



河北师范大学软件学院
Software College of Hebei Normal University



Android 基础开发

第一讲 Android简介与环境搭建



Java与移动智能设备开发



目录

1 课程介绍

2 Android简介

3 Android学习路线

4 Android系统框架

5 Android开发环境

6 第一个Android程序



课程意义

- 为什么学习Android?
 - Android开发人员需求缺口

Android开发工程师	北京天玑科技有限公司	会员
10K-15K 北京 3-5年 本科	股份制企业	20-99人
五险一金 通讯补贴 年终分红 绩效奖金 带薪年假	最新	
android/安卓开发工程师	北京思普瑞特科技发展有限公司	会员
12K-22K 北京 3-5年 大专	民营	100-499人
定期体检 员工旅游 弹性工作 带薪年假 餐补	最新	
android开发工程师	柯锐特互动(北京)科技有限公司	ZP名企
10K-15K 北京 3-5年 本科	民营	100-499人
五险一金 绩效奖金 加班补助 弹性工作 节日福利	最新	
Android开发工程师	嘉网股份	ZP名企
10K-15K 北京-石景山区 3-5年 大专	上市公司	100-499人
五险一金 年底双薪 节日福利	最新	



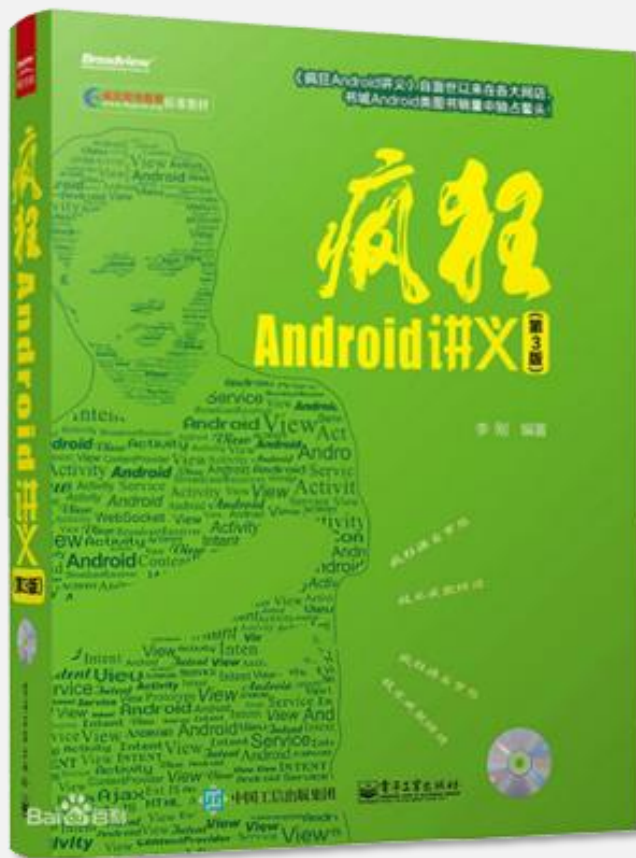
课程介绍



- 上课时间及授课方式
 - 授课时间每周三个半天
- 考核方式
 - 平时成绩40分（作业、综合表现、测试）
 - 期末考试成绩60分



参考资料



百度、谷歌

<http://developer.android.com/index.html>

<http://www.android-studio.org/>

<https://developer.android.google.cn/>



课程目标



- 希望各位同学可胜任如下职位：
 - Android手机应用开发工程师
 - Android系统开发工程师
 - Android测试工程师
 - Android产品策划
 - Android UI设计师
 - Android产品运营管理
 - 智能设备（智能家居、汽车电子、通讯设备、GPS、航天设备、医疗器械）开发工程师



目录

1 课程介绍

2 **Android简介**

3 Android学习路线

4 Android系统框架

5 Android开发环境

6 第一个Android程序



Android简介

- Android是Google公司和开放手机联盟领导及开发的基于Linux平台的、开源的智能操作系统。Android平台由操作系统、中间件、用户界面和应用软件组成。





