



信息工程系

教

案

课程名称： 移动开发基础

教 师： 黄梅佳

总学时： 72

理论学时： 36

实训学时： 36

上课班级： 移动应用开发 241

授课学期： 2025-2026 第一学期

课题名称	第 1 章 Android 基础入门	计划课时	4 课时
教学引入	<p>Android 是 Google 公司基于 Linux 平台开发的手机及平板电脑的操作系统，它自问世以来，受到了前所未有的关注，并迅速成为移动平台最受欢迎的操作系统之一。Android 手机随处可见，如果能加入 Android 开发者行列，编写自己的应用程序供他人使用，想必是件诱人的事情。从今天开始，我们将开启 Android 开发之旅，逐渐成为一名出色的 Android 开发者。</p>		
教学目标	<ul style="list-style-type: none"> ● 使学生了解 1G~5G 的通信技术，能够阐述 1G~5G 技术的发展内容 ● 使学生掌握 Android Studio 开发环境的搭建步骤，能够独立搭建 Android Studio 开发环境 ● 使学生掌握编写简单 Android 程序的步骤，能够编写一个 Hello World 程序 ● 使学生掌握资源的管理与使用方式，能够灵活使用程序中的资源 ● 使学生掌握单元测试与 Logcat 的使用方式，能够对 Android 程序进行调试 		
教学重点	<ul style="list-style-type: none"> ● Android Studio 安装 ● 模拟器创建 ● Android 程序结构 ● 图片资源 ● 主题和样式资源 ● 布局资源 ● 字符串资源 ● 颜色资源 ● 尺寸资源 		
教学难点	无		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	<p>在讲解 Android 系统发展及通信技术演进（1G-5G）时，结合我国在 5G 领域的领先地位及 Android 生态中中国开发者的贡献，引导学生认识到科技创新对国家发展的重要性，激发学生投身科技事业、为国家科技进步贡献力量的责任感。</p> <p>在介绍 Android Studio 开发环境搭建、程序编写与调试等实操内容时，强调代码编写的规范性、调试过程的严谨性，培养学生精益求精、耐心细致的工匠精神，让学生明白只有脚踏实地、注重细节，才能开发出高质量的应用程序。</p>		

<p>教 学 过 程</p>	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">(通信技术、Android 发展历史、Android 体系结构、Dalvik 虚拟机)</p> <p>一、创设情景，导入新课</p> <p>在学习 Android 之前，我们需要了解什么是 Android，我们日常生活中是否用到过 Android 系统。教师结合实际生活中的例子针对 Android 系统进行讲解。例如，大家都在使用智能手机，通常都是苹果、小米、vivo 等。其中苹果使用的是 iOS 系统，其他手机使用的都是 Android 系统。Android 是 Google 公司基于 Linux 平台开发的手机及平板电脑的操作系统，了解 Android 系统之后，接着引出本节课要讲的通信技术、Android 发展历史、Android 体系结构、Dalvik 虚拟机。</p> <p>二、新课讲解</p> <p>知识点 1-通信技术</p> <p>教师通过 PPT 讲解通信技术。</p> <p>(1) 介绍通信技术的发展情况。</p> <p>(2) 介绍 1G、2G、3G、4G、5G 通信技术的具体区别。</p> <p>知识点 2-Android 发展历史</p> <p>教师通过 PPT 讲解 Android 发展历史。</p> <p>(1) 介绍 Android 的诞生。</p> <p>(2) 介绍 Android 各版本发布时间及代号。</p> <p>知识点 3-Android 体系结构</p> <p>教师通过 PPT 讲解 Android 体系结构。</p> <p>(1) Android 系统采用分层架构。</p> <ul style="list-style-type: none"> ● 应用程序层 (Applications)。 ● 应用程序框架层 (Application Framework)。 ● 核心类库 (Libraries)。 ● Linux 内核 (Linux Kernel)。 <p>(2) 展示 Android 体系结构图。</p> <p>(3) 根据 Android 体系结构图对 Android 体系结构进行讲解。</p> <p>知识点 4-Dalvik 虚拟机</p> <p>教师通过 PPT 讲解 Dalvik 虚拟机。</p> <p>(1) Dalvik 虚拟机的作用。</p> <p>(2) Dalvik 虚拟机编译文件的过程。</p> <p>三、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p style="text-align: center;">第二课时</p> <p style="text-align: center;">(Android Studio 安装、模拟器创建、在 Android Studio 中下载 SDK、开发第一个 Android 程序、Android 程序结构)</p> <p>一、复习巩固</p> <p>知识点回顾。</p> <p>二、通过需求引入的方式导入新课</p> <p>如果想要开发 Android 应用程序，那么我们首先需要安装 Android Studio 开发工具并创建模拟器、下载 SDK，然后通过 Android Studio 工具开发 Android 应用程序，将开发完成的程序运行到模拟器上，来展示 Android 程序的效果。</p>
----------------------------	---

通过这些内容引出本节课要讲的 Android Studio 安装、模拟器创建、在 Android Studio 中下载 SDK、开发第一个 Android 程序和 Android 程序结构。

三、新课讲解

知识点 1-Android Studio 安装

教师通过 PPT 结合实际操作的方式讲解 Android Studio 安装。

- (1) Android Studio 的下载。
- (2) Android Studio 的安装过程。
- (3) Android Studio 的配置。

知识点 2-模拟器创建

教师通过 PPT 结合实际操作的方式讲解模拟器创建。

- (1) 弹出 Device Manager 对话框。
- (2) 选择模拟设备。
- (3) 下载 SDK System Image。
- (4) 创建模拟设备。
- (5) 打开模拟设备。

知识点 3-在 Android Studio 中下载 SDK

教师通过 PPT 结合实际操作的方式讲解在 Android Studio 中下载 SDK。

- (1) 什么情况下需要下载 SDK。
- (2) 下载 SDK 版本。
- (3) 下载 SDK Build-Tools 工具。

知识点 4-开发第一个 Android 程序

教师通过 PPT 结合实际操作的方式讲解开发第一个 Android 程序。

- (1) 创建 Hello World 程序。
- (2) 运行程序。

知识点 5-Android 程序结构

教师通过 PPT 讲解 Android 程序结构。

- (1) app。
 - libs。
 - src\androidTest。
 - src\main\java。
 - src\main\res。
 - src\main\AndroidManifest.xml。
 - app\build.gradle。
- (2) build.gradle。
- (3) local.properties。
- (4) settings.gradle。

存放程序的代码和资源等内容

程序的gradle构建脚本

指定项目中所使用的SDK路径

配置在Android中使用的子项目(Moudle)

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业。

第三课时

(图片资源、主题和样式资源、布局资源、字符串资源、颜色资源、尺寸资源)

一、复习巩固

复习上节课程内容要点。

二、通过直接引入的方式导入新课

Android 程序中，资源指的是可以在代码中使用的外部文件，这些文件作为应用程序的一部分，被编译到 App 中。在 Android 程序中，资源文件都保存在 res 目录下。接下来，本节课针对 res 目录下的资源进行详细介绍。

三、新课讲解

知识点 1-图片资源

教师通过 PPT 结合实际操作的方式讲解图片资源。

(1) Android 中的图片资源的扩展名和存放位置。

(2) res 目录中的 mipmap 文件夹和 drawable 文件夹的匹配规则。

(3) 资源文件的调用方式。

① 通过 Java 代码调用图片资源。

```
//调用 mipmap 文件夹中资源文件
getResources().getDrawable(R.mipmap.ic_launcher);
//调用以 drawable 开头的文件夹中的资源文件
getResources().getDrawable(R.drawable.icon);
```

② 在 XML 布局文件中调用图片资源。

@mipmap/ic_launcher //调用 mipmap 文件夹中的资源文件

@drawable/icon //调用以 drawable 开头的文件夹中的资源文件

知识点 2-主题和样式资源

教师通过 PPT 结合实际操作的方式讲解主题和样式资源。

(1) 主题。

① 什么是主题。

主题是包含一种或多种格式化属性的集合，在程序中调用主题资源可改变窗体的样式。

主题资源定义位置：在 res\values 目录下的 themes.xml 文件中定义主题资源的标签：

② 主题的存放位置。

主题资源定义在 res\values\themes.xml 文件中，示例代码如下：

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.HelloWorld" parent="Theme.MaterialComponents.
DayNight.DarkActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    .....
  </style>
</resources>
```

③ 在 AndroidManifest.xml 文件中设置主题。

④ 在 Java 代码中设置主题

(2) 样式。

① 什么情况下需要创建样式。

② 样式的存放位置。

样式资源定义位置：res\values 目录下的 styles.xml 文件中

③ 在创建好的 styles.xml 文件中添加一个名为 textViewStyle 的样式。

④ 在布局文件的 View 控件中通过 style 属性引用 textViewStyle 样式。

知识点 3-布局资源

教师通过 PPT 结合实际操作的方式讲解布局资源。

(1) 通过 Java 代码调用布局资源文件。

(2) 在 XML 布局文件中调用布局资源文件。

- 教师通过 PPT 结合实际操作的方式讲解。布局资源用于定义应用程序界面中控件的排列方式和位置。
- 通过 Java 代码调用布局资源文件：在 Activity 的 `onCreate()` 方法中，使用 `setContentView(R.layout.activity_main)` 方法加载布局文件，其中 `activity_main` 是布局文件的名称。
- 在 XML 布局文件中调用布局资源文件：可以通过 `<include>` 标签复用其他布局文件，如 `<include layout="@layout/header"/>`，将 `header.xml` 布局文件包含到当前布局中。
- PPT 中会展示不同布局类型（如 `LinearLayout`、`RelativeLayout`、`ConstraintLayout`）的示例和效果。

知识点 4-字符串资源

教师通过 PPT 结合实际操作的方式讲解字符串资源。

- （1）通过 Java 代码调用字符串资源。
- （2）在 XML 布局文件中调用字符串资源。

字符串资源：用于显示界面上的文本信息。

字符串资源定义位置：res/values 目录下的 strings.xml 文件中

定义字符串的标签：`<string></string>`，通过 `name` 属性指定字符串的名称。

在 res/values/strings.xml 文件中定义字符串，示例代码如下：

xml

```
<resources>
    <string name="app_name">我的应用</string>
    <string name="welcome">欢迎使用</string>
</resources>
```

通过 Java 代码调用字符串资源：`String welcome = getResources().getString(R.string.welcome);`

在 XML 布局文件中调用字符串资源：`android:text="@string/welcome"`

知识点 5-颜色资源

教师通过 PPT 结合实际操作的方式讲解颜色资源。

- （1）通过 Java 代码调用颜色资源文件。
- （2）在 XML 布局文件中调用颜色资源文件。

颜色资源定义位置：通常在 res/values 目录下的 colors.xml 文件中，使用 `<color>` 标签定义，如：

xml

```
<resources>
    <color name="purple_500">#FF6200EE</color>
```

```
<color name="black">#FF000000</color>
</resources>
```

通过 Java 代码调用颜色资源文件：`int color = getResources().getColor(R.color.purple_500);`
在 XML 布局文件中调用颜色资源文件：
`android:background="@color/purple_500"` 或
`android:textColor="@color/black"`

知识点 6-尺寸资源

教师通过 PPT 讲解尺寸资源。

- (1) 通过 Java 代码调用尺寸资源。
- (2) 在 XML 布局文件中调用尺寸资源。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

尺寸资源用于定义应用程序中各种元素的大小，如控件的宽度、高度、字体大小等，使用统一的尺寸资源便于适配不同屏幕。

尺寸资源定义位置：通常在 `res\values` 目录下的 `dimens.xml` 文件中，使用 `<dimen>` 标签定义，如：

```
xml

<resources>
    <dimen name="text_size">16sp</dimen>
    <dimen name="margin">16dp</dimen>
</resources>
```

通过 Java 代码调用尺寸资源：`float textSize = getResources().getDimension(R.dimen.text_size);`

在 XML 布局文件中调用尺寸资源：
`android:textSize="@dimen/text_size"` 或
`android:layout_margin="@dimen/margin"`

PPT 中会说明不同尺寸单位（如 `dp`、`sp`、`px`）的区别和适用场景，`dp` 用于布局尺寸，`sp` 用于字体大小，`px` 不推荐使用。

第四课时

（单元测试、Logcat 的使用）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

在实际开发中，每个 Android 程序都需要进行一系列的测试工作，确保程序能够正常运行。调试 Android 程序有多种方式，例如单元测试和 Logcat（日

志控制台)等,本节课将针对单元测试和 Logcat 的使用进行详细讲解。

三、新课讲解

知识点 1-单元测试

教师通过 PPT 结合实际操作的方式讲解单元测试。

- (1) 什么是单元测试。
- (2) 分别对 Android Studio 单元测试类 ExampleInstrumentedTest 和 Junit 单元测试类 ExampleUnitTest 的用法进行讲解。
 - Android 单元测试类。
 - Junit 单元测试类。

知识点 2-Logcat 的使用

教师通过 PPT 结合实际操作的方式讲解 Logcat 的使用。

- (1) 什么是 Logcat。
- (2) Log 类所输出的日志内容分为六个级别。
- (3) 在 Android Studio 的 Logcat 中会输出不同级别的日志信息,并且这些日志信息显示的颜色也是不同的。
 - verbose(V)。
 - debug(D)。
 - info(I)。
 - warning(W)。
 - error(E)。
 - assert。
- (4) 在 HelloWorld 程序中编译 MainActivity 代码输出 Log 信息。

什么是 Logcat: Logcat 是 Android 系统提供的日志工具,用于收集和查看系统及应用程序运行时的日志信息,帮助开发者调试程序。

Log 类所输出的日志内容分为六个级别,从低到高依次为:

verbose (V): 最详细的日志信息,通常用于调试过程中的详细跟踪。

debug (D): 调试信息,用于记录程序运行过程中的关键步骤和变量值。

info (I): 一般信息,用于记录程序的正常运行状态。

warning (W): 警告信息,表示程序可能存在潜在问题,但不会导致程序崩溃。

error (E): 错误信息,表示程序运行过程中出现了错误,可能导致功能异常。

assert: 断言失败信息,用于验证程序中的假设条件是否成立。

在 Android Studio 的 Logcat 中,不同级别的日志信息显示的颜色不同,例如 verbose 和 debug 为黑色,info 为蓝色,warning 为黄色,error 和 assert 为红色,便于区分和筛选。

在 HelloWorld 程序中编译 MainActivity 代码输出 Log 信息,示例代码:

