



# 面向对象的软件构造实践

2024春



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ



需求分析

## 安卓平台飞机大战APP

单机

- 三种难度模式，三种敌机
- 碰撞检测，生命值及得分计算
- 道具生成及生效
- 排行榜
- 音效

网络

- 联机对战同步得分

模块分解

用户界面

事件处理

图形系统

数据存储  
与展示

音乐音效

网络编程

飞机大战APP模块图

用户界面

事件处理

图形系统

数据存储  
与展示

音乐音效

网络编程

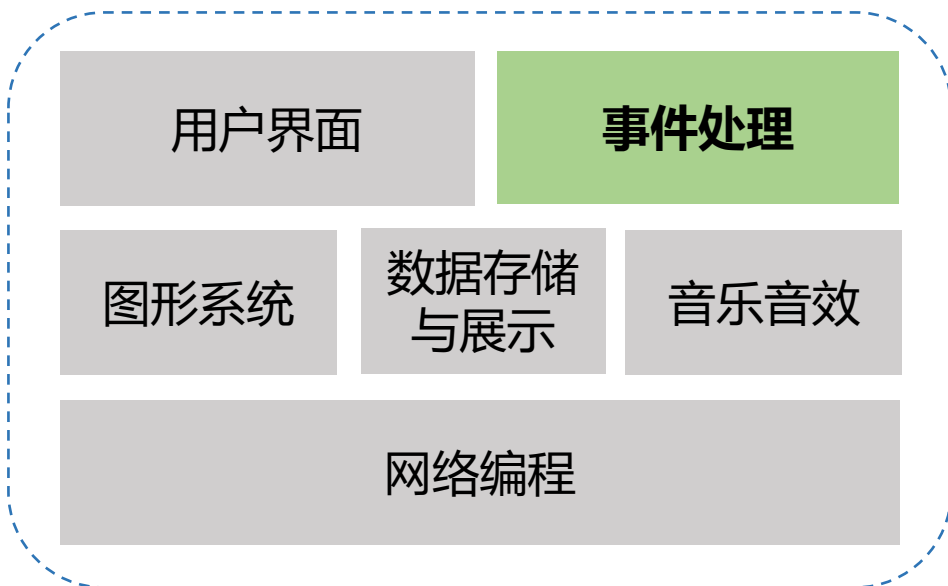
## 模块功能：完成静态页面的设计

① 完成游戏首页的  
界面设计



② 完成游戏难度选择  
页面的界面设计





**模块功能：完成页面切换和数据传递**

- ① 使用事件监听机制，完成从游戏首页到难度选择页面的跳转和数据传递



用户界面

事件处理

**图形系统**

数据存储  
与展示

音乐音效

网络编程

**模块功能：使用图形处理API完成绘制**

- ① 使用SurfaceView和图形处理API，完成不同难度下游戏主界面的开发及实时分数和生命值的显示



用户界面

事件处理

图形系统


数据存储  
与展示

音乐音效

网络编程

模块功能：完成数据存储和展示

① 完成用户数据的存储，及得分页面的展示



排名	用户	得分	时间
1	test	70	Mon Jul 17 02:57:08 GMT 2023
2	test	60	Mon Jul 17 02:58:36 GMT 2023
3	test	50	Mon Jul 17 02:56:46 GMT 2023
4	test	20	Mon Jul 17 02:56:30 GMT 2023
5	test	20	Mon Jul 17 02:58:58 GMT 2023
6	test	0	Mon Jul 17 02:57:20 GMT 2023
7	test	0	Mon Jul 17 03:01:08 GMT 2023
8	test	0	Mon Jul 17 03:02:11 GMT 2023
9	test	0	Mon Jul 17 08:18:50 GMT 2023

用户界面

事件处理

图形系统

数据存储  
与展示

**音乐音效**

网络编程

## 模块功能：实现音乐音效控制

### ① 实现各种音乐音效的播放和停止

```
? bgm.wav  
? bgm_boss.wav  
? bomb_explosion.wav  
? bullet.wav  
? bullet_hit.wav  
? game_over.wav  
? get_supply.wav
```

用户界面

事件处理

图形系统

数据存储  
与展示

音乐音效

网络编程

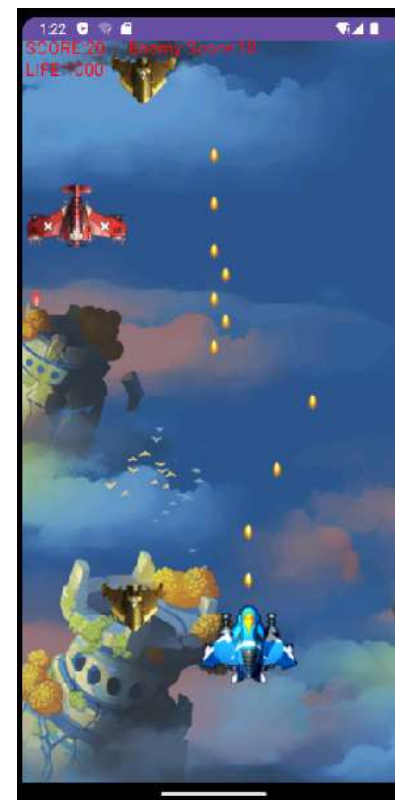
## 模块功能：通过网络传递数据

- ① 完成服务器端的开发，实现联机对战功能，在对战过程中实时显示对手分数

玩家1



玩家2





实验项目	学时	授课内容	实验内容
实验1	2	<ul style="list-style-type: none"><li>• Android发展历史</li><li>• Android架构</li><li>• Android开发环境</li><li>• 开发程序的一般过程</li><li>• Android项目结构</li><li>• Logcat工具</li></ul>	<ul style="list-style-type: none"><li>• 熟悉实验环境</li><li>• 创建一个空项目</li></ul>
实验2	4	<ul style="list-style-type: none"><li>• 布局</li><li>• 视图控件</li><li>• 活动Activity</li><li>• Android事件处理</li><li>• 意图Intent</li></ul>	<ul style="list-style-type: none"><li>• 创建游戏首页和单机模式选择页面</li><li>• 通过按钮点击事件实现从游戏首页到单机模式选择页面的跳转</li></ul>
实验3	2	<ul style="list-style-type: none"><li>• Android图形处理API</li><li>• Android SurfaceView</li></ul>	<ul style="list-style-type: none"><li>• 导论代码移植</li><li>• 使用SurfaceView实现游戏主界面</li></ul>

实验项目	学时	授课内容	实验内容
实验4	4	<ul style="list-style-type: none"><li>• Android数据存储</li><li>• 异步通信机制Handler</li><li>• 视图控件AlertDialog, Toast</li><li>• Android上下文Context</li></ul>	<ul style="list-style-type: none"><li>• 使用ListView和SimpleAdapter实现排行榜页面</li><li>• 使用文件存储排行榜数据</li></ul>
实验5	2	<ul style="list-style-type: none"><li>• Media Player</li><li>• Sound Pool</li></ul>	<ul style="list-style-type: none"><li>• 播放背景音乐和音效</li></ul>
中期检查	2		<ul style="list-style-type: none"><li>• 检查单机功能完成情况</li></ul>
实验6	6	<ul style="list-style-type: none"><li>• Android网络编程</li></ul>	<ul style="list-style-type: none"><li>• 实现联机对战、同步得分</li></ul>
结题检查	2		<ul style="list-style-type: none"><li>• 检查全部实验任务完成情况</li></ul>
总计	24		

考察点	分数	评分标准
任务书	10分	见任务书模板
中期检查	20分	见中期检查细则
结题检查	50分	见结题检查细则
结题报告	20分	见结题报告模板

# 中期检查标准

项目	分数	检查点	分数
实验1	20分	• 所有团队成员均能在模拟器中成功运行项目	20分
实验2	20分	• 游戏首页有一个“开始游戏”按钮和一组音乐开关，默认关闭音乐	10分
		• 单机难度选择界面有三个难度选择按钮	5分
		• 点击开始游戏，从游戏首页跳转到单机难度选择界面	5分
实验3	20分	• 根据游戏难度的不同，进入不同的游戏界面	8分
		• 实现飞机大战游戏的基本功能，生成敌机、子弹发射、碰撞检测、道具生成等	6分
		• 在左上角实时显示英雄机的得分和血量	6分

# 中期检查标准

项目	分数	检查点	分数
实验4	20分	• 正确显示当前难度	4分
		• 显示当前难度对应的排行榜数据，包括排名，姓名，得分，时间	6分
		• 可以删除排行榜数据，有删除提示，删除后正确更新视图显示	4分
		• 能够正确保存当前排行榜数据，保存成功有提示	6分
实验5	20分	• 若音乐开关打开，游戏开始时播放背景音乐	4分
		• 若音乐开关关闭，游戏时不播放任何音频	4分
		• 若音乐开关打开，boss机出现时播放boss机音乐，背景音乐暂停，boss机消失后背景音乐继续，boss机音乐停止	4分
		• 若音乐开关打开，游戏结束时停止播放所有音频	4分
		• 若音乐开关打开，游戏时正确播放各种音效	4分
总计	100分		100分

项目	分数	检查点	分数
实验6	100分	• 点击联机对战进入匹配过程	20分
		• 匹配成功后进入游戏主界面	20分
		• 游戏过程中实时更新对方分数	20分
		• 本玩家死亡后，仍然实时更新对方分数	20分
		• 双方均死亡后，进入对战结果页面	20分