Android常见机型





目录

1 课程介绍

2 Android简介

3 Android学习路线

4 Android系统框架

5 Android开发环境

6 第一个Android程序



Android学习路线



JAVA基础

开发环境

编程基础

面向对象基础

面向对象高级

线程

IO

常见类库

Android
应用开发

Android
基础入门

Android
用户界面

Android
基础组件

Android
数据存储

Android
多线程

Android
数据解析

Android
多媒体应用

Android
互联网应用

Android
定位应用



目录

1 课程介绍

2 Android简介

3 Android学习路线

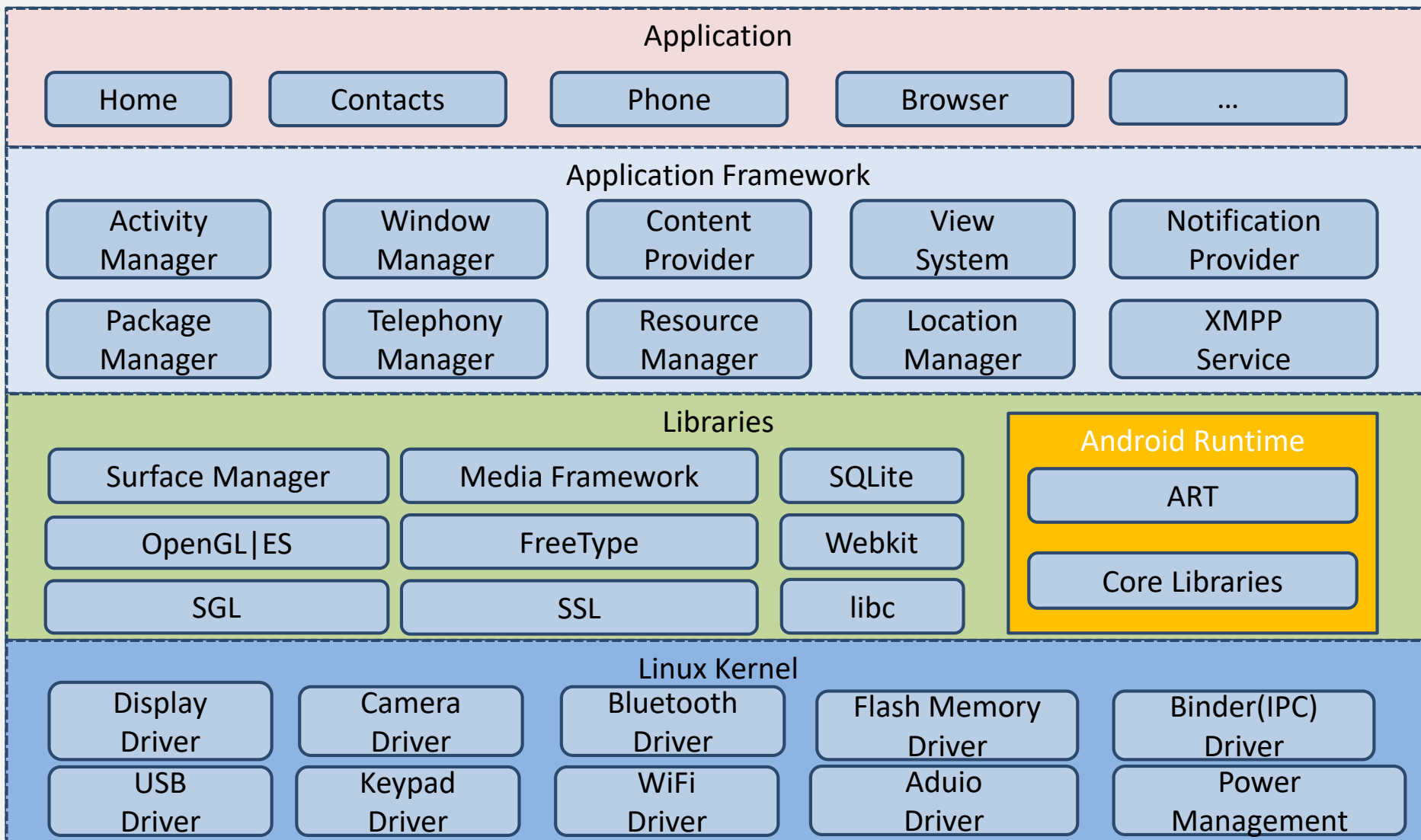
4 Android系统框架

5 Android开发环境

6 第一个Android程序



Android系统框架





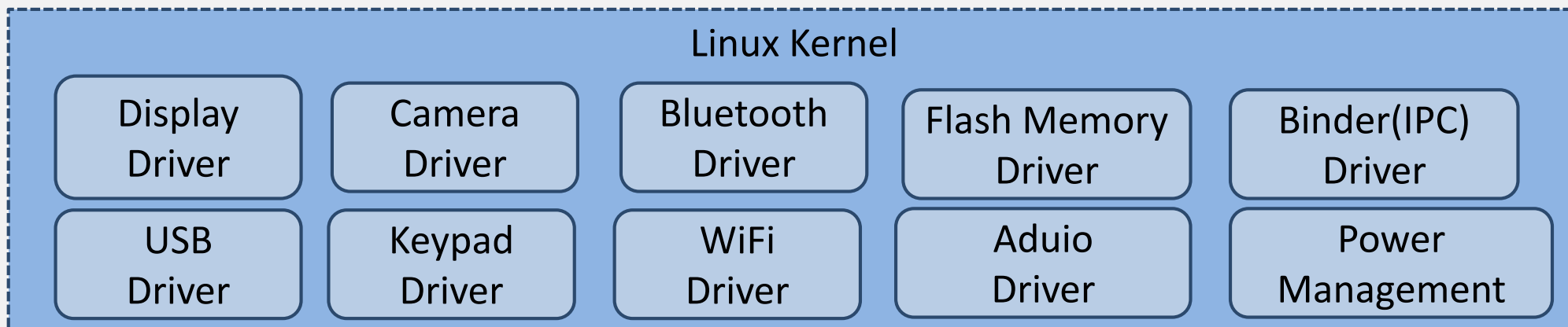
Android系统框架

- 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





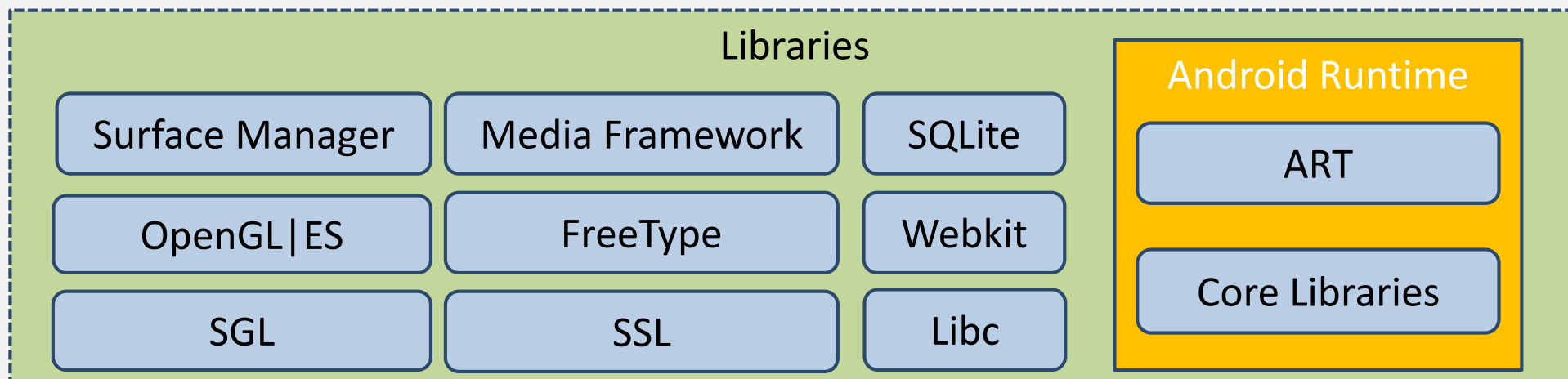
Android系统框架