```
java  
运行
```

```
public class MainActivity extends AppCompatActivity {
```

	<pre> private static final String TAG = "MainActivity"; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); Log.v(TAG, "verbose 日志信息"); Log.d(TAG, "debug 日志信息"); Log.i(TAG, "info 日志信息"); Log.w(TAG, "warning 日志信息"); Log.e(TAG, "error 日志信息"); } } </pre> <p>运行程序后，在 Android Studio 的 Logcat 窗口中可以查看输出的日志信息，通过设置过滤器（如按标签 TAG 筛选）可以快速找到需要的日志。</p> <p>四、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p> <p>教师通过高校教辅平台（http://tch.ityxb.com）布置本节课作业以及下节课的预习作业。</p> <p>上机：（考察知识点为 Android Studio 安装、模拟器创建、开发第一个 Android 程序）</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照以下要求完成操作，要求如下：</p> <ol style="list-style-type: none"> （1）下载并安装 Android Studio 开发工具。 （2）创建 Android 模拟器。 （3）开发第一个 Android 程序 HelloWorld。
教学后记	<p>对于 Android Studio 安装、模拟器创建、Hello World 程序开发等实操内容，多数学生能够按照指导步骤完成操作，但部分学生在环境配置和程序运行环节遇到问题，经过个别指导后得以解决，整体掌握情况基本达到教学目标。</p> <p>资源管理与使用部分，由于涉及多种资源类型及调用方式，部分学生对不同资源在 Java 代码和 XML 布局文件中的调用区别理解不够透彻，需要在后续练习中进一步巩固。</p>

课题名称	第 2 章 Android 常见界面布局	计划课时	4 课时
教学引入	在 Android 应用中，界面由布局和控制件组成。布局好比是建筑里的框架，控件相当于建筑里的砖瓦。针对界面中控件不同的排列位置，Android 定义了相应的布局进行管理。本章将针对 Android 的常见界面布局进行详细讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生了解 View 与 ViewGroup 的简介，能够说出 View 与 ViewGroup 的作用和关系 ● 使学生掌握界面布局的编写方式，能够独立在 XML 文件中与 Java 代码中编写布局 ● 使学生掌握常见界面布局的特点及使用，能够搭建简单的界面布局 		
教学重点	<ul style="list-style-type: none"> ● 在 XML 文件中编写界面布局 ● 在 Java 代码中编写界面布局 ● 布局的通用属性 ● RelativeLayout ● LinearLayout ● TableLayout ● FrameLayout ● ConstraintLayout 		
教学难点	无		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	<ol style="list-style-type: none"> 1. 在讲解界面布局的美观性和合理性时，引导学生认识到良好的用户体验不仅需要技术支撑，还需要具备审美素养和以人为本的设计理念，培养学生注重细节、追求卓越的职业素养。 2. 在介绍各种布局的适用场景时，强调根据实际需求选择合适技术的重要性，培养学生实事求是的思维方式和解决问题的能力，让学生明白技术的价值在于应用于实际并创造价值。 		

第一课时

(View、在 XML 文件中编写界面布局、在 Java 代码中编写界面布局)

一、复习巩固

教师通过检查上节课作业的完成情况，了解学生对 Android 开发环境搭建、资源管理等知识点的掌握程度。对于学生在作业中暴露出的问题，如资源调用错误、模拟器启动失败等，进行集中讲解和再次巩固，确保学生为本次课的学习做好铺垫。

二、通过需求引入的方式导入新课

通常情况下，Android 程序的界面非常美观，而且特别吸引人，让大家对其很有兴趣，例如 QQ 界面、微信界面、网易新闻等，这些界面上控件的摆放都是通过布局来实现的。Android 程序开发最重要的一个环节就是界面处理，界面的美观度直接影响用户的第一印象，因此，开发一个整齐、美观的界面是至关重要的，本节课将针对 View、在 XML 文件中编写界面布局和在 Java 代码中编写界面布局进行详细地讲解。

三、新课讲解

知识点 1-View

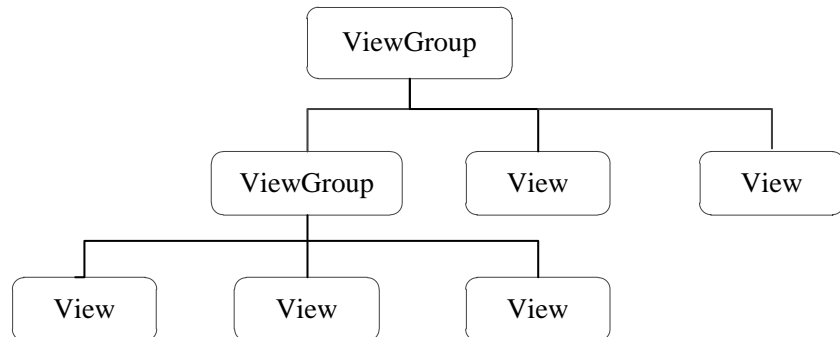
教师通过 PPT 讲解 View。

(1) 什么是 View。

View 是 Android 程序中可视化的界面组件的实体，它主要提供了组件绘制和事件处理的方法。

(2) View 的容器类型和实体类型。

(3) Android 程序中 ViewGroup 容器和实体类型的 View 的包含关系。



知识点 2-在 XML 文件中编写界面布局

教师通过 PPT 结合实际操作的方式讲解在 XML 文件中编写界面布局。

(1) 使用 XML 文件编写界面布局的优势。

在 Android 程序中可以使用 XML 文件控制界面布局，从而有效地将界面中布局的代码和 Java 代码隔离，使程序的结构更加清晰。因此多数 Android 程序采用这种方式编写布局。

前面讲解过布局文件通常放在 res\layout 文件夹中，我们可以在该文件夹中的 XML 文件中编写布局。

(2) 演示如何在 XML 文件中编写界面布局。

相对布局继承自ViewGroup

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="使用XML布局文件控制UI界面"
        android:textColor="#ff0000"
        android:textSize="18sp"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

TextView控件继承自View

设置文字的样式

知识点 3-在 Java 代码中编写界面布局

教师通过 PPT 结合实际操作的方式讲解在 Java 代码中编写界面布局。

- (1) 在 Java 代码中编写界面布局的方法。
- (2) 演示如何在 Java 文件中编写界面布局。

```
RelativeLayout relativeLayout = new RelativeLayout(this);
RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(
    RelativeLayout.LayoutParams.WRAP_CONTENT,
    RelativeLayout.LayoutParams.WRAP_CONTENT);
params.addRule(RelativeLayout.CENTER_IN_PARENT); //设置布局中的控件居中显示
TextView textView = new TextView(this); //创建TextView控件
textView.setText("Java 代码实现界面布局"); //设置TextView的文字内容
textView.setTextColor(Color.RED); //设置TextView的文字颜色
textView.setTextSize(18); //设置TextView的文字大小
relativeLayout.addView(textView, params); //添加TextView对象和TextView的布局属性
setContentView(relativeLayout); //设置在Activity中显示RelativeLayout
```

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第二课时

(布局的通用属性、RelativeLayout、LinearLayout)

一、复习巩固

对上一节知识点进行再次巩固讲解。

三、通过需求引入的方式导入新课

为了适应不同的界面风格，Android 系统为每个布局都提供了属性，但是有一些属性是所有布局通用的，这些属性就被称为布局的通用属性，在学习布局之前我们首先需要了解一下这些通用属性，便于后续更好地设置布局。接下来将针对布局中的通用属性、RelativeLayout 和 LinearLayout 进行详细讲解。

四、新课讲解

知识点 1-布局的通用属性

教师通过 PPT 讲解布局的通用属性。

- (1) android:id。
- (2) android:layout_width。

- (3) android:layout_height。
- (4) android:background。
- (5) android:layout_margin。
- (6) android:padding。

属性名称	功能描述
android:id	设置布局的标识
android:layout_width	设置布局的宽度
android:layout_height	设置布局的高度
android:background	设置布局的背景
android:layout_margin	设置当前布局与屏幕边界或与周围控件的距离
android:padding	设置当前布局与该布局中控件的距离

知识点 2-RelativeLayout

教师通过 PPT 结合实际操作的方式讲解 RelativeLayout。

- (1) 什么是相对布局。
- (2) 相对布局的语法格式。

RelativeLayout 通过相对定位的方式指定子控件的位置。在 XML 布局文件中定义相对布局时使用<RelativeLayout>标签，基本语法格式如下所示。

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    属性 = "属性值"
    .....>
</RelativeLayout>
```

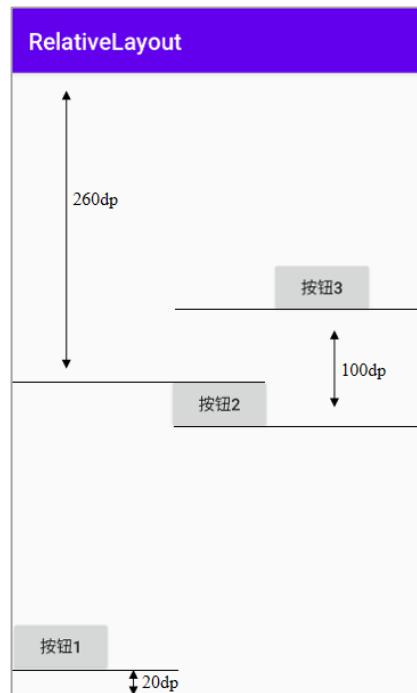
- (3) RelativeLayout 中子控件的属性。

属性名称	功能描述
android:layout_centerInParent	设置当前控件位于父布局的中间位置
android:layout_centerVertical	设置当前控件位于父布局的垂直居中位置
android:layout_centerHorizontal	设置当前控件位于父控件的水平居中位置
android:layout_above	设置当前控件位于某控件上方
android:layout_below	设置当前控件位于某控件下方
android:layout_toLeftOf	设置当前控件位于某控件左侧
android:layout_toRightOf	设置当前控件位于某控件右侧
android:layout_alignParentTop	设置当前控件是否与父控件顶端对齐
android:layout_alignParentLeft	设置当前控件是否与父控件左边对齐
android:layout_alignParentRight	设置当前控件是否与父控件右边对齐
android:layout_alignParentBottom	设置当前控件是否与父控件底部对齐
android:layout_alignTop	设置当前控件的上边界与某控件的上边界对齐
android:layout_alignBottom	设置当前控件的下边界与某控件的下边界对齐
android:layout_alignLeft	设置当前控件的左边界与某控件的左边界对齐
android:layout_alignRight	设置当前控件的右边界与某控件的右边界对齐

(4) 通过一个案例来演示在相对布局中指定三个按钮的位置。

- 创建程序。
- 修改程序主题样式。
- 添加界面控件。

通过一个案例来演示如何在相对布局中指定 3 个按钮的位置。本案例中使用了相对布局 `RelativeLayout`，在相对布局中添加了 3 个按钮，这 3 个按钮以不同的位置进行显示，相对布局界面的效果如下图所示。



注意：在 `RelativeLayout` 布局中定义的控制默认与父布局左上角对齐。

为了让 Android 程序拥有更好的屏幕适配能力，在设置控件和布局的宽度与高度时最好使用“`match_parent`”或“`wrap_content`”，尽量避免将控件的宽与高设置为固定值。因为控件在很多情况下会相互挤压，从而导致控件变形。但特殊情况下需要使用指定的宽度与高度的值时，可以选择使用 `px`、`pt`、`dp`、`sp` 等四种类型的单位。例如：`android:layout_width="20dp"`，表示控件宽度为 20dp。

知识点 3-`LinearLayout`

教师通过 PPT 结合实际操作的方式讲解 `LinearLayout`。

(1) 什么是线性布局。

`LinearLayout`（线性布局）通常指定布局内的子控件水平或垂直排列。

在 XML 布局文件中定义线性布局的基本语法格式如下：

(2) 线性布局的语法格式。

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
        属性 = "属性值"  
        .....>  
</LinearLayout>
```

(3) 线性布局中的常用属性。

- android:orientation。
- android:layout_weight。

android:orientation 属性的可选值为 vertical 和 horizontal。

(1) vertical: 表示 LinearLayout 布局内控件依次从上到下垂直排列。

(2) horizontal: 表示 LinearLayout 布局内控件依次从左到右水平排列。

android:layout_weight 属性。

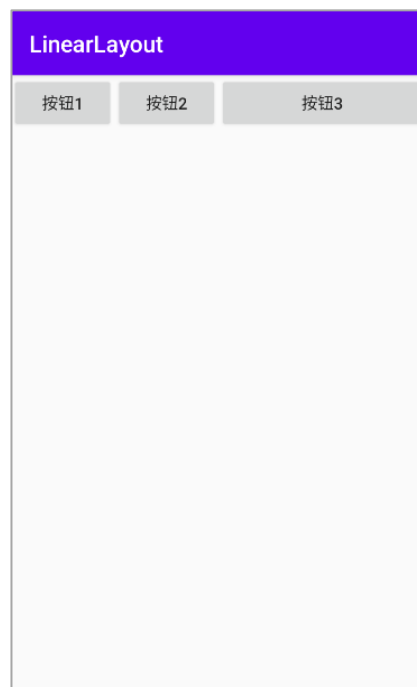
(1) 该属性被称为权重，通过设置该属性值，可使布局内的控件按照权重比显示大小。

(2) 在进行屏幕适配时起到关键作用。

(4) 演示如何使用 android:layout_weight 属性为 LinearLayout 中的控件分配权重。

- 创建程序。
- 添加界面控件。

通过一个案例来演示如何使用 android:layout_weight 属性为 LinearLayout 中的控件分配权重。本案例中使用了线性布局 LinearLayout，在线性布局中添加了三个按钮，这三个按钮的宽度在水平方向的比重是 1:1:2，线性布局界面的效果如下图所示。



LinearLayout 布局中的 android:layout_width 属性值不可设为 wrap_content。

这是因为 LinearLayout 的优先级比 Button 高，如果设置为 wrap_content，则 Button 控件的 android:layout_weight 属性会失去作用。

当设置了 Button 控件的 android:layout_weight 属性时，控件的 android:layout_width 属性值一般设置为 0dp 才会有权重占比的效果。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业。

第三课时

(TableLayout、FrameLayout、ConstraintLayout)

一、复习巩固

对知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

Android 中常见的界面布局除了前面讲的相对布局和线性布局外，还有表格布局、帧布局和约束布局，通过这些布局来显示不同的界面效果。本节课将针对 TableLayout、FrameLayout 和 ConstraintLayout 进行详细讲解。

三、新课讲解

知识点 1-TableLayout

教师通过 PPT 结合实际操作的方式讲解 TableLayout。

(1) 什么是表格布局。

TableLayout（表格布局）采用行、列的形式来管理控件，它不需要明确声明包含多少行、多少列，而是通过在 TableLayout 布局中添加 TableRow 布局或控件来控制表格的行数，在 TableRow 布局中添加控件来控制表格的列数。在 XML 布局文件中定义表格布局的基本语法格式如下：

(2) 表格布局的语法格式。

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    属性 = "属性值">
    <TableRow>
        UI 控件
    </TableRow>
    UI 控件
    .....
</TableLayout>
```