- 中期检查后完成的单机功能只能获得80%的分数。如某队中期检查得分80分，中期检查后完成了剩下的单机功能，则最终中期检查得分： $80+20*0.8=96$ 。
- 中期检查和结题检查均不设二次检查。

小组成员每人一份



班 号		学 号		姓 名	
院 系			专 业		
组 号		同组人员姓名			
任务书评分 (A、B、C、D、E 五级)					
<p>1. 请谈谈你对 Android 开发的理解。 (50 分)</p> <p>2. 请分析飞机大战 APP 的核心功能。</p> <p>(50 分)</p>					

说明：任务书及课程报告均需独立完成，依据各自理解及具体分工来撰写，不可雷同。

小组成员每人一份

- **需求分析与任务分工 (20分)**

描述每次实验完成的主要功能，列出由自己完成的任务序号，如1.1,2.1等。

- **网络通信方案 (30分)**

使用流程图或伪代码方式描述网络通信方案，包括使用的数据结构、核心算法等。

- **调试分析 (20分)**

测试时出现问题的调试分析过程、模块设计及算法设计存在的问题及改进方法。

- **系统核心功能运行结果与分析 (20分)**

软件运行的关键界面截图。

- **总结与建议 (10分)**

课程任务完成过程中的收获和体会，对课程设计的想法和建议。



- 自由组队，**每组1~2人**，每次实验均有两个任务，1人的队伍需独立完成所有任务，2人队伍可每人一个任务，组队成功后填写QQ群中的**共享文档**记录组队信息；
- 作业提交到作业系统<http://grader.tery.top:8001/>，用户名和初始密码均为学号，初次登录后请修改密码；
- 实验一结束一周后提交任务书，**每人一份**，组内成员请勿雷同；
- 结题检查一周后提交最终项目代码和结题报告，**项目代码可每组提交一份，结题报告每人一份。**

任务书（每位同学提交一份）命名格式：学号-姓名-任务书



# 面向对象的软件构造实践

实验一  
(2学时)

2024春



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ

- 了解Android发展、现状和未来趋势；
- 理解Android系统架构；
- 掌握Android简单的Android应用程序的创建及基本目录结构，掌握AndroidManifest文件的配置；
- 掌握Android开发环境Android Studio的安装和使用；
- 掌握Logcat工具，能够使用Logcat和调试器进行调试分析。

1.1 Android系统概述

1.2 Android平台架构

1.3 Android应用程序的一般过程

1.4 Android项目结构

1.5 Android调试工具

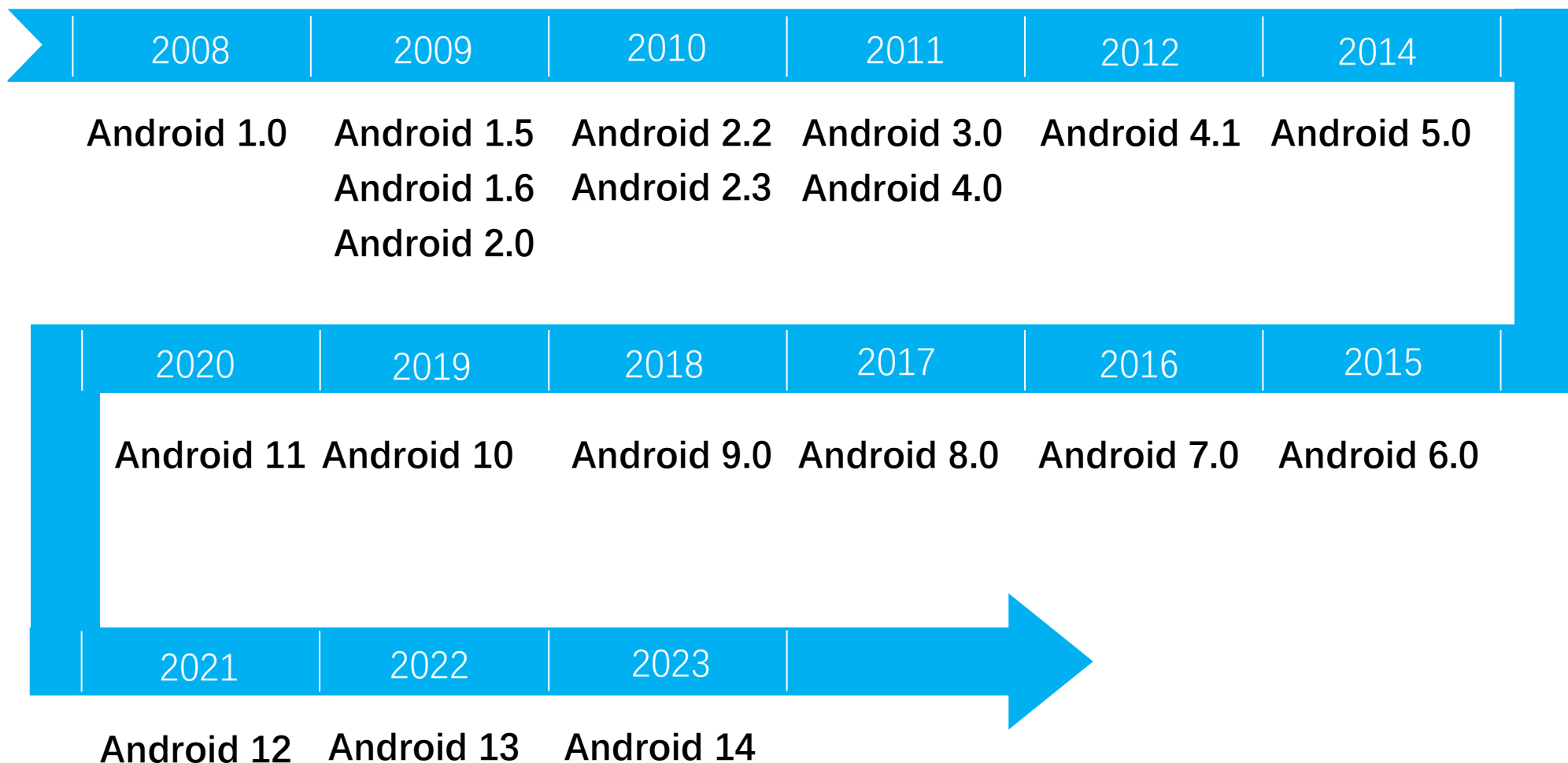
# 1.1 Android系统概述



- 由谷歌发布的开源操作系统
- 第一个完全定制、免费、开放的手机平台
- 较好的可移植性
  - 手机、平板电脑、电视、手表.....

# 1.1 Android系统概述

## Android平台的发展



# 1.1 Android系统概述

## Android API

**API级别：**唯一标识Android 平台版本提供的框架 API 。

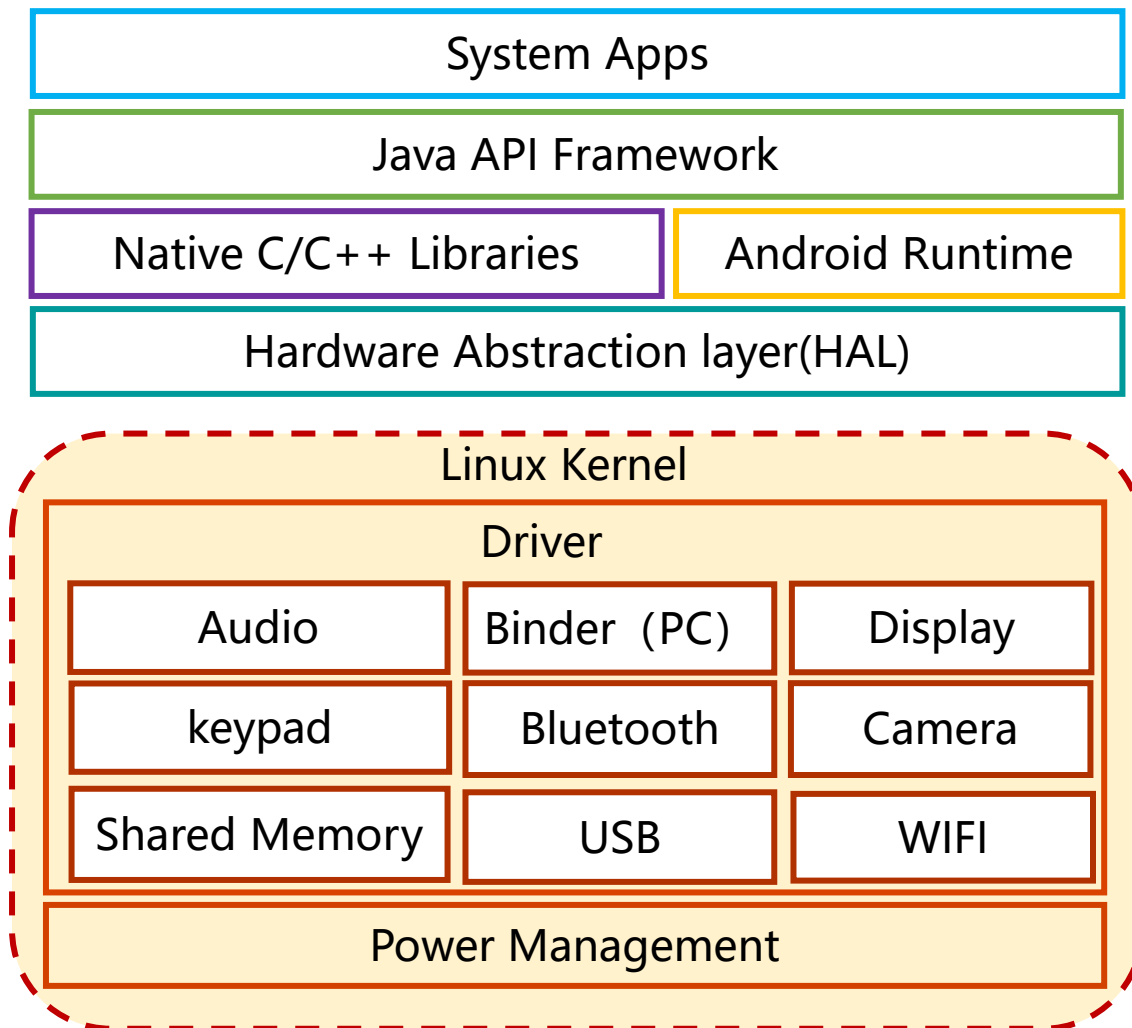
Android版本	API级别/SDK
Android 5.0	21
Android 5.1	22
Android 6.0	23
Android 7.0	24
Android 7.1	25
Android 8.0	26
Android 8.1	27
Android 9	28
Android 10	29
Android 11	30
Android 12	31
Android 12L	32
Android 13	33
Android 14	34