- 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



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





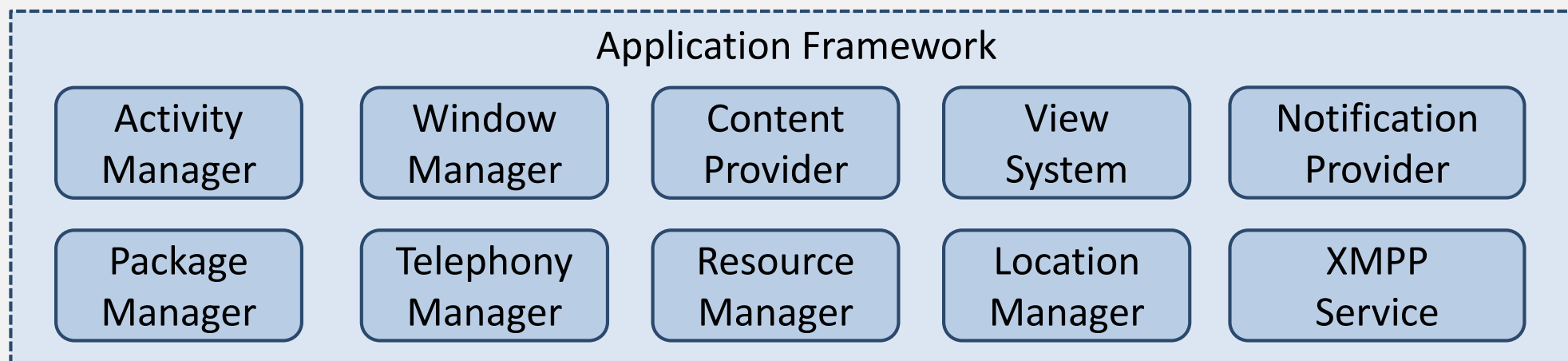
Android系统框架

- 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

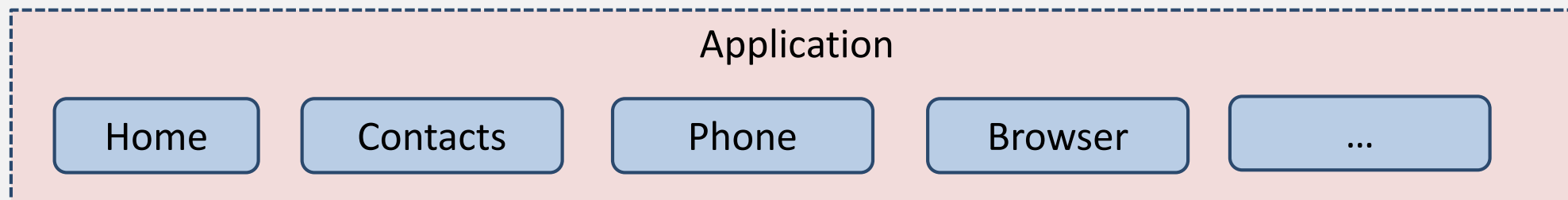
- 应用程序框架
 - Resource Manager, 允许应用程序使用非代码资源
 - Location Manager, 管理与地图相关的服务功能
 - Notification Manager, 允许应用程序在状态栏中显示提示信息





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





目录

1 课程介绍

2 Android简介

3 Android学习路线

4 Android系统框架

5 Android开发环境

6 第一个Android程序



Android Studio环境配置

- 安装JDK，配置基本Java环境
- 安装Android Studio 3.1.4，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简介