(3) 表格布局的常用属性。

- android:stretchColumns。
- android:shrinkColumns。
- android:collapseColumns。

(4) 表格布局中控件的常用属性。

- android:layout_column。
- android:layout_span。

表格布局属性

属性名称	功能描述
android:stretchColumns	设置可被拉伸的列
android:shrinkColumns	设置可被收缩的列
android:collapseColumns	设置可被隐藏的列

表格布局中控件的常用属性

属性名称	功能描述
android:layout_column	设置该控件显示的位置
android:layout_span	设置该控件占据几列，默认为1列

(5) 演示如何设置三行三列的表格。

- 创建程序。
- 添加界面控件。

知识点 2-FrameLayout

教师通过 PPT 结合实际操作的方式讲解 FrameLayout。

- (1) 什么是帧布局。
- (2) 帧布局的语法格式。

FrameLayout（帧布局）用于在屏幕上创建一块空白区域，添加到该区域中的每个子控件占一帧，这些帧会一个一个叠加在一起，后加入的控件会叠加在上一个控件上层。默认情况下，帧布局中的所有控件会与左上角对齐。在 XML 布局文件中定义 FrameLayout 的基本语法格式如下所示。

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    属性 ="属性值">
</FrameLayout>
```

(3) 除了布局的通用属性之外，帧布局还有两个特殊属性。

- android:foreground。
- android:foregroundGravity。

FrameLayout的特殊属性

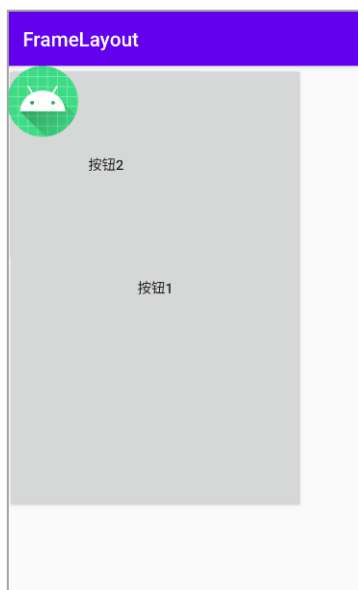
属性名称	功能描述
android:foreground	设置帧布局容器的前景图像（始终在所有子控件之上）
android:foregroundGravity	设置前景图像显示的位置

(4) 演示指定帧布局中的前景图像与前景图像的位置。

- 创建程序。
- 添加界面控件。

- 运行效果。

通过一个案例来讲解如何在帧布局中使用属性 `android:foreground` 和 `android:foregroundGravity` 指定帧布局中的前景图像与前景图像的位置。本案例中使用了帧布局 `FrameLayout`，在帧布局中添加两个按钮，分别是按钮 1 和按钮 2，按钮 2 在按钮 1 的上一层进行显示，帧布局界面的效果如下图所示。



知识点 3-ConstraintLayout

教师通过 PPT 结合实际操作的方式讲解 `ConstraintLayout`。

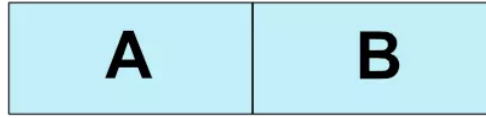
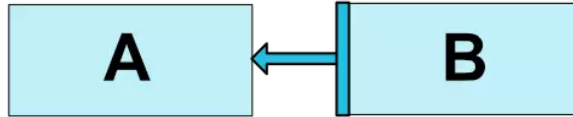
(1) 什么是 `ConstraintLayout`。

`ConstraintLayout`（约束布局）是 `Android Studio 2.2` 新添加的布局。与前面介绍的界面布局相比，`ConstraintLayout` 并不太适合使用 XML 代码的方式编写布局，但是它非常适合使用可视化的方式编写界面布局。当然，可视化操作的背后仍然是使用 XML 代码实现的，只不过这些代码是 `Android Studio` 根据我们的操作自动生成的。

(2) 相对于传统布局，`ConstraintLayout` 提供了一些新的特性。

- 相对定位。

相对定位是在 `ConstraintLayout` 中创建布局的基本构建方法之一。相对定位即一个控件相对于另一个控件进行定位，`ConstraintLayout` 布局中的控件可以在横向和纵向上以添加约束关系的方式进行相对定位，其中，横向边包括 `left`、`start`、`right`、`end`，纵向边包括 `top`、`bottom`、`baseline`（文本底部的基准线），控件中横向和纵向的边的分布如下图所示。



- 居中定位和倾向。

在 `ConstraintLayout` 布局中，两个控件之间通过添加约束的方式不仅可以确定控件的相对位置，而且还可以确定控件在父布局（`ConstraintLayout`）中的相对位置。在相同方向上（横向或纵向），控件的左右两边或上下两边同时在 `ConstraintLayout` 中添加约束，则控件在添加约束的方向上居中显示。例如，设置控件在父布局中与父布局左边对齐，与父布局右边也对齐，此时控件在父布局中横向居中显示，如下图所示。

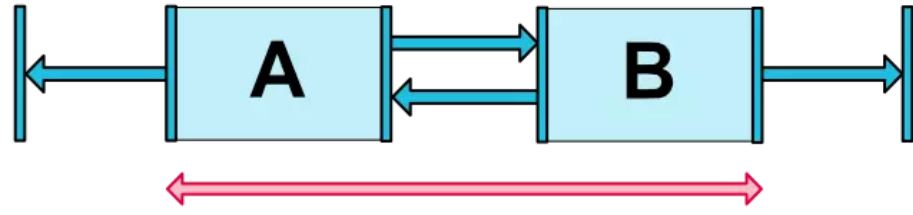


在横向或纵向方向上，添加的约束是相反的情况下，默认控件是居中的，当添加的两个约束的力大小不等时，就会产生倾向，设置倾向的属性如下表所示。

属性名称	功能描述
<code>layout_constraintHorizontal_bias</code>	横向的倾向
<code>layout_constraintVertical_bias</code>	纵向的倾向

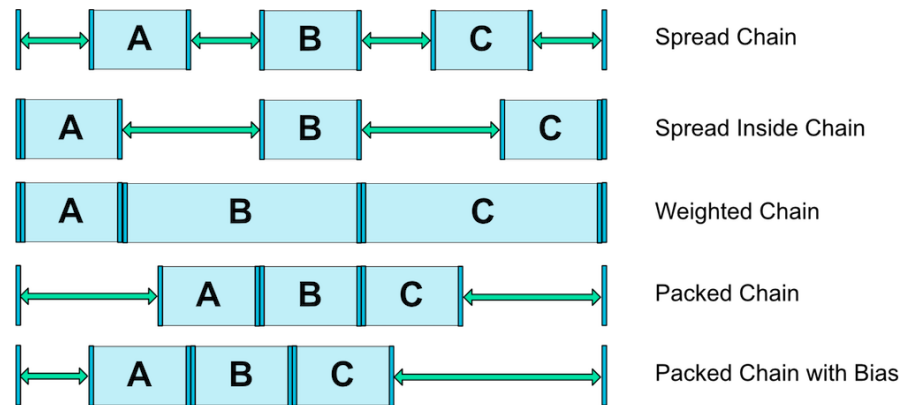
- Chain。

Chain（链）是一种特殊的约束，它使我们能够对一组水平或竖直方向互相关联的控件进行统一管理。一组控件通过一个双向的约束关系链接起来，就能形成一个 **Chain**。形成的 **Chain** 约束如下图所示。



Chain

上一页图的 Chain 中的第一个控件 A 称为头控件，Chain 的头控件可以通过 `layout_constraintHorizontal_chainStyle` 和 `layout_constraintVertical_chainStyle` 属性设置水平链条和垂直链条的样式。其属性值为 `spread`、`spread_inside` 和 `packed`。ConstraintLayout 布局中，当控件宽或者高的属性设置为 `0dp` 时，Chain 的三种样式可以搭配 `layout_constraintHorizontal_weight` 属性形成 Weighted Chain 的样式。`packed` 可以搭配 `layout_constraintHorizontal_bias` 属性控制 Chain 与父容器的间距，从而形成 Packed Chain With Bias 样式。Chain 的五种样式如下图所示。



四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第四课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：（考察知识点为 `RelativeLayout`、`LinearLayout`、`TableLayout`、`FrameLayout`）

形式：独立完成

题目：

通过不同的小案例演示相对布局、线性布局、表格布局和帧布局，要求如下：

(4) 在相对布局中指定三个按钮的位置。

(5) 在线性布局中添加三个按钮，这三个按钮的宽度在水平方向的比重是 1:1:2。

	<p>(6) 使用表格布局设置三行三列的表格。</p> <p>(7) 指定帧布局中的前景图像与前景图像的位置。</p>
教学后记	对于 LinearLayout 中权重的理解和使用，部分学生还不够熟练，在后续教学中应结合更多实例进行说明。

课题名称	第3章 Android 常见界面控件	计划课时	8 课时
教学引入	几乎每一个 Android 程序都是通过界面控件与用户交互的, Android 系统提供了非常丰富的界面控件, 借助这些控件, 我们可以很方便地进行用户界面开发。接下来, 本章将针对 Android 中常见的界面控件进行讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生掌握简单控件的使用, 能够搭建简单的界面 ● 使学生掌握 ListView 控件与 RecyclerView 控件的使用, 能独立搭建列表界面 ● 使学生掌握自定义 View 的定义方式, 能够自定义一个简单的控件 		
教学重点	<ul style="list-style-type: none"> ● TextView 控件 ● EditText 控件 ● Button 控件 ● ImageView 控件 ● RadioButton 控件 ● CheckBox 控件 ● Toast 类 ● ListView 控件的使用 ● RecyclerView 控件的使用 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——超市界面 ● 实战演练——相册界面 ● 自定义 View 		
教学方式	课堂教学以 PPT 讲授为主, 并结合多媒体进行教学		
思政元素	在讲解控件属性配置 (如 TextView 的 android:id 唯一标识、ListView 的性能优化) 时, 强调代码规范的重要性。例如, 在编写 ViewHolder 缓存控件时, 引导学生理解 “重复代码简化” “资源复用” 背后的工程思维 —— 如同工业生产中 “标准化流程”, 减少冗余、提升效率, 培养学生严谨细致的编码习惯, 树立 “工匠精神”, 让每一行代码都经得起调试与迭代。		
教学过程	<p style="text-align: center;">第一、二课时</p> <p style="text-align: center;">(TextView 控件、EditText 控件、Button 控件、ImageView 控件)</p> <p>一、复习巩固 教师通过上节课作业的完成情况, 对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过需求引入的方式导入新课 在 Android 程序的界面上, 我们通常会看到按钮、输入框、文字、图片、单选按钮、复选框等信息, 这些信息是通过哪些方式来实现的呢? 为了更方便去显示与操作界面上的这些信息, Android 系统提供了一些控件来显示这些信息, 每个控件都有对应的属性来设置不同的效果。这些控件就是我们本节课需要讲的 TextView 控件、EditText 控件、Button 控件和 ImageView 控件。</p> <p>三、新课讲解</p> <p>知识点 1-TextView 控件 教师通过 PPT 结合实际操作的方式讲解 TextView 控件。</p>		

(1) TextView 控件的作用。

(2) TextView 控件的常用属性。

TextView 控件用于显示文本信息，我们可以在 XML 布局文件中以添加属性的方式来控制 TextView 控件的样式，TextView 控件的常用属性如下表所示。

属性名称	功能描述
android:layout_width	设置控件的宽度
android:layout_height	设置控件的高度
android:id	设置控件的唯一标识
android:background	设置控件的背景
android:layout_margin	设置当前控件与屏幕边界或周围控件、布局的距离
android:padding	设置TextView控件与该控件中内容的距离
android:text	设置文本内容
android:textColor	设置文字显示的颜色
android:textSize	设置文字大小，推荐单位为sp

属性名称	功能描述
android:gravity	设置文本内容的位置
android:maxLength	设置文本最大长度，超出此长度的文本不显示
android:lines	设置文本的行数，超出此行数的文本不显示
android:maxLines	设置文本的最大行数，超出此行数的文本不显示
android:ellipsize	设置当文本超出TextView规定的范围的显示方式
android:drawableTop	在文本的顶部显示图像
android:lineSpacingExtra	设置文本的行间距
android:textStyle	设置文本样式，如bold (粗体)，italic (斜体)，normal (正常)

(3) 通过案例讲解如何将 TextView 控件中的文本信息居中，并且将文本的字体设置为斜体进行显示。

- 创建程序。
- 添加界面控件。

TextView

TextView控件显示的文本信息

知识点 2-EditText 控件

教师通过 PPT 结合实际操作的方式讲解 EditText 控件。

(1) EditText 控件的作用。

EditText 控件用于显示编辑框，它是 TextView 的子类，用户可在此控件中输入信息。除了支持 TextView 控件的属性外，EditText 控件还支持一些其他的常用属性，这些常用属性如下表所示。

(2) EditText 控件的常用属性。

属性名称	功能描述
android:hint	控件中内容为空时显示的提示文本信息
android:textColorHint	控件中内容为空时显示的提示文本信息的颜色
android:password	输入文本框中的内容显示为 "."
android:phoneNumber	设置输入文本框中的内容只能是数字
android:maxLines	设置文本的最大行数
android:scrollHorizontally	设置文本信息超出EditText的宽度情况下，是否出现横拉条
android:editable	设置是否可编辑

(3) 通过一个案例来讲解如何使用 EditText 控件编辑文本信息。

- 创建程序。
- 添加界面控件。



知识点 3-Button 控件

教师通过 PPT 结合实际操作的方式讲解 Button 控件。

(1) Button 控件的作用。

Button 控件表示按钮，它继承自 TextView 控件，既可以显示文本，又可以显示图片，同时也允许用户通过点击来执行操作，当 Button 控件被点击时，被按下与弹起的背景会有一个动态的切换效果，这个效果就是点击效果。

(2) Button 控件的点击效果。

(3) Button 控件常用的点击事件。

- 通过在布局文件中指定 onClick 属性的方式设置 Button 控件的点击事件。
- 通过使用匿名内部类的方式设置 Button 控件的点击事件。
- 通过将 Activity 实现 OnClickListener 接口的方式设置 Button 控件的点击事件。

通过在布局文件中指定 onClick 属性的方式设置 Button 控件的点击事件

<Button

.....

android:onClick="click" />

通过使用匿名内部类的方式设置 Button 控件的点击事件