## 1.2 Android平台架构





## 1.2 Android平台架构



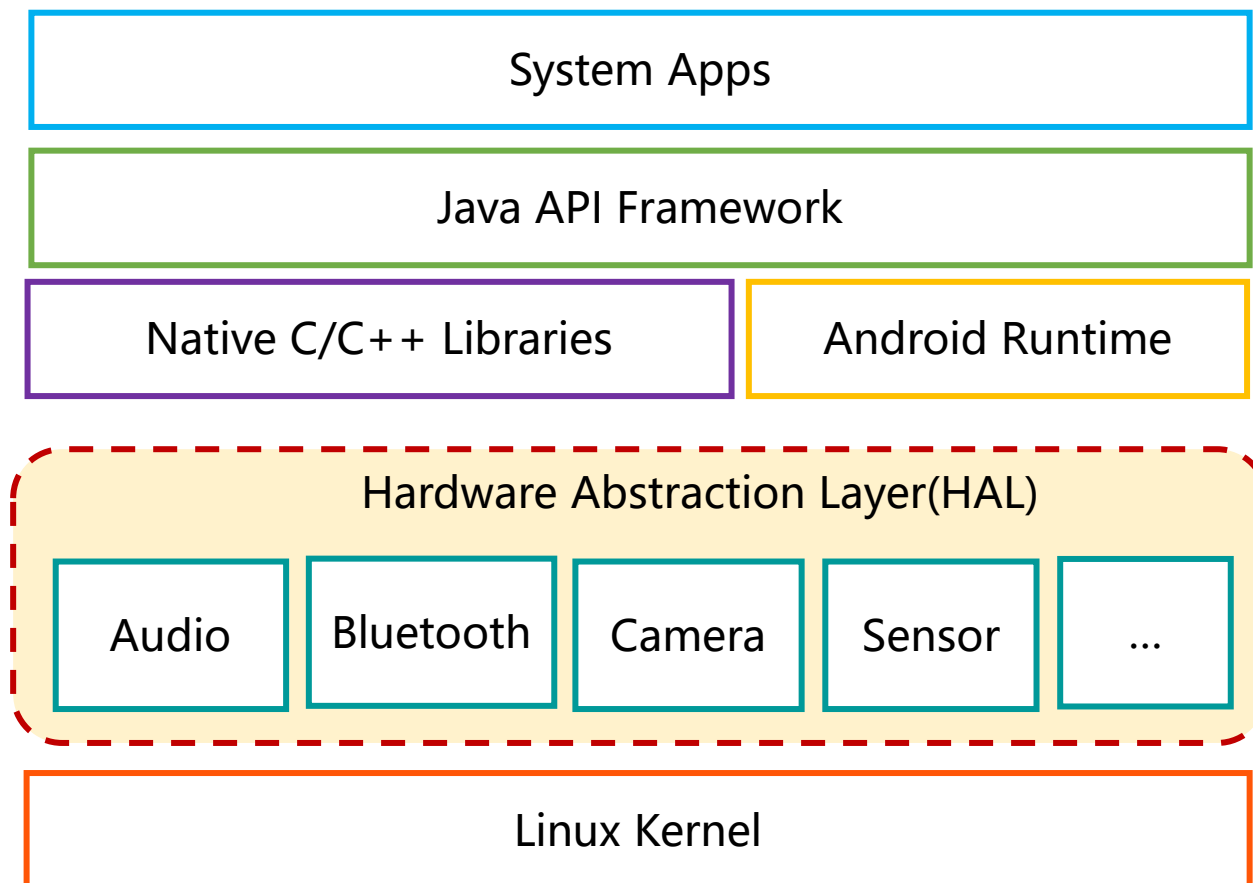
Android 是Google开发的基于Linux平台的开源手机操作系统。

### 1. Linux 内核

通过借助Linux内核服务实现一些核心功能：

- 硬件设备驱动
- 进程和内存管理
- 网络协议
- 电源管理
- 无线通信等。

# 1.2 Android平台架构



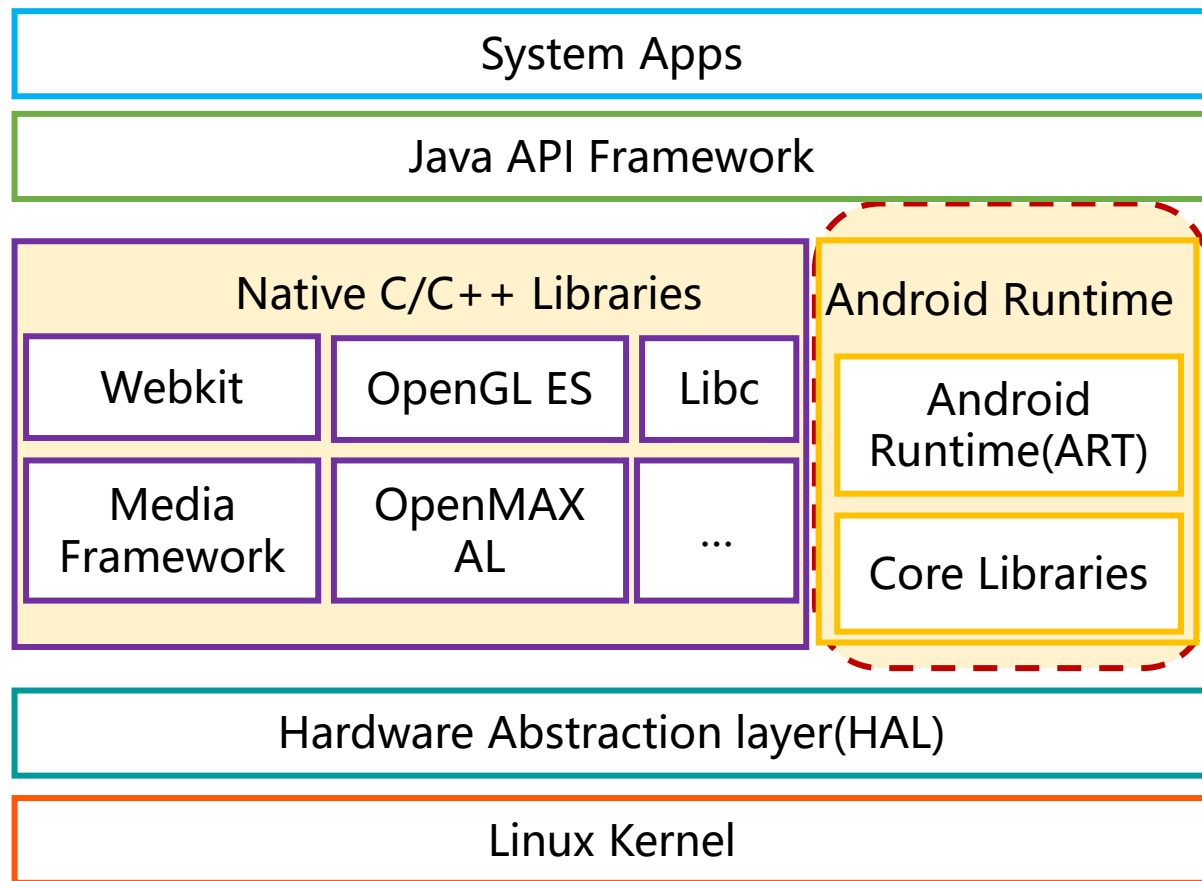
## 2. 硬件抽象层 (HAL)