- Android提供的SDK有三个版本：
 - Windows
 - Mac OS X
 - Linux



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



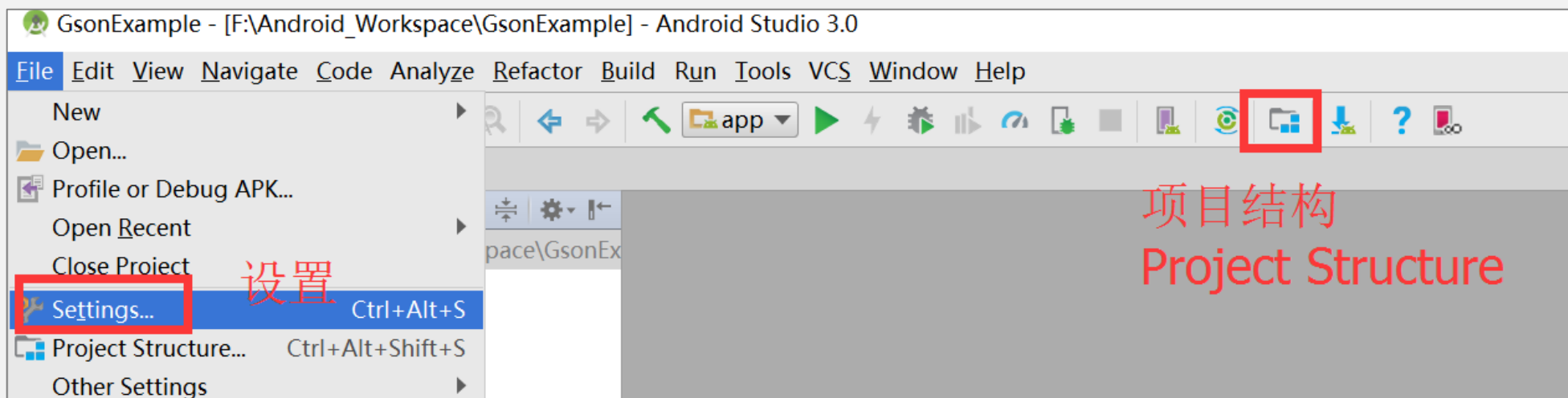
Gradle环境配置

- 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

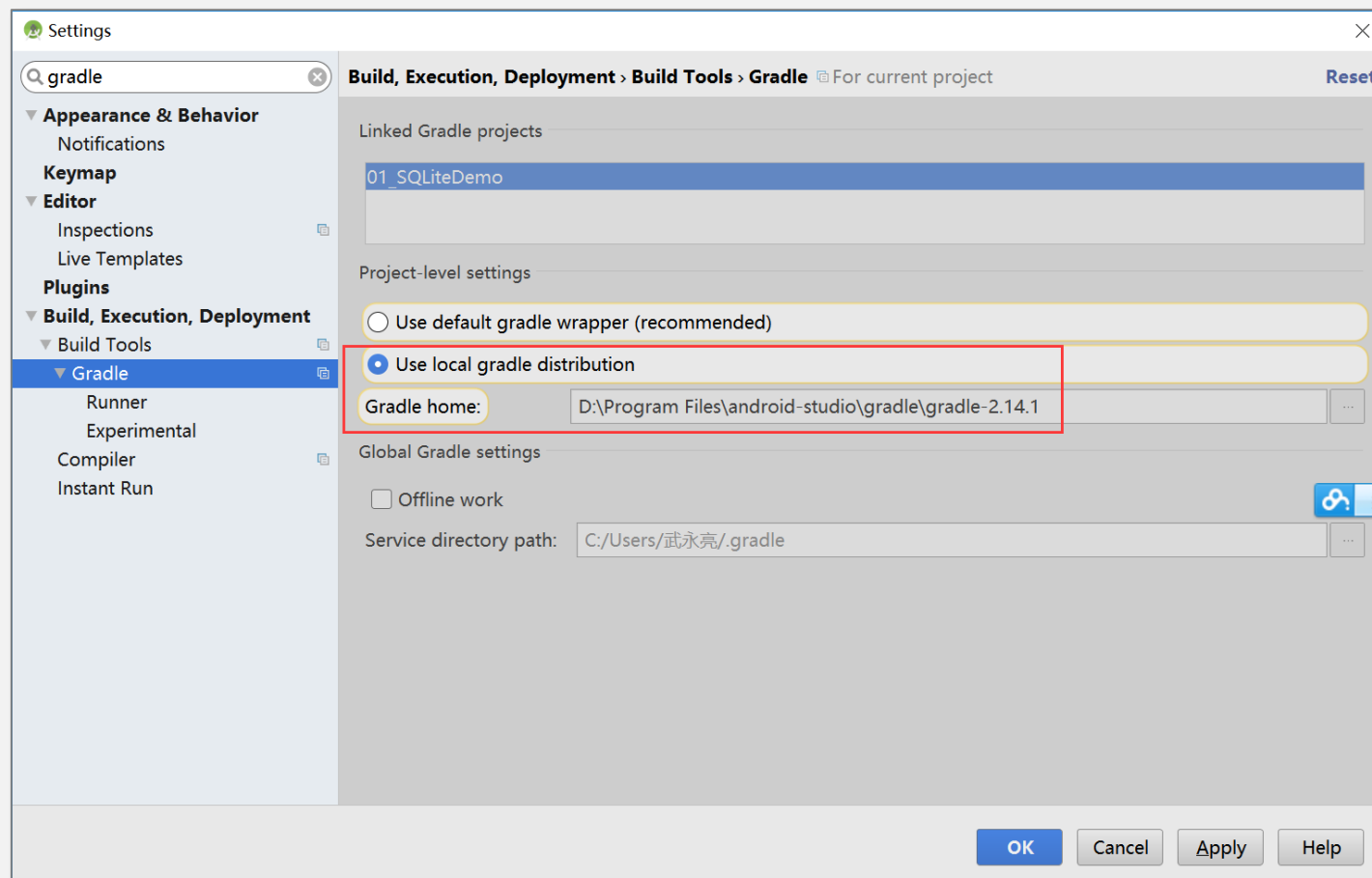
- 在Android studio中配置工程使用Gradle





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



Android程序调试环境



- Genymotion虚拟机安装配置
- <http://jingyan.baidu.com/article/3ea51489e7d8bd52e61bba36.html>