```
btn.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View view) {
```

```
        //实现点击事件的代码
```

```
    }
```

```
});
```

通过将 Activity 实现 OnClickListener 接口的方式设置 Button 控件的点击事件

```
public class Activity extends AppCompatActivity implements
```

```
View.OnClickListener{
```

```
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    .....
    btn.setOnClickListener(this); // 设置
Button 控件的点击监听事件
}
@Override
public void onClick(View view) {
    // 实现点击事件的代码
}
}

```

(4) 通过一个案例来讲解如何以三种方式为按钮设置点击事件。

- 创建程序。
- 添加界面控件。
- 实现按钮的点击事件。
- 运行结果。



在 MainActivity 中分别采用三种方式实现点击事件

知识点 4-ImageView 控件

教师通过 PPT 结合实际操作的方式讲解 ImageView 控件。

- (1) ImageView 控件的作用。
- (2) ImageView 控件的常用属性。

ImageView控件表示图片，它继承自View，可以加载各种图片资源。ImageView控件的常用属性如下表所示。

属性名称	功能描述
android:layout_width	设置控件的宽度
android:layout_height	设置控件的高度
android:id	设置控件的唯一标识
android:background	设置控件的背景
android:layout_margin	设置当前控件与屏幕边界或周围控件的距离
android:src	设置控件需要显示的图片资源
android:scaleType	将图片资源缩放或移动，以适应ImageView控件的宽高
android:tint	将图片渲染成指定的颜色

(3) 通过一个案例来讲解如何使用 ImageView 控件显示图片。

- 创建程序。
- 导入图片资源。
- 添加界面控件。



四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第三课时

(RadioButton 控件、CheckBox 控件、Toast 类)

一、复习巩固

对知识点进行再次巩固讲解。

四、通过直接引入的方式导入新课

通常情况下，Android 程序的界面上除了显示图片和文本信息，还可以显示单选按钮和复选框，同时需要提示用户一些信息时，也可以在界面上弹出一个提示信息。这些是如何显示的呢？这些就是通过我们本节课要讲的 **RadioButton** 控件、**CheckBox** 控件和 **Toast** 类来实现的。

五、新课讲解

知识点 1-RadioButton 控件

教师通过 PPT 结合实际操作的方式讲解 **RadioButton** 控件。

(1) **RadioButton** 控件的作用。

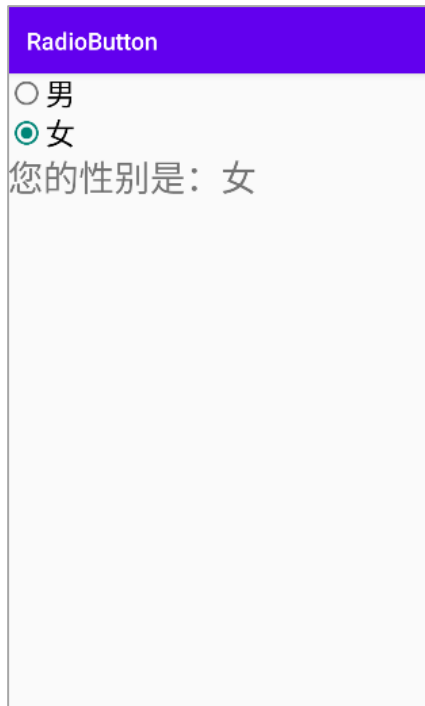
(2) 在 XML 布局文件中，**RadioGroup** 和 **RadioButton** 配合使用的语法格式。

RadioButton 表示单选按钮，它是 **Button** 的子类。每一个单选按钮都有“选中”和“未选中”两种状态，这两种状态是通过 **android:checked** 属性指定的。当可选值为 **true** 时，表示选中状态，否则，表示未选中状态。

```
<RadioGroup
    android:属性名称 ="属性值"
    .....>
    <RadioButton
        android:属性名称 ="属性值"
        ..... />
    .....
</RadioGroup/>
```

(3) 通过一个案例来讲解如何使用 **RadioGroup** 和 **RadioButton** 实现单选按钮的功能。

- 创建程序。
- 添加界面控件。
- 设置 **RadioGroup** 的监听事件。
- 运行结果。



知识点 2-CheckBox 控件

教师通过 PPT 结合实际操作的方式讲解 CheckBox 控件。

(1) CheckBox 控件的作用。

CheckBox 表示复选框，它是 Button 的子类，用于实现多选功能。每一个复选框都有“选中”和“未选中”两种状态，这两种状态是通过 android:checked 属性指定的，当该属性的值为 true 时，表示选中状态，否则，表示未选中状态。

(2) 通过一个案例讲解如何使用 CheckBox 控件统计用户的兴趣爱好。

- 添加界面控件。
- 实现 CheckBox 控件的点击事件。
- 运行结果。

知识点 3-Toast 类

教师通过 PPT 结合实际操作的方式讲解 Toast 类。

(1) Toast 类的作用。

(2) 使用 Toast 显示提示信息的示例代码。

(3) 演示使用 Toast 提示用户“WIFI 已断开”的信息。

Toast 是 Android 系统提供的轻量级信息提醒机制，用于向用户提示即时消息，它显示在应用程序界面的最上层，显示一段时间后自动消失不会打断当前操作，也不获得焦点。

使用 Toast 显示提示信息的示例代码如下：

```
Toast.makeText(Context,Text,Time).show();
```

关于 makeText()方法中参数的相关介绍具体如下：

Context：表示应用程序环境的信息，即当前组件的上下文环境。

Text：表示提示的字符串信息。

Time：表示显示信息的时长，其属性值包括 `Toast.LENGTH_SHORT` 和 `Toast.LENGTH_LONG`，分别表示显示较短时间和较长时间。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业以及下节课的预习作业。

第四、五、六课时

(ListView 控件的使用、常用数据适配器 (Adapter)、实战演练——超市界面)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过需求引入的方式导入新课

在日常生活中，大家经常会使用微信、京东、淘宝等应用程序，这些程序中通常会出现一些列表界面，如微信的通讯录界面、淘宝的待发货界面等，这些界面中展示了多个条目，并且每个条目的布局风格一致，这样的效果该如何实现呢？其实可以通过本节课要讲的 `ListView` 控件的使用和常用数据适配器 (`Adapter`) 来实现，同时本节课还通过一个超市界面的案例演示 `ListView` 控件的使用。

三、新课讲解

知识点 1-ListView 控件的使用

教师通过 PPT 结合实际操作的方式讲解 `ListView` 控件的使用。

(1) ListView 控件的作用。

在 `Android` 开发中，`ListView` 是一个比较常用的控件，它以列表的形式展示数据内容，并且能够根据列表的高度自适应屏幕显示。`ListView` 控件的样式是由属性决定的，它的常用属性如下表所示。

属性名称	功能描述
<code>android:listSelector</code>	当条目被点击后，改变条目的背景颜色
<code>android:divider</code>	设置分割线的颜色
<code>android:dividerHeight</code>	设置分割线的高度
<code>android:scrollbars</code>	是否显示滚动条
<code>android:fadingEdge</code>	去掉上边和下边的黑色阴影

(2) ListView 控件的常用属性。

- `android:listSelector`。
- `android:divider`。
- `android:dividerHeight`。
- `android:scrollbars`。

- android:fadingEdge。

(3) 在 XML 文件的 RelativeLayout 布局中添加 ListView 控件。

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    .....>
<ListView
    android:id="@+id/lv_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:listSelector="#fefeefe"
    android:scrollbars="none">
</ListView>
</RelativeLayout>
```

知识点 2-常用数据适配器 (Adapter)

教师通过 PPT 结合实际操作的方式讲解常用数据适配器 (Adapter)。

- (1) 数据适配器的作用。
- (2) 常用的数据适配器。

- BaseAdapter。

BaseAdapter，顾名思义，是基本的适配器。它实际上是一个抽象类，通常在自定义适配器时会继承 BaseAdapter，该类拥有四个抽象方法，根据这几个抽象方法对 ListView 控件进行数据适配。

方法名称	功能描述
getCount()	获取列表条目的总数
getItem(int position)	根据position (位置) 获取某个条目的对象
getItemId(int position)	根据position (位置) 获取某个条目的id
getView(int position, View convertView, ViewGroup parent)	获取相应position对应的条目视图, position是当前条目的位置, convertView用于复用旧视图, parent用于加载XML布局

- SimpleAdapter。

SimpleAdapter 继承自 BaseAdapter，实现了 BaseAdapter 的四个抽象方法并进行封装。

- ArrayAdapter。

ArrayAdapter 也是 BaseAdapter 的子类，用法与 SimpleAdapter 类似，开发者只需要在构造方法里面传入相应参数即可。ArrayAdapter 通常用于适配 TextView 控件，例如 Android 系统中的 Setting (设置菜单)。ArrayAdapter 有多个构造方法，构造方法的具体信息如下所示。

```
public ArrayAdapter(Context context,int resource);
```

```
public ArrayAdapter(Context context,int resource, int textViewResourceId);
public ArrayAdapter(Context context,int resource,T[] objects);
public ArrayAdapter(Context context,int resource,int textViewResourceId,T[]
objects);
public ArrayAdapter(Context context,int resource,List<T> objects);
public ArrayAdapter(Context context,int resource,int textViewResourceId,
List<T> objects)
```

在创建适配器后，可以通过 ListView 对象的 setAdapter()方法添加适配器，如将继承 BaseAdapter 的 MyBaseAdapter 实例添加到 ListView 中，示例代码如下：

```
ListView mListView = (ListView)findViewById(R.id.lv_list);    // 初始化
ListView 控件
MyBaseAdapter mAdapter = new MyBaseAdapter();                // 创建一个
Adapter 的实例
mListView.setAdapter(mAdapter);
```

知识点 3-实战演练——超市界面

教师通过 PPT 结合实际操作的方式讲解实战演练——超市界面。

- (1) 创建程序。
- (2) 导入界面图片。
- (3) 去掉默认标题栏。
- (4) 添加界面控件。
- (5) 创建超市列表条目界面。
- (6) 实现超市界面的显示效果。
- (7) 优化 ListView 控件加载数据的逻辑。
 - 当滑动屏幕时，不断地创建列表条目对象。
 - 不断执行 findViewById()方法初始化控件。

参考源码：

```
public class MainActivity extends Activity {
    private ListView mListView;
    //商品名称与价格数据集合
    private String[] titles = {"桌子", "苹果", "蛋糕", "线衣", "
猕猴桃", "围巾"};
    private String[] prices = {"1800 元", "10 元/kg", "300 元", "350
元", "10 元/kg",
        "280 元"};
    //图片数据集合
    private int[] icons = {R.drawable.table, R.drawable.apple,
```

```

R.drawable.cake,
        R.drawable.wireclothes, R.drawable.kiwifruit,
R.drawable.scarf});

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mListView = findViewById(R.id.lv); //初始化 ListView 控
    件
    MyBaseAdapter mAdapter = new MyBaseAdapter(); //创建一个 Adapter 的实例
    mListView.setAdapter(mAdapter); //设置 Adapter
}

```

```

class MyBaseAdapter extends BaseAdapter {
    @Override
    public int getCount() { //获取条目的总数
        return titles.length; //返回条目的总数
    }
}

```

```

@Override
public Object getItem(int position) {
    return titles[position]; //返回条目的数据对象
}

```

```

@Override
public long getItemId(int position) {
    return position; //返回条目的 Id
}
//获取条目的视图
@Override
public View getView(int position, View convertView,
    ViewGroup parent) {
    ViewHolder holder = null;
    if (convertView == null) {
        //将 list_item.xml 文件找出来并转换成 View 对象
        convertView = View.inflate(MainActivity.this,
R.layout.list_item, null);
        //找到 list_item.xml 中创建的 TextView
        holder = new ViewHolder();
        holder.title
= convertView.findViewById(R.id.title);
        holder.price =

```

```

convertView.findViewById(R.id.price);
        holder.iv = convertView.findViewById(R.id.iv);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }
    holder.title.setText(titles[position]);
    holder.price.setText(prices[position]);
    holder.iv.setBackgroundResource(icons[position]);
    return convertView;
}
class ViewHolder {
    TextView title, price;
    ImageView iv;
}
}
}

```

设置点击事件:

```

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // 点击后触发的逻辑: position 就是点击的 Item 位置
        String clickedText = dataList.get(position); // 拿到点击项的数据
        Toast.makeText(MainActivity.this, "点击了: " + clickedText, Toast.LENGTH_SHORT).show();
    }
});

```

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业。

扩展:

RecyclerView 是 Android 开发中用于展示大量数据列表的核心控件，其使用流程可以分为 7 个关键步骤，从依赖配置到数据更新，每个环节都有明确的职责。以下是详细流程:

一、添加依赖

首先需要在项目中引入 RecyclerView 库（基于 AndroidX），确保能正常使用相关类。

打开模块级别的 build.gradle（通常是 app/build.gradle）

在 dependencies 中添加依赖:

```
gradle
```

```
implementation 'androidx.recyclerview:recyclerview:1.3.2'  
// 可选: 如果需要使用默认动画或更多功能  
implementation  
'androidx.recyclerview:recyclerview-selection:1.1.0'
```

同步项目 (Sync Now)

二、定义 Item 布局

每个列表项 (Item) 的 UI 样式需要单独定义, 例如展示文字、图片或复杂布局。

示例 (res/layout/item_list.xml):

xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="80dp"  
    android:padding="16dp"  
    android:orientation="horizontal">  
  