隐藏底层硬件细节为上层提供统一硬件功能服务。

解释：对Linux内核驱动程序进行的**封装**，将硬件**抽象化**，屏蔽掉了底层的实现细节。

**HAL**包含多个库模块，每个模块为特定类型的硬件组件实现一个界面，例如相机或者蓝牙模块。当框架API要求访问设备硬件时，Android系统将为该硬件组件加载库模块。

## 1.2 Android平台架构

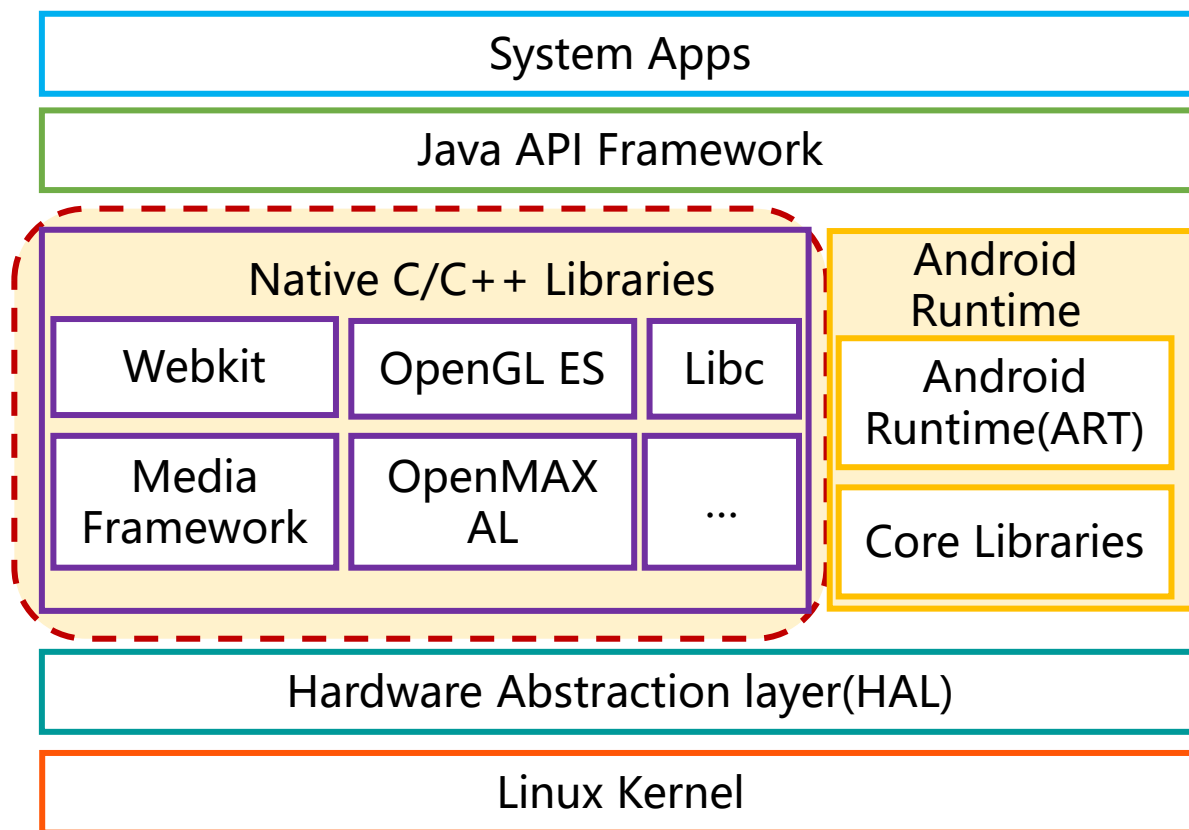


### 3. Android Runtime

- Android Runtime (缩写为ART) , 是一种在Android操作系统上的运行环境。
- ART主要作用是将APK文件中的DEX (一种专门为Android设计的字节码格式) 文件的字节码 (byte code) 翻译成机器可以直接执行的机器码 (machine code) 。

(Android APK: Android操作系统上用于安装和分发应用程序的文件格式)

## 1.2 Android平台架构

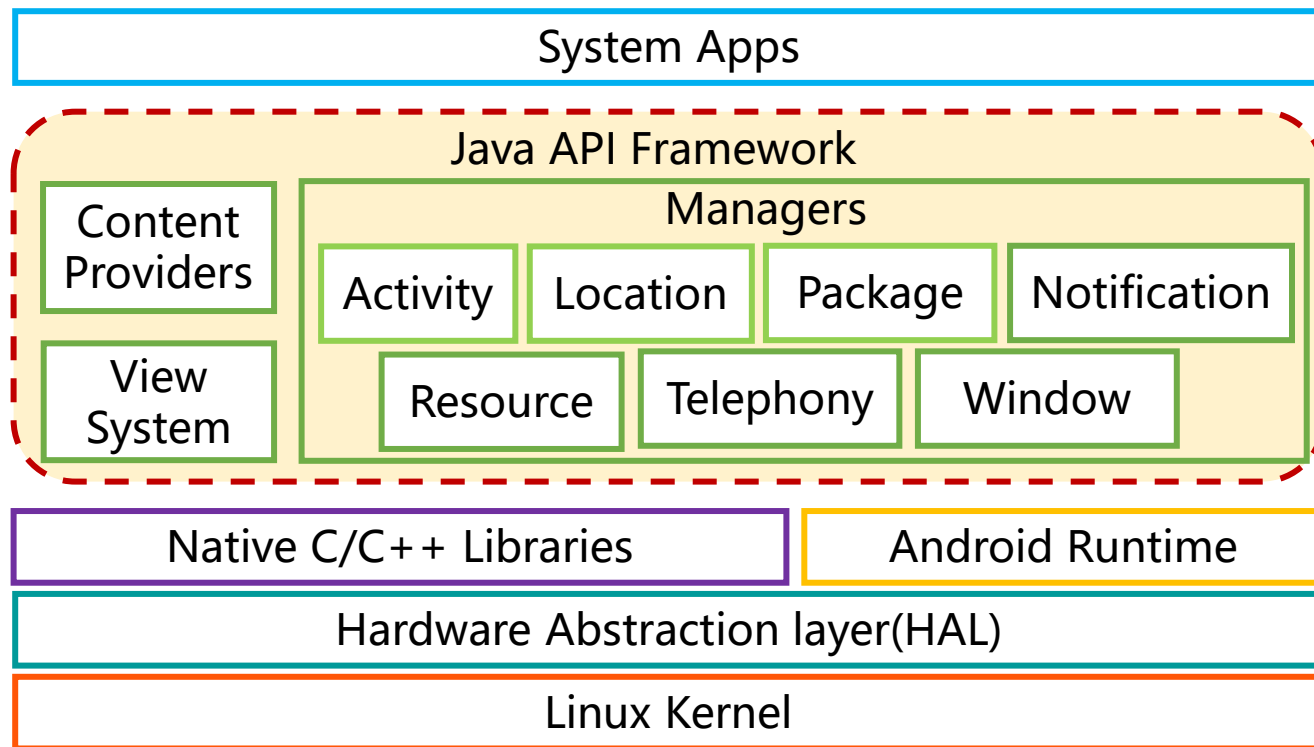


### 4. 原生C/C++库

原生C/C++是系统类库，许多核心Android系统组件和服务构建自原生代码，大部分由C/C++编写，所提供的功能通过**Android应用程序框架层**为开发者所使用。

- **OpenGL ES**: 可以通过Android框架的Java OpenGL API访问OpenGL ES，以支持在应用中绘制和操作2D和3D图形。
- **SQLite**: 本地小型关系数据库，Android提供了一些新的SQLite数据库API;

## 1.2 Android平台架构



资源管理器Resources:用于访问非代码资源，例如本地化的字符串、图形和布局文件。

通知管理器Notification: 让所有应用在状态栏中显示自定义提醒。

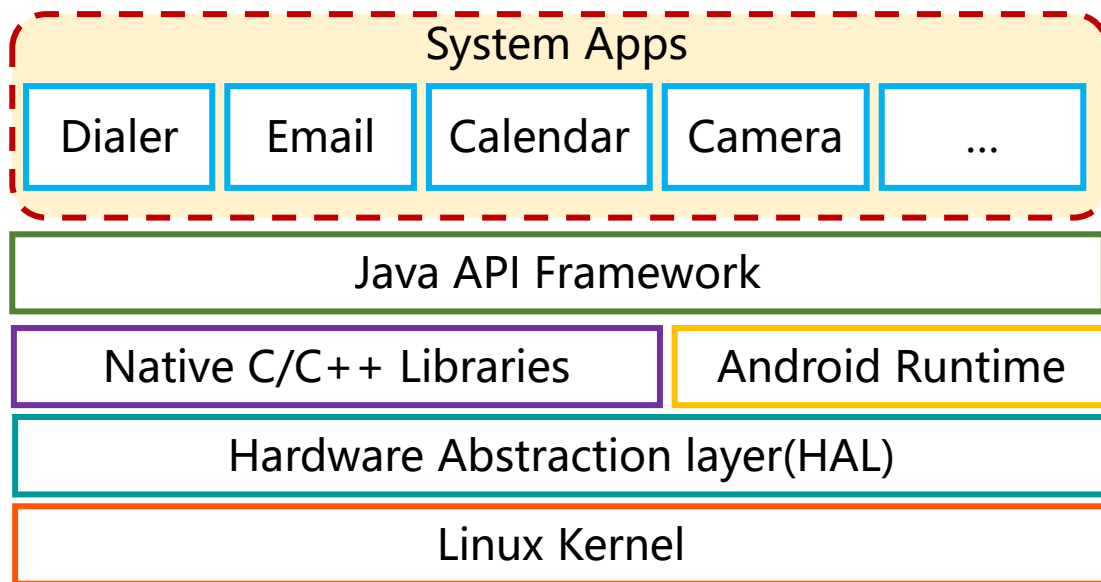
### 5. Java API Framework

- 通过以 Java 语言（一部分已由Kotlin实现）编写的 API 使用 Android OS 的整个功能集。
- 包含一系列类库，可以直接使用其中的组件来进行快速的应用开发，也可以通过继承实现个性化的拓展。

内容提供程序Content Providers: 可让应用访问其他应用（如“联系人”）中的数据或者共享自己的数据。

丰富、可扩展的视图系统View System: 包括列表、网格、文本框等。

## 1.2 Android平台架构



### 6. 系统应用层

- 主要由运行在Android设备上的所有应用程序构成，包括预装的系统应用和用户自己安装的第三方应用  
(通话、短信服务、日历日程、浏览器、联系人等)
- 应用层也支持应用程序之间的数据共享和交互
- 应用层还包括一些系统服务，如通知管理器
- 同时为开发者提供了强大的开发工具和API支持

