个工程的常用文件夹
 - Gradle文件单独列出，方便修改
 - 推荐使用此视图
- **Project视图:**
 - 以Project为根目录列出
 - 一般的工程常用的是一个app模块
 - 可看到工程中所有文件
- **Packages视图:**
 - 以包为单位列出所有的源代码及资源



Android程序结构



- **Java目录**是源代码目录，所有的java文件都保存在此目录
- MainActivity.java文件的代码如下

```
public class HelloAndroid extends AppCompatActivity {  
    /** Called when the activity is first created. */  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

Activity VS AppCompatActivity

- 在早期的Eclipse环境中，默认继承Activity
- 在最新的AS环境中，API Level >19时，默认继承AppCompatActivity
- MainActivity.java文件的代码如下：

```
public class HelloAndroid extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

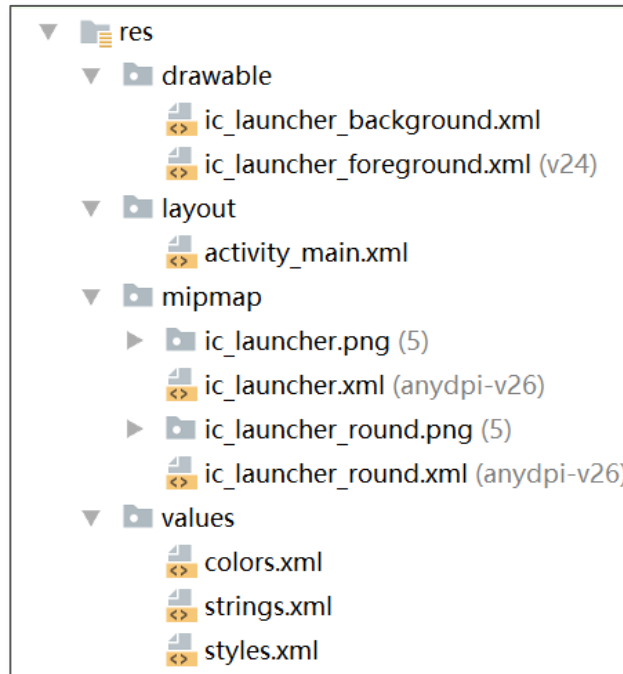
```
public class HelloAndroid extends AppCompatActivity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

Android程序结构

- **build目录**用来保存编译后**自动生成**的文件，例如：
R.java或AIDL文件
- R.java文件是自动生成的文件，包含对drawable、layout和values目录内的资源的引用ID，Android程序能够直接通过R类引用目录中的资源
 - R.java文件不能手工修改，如果资源目录中增加了资源文件后会自动添加
 - R.java文件在**AS中默认不可见**

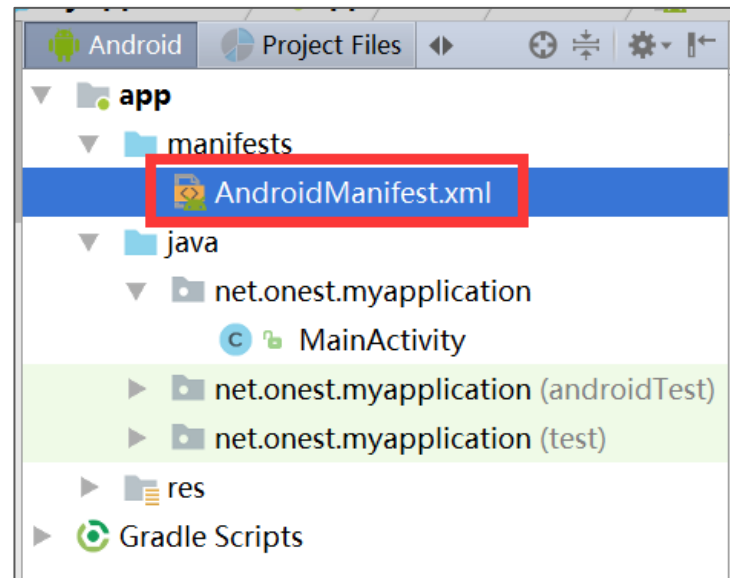
Android程序结构

- res: 目录中存储工程的所有资源
 - drawable: 图片资源
 - layout: 布局资源
 - mipmap: 图片资源
 - values: 常量配置资源
 - colors: 颜色
 - dimens: 距离, 长度
 - strings: 字符常量
 - styles: 主题, 样式



Android程序结构

- `AndroidManifest.xml`是XML格式的Android程序声明文件，包含了运行Android程序的重要信息，这些信息包含应用程序名称、图标、包名称、模块组成、授权和SDK最低版本等
- 每个Android程序必须包含一个`AndroidManifest.xml`文件



实例-加法器

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button cal = (Button) findViewById(R.id.button);

    cal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText et1 = (EditText) findViewById(R.id.editText1);
            EditText et2 = (EditText) findViewById(R.id.editText2);
            int a = Integer.parseInt(et1.getText().toString());
            int b = Integer.parseInt(et2.getText().toString());
            String s = Integer.toString(add(a, b));
            Toast.makeText(Caculator.this, s, Toast.LENGTH_LONG).show();
        }
    });
}
```