    <ImageView  
        android:id="@+id/iv_icon"  
        android:layout_width="48dp"  
        android:layout_height="48dp"  
        android:src="@mipmap/ic_launcher" />  
  
    <TextView  
        android:id="@+id/tv_title"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginStart="16dp"  
        android:layout_gravity="center_vertical"  
        android:textSize="18sp"  
        android:text="Item Title" />  
  
</LinearLayout>
```

三、创建数据模型类

定义数据结构, 用于存储每个 Item 所需的数据 (如标题、图片 URL 等)。

示例 (DataModel.java):

java

运行

```
public class DataModel {
    private String title;
    private int iconRes; // 图标资源 ID

    // 构造方法
    public DataModel(String title, int iconRes) {
        this.title = title;
        this.iconRes = iconRes;
    }

    // getter 和 setter
    public String getTitle() { return title; }
    public int getIconRes() { return iconRes; }
}
```

四、实现 ViewHolder

ViewHolder 用于缓存 Item 布局中的控件引用，避免频繁调用 findViewById，提升性能。

示例 (MyViewHolder.java):

java
运行

```
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;

public class MyViewHolder extends RecyclerView.ViewHolder {
    // 声明 Item 中的控件
    ImageView ivIcon;
    TextView tvTitle;

    // 构造方法：初始化控件
    public MyViewHolder(View itemView) {
        super(itemView);
        ivIcon = itemView.findViewById(R.id.iv_icon);
        tvTitle = itemView.findViewById(R.id.tv_title);
    }
}
```

五、实现 Adapter 适配器

Adapter 是连接数据与 RecyclerView 的桥梁，负责创建 ViewHolder、绑定数据到视图，并返回数据数量。

需继承 RecyclerView.Adapter<MyViewHolder>，并实现 3 个核心方法：

示例 (MyAdapter.java)：

java

运行

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class MyAdapter extends RecyclerView.Adapter<MyViewHolder>
{
    // 数据源
    private List<DataModel> mDataList;

    // 构造方法：接收数据源
    public MyAdapter(List<DataModel> dataList) {
        this.mDataList = dataList;
    }

    // 1. 创建 ViewHolder（当需要新的 Item 视图时调用）
    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        // 加载 Item 布局
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_list, parent, false);
        return new MyViewHolder(view);
    }

    // 2. 绑定数据到 ViewHolder（复用已有视图时调用）
    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int
position) {
        // 获取当前位置的数据
```

```

        DataModel data = mDataList.get(position);
        // 绑定数据到控件
        holder.tvTitle.setText(data.getTitle());
        holder.ivIcon.setImageResource(data.getIconRes());
    }

    // 3. 返回数据总数
    @Override
    public int getItemCount() {
        return mDataList == null ? 0 : mDataList.size();
    }
}

```

六、在 Activity/Fragment 中初始化 RecyclerView
完成上述准备后，在界面中初始化 RecyclerView，设置布局管理器和适配器。

示例 (MainActivity.java):

java
运行

```

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private RecyclerView mRecyclerView;
    private MyAdapter mAdapter;
    private List<DataModel> mDataList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // 1. 初始化数据源
        initData();

        // 2. 获取 RecyclerView 实例
        mRecyclerView = findViewById(R.id.recycler_view);
    }
}

```

```

// 3. 设置布局管理器（决定 Item 的排列方式）
// 线性布局（纵向列表）
mRecyclerView.setLayoutManager(new
LinearLayoutManager(this));
// 可选：网格布局（3 列）
//
//                                mRecyclerView.setLayoutManager(new
GridLayoutManager(this, 3));
// 可选：瀑布流布局（2 列，纵向）
//
//                                mRecyclerView.setLayoutManager(new
StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.VERTICAL));

// 4. 设置适配器
mAdapter = new MyAdapter(mDataList);
mRecyclerView.setAdapter(mAdapter);
}

// 初始化测试数据
private void initData() {
    mDataList = new ArrayList<>();
    for (int i = 0; i < 20; i++) {
        mDataList.add(new    DataModel("Item    "    +    i,
R.mipmap.ic_launcher));
    }
}
}
}

```

七、在 Activity 布局中添加 RecyclerView
在界面布局中定义 RecyclerView 控件，作为列表的容器。

示例（res/layout/activity_main.xml）：

```

xml

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

```

	<pre></androidx.constraintlayout.widget.ConstraintLayout></pre> <p style="text-align: center;">第七、八课时 (上机练习)</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机：（考察知识点为 TextView 控件、EditText 控件、Button 控件、ImageView 控件、RadioButton 控件、CheckBox 控件）</p> <p>形式：独立完成</p> <p>题目：</p> <p>通过不同的案例演示 TextView 控件、EditText 控件、Button 控件、ImageView 控件、RadioButton 控件和 CheckBox 控件的使用，要求如下：</p> <ul style="list-style-type: none"> (8) 通过 TextView 控件演示一个显示斜体文本的界面。 (9) 通过 EditText 控件演示如何在界面上编辑文本信息。 (10) 通过三种方式为界面上的 Button 控件设置点击事件。 (11) 通过 ImageView 控件显示一张图片。 (12) 通过 RadioGroup 和 RadioButton 控件实现单选按钮的功能。 (13) 通过 CheckBox 控件统计用户的兴趣爱好。 <p>上机：（考察知识点为实战演练——超市界面、RecyclerView 控件的使用、实战演练——相册界面、自定义 View）</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照以下要求完成操作，要求如下：</p> <ul style="list-style-type: none"> (1) 使用 ListView 控件实现一个超市界面的案例。
教学后记	<p>在讲解 ListView 性能优化时，仅通过代码示例演示 ViewHolder 的使用，未深入剖析“不复用视图会导致的内存开销”（如滑动时频繁创建 View 对象的内存变化），导致部分学生只知“要这么写”，不知“为什么这么写”，后续需结合 Android Studio 的 Profiler 工具，直观展示复用前后的内存占用差异，帮助学生理解原理。</p>

课题名称	第 4 章 程序活动单元 Activity	计划课时	10 课时
教学引入	Android 中的四大组件分别是 Activity、Service、ContentProvider 和 BroadcastReceiver，其中，Activity 是一个负责与用户交互的组件，每个 Android 应用中都会用 Activity 来显示界面以及处理界面上一些控件的事件。本章将针对 Activity 组件进行详细讲解，其他组件的介绍会在后续章节中讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生掌握 Activity 生命周期的作用，能够正确使用每个方法 ● 使学生掌握 Activity 的创建、配置、启动和关闭的方式，能够完成创建、配置、启动和关闭 Activity ● 使学生掌握 Intent 和 IntentFilter 的用法，能够灵活使用 Intent 与 IntentFilter ● 使学生掌握 Activity 之间的跳转方式，能够独立实现 Activity 之间的跳转功能 ● 使学生掌握 Activity 的任务栈和四种启动模式的作用，能够正确使用任务栈和四种启动模式 ● 使学生掌握 Fragment 的使用，能够完成在 Activity 中添加 Fragment 		
教学重点	<ul style="list-style-type: none"> ● 生命周期方法 ● 创建 Activity ● 配置 Activity ● 启动和关闭 Activity ● Intent ● IntentFilter ● 在 Activity 之间的数据传递 ● Activity 之间的数据回传 ● Android 中的任务栈 ● Activity 的启动模式 ● Fragment 的生命周期 ● 创建 Fragment ● 在 Activity 中添加 Fragment 		
教学难点	<ul style="list-style-type: none"> ● 在 Activity 之间的数据传递 ● 实战演练——小猴子摘桃 ● Activity 的启动模式 ● 实战演练——订餐菜单 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	讲解 Activity 生命周期时，以“人类生命周期”类比的基础上，进一步拓展“系统思维”：Activity 的每一个状态切换（如从运行到暂停、停止到重启）都不是孤立的，需结合用户操作（如切换应用、锁屏）和系统资源（如内存不足）综合判断。例如，onPause() 方法中需暂停耗时操作、onDestroy() 中需释放资源，这如同“人在不同人生阶段需承担不同责任、做好相应准备”。引导学生学会从“用户操作 - 系统响应 - 代码处理”的全局视角分析问题，避免仅关注单一方法实现，培养“整体大于部分之和”的系统观念。		

第一课时

(生命周期状态、生命周期方法)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、创设情景，导入新课

每个人都是有生命的，每个人从出生到老去的过程中都需要经历少儿期、青年期、中年期和老年期四个阶段。同样，Activity 也是有“生命”的，Activity 从创建到销毁的整个过程就是 Activity 的生命周期，Activity 的生命周期包含五种状态，这五种状态好比人类生命过程中经历的四个阶段。本节课将针对 Activity 生命周期中的五种状态和这些状态对应的生命周期方法进行详细讲解。

三、新课讲解

知识点 1-生命周期状态

教师通过 PPT 讲解生命周期状态。

- (1) 启动状态。
- (2) 运行状态。
- (3) 暂停状态。
- (4) 停止状态。
- (5) 销毁状态。



知识点 2-生命周期方法

教师通过 PPT 结合实际操作的方式讲解生命周期方法。

- (1) Activity 中的回调方法。

onCreate(): Activity 创建时调用，通常做一些初始化设置。

onStart(): Activity 即将可见时调用。

onResume(): Activity 获取焦点时调用。

onPause(): 当前 Activity 被其他 Activity 覆盖或屏幕锁屏时调用。

onStop(): Activity 对用户不可见时调用。

onRestart(): Activity 从停止状态到再次启动时调用。

onDestroy(): Activity 销毁时调用。

- (2) Activity 的生命周期模型。

(2) 创建 Activity, 展示 Activity 的代码。

知识点 2-配置 Activity

教师通过 PPT 结合实际操作的方式讲解配置 Activity。(了解)

(1) 为什么要配置 Activity。

(2) 演示 Activity 的配置。

通过第二种方式创建的 Activity 需要手动在清单文件中配置创建的 Activity。在 AndroidManifest.xml 文件的<application>标签中配置 SecondActivity, 示例代码如下:

```
<activity android:name="cn.itcast.activitylifecycle.SecondActivity" />
```

知识点 3-启动和关闭 Activity

教师通过 PPT 讲解启动和关闭 Activity。

(1) 如何启动 Activity。

```
Intent intent = new Intent(MainActivity.this,SecondActivity.class);
startActivity(intent);
```

(2) 如何关闭 Activity。

如果想要关闭当前的 Activity, 可以调用 Activity 提供的 finish()方法。

四、归纳总结

教师回顾本节课所讲的内容, 并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通布置本节课作业以及下节课的预习作业。

第三、四课时

(Intent、IntentFilter)

一、复习巩固

教师通过上节课作业的完成情况, 对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

在 Android 程序中, Activity、Service 和 BroadcastReceiver 这三种核心组件都需要使用 Intent 进行操作, 例如, 如果用户需要从一个 Activity 切换到另一个 Activity, 则必须使用 Intent 来进行切换。Intent 用于相同或者不同应用程序组件间的绑定。本节课将针对 Intent 和 IntentFilter 进行详细讲解。

三、新课讲解

知识点 1-Intent

教师通过 PPT 结合实际操作的方式讲解 Intent。

(1) 通过邮递员的例子引出组件之间怎么进行数据传递。

(2) Intent 的概念和作用。

Intent 被称为意图, 是程序中各组件间进行交互的一种重要方式, 它不仅指定当前组件要执行的动作, 还可以在不同组件之间进行数据传递。根据开启目标组件的方式不同, Intent 被分为两种类型, 分别为显式 Intent 和隐式 Intent。

(3) Intent 的显式意图和隐式意图。

显式 Intent 指的是直接指定目标组件，例如，使用 Intent 显式指定要跳转的目标 Activity，示例代码如下：

```
Intent intent = new Intent(this, SecondActivity.class);
startActivity(intent);
```

隐式 Intent 不会明确指出需要开启的目标组件，它被广泛地应用在不同应用程序之间传递数据。

Android 系统会使用 IntentFilter 匹配属性 action、data、category，这 3 个属性的具体介绍如下：

action: 表示 Intent 对象要完成的动作。

data: 表示 Intent 对象中传递的数据。

category: 表示为 action 添加的额外信息。

```
<activity android:name=".SecondActivity">
    <intent-filter>
        <action android:name="cn.itcast.START_ACTIVITY"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

```
Intent intent = new Intent();
intent.setAction("cn.itcast.START_ACTIVITY");
startActivity(intent);
```

需要注意的是，在使用隐式 Intent 开启 Activity 时，系统会默认为 Intent 配置 category，将 category 的属性 name 的值设置为

“android.intent.category.DEFAULT”，因此为了被开启的 Activity 能够接收隐式 Intent，必须在 AndroidManifest.xml 文件中的 SecondActivity 对应的<intent-filter> 标签中，将<category />标签中的属性 android:name 的值设置为

“android.intent.category.DEFAULT”。

知识点 2-IntentFilter

教师通过 PPT 结合实际操作的方式讲解 IntentFilter。

(1) action 属性匹配规则。

```
<intent-filter>
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.VIEW" />
    .....
</intent-filter>
```

只要 Intent 携带的 action 与其中一个<intent-filter>标签中 action 的声明相同，action 属性就匹配成功。

在清单文件中为 Activity 添加<intent-filter>标签时，必须添加 action 属性，否则隐式 Intent 无法开启该 Activity。

(2) data 属性匹配规则。

data 属性用来指定数据的 URI 或者数据 MIME 类型，它的值通常与 Intent 的 action 属性有关联，在清单文件中设置 data 属性的示例代码如下：

```
<intent-filter>
    <data android:mimeType="video/mpeg" android:scheme="http....." />
    <data android:mimeType="audio/mpeg" android:scheme="http....." />
    .....
</intent-filter>
```

隐式 Intent 携带的 data 数据只要与 IntentFilter 中的任意一个 data 声明相同，data 属性就匹配成功。

(3) category 属性匹配规则。