### 了解Android系统架构分层的意义：

- ① 采用分层架构的思想，架构清晰，层次分明，协同工作；
- ② 不仅从宏观上认识了Android系统，同时也给我们的学习与实践指明了方向。
  - 若是从事Android应用开发，那应该研究Android的应用框架层和应用程序层；
  - 若是从事Android系统开发，那应该研究Android的系统库和Android运行时。
  - 若是从事Android驱动开发，那应该研究Android的Linux内核。

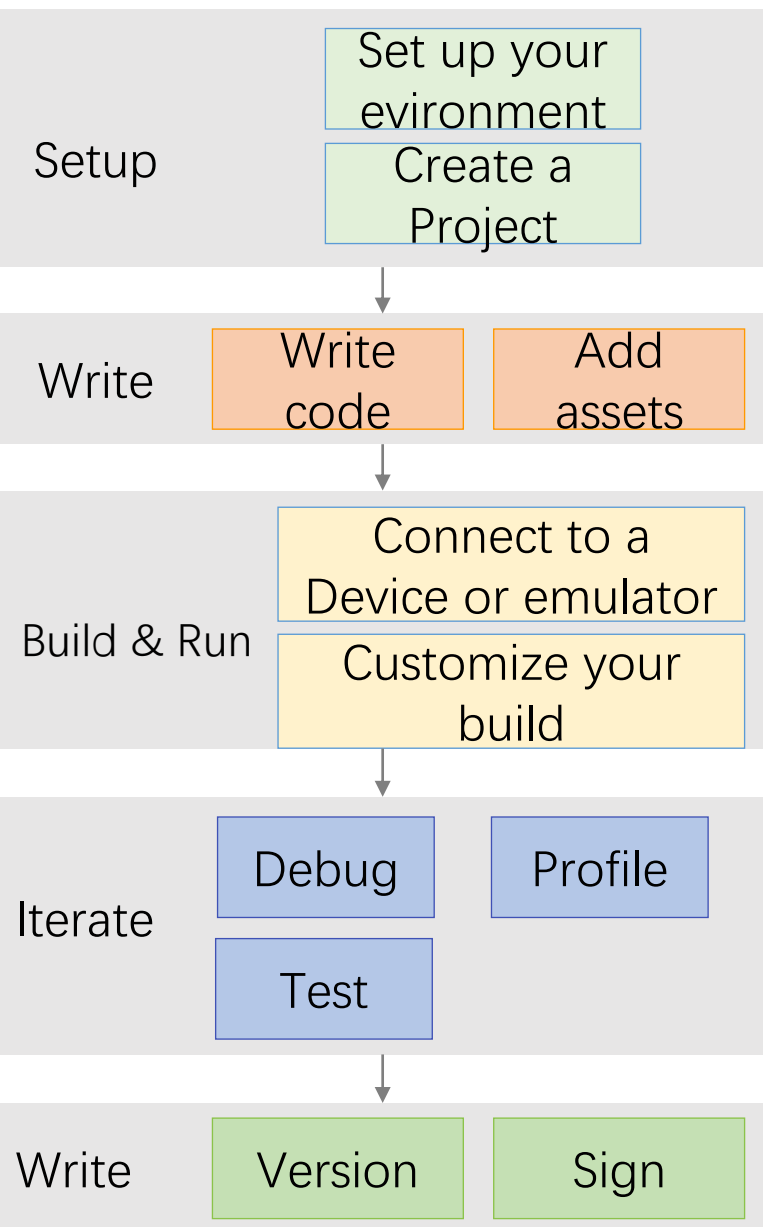
## 1.2 Android平台架构

### Android开发语言：Java和Kotlin对比

对比项	Java	Kotlin
语言特点	面向对象、静态类型、多平台	面向对象、静态类型、跨平台、简洁
语法简洁性	适中，相对冗长	简洁、减少样板代码
空安全	需要手动处理空指针异常	内建空安全特性，减少空指针异常
Lambda 表达式	支持、但语法较繁琐	支持、语法简洁
性能	优秀、广发测试和优化	与 Java 相当,因为最终编译为 Java 字节码
社区支持	庞大、历史悠久	迅速增长、官方推荐
学习曲线	适中	较平缓、入门容易精通困难
与 Android Studio 集成	原生支持	原生支持
与 Java 库和框架兼容性	完全兼容	完全兼容,并可轻松转换成 Java 代码



# 1.3 Android应用开发基本流程



## 1. 设置工作区

安装 Android Studio 并**创建项目**。

## 2. 编写应用

设计应用UI、交互、完成网络请求、数据处理等**业务逻辑编写**。

## 3. 构建并运行

**编译运行**，将项目构建成一个可调试的 APK 软件包，以便在模拟器或 Android 设备上安装和运行。

## 4. 调试、剖析和测试

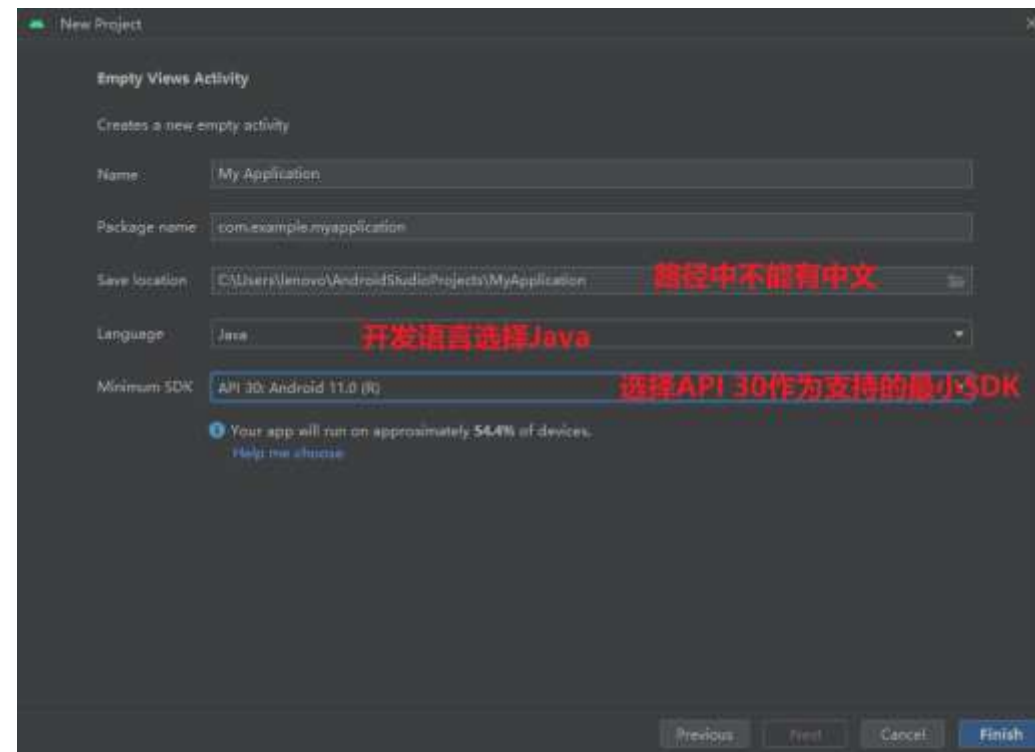
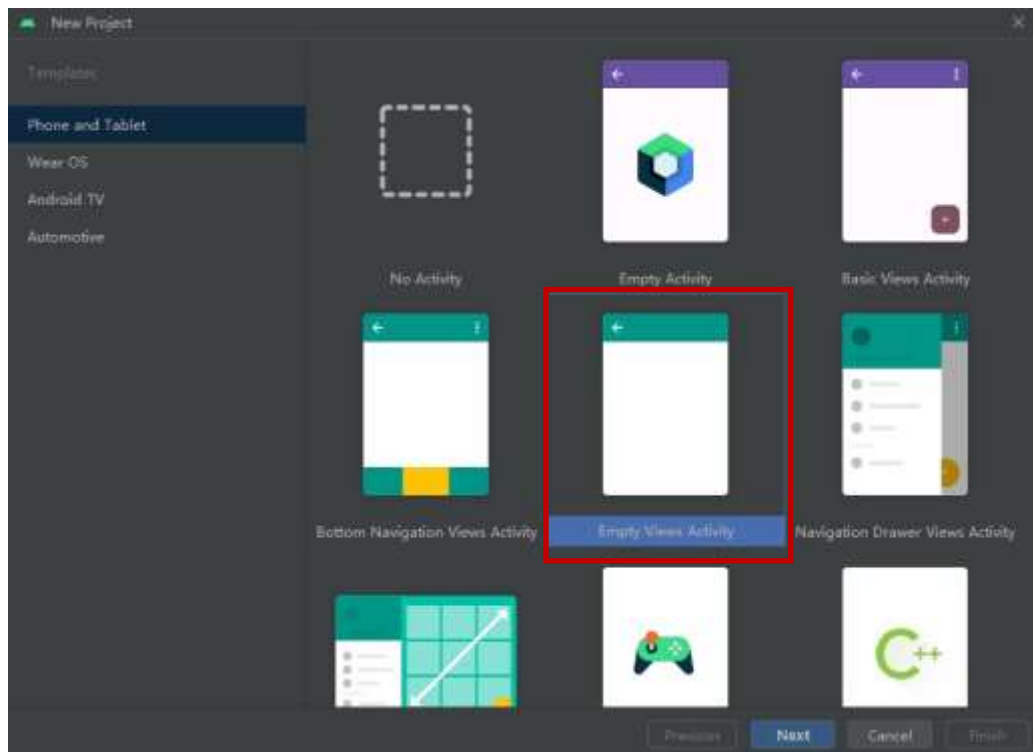
**消除错误并优化应用性能**。查看和分析各种性能指标（如内存使用情况、网络流量、CPU占用等）。