目录

1 课程介绍

2 Android简介

3 Android学习路线

4 Android系统框架

5 Android开发环境

6 第一个Android程序



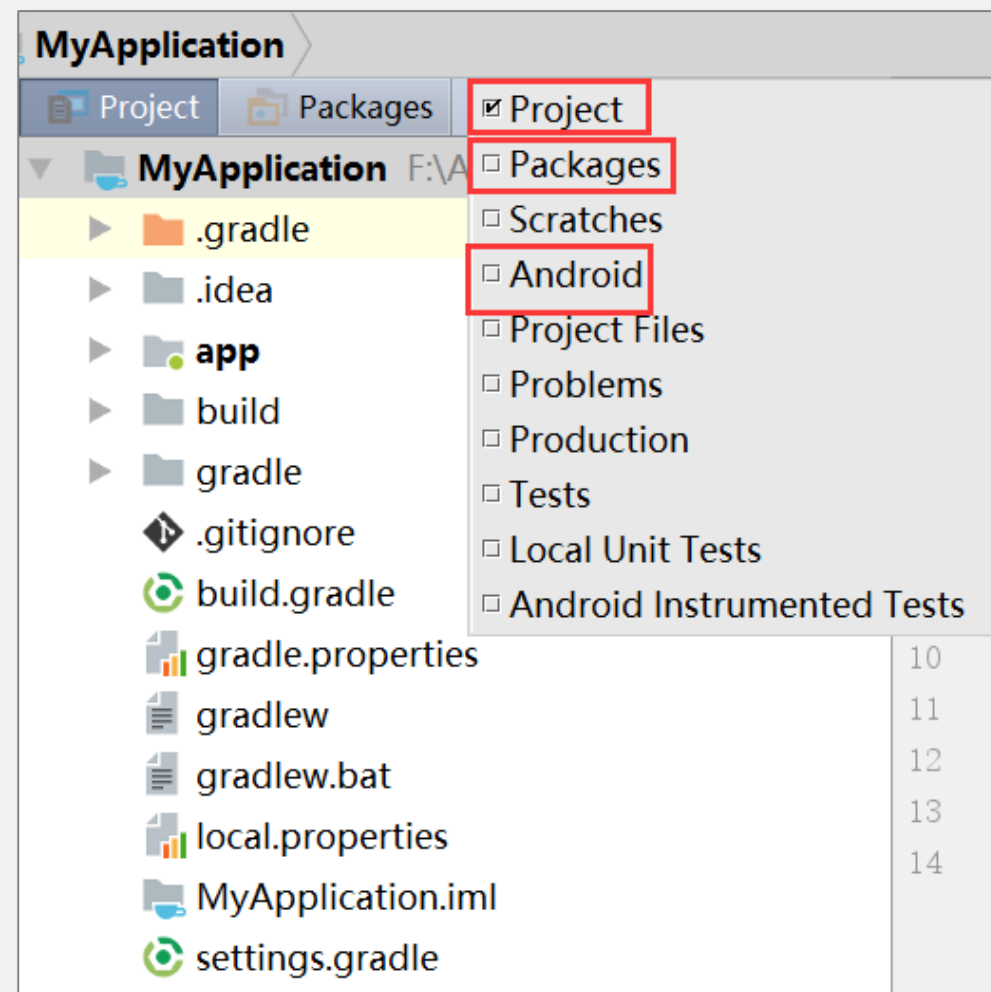
Android工程建立

- Android工程建立演示
- 第一个Android程序的编写运行
- Android工程导出
- Android工程导入



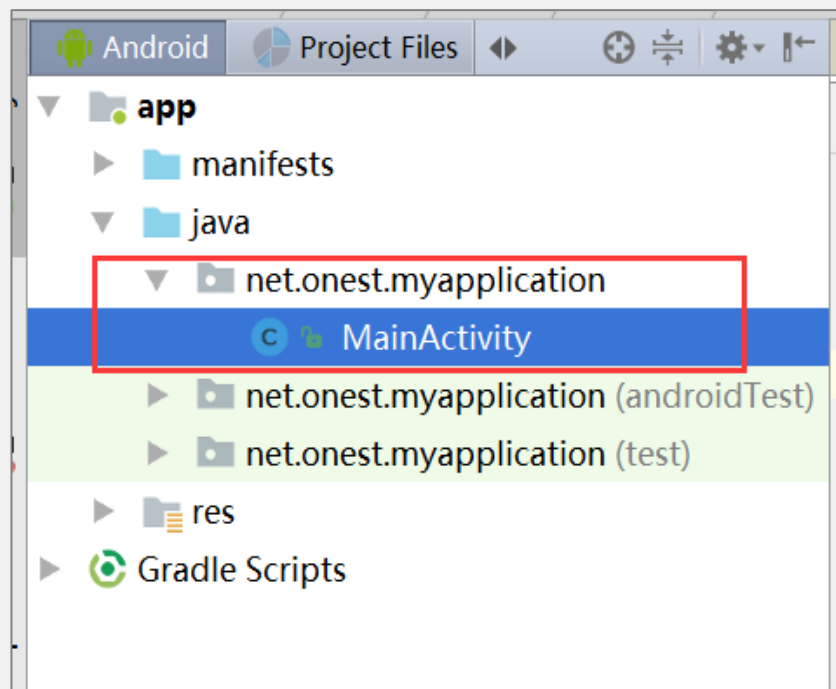
Android Studio视图

- **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);  
    }  
}
```



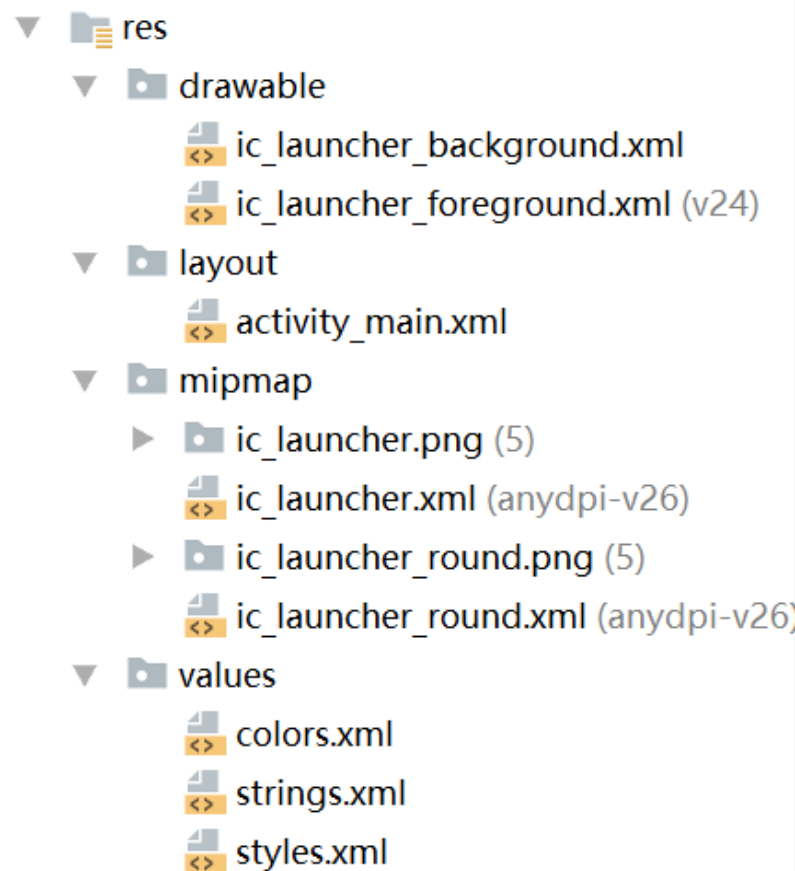
Android程序结构

- **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

- main.xml文件的代码

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.onest.myapplication.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```



Android程序结构

- strings.xml文件的代码

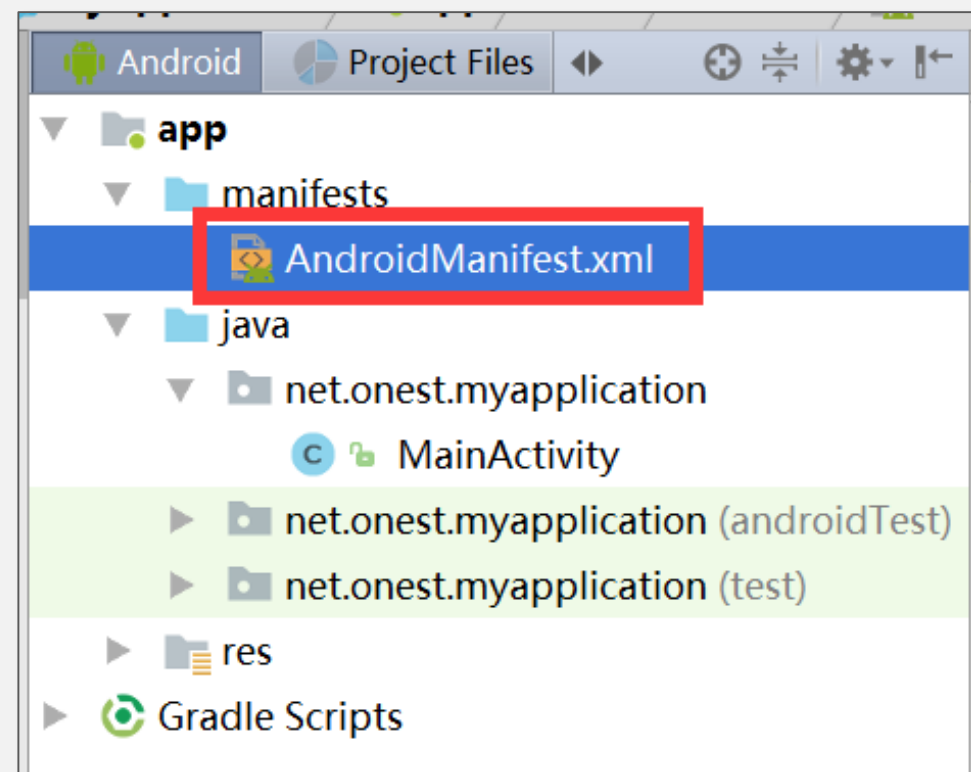
```
<resources>  
    <string name="app_name">My Application</string>  
</resources>
```

- 定义项目中用到的字符串常量



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





Android程序结构