```
<intent-filter>
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    .....
</intent-filter>
```

隐式 Intent 中声明的 category 必须全部能够与某一个 IntentFilter 中的 category 匹配才算匹配成功。

需要注意的是，IntentFilter 中罗列的 category 属性数量必须大于或者等于隐式 Intent 携带的 category 属性数量时，category 属性才能匹配成功。如果一个隐式 Intent 没有设置 category 属性，那么他可以通过任何一个 IntentFilter（过滤器）的 category 匹配。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师布置本节课作业。

第五、六课时

(Activity 之间的数据传递、Activity 之间的数据回传、实战演练——小猴子摘桃)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

一个 Android 程序通常会包含多个 Activity，这些 Activity 之间可以互相跳转并传递数据。接下来，本节课将针对 Activity 之间的跳转及数据传递进行详细讲解。

三、新课讲解

知识点 1-Activity 之间的数据传递

教师通过 PPT 结合实际操作的方式讲解在 Activity 之间的数据传递。

(1) 使用 Intent 的 putExtra()方法传递数据。

```
Intent intent = new Intent();
intent.setClass(MainActivity.this,SecondActivity.class);
intent.putExtra("studentName","王晓明");
intent.putExtra("englishScore",98);
intent.putExtra("isPassed",true);
startActivity(intent);

Intent intent = getIntent();
String name = intent.getStringExtra("studentName");
int englishScore = intent.getIntExtra("englishScore",0);
boolean isPassed = intent.getBooleanExtra("isPassed",true);
```

(2) 使用 Bundle 类传递数据。

```
Intent intent = new Intent();
intent.setClass(this,SecondActivity.class);
Bundle bundle = new Bundle();
bundle.putString("account", "王小明");
intent.putExtras(bundle);
startActivity(intent);

Bundle bundle = getIntent().getExtras();
String account = bundle.getString("account");
```

知识点 2-Activity 之间的数据回传

教师通过 PPT 结合实际操作的方式讲解 Activity 之间的数据回传。

(1) 低于 Android 10 版本中，Activity 之间的数据回传。

- startActivityForResult()方法。
- setResult()方法。
- onActivityResult()方法。

(1) startActivityForResult()方法

用于开启一个Activity，当开启的Activity销毁时，会从销毁的Activity中返回数据，该方法的语法格式如下：

```
startActivityResult(Intent intent, int requestCode)
```

意图对象 表示请求码，用于标识请求的来源

(2) setResult() 方法

用于携带数据进行回传，该方法的语法格式如下：

```
setResult(int resultCode, Intent intent)
```

表示返回码，用于标识返回的数据来自哪一个Activity

(3) onActivityResult()方法

用于接收回传的数据，该方法的语法格式如下：

```
onActivityResult(int requestCode, int resultCode, Intent data)
```

请求码 回传码 回传数据

程序会根据传递的参数requestCode与resultCode来识别数据的来源。

```
Intent intent = new Intent(MainActivity.this,SecondActivity.class);
startActivityResult(intent,1);

Intent intent = new Intent();
intent.putExtra("data","Hello MainActivity");
setResult(2,intent);
finish();

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1&&resultCode == 2){
        String acquiredData= data.getStringExtra("data");
        Toast.makeText(MainActivity.this,acquiredData,Toast.LENGTH_SHORT).show();
    }
}
```

(2) Android 10 或以上版本中，Activity 之间的数据回传。（难点）

- registerActivityResult()方法。
- launch()方法。

使用registerActivityResult()方法替代了startActivityForResult()方法，简化了数据回传的编写方式。还使用了launch()方法与setResult()方法，其中setResult()方法用于携带数据进行回传。

(1) registerActivityResult()方法

用于接收回传的数据，并获取ActivityResultLauncher类的对象，语法格式如下所示。

```
registerActivityResult(ActivityResultContract<I, O> contract, 协议,  
ActivityResultCallback<O> callback)
```

需要注意的是，registerActivityResult()方法的返回值是ActivityResultLauncher类的对象，该对象必须在onCreate()方法或者onAttach()方法中初始化。ActivityResultLauncher类是一个启动器，可以调用launch()方法启动Activity。

launch()方法用于启动Activity，该方法的语法格式如下所示。

launch(I input)

```
Intent data = result.getData();  
String acquiredData = data.getStringExtra("data"); //获取回传的数据  
Toast.makeText(MainActivity.this,acquiredData,  
Toast.LENGTH_SHORT).show();  
Intent intent = new Intent(MainActivity.this,SecondActivity.class);  
launcher.launch(intent);
```

实例：

以下是使用传统匿名类（不使用箭头函数）实现的 Android 10+ Activity 数据回传示例：

1. 主 Activity (发送数据并等待回传)

java

```
public class MainActivity extends AppCompatActivity {  
    // 定义 Activity Result Launcher  
    private ActivityResultLauncher<Intent> someActivityResultLauncher;  
    private TextView resultTextView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        resultTextView = findViewById(R.id.result_text);  
        Button launchButton = findViewById(R.id.launch_button);
```

```

// 注册 Activity 结果启动器（使用传统匿名类）
someActivityResultLauncher = registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
    new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if (result.getResultCode() == Activity.RESULT_OK) {
                Intent data = result.getData();
                if (data != null) {
                    // 获取回传的数据
                    String returnedData =
data.getStringExtra("return_data");
                    resultTextView.setText("回传数据：" +
returnedData);
                }
            }
        }
    });

launchButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 创建 Intent 并添加数据
        Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
        intent.putExtra("extra_data", "这是从 MainActivity 发送的数据");

        // 启动 Activity 并等待结果
        someActivityResultLauncher.launch(intent);
    }
});
}
}

```

2. 第二个 Activity (接收数据并回传)

java

```

public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_second);

// 获取从 MainActivity 传递过来的数据
final String receivedData = getIntent().getStringExtra("extra_data");
TextView receivedTextView = findViewById(R.id.received_text);
receivedTextView.setText("接收到的数据: " + receivedData);

Button returnButton = findViewById(R.id.return_button);
returnButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 创建回传的 Intent
        Intent returnIntent = new Intent();
        returnIntent.putExtra("return_data", "这是从 SecondActivity 回
传的数据");

        // 设置结果并结束当前 Activity
        setResult(Activity.RESULT_OK, returnIntent);
        finish();
    }
});
}
}

```

知识点 3-实战演练——小猴子摘桃（了解）

教师通过 PPT 结合实际操作的方式讲解实战演练——小猴子摘桃。

- (1) 搭建首页界面布局。
- (2) 创建程序。
 - 导入界面图片。
 - 添加界面控件。
 - 去掉默认标题栏。
- (3) 搭建桃园界面布局。
 - 创建桃园界面。
 - 导入界面图片。
 - 添加界面控件。
- (4) 实现小猴子摘桃的功能。
 - 实现首页界面的显示效果。

- 实现桃园界面的摘桃效果。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第七课时

(Android 中的任务栈、Activity 的启动模式、Fragment 简介、Fragment 的生命周期)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

上节课讲解了 Activity 之间的数据传递与数据回传，接下来本节课将针对 Android 中的任务栈、Activity 的启动模式、Fragment 简介和 Fragment 的生命周期进行讲解。

三、新课讲解

知识点 1-Android 中的任务栈

教师通过 PPT 讲解 Android 中的任务栈。

- (1) 什么是任务栈。
- (2) 任务栈的两个基本操作，压栈和出栈。
- (3) Activity 在任务栈中的存放情况。

任务栈：一种用来存放 Activity 实例的容器

特点：“先进后出”

操作：压栈和出栈

知识点 2-Activity 的启动模式（了解）

教师通过 PPT 讲解 Activity 的启动模式。

- (1) standard 模式。
- (2) singleTop 模式。
- (3) singleTask 模式。
- (4) singleInstance 模式。

知识点 3-Fragment 简介

教师通过 PPT 讲解 Fragment 简介。

- (1) Fragment 的作用。
- (2) 通过图例方式讲解 Fragment 的用途。

知识点 4-Fragment 的生命周期

教师通过 PPT 讲解 Fragment 的生命周期。

- (1) 通过图例的方式进行讲解 Fragment 的生命周期。

Fragment（碎片）是一种嵌入在 Activity 中的 UI 片段，它可以用来描述 Activity 中的一部分布局。

Fragment 不能独立存在，必须嵌入到 Activity 中使用，所以 Fragment 生命周期状态直接受其所属 Activity 的生命周期状态影响。

当在 Activity 中创建 Fragment 时，Fragment 处于启动状态；

当 Activity 被暂停时，其中的所有 Fragment 也被暂停；
当 Activity 被销毁时，所有在该 Activity 中的 Fragment 也被销毁。
当一个 Activity 处于运行状态时，可以单独地对每一个 Fragment 进行操作，如添加或删除，当添加时，Fragment 处于启动状态。当删除时，Fragment 处于销毁状态。

(2) Fragment 生命周期比 Activity 多了以下几个方法。

- onAttach()。
- onCreateView()。
- onActivityCreated()。
- onDestroyView()。
- onDetach()。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并予以指导。

第八、九课时

(创建 Fragment、在 Activity 中添加 Fragment、实战演练——订餐菜单)

一、复习巩固

对知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

上节讲解了 Fragment 简介与 Fragment 的生命周期，本节课将针对创建 Fragment 与在 Activity 中添加 Fragment 的内容进行详细讲解，并结合学习的 Fragment 内容，实现一个订餐菜单的案例。

三、新课讲解

知识点 1-创建 Fragment

教师通过 PPT 讲解创建 Fragment。

(1) 创建 Fragment 时必须创建一个类继承自 Fragment。

(2) 演示创建 NewsListFragment 类的示例代码。

与 Activity 类似，创建 Fragment 时必须创建一个类继承 Fragment。

```
public class NewsListFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment, container, false);
        return v;
    }
}
```

在布局文件中添加Fragment

```
<fragment
    android:name="cn.itcast.NewsListFragment"
    android:id="@+id/newslst"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

Fragment的全路径名称

在Activity中动态加载Fragment

当Activity运行时，也可以将Fragment动态添加到Activity中，具体步骤如下：

- (1) 创建一个Fragment的实例对象。
- (2) 获取FragmentManager (Fragment管理器)的实例。
- (3) 开启FragmentTransaction(事务)。
- (4) 向Activity的布局容器(一般为FrameLayout)中添加Fragment。
- (5) 通过commit()方法提交事务。

在Activity中动态加载Fragment

```
NewsListFragment fragment = new NewsListFragment
FragmentManager fm = getFragmentManager();
FragmentTransaction beginTransaction = fm.beginTransaction();
beginTransaction.replace(R.id.ll,fragment);
beginTransaction.commit();
```

开启事务
添加一个Fragment
提交事务

知识点2-在 Activity 中添加 Fragment

教师通过 PPT 结合实际操作的方式讲解在 Activity 中添加 Fragment。在 Activity 中添加 Fragment 有两种方式，具体内容如下所示。

- (1) 在布局文件中添加 Fragment。
- (2) 在 Activity 中动态加载 Fragment。
 - 创建一个 Fragment 的实例对象。
 - 获取 FragmentManager (Fragment 管理器) 的实例。
 - 开启 FragmentTransaction (事务)。
 - 向 Activity 的布局容器 (一般为 FrameLayout) 中添加 Fragment。
 - 通过 commit()方法提交事务。

知识点3-实战演练——订餐菜单

	<p>教师通过 PPT 结合实际操作的方式讲解实战演练——订餐菜单。</p> <p>(1) 搭建菜单界面布局。</p> <ul style="list-style-type: none"> ● 创建程序。 ● 导入界面图片。 ● 添加浅灰色与深灰色颜色值。 ● 创建文本样式。 ● 添加界面控件。 ● 搭建左侧菜单栏界面布局。 ● 搭建右侧菜单列表界面布局。 ● 搭建菜单列表界面的条目布局。 ● 修改默认标题栏的名称。 <p>(2) 实现菜单界面功能。</p> <ul style="list-style-type: none"> ● 封装菜品信息的实体类。 ● 加载左侧菜单栏界面布局。 ● 编写菜单列表的适配器。 ● 加载右侧菜单栏界面布局。 ● 实现显示菜单的效果。 <p>(3) 运行程序。</p> <p>四、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p> <p>布置本节课作业以及下节课的预习作业。</p> <p style="text-align: center;">第十课时 (上机练习)</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机：(考察知识点为启动和关闭 Activity、Intent、IntentFilter、Activity 之间的数据传递、Activity 之间的数据回传、实战演练——小猴子摘桃、Activity 的启动模式)</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照以下要求完成操作，要求如下：</p> <ul style="list-style-type: none"> (2) 实现启动和关闭 Activity。 (3) 使用 Intent 与 IntentFilter 实现 Activity 之间的跳转。 (4) 实现 Activity 之间的数据传递与回传。 (5) 测试 Activity 的启动模式。
教学后记	<p>Android 10 + 与低版本的数据回传方法差异较大，教学中虽分别讲解了 startActivityResult() 和 registerActivityResult(), 但未深入对比两者的“设计理念差异”(如后者通过“协议 + 回调”解耦，前者依赖 onActivityResult() 集中处理)，也未强调“低版本方法已过时”的兼容性问题，导致部分学生仍执着于使用旧方法，缺乏“关注技术迭代”的意识。</p>

课题名称	第 5 章 数据存储	计划课时	8 课时
教学引入	大部分应用程序都会涉及数据存储，Android 程序也不例外。Android 中的数据存储方式有五种，分别为文件存储、SharedPreferences 存储、SQLite 数据库存储、ContentProvider 存储以及网络存储。由于 ContentProvider 存储与网络存储会在后续章节中讲解，所以本章将重点针对文件存储、SharedPreferences 和 SQLite 数据库的知识进行讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生熟悉数据存储方式，能够归纳五种数据存储方式的特点 ● 使学生掌握文件存储的方式，能够使用文件存储数据 ● 使学生掌握 SharedPreferences 的使用，能够存储、读取和删除数据 ● 使学生掌握 SQLite 数据库的使用，能够实现数据的添加、删除、修改和查询功能 		
教学重点	<ul style="list-style-type: none"> ● 将数据存入文件中 ● 从文件中读取数据 ● 将数据存入 SharedPreferences 中 ● 读取与删除 SharedPreferences 中的数据 ● SQLite 数据库的创建 ● SQLite 数据库的基本操作 ● SQLite 数据库中的事务 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——保存 QQ 账号与密码（文件存储） ● 实战演练——保存 QQ 账号与密码（SharedPreferences 存储） ● SQLite 数据库的基本操作 ● 实战演练——绿豆通讯录 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	在讲解文件存储的 MODE_PRIVATE 模式时，结合“数据隐私保护”的价值观：该模式下文件仅当前程序可读写，如同“个人日记本加锁，只有自己能看到”。对比 MODE_WORLD_READABLE（其他程序可读取）的风险，举例说明：若 QQ 账号密码用开放模式存储，可能被恶意程序窃取，导致用户账号安全隐患。引导学生理解“默认私有、按需开放”的存储原则，培养“将用户隐私放在首位”的职业操守，避免因图方便使用不安全的存储模式。		
教学过程	<p style="text-align: center;">第一、二课时</p> <p style="text-align: center;">（数据存储方式、将数据存入文件中、从文件中读取数据、实战演练——保存 QQ 账号与密码（文件存储））</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过直接引入的方式导入新课</p> <p>当我们开发一个 Android 应用时，需要在有网络时将获取的首页数据保存到本地，从而避免没有网络时首页界面显示空白的情况。这些页面的数据存储有哪些方式呢？数据是否可以存放在文件中呢？数据该如何存放呢？本节课我们将针对数据存储方式、将数据存入文件中、从文件中读取数据和实战演练——保存 QQ 账号与密码（文件存储）进行详细讲解。</p>		