## 5. 发布

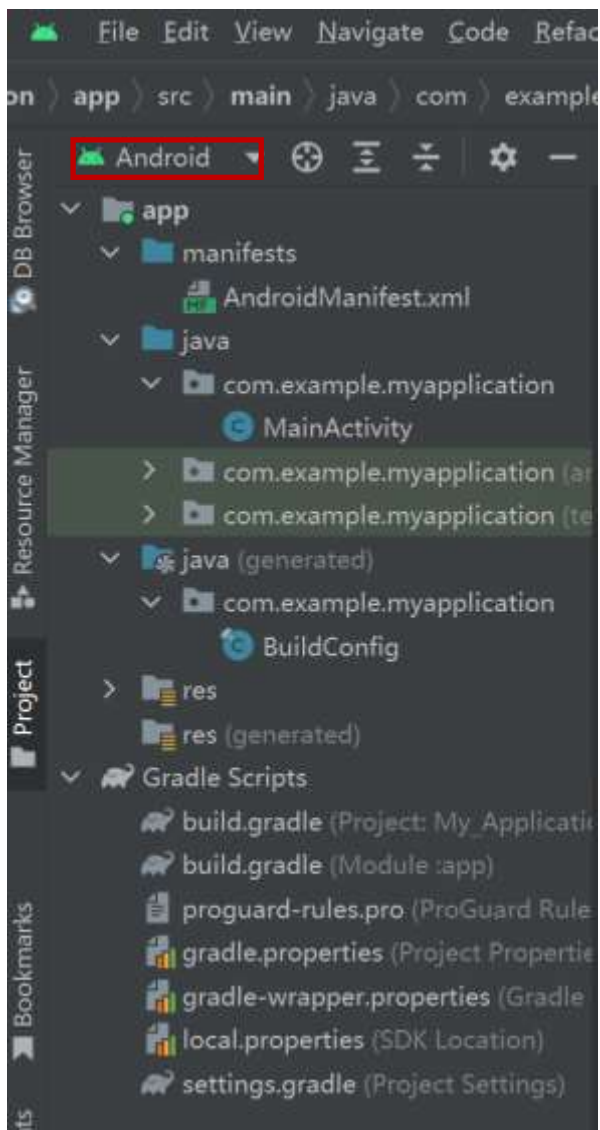
当准备向用户发布应用时，需要考虑管理应用版本、运营、构建 APK包、使用密钥为应用签名。

## 1.4 Android项目结构

### 创建 “Empty Views Activity” 项目



## 1.4 Android项目结构



### 以“Android”模式展示项目结构

- **app/manifests目录**

AndroidManifest.xml文件：提供软件包名，描述应用组件，权限声明等

- **app/java目录**

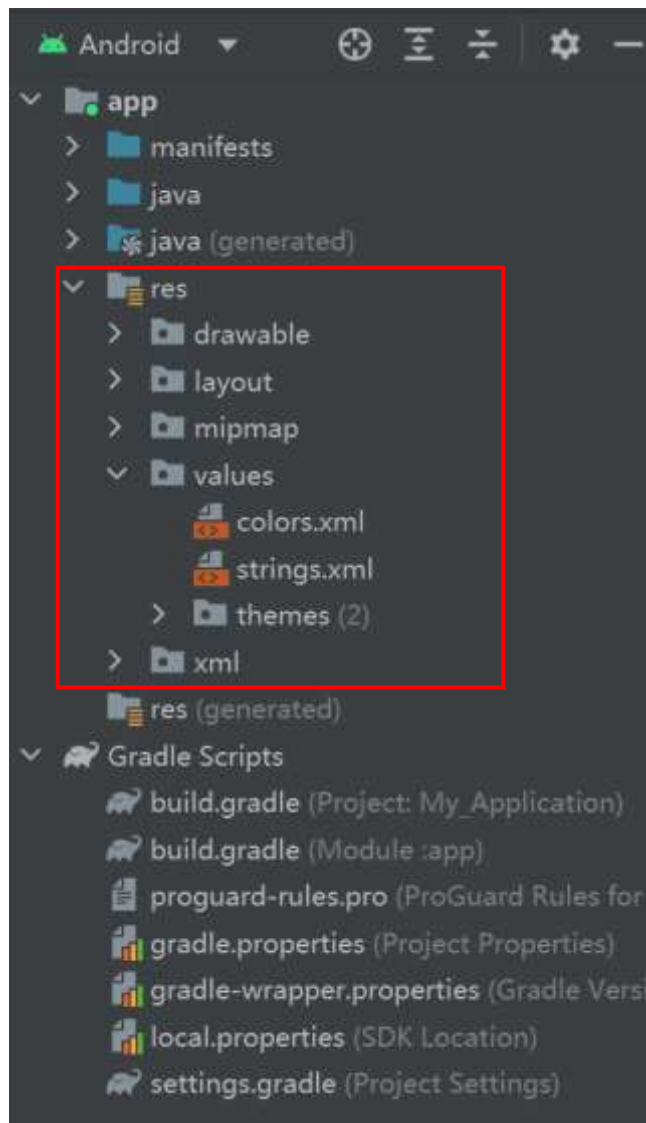
该目录用于存放Android应用项目中的所有Java代码文件。Java代码以用户声明的包进行自动地组织。其中MainActivity.java是默认的首个被运行的代码文件。

- **app/res目录：**

该目录存放整个项目经常使用的资源文件，称为资源目录。该目录包括项目中使用到的所有图标、图片、布局、声音、字符串、颜色、样式等资源参数描述文件。

需要注意的是，res目录中的所有文件名只能是以 a~z、0~9或 “\_” 字符命名，不能包含大写字母，且必须以字母开头，否则会导致错误。

# 1.4 Android项目结构



## res目录结构

- **图片资源目录**: 以drawable开头的目录用来存放图片文件, 有时还可以存放一些其他的drawable类型的XML文件。
- **图标资源目录**: 以mipmap开头的目录用来存放应用的图标。应用项目会根据不同分辨率设备, 启动相应分辨率的mipmap目录下的图标。
- **layout目录**: 该目录存放应用项目的布局文件, 文件类型为XML格式。新建项目时系统会自动创建一个activity\_main.xml文件。
- **values目录**: 该目录存放所有XML格式的资源描述文件, 一般根据英文含义来命名。例如:
  - colors.xml: 定义颜色资源。
  - string.xml: 定义字符串资源。
  - styles.xml: 定义主题资源。