- AndroidManifest.xml文件的代码

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.onest.myapplication">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```




Android程序结构

- AndroidManifest.xml文件的根元素是manifest，子属性包括xmlns:android、package、android:versionCode和android:versionName等
 - xmlns:android定义了Android的命名空间，值为<http://schemas.android.com/apk/res/android>
 - package定义了应用程序的包名称
 - android:versionCode定义了应用程序的版本号，是一个整数值，数值越大说明版本越新，但仅在程序内部使用，并不提供给应用程序的使用者
 - android:versionName定义了应用程序的版本名称，是一个字符串，仅限于为用户提供一个版本标识



Android程序结构

- manifest元素仅能包含一个application元素，application元素中能够声明Android程序中最重要四个组成部分，包括Activity、Service、BroadcastReceiver和ContentProvider，所定义的属性将影响所有组成部分
 - 属性android:icon定义了Android应用程序的图标，其中@mipmap/ic_launcher是一种资源引用方式，表示资源类型是图像，资源名称为ic_launcher，对应的资源文件为res/mipmap目录下的ic_launcher.png
 - 属性android:label则定义了Android应用程序的标签名称



Android程序结构

- activity元素是对Activity子类的声明，必须在AndroidManifest.xml文件中声明的Activity才能在用户界面中显示
 - 属性android:name定义了实现Activity类的名称，可以是完整的类名称，也可以是简化后的类名称
 - 属性android:label则定义了Activity的标签名称，标签名称将在用户界面的Activity上部显示，@string/app_name属于资源引用，表示资源类型是字符串，资源名称为app_name，资源保存在res/values目录下的strings.xml文件中

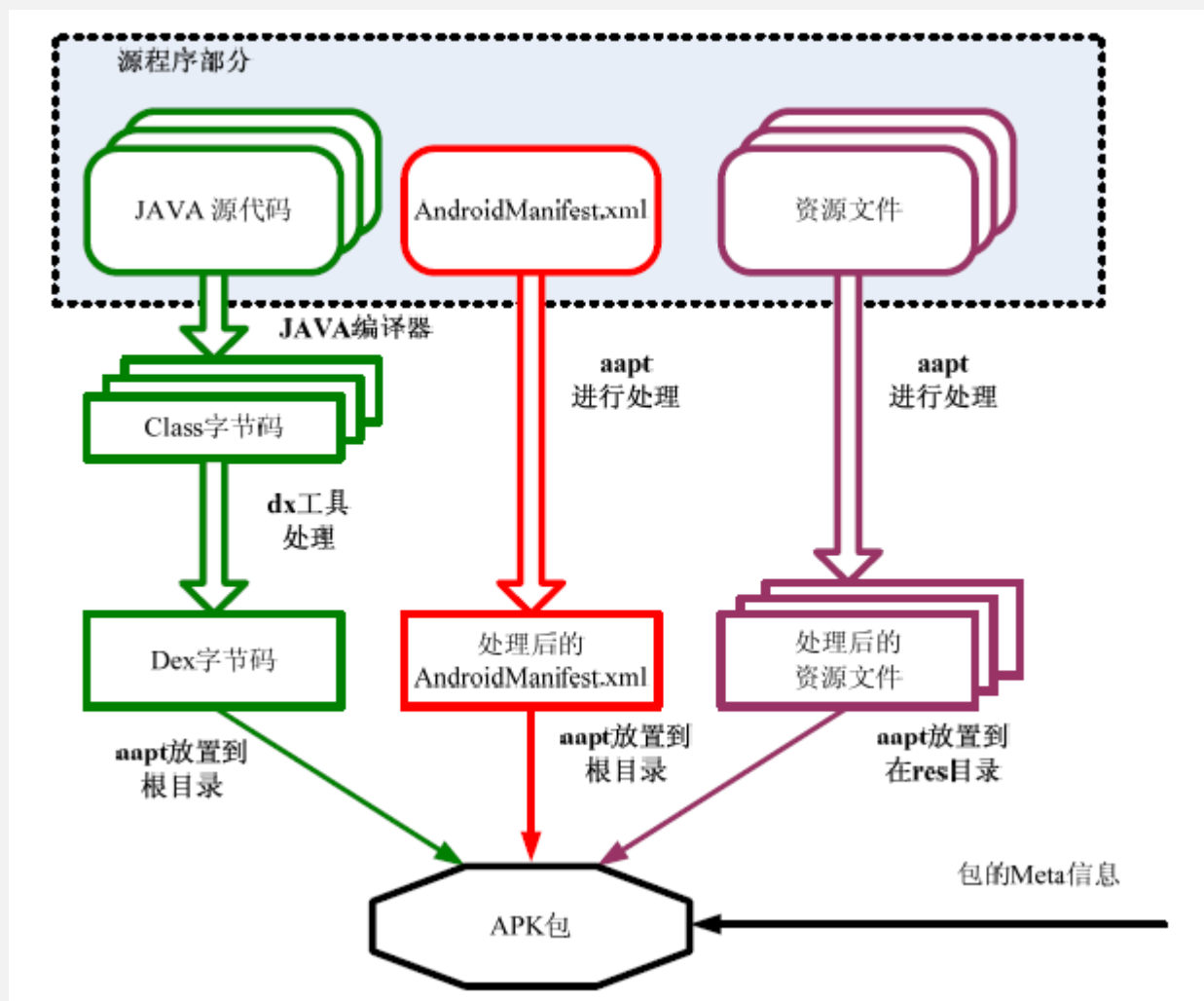


Android程序结构

- intent-filter中声明了两个子元素action和category, intent-filter使程序在启动时, 将MainActivity这个Activity作为默认启动模块



Android应用程序的构成

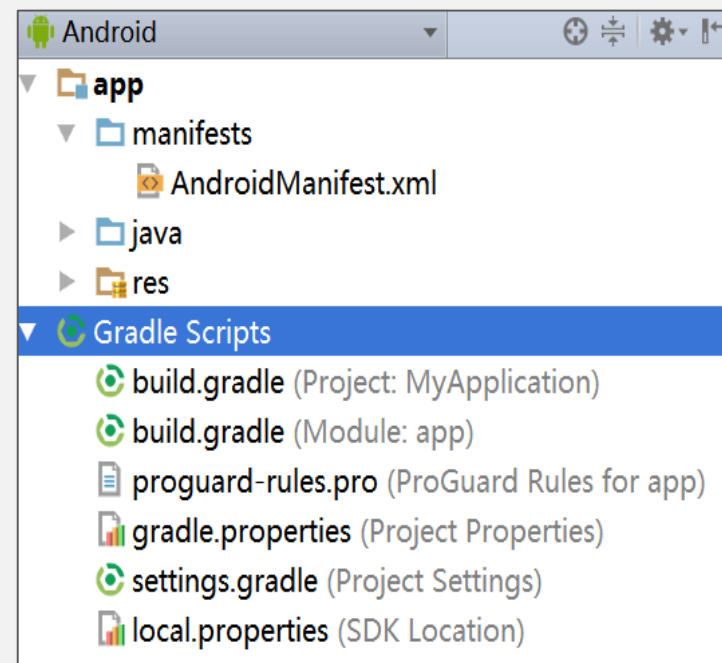




Android程序结构



- Gradle相关文件
- Project: build.gradle
 - Module1: build.gradle
 - Module2: build.gradle
 - Module3: build.gradle
 - ...





Android课堂练习

- 第一个Android程序编写运行，显示自己的名字
- Android工程导出，找到工程目录复制
- Android工程导入
 - AS工程导入AS
 - Eclipse工程导入Android Studio（问题会多）
 - 修改Android SDK的版本
 - 修改gradle的版本



目录

1 课程介绍

2 Android简介

3 Android学习路线

4 Android系统框架

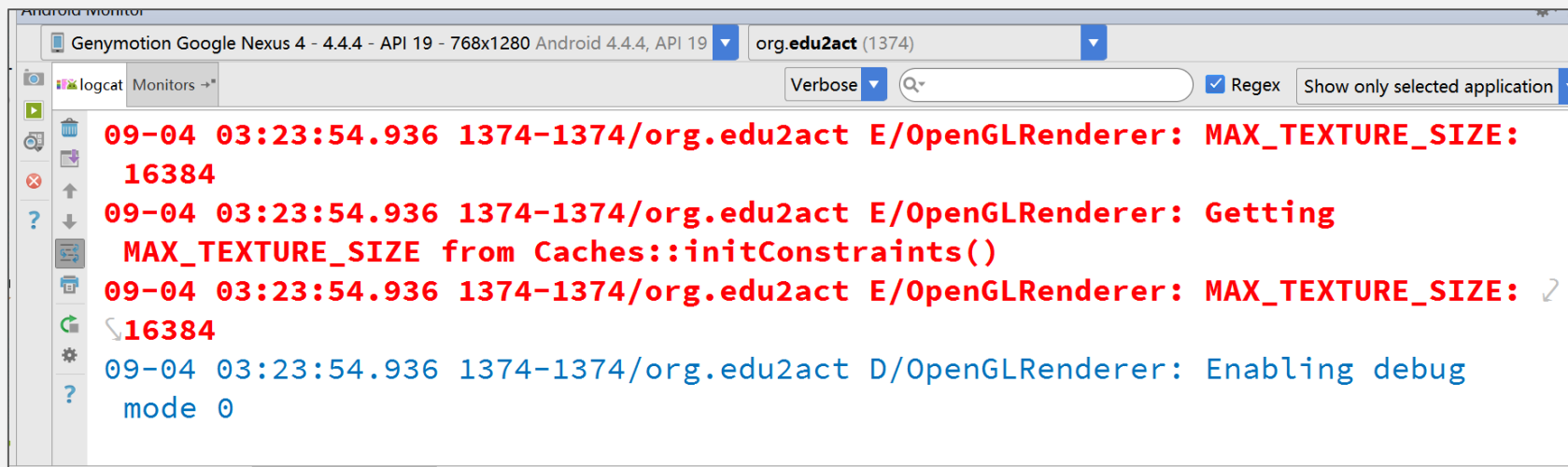