三、新课讲解

知识点 1-数据存储方式

教师通过 PPT 讲解数据存储方式。

- (1) 文件存储。
- (2) SharedPreferences 存储。
- (3) SQLite 数据库存储。
- (4) ContentProvider 存储。
- (5) 网络存储。

知识点 2-将数据存入文件中

教师通过 PPT 结合实际操作的方式讲解将数据存入文件中。

Android 中的文件存储分为内部存储和外部存储。

- (1) 内部存储的存储方式和示例代码。

内部存储

```
FileOutputStream fos = openFileOutput(String name, int mode);  
FileInputStream fis = openFileInput(String name);
```

mode取值：
打开应用程序对应的输入流，
读取指定文件中的数据

- MODE_PRIVATE: 该文件只能被当前程序读写;
- MODE_APPEND: 该文件的内容可以追加;
- MODE_WORLD_READABLE: 该文件的内容可以被其他程序读取;
- MODE_WORLD_WRITEABLE: 该文件的内容可以被其他程序写入。

打开应用程序中对应的输出流，
将数据存储到指定的文件中
参数name: 文件名
参数mode: 文件的操作模式

注意：Android系统有一套自己的安全模式，默认情况下任何应用
创建的文件都是私有的，其他程序无法访问。

内部存储

```
String fileName = "data.txt"; //文件名称  
String content = "helloworld"; //保存数据  
FileOutputStream fos = null;  
try {  
    fos = openFileOutput(fileName, MODE_PRIVATE);  
    fos.write(content.getBytes());  
} catch (Exception e) {  
    e.printStackTrace();  
}.....
```

将数据写入文件中

- (2) 外部存储的存储方式和示例代码。

外部存储

```
String state = Environment.getExternalStorageState();
if (state.equals(Environment.MEDIA_MOUNTED)) {
    String SDPath;
    if (Build.VERSION.SDK_INT > 29) {
        SDPath = getExternalFilesDir(null).getAbsolutePath();
    } else {
        SDPath = Environment.getExternalStorageDirectory().getPath();
    }
    File file = new File(SDPath, "data.txt");
    String data = "HelloWorld";
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(file);
        fos.write(data.getBytes());
    } .....
```

获取外部设备状态
判断外部设备是否可用
获取SD卡目录
获取SD卡目录

知识点 3-从文件中读取数据

教师通过 PPT 结合实际操作的方式讲解从文件中读取数据。

(1) 读取内部存储中的文件数据。

读取内部存储中的文件数据

```
String content = "";
FileInputStream fis = null;
try {
    fis = openFileInput("data.txt");
    byte[] buffer = new byte[fis.available()];
    fis.read(buffer);
    content = new String(buffer);
} catch (Exception e) {
    e.printStackTrace();
}
```

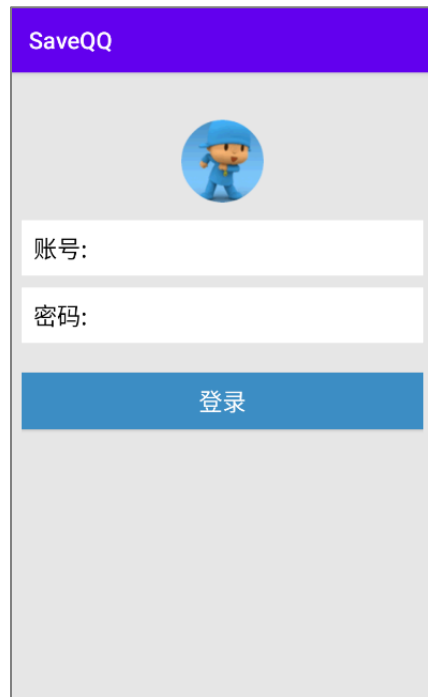
获取文件输入流对象
创建缓冲区并获取文件长度
将文件内容读取到buffer缓冲区

(2) 读取外部存储中的文件数据。(了解)

知识点 4-实战演练——保存 QQ 账号与密码 (文件存储)

教师通过 PPT 结合实际操作的方式讲解实战演练——保存 QQ 账号与密码 (文件存储)。

- (1) 创建程序。
- (2) 导入界面图片。
- (3) 添加界面控件。
- (4) 创建工具类。
- (5) 实现 QQ 账号和密码的保存与读取功能。
- (6) 运行程序。



```
public class FileSaveQQ {  
    public static boolean saveUserInfo(Context context, String account, String  
        password) {  
        FileOutputStream fos = null;  
        try {  
            //获取文件的输出流对象 fos
```

```

        fos = context.openFileOutput("data.txt",
Context.MODE_PRIVATE);
        //将数据转换为字节码的形式写入 data.txt 文件中
        fos.write((account + ":" + password).getBytes());
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    } finally {
        try {
            if(fos != null){
                fos.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

public static Map<String, String> getUserInfo(Context context) {
    String content = "";
    FileInputStream fis = null;
    try {
        //获取文件的输入流对象 fis
        fis = context.openFileInput("data.txt");
        //将输入流对象中的数据转换为字节码的形式
        byte[] buffer = new byte[fis.available()];
        fis.read(buffer);//通过 read()方法读取字节码中的数据
        content = new String(buffer); //将获取的字节码转换为字符串
        Map<String, String> userMap = new HashMap<String,
String>();
        String[] infos = content.split(":");//将字符串以 “：” 分隔后形
成一个数组的形式
        userMap.put("account", infos[0]); //将数组中的第一个数据
放入 userMap 集合中
        userMap.put("password", infos[1]); //将数组中的第二个数据放
入 userMap 集合中
        return userMap;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    } finally {
        try {
            if(fis != null){
                fis.close();
            }
        }
    }
}

```


知识点 2-读取与删除 SharedPreferences 中的数据

教师通过 PPT 结合实际操作的方式讲解读取与删除 SharedPreferences 中的数据。

- (1) 读取 SharedPreferences 中的数据。
- (2) 删除 SharedPreferences 中的数据。

读取 SharedPreferences 中的数据：

```
SharedPreferences sp = getSharedPreferences("data",MODE_PRIVATE);  
String data = sp.getString("name","");
```

删除 SharedPreferences 中的数据：

```
editor.remove("name");  
editor.clear();
```

知识点 3-实战演练——保存 QQ 账号与密码(SharedPreferences 存储)

教师通过 PPT 结合实际操作的方式讲解实战演练——保存 QQ 账号与密码 (SharedPreferences 存储)。

- (1) 创建工具类。
- (2) 实现 QQ 账号和密码的保存与读取功能。
- (3) 运行程序。

```
public class SPsaveQQ{  
    // 保存 QQ 账号和登录密码到 data.xml 文件中  
    public static boolean saveUserInfo(Context context, String account, String  
password) {  
        SharedPreferences sp = context.getSharedPreferences("data",  
Context.MODE_PRIVATE);  
        SharedPreferences.Editor edit = sp.edit();  
        edit.putString("userName", account);  
        edit.putString("pwd", password);  
        edit.commit();  
        return true;  
    }  
    //从 data.xml 文件中获取保存的 QQ 账号和密码  
    public static Map<String, String> getUserInfo(Context context) {  
        SharedPreferences sp = context.getSharedPreferences("data",  
Context.MODE_PRIVATE);  
        String account = sp.getString("userName", null);  
        String password = sp.getString("pwd", null);  
        Map<String, String> userMap = new HashMap<String, String>();  
        userMap.put("account", account);
```

```

        userMap.put("password", password);
        return userMap;
    }
}

```

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业。

第五、六课时

(SQLite 数据库的创建、SQLite 数据库的基本操作、SQLite 数据库中的事务、实战演练——绿豆通讯录)

一、复习巩固

教师对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

前面介绍了如何使用文件以及 SharedPreferences 存储数据，这两种方式适合存储简单数据，当需要存储大量数据时显然是不适合的。为此 Android 系统提供了 SQLite 数据库，它可以存储应用程序中的大量数据，并对数据进行管理和维护。本节课将针对 SQLite 数据库的创建、SQLite 数据库的基本操作、SQLite 数据库中的事务、实战演练——绿豆通讯录进行详细讲解。

三、新课讲解

知识点 1-SQLite 数据库的创建

教师通过 PPT 结合实际操作的方式讲解 SQLite 数据库的创建。

(1) 创建一个类继承 SQLiteOpenHelper 类。

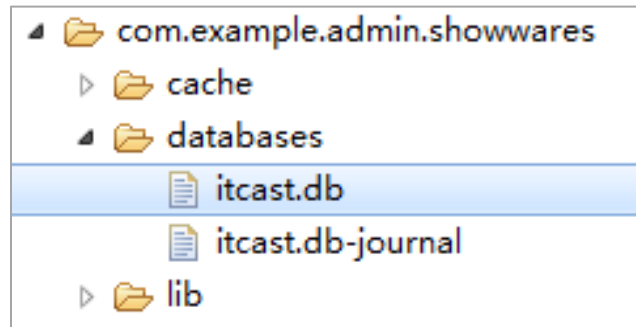
(2) 在该类中重写 onCreate()方法和 onUpgrade()方法即可创建一个 SQLite 数据库。

```

public class MyHelper extends SQLiteOpenHelper {
    public MyHelper(Context context) {
        super(context, "itcast.db", null, 2);
    }
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE information(_id
INTEGER PRIMARY
        KEY AUTOINCREMENT, name VARCHAR(20), price
INTEGER)");
    }
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    }
}

```

查看已经创建好的数据库文件，首先在 Device File Explorer 视图找到数据库文件所在目录 data/data/“项目包名全路径”/databases，数据库文件 itcast.db 如下图所示。



知识点 2-SQLite 数据库的基本操作

教师通过 PPT 结合实际操作的方式讲解 SQLite 数据库的基本操作。

- (1) 添加数据。
- (2) 删除数据。
- (3) 修改数据。
- (4) 查询数据。

```
public void insert(String name,String price) {
    MyHelper helper = new MyHelper(MainActivity.this);
    SQLiteDatabase db = helper.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("name", name);
    values.put("price", price);
    long id = db.insert("information",null,values);
    db.close();
}

public int delete(long id){
    SQLiteDatabase db = helper.getWritableDatabase();
    int number = db.delete("information", "_id=?", new
String[]{id+""});
    db.close();
    return number;
}

public int update(String name, String price) {
    SQLiteDatabase db = helper.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("price", price);
```

```

        int number = db.update("information", values, "name=?",
new String[]{name});
        db.close();
        return number;
    }

    public void find(int id){
        MyHelper helper = new MyHelper(MainActivity.this);
        SQLiteDatabase db = helper.getReadableDatabase();
        Cursor cursor = db.query("information", null, "_id=?", new
String[]{id+""}, null, null, null);
        if (cursor.getCount() != 0){
            while (cursor.moveToNext()){
                String _id = cursor.getString(cursor.getColumnIndex("_id"));
                String name =
cursor.getString(cursor.getColumnIndex("name"));
                String price =
cursor.getString(cursor.getColumnIndex("price"));
            }
            .....
        }
    }

```

使用 `execSQL()` 方法通过 SQL 语句对数据库进行操作，示例代码如下：

```

//增加一条数据
db.execSQL("insert into information (name, price) values (?,?)", new
Object[]{name, price });
//删除一条数据
db.execSQL("delete from information where _id = 1");
//修改一条数据
db.execSQL("update information set name=? where price =?", new
Object[]{name, price });
//执行查询的 SQL 语句
Cursor cursor = db.rawQuery("select * from information where name=?",
new String[]{name});

```

知识点 3-SQLite 数据库中的事务

教师通过 PPT 结合实际操作的方式讲解 SQLite 数据库中的事务。

- (1) 什么是数据库事务。
- (2) 数据库事务正确执行的四个基本要素。
 - 原子性。

- 一致性。
- 隔离性。
- 持久性。

```

PersonSQLiteOpenHelper helper = new PersonSQLiteOpenHelper (getApplication());
SQLiteDatabase db = helper.getWritableDatabase();
db.beginTransaction(); // 开启数据库事务
try {
    db.execSQL("update person set account = account-1000 where name =?", new Object[] { "张三" });
    db.execSQL("update information set account = account + 1000 where name =?", new Object[] { "王五" });
    db.setTransactionSuccessful();
} catch (Exception e) { // 标记数据库事务执行成功
    Log.i("事务处理失败", e.toString());
} finally {
    db.endTransaction();
    db.close(); // 关闭函数 关闭事务
}

```

(3) 通过张三与王五取钱和存钱的例子，使用 SQLite 的事务模拟银行转账功能。

知识点 4-实战演练——绿豆通讯录

教师通过 PPT 结合实际操作的方式讲解实战演练——绿豆通讯录。

- (1) 创建程序。
- (2) 导入界面图片。
- (3) 添加界面控件。
- (4) 实现绿豆通讯录界面的功能。
 - 获取界面控件。
 - 创建绿豆通讯录的数据库与数据表。
 - 查询数据表 information 中的数据。
 - 实现通讯录界面控件的点击事件。
- (5) 运行程序。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

布置本节课作业。

第七、八课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：(考察知识点为实战演练——保存 QQ 账号与密码(文件存储)、实战演练——保存 QQ 账号与密码 (SharedPreferences 存储))