### AndroidManifest.xml

- 每个 Android 项目必须有一个 AndroidManifest.xml 文件，当创建一个新的应用项目时，系统会自动生成此文件。
- 作用：
  - 描述应用项目的每个组件的类名称和组件能力（属性），帮助Android 系统了解这些组件以及在何种条件下可以启动这些组件。
  - 声明应用项目自身应该具有的权限，如应用程序需要访问系统功能（短信、联系人、相机和网络等），需要获取相关权限。

```
<?xml version="1.8" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

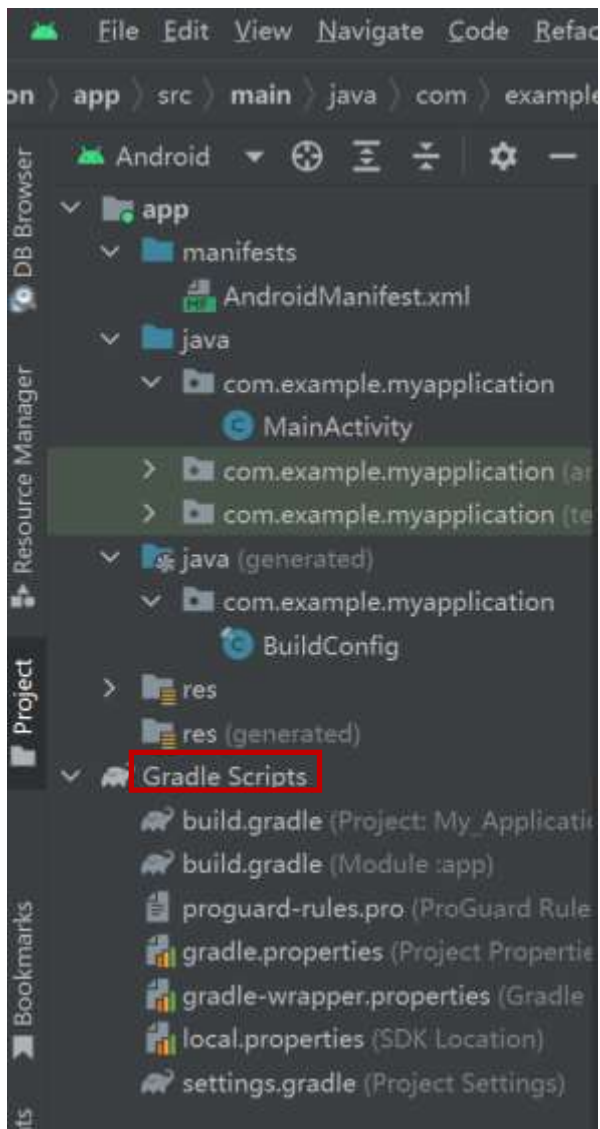
### AndroidManifest.xml文件说明

代码元素	说 明
<b>manifest</b>	xml文件的根结点，包含了package中所有的内容
<b>xmlns:android</b>	命名空间的声明。使得Android中各种标准属性能在文件中使用。
<b>uses-sdk</b>	声明应用程序所使用的Android SDK版本
<b>application</b>	<b>application</b> 级别组件的根结点。声明一些全局或默认的属性，如标签、图标、必要的权限等。
<b>android:icon</b>	应用程序图标
<b>android:label</b>	应用程序名称
<b>activity</b>	<b>Activity</b> 是一个应用程序与用户交互的图形界面。每一个 <b>Activity</b> 必须有一个< <b>activity</b> > 标记对应。
<b>android:name</b>	应用程序默认启动的活动程序 <b>Activity</b> 界面
<b>intent-filter</b>	声明一组组件支持的Intent值。在Android中，组件之间可以相互调用，协调工作，Intent提供组件之间通讯所需要的相关信息
<b>action</b>	声明目标组件执行的Intent动作。
<b>category</b>	指定目标组件支持的Intent 类别

- 思考：新增加一个Activity该如何修改AndroidManifest文件？



## 1.4 Android项目结构



### Gradle Scripts

Android 应用程序采用 Gradle 作为构建工具

**build.gradle(项目级别)**: 是项目全局的gradle构建脚本, 该文件的内容通常不需要修改。

**build.gradle (模块级别)**: app模块的gradle构建脚本, 一般用来管理app包名以及添加和修改依赖库。

**gradle.properties**: 全局的gradle配置文件, 一般不需要修改。

**gradle-wrapper.properties**: 配置使用的gradle版本、保存地址以及下载地址等。

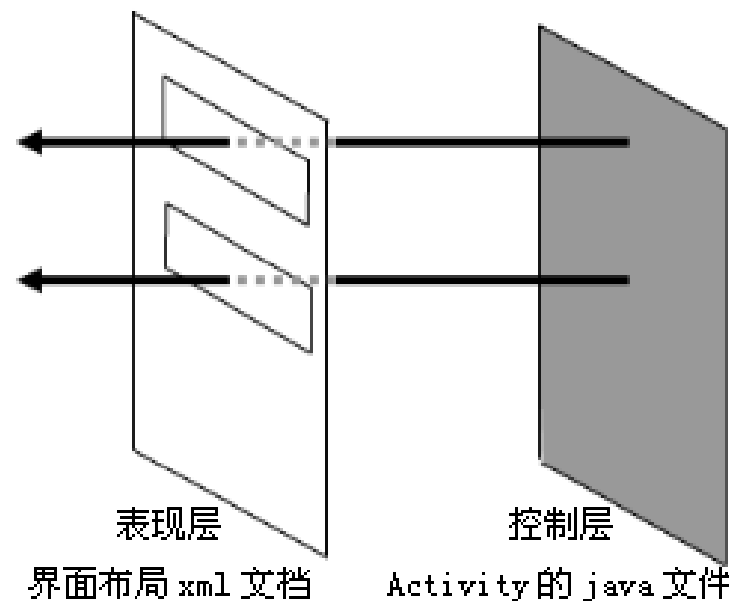
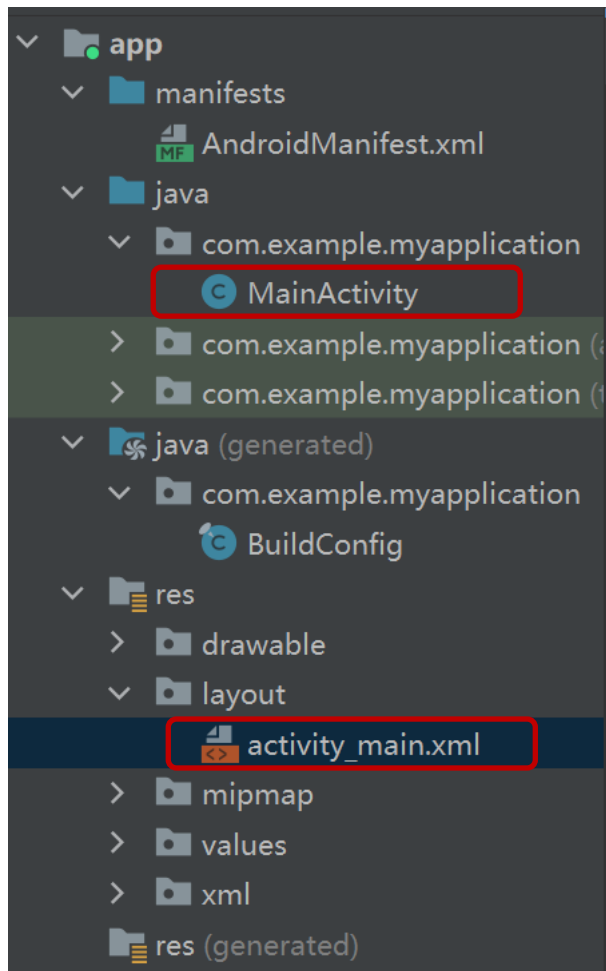
**local.properties**: 该文件用于指定本机中的Android SDK路径, 由系统自动生成。当Android SDK位置变化时, 需要在该文件更新路径。

**settings.gradle**: 该文件用于指定项目中所有引入的模块。

## 1.4 Android项目结构

### Android应用程序结构分析

- 一个简单的Android应用程序通常由Activity类程序（Java程序）和用户界面布局XML文件组成。
- 在Android应用程序中，逻辑控制层与表现层是分开的设计的。逻辑控制层由Java应用程序实现，表现层由XML文档描述。





### Android应用程序结构分析



#### MainActivity.java

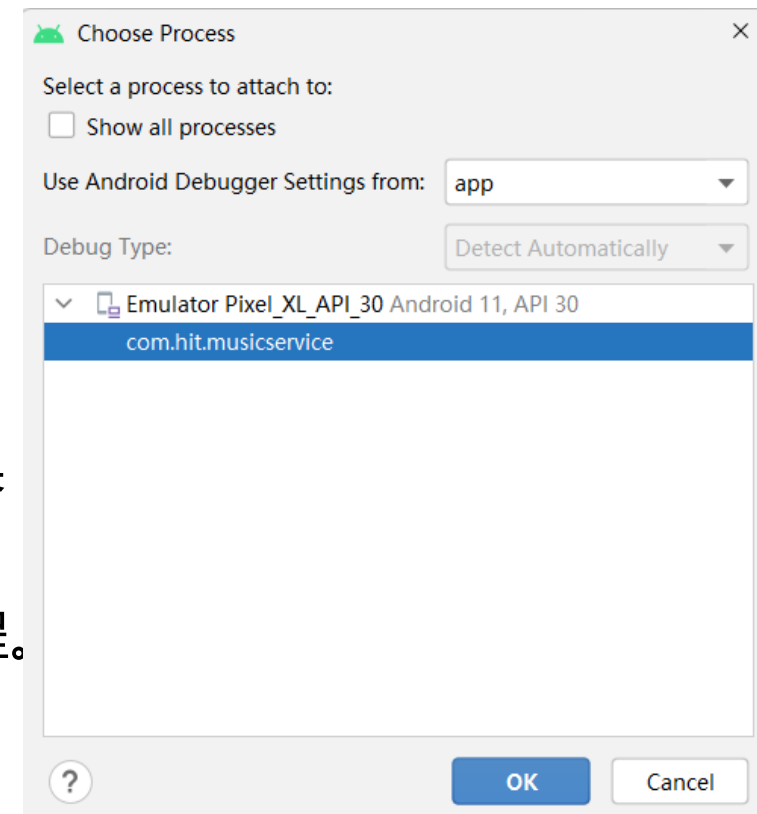
```
1  package com.example.HelloAndroid;    ← 包声明语句

2  import androidx.appcompat.AppCompatActivity;  ] ← 导入包
3  import android.os.Bundle;