5 Android开发环境

6 第一个Android程序



LogCat的使用

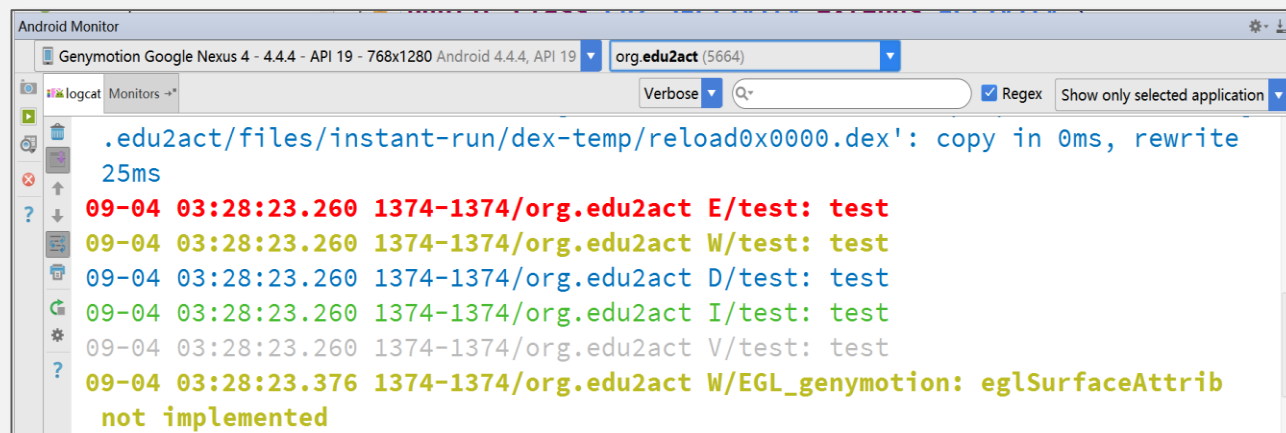
- LogCat工具是查看系统Log信息的工具，可以获得Android系统运行时打印的log信息。





LogCat的使用

- LogCat的右上方的五个字母表示五种不同类型的日志信息，他们的级别依次增高
 - [V]: 详细 (Verbose) 信息
 - [D]: 调试 (Debug) 信息
 - [I]: 通告 (Info) 信息
 - [W]: 警告 (Warn) 信息
 - [E]: 错误 (Error) 信息





LogCat的使用

- 在LogCat中，用户可以通过五个字母图标选择显示的信息类型，级别高于所选类型的信息也会在LogCat中显示，但级别低于所选类型的信息则不会被显示

Create New Logcat Filter

Filter Name:

Specify one or several filtering parameters:

Log Tag: ☒ Regex

Log Message: ☒ Regex

Package Name: ☒ Regex

PID:

Log Level:

OK Cancel



LogCat的使用

- LogCat程序调试步骤
 - 引入android.util.Log包
 - 使用Log.v()、 Log.d()、 Log.i() 、 Log.w() 和 Log.e()五个函数在程序中设置 “日志点”
 - 当程序运行到 “日志点” 时，应用程序的日志信息便被发送到 LogCat中
 - 判断 “日志点” 信息与预期的内容是否一致
 - 进而判断程序是否存在错误
 - Log.v()用来记录详细信息， Log.d()用来记录调试信息， Log.i()用来记录通告信息， Log.w()用来记录警告信息， Log.e()用来记录通错误信息



Thank you!