4  public class MainActivity extends AppCompatActivity  ← 类声明语句
5  {                                                    类名
6      public void onCreate(Bundle savedInstanceState) ← 重写 onCreate() 方法
7      {
8          super.onCreate(savedInstanceState); ← 调用父类 Activity 的 onCreate() 方法
9          setContentView(R.layout.activity_main);
10     }                                                在屏幕上显示内容的方法
11 }
```

### Android Studio Debugger

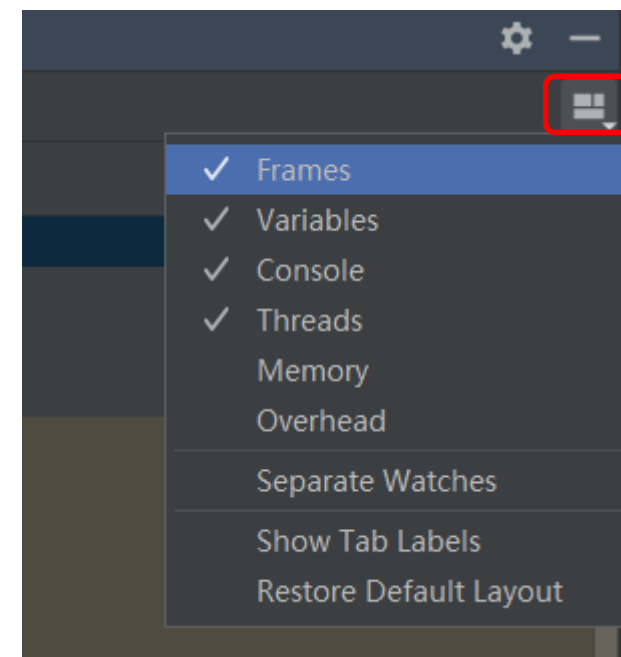
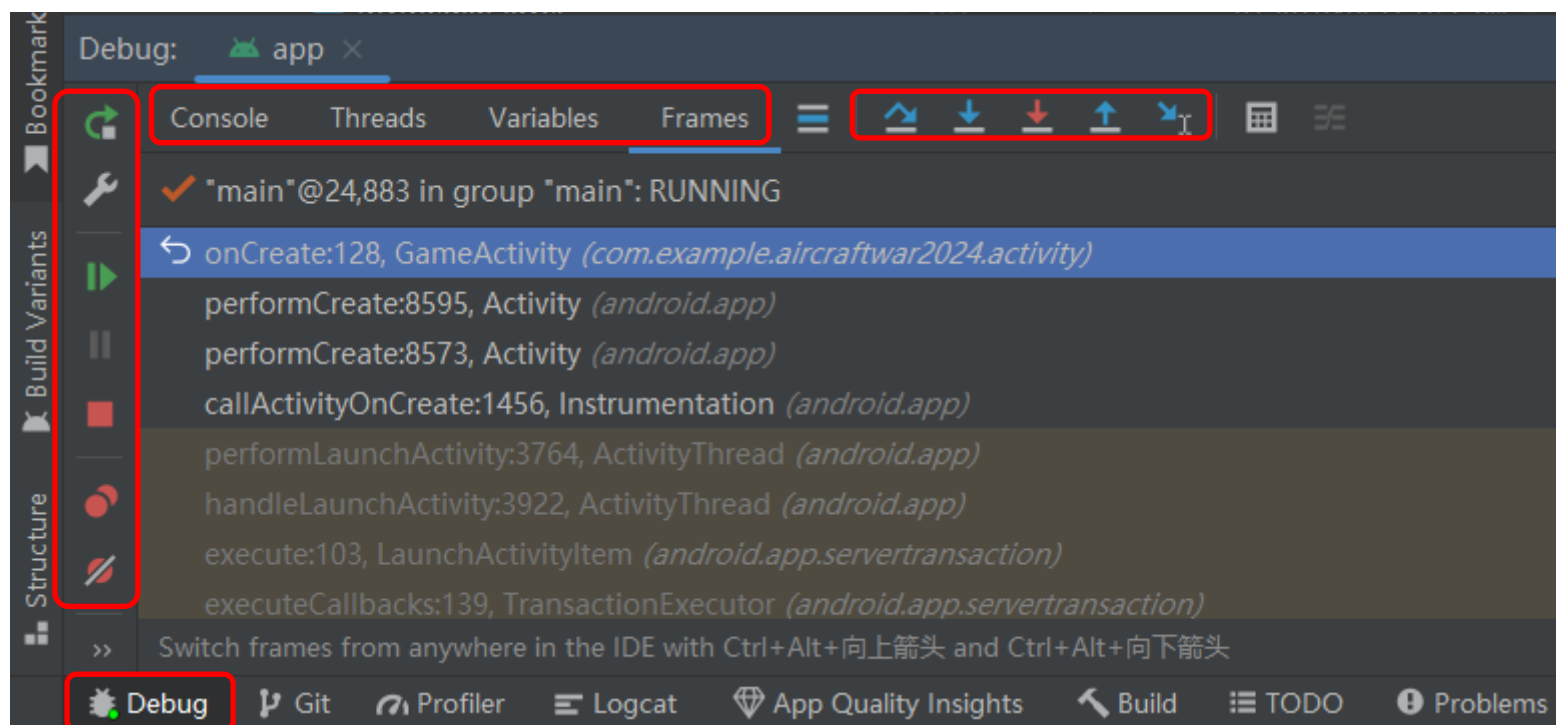
- 在有可能出问题的代码处设置断点。
- 单击工具 “Debug app” 按钮  开始调试
- “Debug app” 按钮会重启应用程序，需要等待较长时间。如果程序已经运行，可以点击 “Attach Debugger To Android Process” 按钮  会弹出一个选择进程的对话框，选择项目进程。



## 1.5 Android调试工具

### Android Studio Debugger

- 使用调试工具，监视代码执行过程



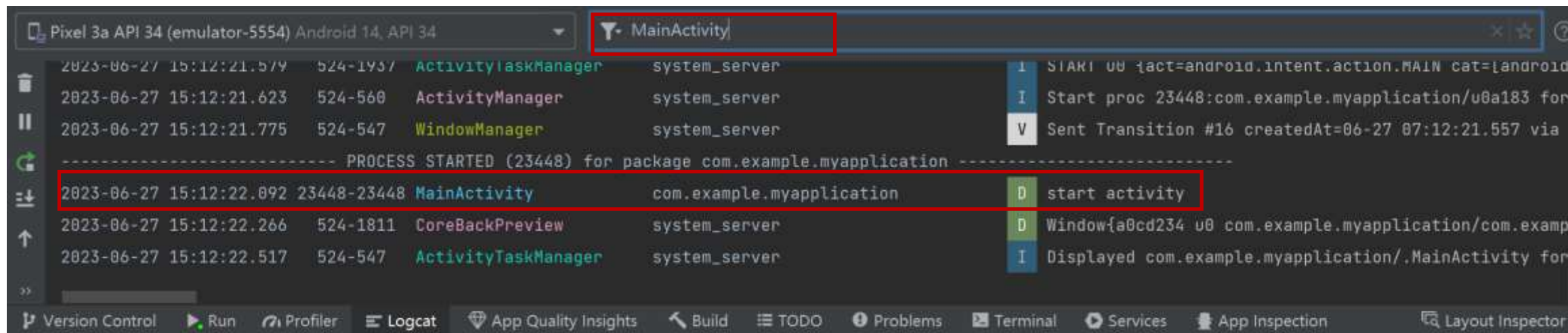
## 1.5 Android调试工具

**Log 是 Android Studio 中的日志工具类，有 6 个级别**

级别	作用	Java方法
Verbose	显示所有日志消息（默认）	Log.v()
Debug	显示仅在开发期间有用的调试日志消息，以及此列表中较低的消息级别。	Log.d()
Info	显示常规使用情况的预期日志消息，以及此列表中较低的消息级别。	Log.i()
Warn	显示尚不是错误的潜在问题，以及此列表中较低的消息级别。	Log.w()
Error	显示已经引发错误的问题，以及此列表中较低的消息级别。	Log.e()
Assert	显示严重问题，可能导致程序立即终止。	Log.wtf()

### Logcat

- Android应用程序运行时会在Logcat窗口打印日志
- 用颜色区分不同级别的日志
- 使用搜索框过滤日志



## Android Studio Flamingo

属性		值	设置方式
Android SDK		Android 14 API 34	Tools → SDK Manager → SDK Platforms
Gradle	Gradle JDK	JetBrains Runtime Version 17.0.6	File → Settings → Gradle → Gradle JDK
	Gradle Plugin Version	8.0.2	File → Project Structure → Project
	Gradle Version	8.0	
Emulator		Pixel 3a API 34	创建模拟器时选择相应API级别和型号

安装和配置参考实验指导书

- 安装Android Studio Flamingo，按上页表格配置开发环境；
- 在Android Studio中创建一个名为“AircarftWar2024”的项目，注意项目路径**不要有中文**；
- 项目语言为Java；
- 支持的最小API级别为30；
- 将资源目录res/layout/activity\_main.xml中TextView的内容修改为自己的姓名和学号；
- 使用Pixel 3a API 34的模拟器运行此应用程序；
- 熟悉Android Studio开发环境和应用程序的目录结构；
- 完成任务书。



群名称:2024-软件构造实践-2群  
群 号:878198558



## 计算计7班:

上课教室 (T2604)

11周: 周二 5-6节 (实验1)

12周: 周三 1-2节 (实验2)

13周: 周二 1-2节 (实验3)

14周: 周二 5-6节

15周: 周四 5-6节 (中期)

16周: 周二 1-2节

周五 1-2节

周五 7-8节 (实验4)

周四 5-6节 (实验5)

周六1-4节 (实验6)

周五 7-8节 (结题)

## 计算机8班:

上课教室 (T2604)

11周: 周三 1-2节 (实验1)

12周: 周四 1-4节 (实验2)

13周: 周一 5-6节 (实验3)

14周: 周一 1-2节 (实验5)

15周: 周四 1-4节 (实验6)

16周: 周三 1-2节

周四 1-4节 (实验4)

周五 7-8节 (中期)

周五 5-6节 (结题)

## 计算机9班:

上课教室 (T2604)

11周: 周三 11-12节 (实验1)

12周: 周一 1-4节 (实验2)

13周: 周一 1-2节 (实验3)      周三 1-2节    周五 5-6节 (实验4)

14周: 周一 5-6节 (实验5)      周五 1-2节 (中期)

15周: 周五 5-8节 (实验6)

16周: 周一 5-6节      周五 1-2节 (结题)

## 计算机10班:

上课教室 (T2507)

12周: 周二 7-8节 (实验1)

12周: 周四 5-8节 (实验2)

13周: 周二 7-8节 (实验3)      周四5-8节 (实验4)

14周: 周二 7-8节 (实验5)      周五11-12节 (中期)

15周: 周二 5-8节 (实验6)

16周: 周二 7-8节      周四7-8节 (结题)