	<p>形式：单独完成</p> <p>题目： 请按照以下要求完成操作，要求如下： （14）实现保存 QQ 账号与密码到文件中的案例。 （15）实现保存 QQ 账号与密码到 SharedPreferences 中的案例。</p> <p>上机：（考察知识点为实战演练——绿豆通讯录）（可选）</p> <p>形式：单独完成</p> <p>题目： 通过 SQLite 数据库的创建与基本操作，实现一个绿豆通讯录案例，要求如下： （6）创建绿豆通讯录程序 Directory。 （7）搭建绿豆通讯录界面布局。 （8）创建绿豆通讯录的数据库与数据表。 （9）实现绿豆通讯录界面的功能。</p>
教学后记	<p>教材中以“功能实现”为主，对数据安全（如加密存储）、性能优化（如 SQL 索引、流关闭）的讲解较少，导致学生缺乏“安全与优化”的意识。例如，实战“QQ 账号密码存储”时，多数学生未意识到明文存储的风险，教师需额外补充加密知识，占用了课堂时间；SQLite 查询优化仅提及基本操作，未深入讲解 Cursor 复用、避免 SELECT *等细节。</p>

课题名称	第 6 章 阶段案例——记事本	计划课时	6 课时
教学引入	通过前面 1~5 章的学习，我们已经对 Android 的基础知识有了大概了解，本章我们将运用前面章节所学的知识，开发一个记事本程序。记事本的主要功能包含记录事件、查看事件、修改事件和删除事件。接下来我们将正式进入记事本程序的开发。		
教学目标	<ul style="list-style-type: none"> ● 1. 能使用 ListView 实现笔记列表展示 ● 2. 掌握 BaseAdapter 适配器的编写与优化 ● 3. 实现记事本的增删改查完整功能 ● 4. 理解 ViewHolder 模式对 ListView 性能的优化作用 ● 5. 培养模块化开发思维 		
教学重点	<ul style="list-style-type: none"> ● 1. ListView 与 BaseAdapter 的绑定使用 ● 2. ViewHolder 模式的应用 ● 3. SQLite 数据库与界面的联动 ● 4. 列表项点击 / 长按事件处理 		
教学难点	<ul style="list-style-type: none"> ● 1. BaseAdapter 中 getView 方法的优化 ● 2. 长按删除的对话框与数据同步 ● 3. 添加 / 修改界面的数据传递与回传 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	在讲解“长按删除笔记”功能时，强调“二次确认”的必要性：删除操作不可逆，弹出对话框让用户确认（“确定要删除这条笔记吗？”），如同“删除重要文件前的提醒”，避免用户误操作导致数据丢失。通过演示“无确认直接删除”与“有确认删除”的差异，让学生理解：开发者的代码决策直接影响用户权益，需秉持“谨慎负责”的态度，不因“图方便”省略必要的安全校验，培养“对用户负责、对产品负责”的担当精神。		
教学过程	<p style="text-align: center;">第一课时：需求分析与项目初始化</p> <p>一、复习巩固（10 分钟）</p> <p>1. 回顾 ListView 基本概念：</p> <ol style="list-style-type: none"> 1. “ListView 的作用是什么？”（高效展示大量数据列表） 2. “BaseAdapter 的四个核心方法是什么？”（getCount、getItem、getItemId、getView） <p>2. 演示简单 ListView 案例，回顾基础用法。</p> <p>二、需求分析（20 分钟）</p> <p>1. 功能需求</p> <ul style="list-style-type: none"> ● 核心功能：添加、查看、修改、删除笔记 ● 数据存储：本地 SQLite 数据库（确保数据持久化） ● 界面组成： 		

- 主界面：标题栏 + ListView 列表 + 添加按钮
- 编辑界面：标题栏（带返回） + 标题输入框 + 内容输入框 + 清空 / 保存按钮

2. 数据库设计

表名	字段名	类型	说明
note	_id	INTEGER	主键（自增）
	title	VARCHAR(50)	笔记标题（非空）
	content	TEXT	笔记内容
	time	VARCHAR(20)	保存时间（格式： yyyy-MM-dd HH:mm）

三、项目初始化实操（30 分钟）

1. 创建项目

- 项目名：NotePadWithListView
- 包名：cn.itcast.notepad
- 语言：Java
- 最小 SDK：API 21

2. 创建包结构

```
plaintext
cn.itcast.notepad
├── activity    // 界面类
├── adapter    // 适配器类
├── bean       // 实体类
├── db         // 数据库相关
└── utils     // 工具类
```

3. 准备布局文件

1. 主界面布局（activity_main.xml）：

```
xml
<?xml version="1.0" encoding="utf-8"?><LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```

android:layout_height="match_parent"
android:orientation="vertical">

<!-- 标题栏 -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="#3F51B5"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="记事本"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:textStyle="bold" />
</LinearLayout>

<!-- 笔记列表 (ListView) -->
<ListView
    android:id="@+id/lv_notes"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:divider="#E0E0E0"
    android:dividerHeight="1dp" />

<!-- 添加按钮 -->
<Button
    android:id="@+id/btn_add"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="#3F51B5"
    android:text="添加笔记"
    android:textColor="#FFFFFF"
    android:textSize="16sp" />
</LinearLayout>

```

1. 列表项布局 (item_note.xml):

xml

```
<?xml version="1.0" encoding="utf-8"?><LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="12dp">

    <!-- 笔记标题 -->
    <TextView
        android:id="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="测试标题"
        android:textColor="#333333"
        android:textSize="16sp"
        android:textStyle="bold" />

    <!-- 笔记时间 -->
    <TextView
        android:id="@+id/tv_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="2024-08-26 10:30"
        android:textColor="#999999"
        android:textSize="12sp" />
</LinearLayout>
```

四、归纳总结（5 分钟）

1. 检查学生项目结构是否正确
2. 强调布局文件中 `layout_weight` 的使用技巧
3. 预告下一课时将实现实体类和数据库操作

第二课时：实体类与数据库操作

一、复习巩固（10 分钟）

1. 回顾 SQLite 数据库操作流程

2. 提问：“为什么要使用实体类？”（映射数据库表，方便数据处理）

二、实体类编写（15 分钟）

在 bean 包下创建 Note.java:

```
java
运行
package cn.itcast.notepad.bean;
/**
 * 笔记实体类：映射 note 表
 */public class Note {
    private int id;        // 主键_id
    private String title;  // 标题
    private String content; // 内容
    private String time;   // 时间

    // 无参构造
    public Note() {}

    // 带参构造
    public Note(int id, String title, String content, String time)
    {
        this.id = id;
        this.title = title;
        this.content = content;
        this.time = time;
    }

    // Getter 和 Setter 方法
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getTitle() { return title; }
    public void setTitle(String title) { this.title = title; }
    public String getContent() { return content; }
    public void setContent(String content) { this.content =
content; }
    public String getTime() { return time; }
    public void setTime(String time) { this.time = time; }}
```

三、数据库操作类编写（30 分钟）

1. 数据库帮助类（NoteOpenHelper.java）

在 db 包下创建:

```

java
运行
package cn.itcast.notepad.db;
import android.content.Context;import
android.database.sqlite.SQLiteDatabase;import
android.database.sqlite.SQLiteOpenHelper;
/**
 * 数据库帮助类：创建数据库和表
 */public class NoteOpenHelper extends SQLiteOpenHelper {
    // 数据库名
    private static final String DB_NAME = "notes.db";
    // 数据库版本
    private static final int DB_VERSION = 1;

    public NoteOpenHelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // 创建 note 表
        String sql = "CREATE TABLE note (" +
            "_id INTEGER PRIMARY KEY AUTOINCREMENT," +
            "title TEXT NOT NULL," +
            "content TEXT," +
            "time TEXT NOT NULL)";
        db.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        // 版本升级时调用
    }}

```

2. 数据库操作工具类 (NoteDao.java)

在 db 包下创建：

```

java
运行
package cn.itcast.notepad.db;
import android.content.ContentValues;import

```

```

android.content.Context;import android.database.Cursor;import
android.database.sqlite.SQLiteDatabase;
import java.util.ArrayList;import java.util.List;
import cn.itcast.notepad.bean.Note;
/**
 * 数据访问对象：封装数据库增删改查操作
 */public class NoteDao {
    private NoteOpenHelper helper;

    public NoteDao(Context context) {
        helper = new NoteOpenHelper(context);
    }

    // 添加笔记
    public long addNote(Note note) {
        SQLiteDatabase db = helper.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("title", note.getTitle());
        values.put("content", note.getContent());
        values.put("time", note.getTime());
        long id = db.insert("note", null, values);
        db.close();
        return id;
    }

    // 删除笔记
    public int deleteNote(int id) {
        SQLiteDatabase db = helper.getWritableDatabase();
        int rows = db.delete("note", "_id=?", new
String[] {String.valueOf(id)});
        db.close();
        return rows;
    }

    // 更新笔记
    public int updateNote(Note note) {
        SQLiteDatabase db = helper.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("title", note.getTitle());
        values.put("content", note.getContent());
        values.put("time", note.getTime());
        int rows = db.update("note", values, "_id=?",
            new String[] {String.valueOf(note.getId())});
        db.close();
    }
}

```

```

        return rows;
    }

    // 查询所有笔记
    public List<Note> queryAllNotes() {
        List<Note> notes = new ArrayList<>();
        SQLiteDatabase db = helper.getReadableDatabase();
        Cursor cursor = db.query("note", null, null, null,
            null, null, "time DESC");

        while (cursor.moveToNext()) {
            int id = cursor.getInt(cursor.getColumnIndex("_id"));
            String title =
cursor.getString(cursor.getColumnIndex("title"));
            String content =
cursor.getString(cursor.getColumnIndex("content"));
            String time =
cursor.getString(cursor.getColumnIndex("time"));
            notes.add(new Note(id, title, content, time));
        }
        cursor.close();
        db.close();
        return notes;
    }

    // 根据 ID 查询笔记
    public Note queryNoteById(int id) {
        SQLiteDatabase db = helper.getReadableDatabase();
        Cursor cursor = db.query("note", null, "_id=?",
            new String[] {String.valueOf(id)}, null, null,
null);

        Note note = null;
        if (cursor.moveToNext()) {
            String title =
cursor.getString(cursor.getColumnIndex("title"));
            String content =
cursor.getString(cursor.getColumnIndex("content"));
            String time =
cursor.getString(cursor.getColumnIndex("time"));
            note = new Note(id, title, content, time);
        }
        cursor.close();
        db.close();
    }

```

```
        return note;
    }}
}
```

四、时间工具类 (TimeUtils.java)

在 utils 包下创建:

```
java
运行
package cn.itcast.notepad.utils;
import java.text.SimpleDateFormat;import java.util.Date;import
java.util.Locale;
/**
 * 时间工具类: 格式化时间
 */public class TimeUtils {
    public static String getCurrentTime() {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm", Locale.CHINA);
        return sdf.format(new Date());
    }
}
```

五、归纳总结 (5 分钟)

1. 强调数据库操作后关闭 SQLiteDatabase 和 Cursor 的重要性
2. 解释 DAO 模式的优势: 分离数据操作与界面逻辑
3. 布置练习: 测试数据库基本操作方法

第三课时: ListView 适配器与主界面

一、复习巩固 (10 分钟)

1. 回顾 BaseAdapter 的工作原理
2. 提问: "为什么 ListView 需要优化?" (避免频繁创建视图对象导致性能问题)

二、编写适配器 (NoteAdapter.java)

在 adapter 包下创建:

```

java
运行
package cn.itcast.notepad.adapter;
import android.content.Context;import
android.view.LayoutInflater;import android.view.View;import
android.view.ViewGroup;import android.widget.BaseAdapter;import
android.widget.TextView;
import java.util.List;
import cn.itcast.notepad.R;import cn.itcast.notepad.bean.Note;
/**
 * 笔记列表适配器：连接 ListView 与数据
 */public class NoteAdapter extends BaseAdapter {
    private Context context;
    private List<Note> noteList;
    private LayoutInflater inflater;

    // 构造方法
    public NoteAdapter(Context context, List<Note> noteList) {
        this.context = context;
        this.noteList = noteList;
        this.inflater = LayoutInflater.from(context);
    }

    // 获取数据总数
    @Override
    public int getCount() {
        return noteList == null ? 0 : noteList.size();
    }

    // 获取指定位置的数据
    @Override
    public Object getItem(int position) {
        return noteList.get(position);
    }

    // 获取指定位置数据的 ID
    @Override
    public long getItemId(int position) {
        return position;
    }

    // 获取列表项视图（核心方法）
    @Override

```

```

public View getView(int position, View convertView, ViewGroup
parent) {
    // 1. 创建/复用 ViewHolder
    ViewHolder holder;
    if (convertView == null) {
        // 加载布局
        convertView = inflater.inflate(R.layout.item_note,
parent, false);
        // 创建 ViewHolder
        holder = new ViewHolder();
        holder.tvTitle =
convertView.findViewById(R.id.tv_title);
        holder.tvTime =
convertView.findViewById(R.id.tv_time);
        // 缓存 ViewHolder
        convertView.setTag(holder);
    } else {
        // 复用 ViewHolder
        holder = (ViewHolder) convertView.getTag();
    }

    // 2. 绑定数据
    Note note = noteList.get(position);
    holder.tvTitle.setText(note.getTitle());
    holder.tvTime.setText(note.getTime());

    return convertView;
}

// ViewHolder: 缓存列表项控件 (优化性能)
static class ViewHolder {
    TextView tvTitle;
    TextView tvTime;
}

// 刷新数据
public void refreshData(List<Note> newList) {
    this.noteList = newList;
    notifyDataSetChanged();
}}

```

教学重点:

- 讲解 ViewHolder 模式的优化原理：减少 findViewById 的调用次数
- 分析 convertView 的作用：复用已创建的视图对象，减少内存消耗
- 强调 notifyDataSetChanged 的使用：数据变化后通知列表刷新

三、主界面实现 (MainActivity.java)

在 activity 包下创建：

```
java
运行
package cn.itcast.notepad.activity;
import android.content.DialogInterface;import
android.content.Intent;import android.os.Bundle;import
android.view.View;import android.widget.AdapterView;import
android.widget.Button;import android.widget.ListView;import
android.widget.Toast;
import androidx.appcompat.app.AlertDialog;import
androidx.appcompat.app.AppCompatActivity;
import java.util.List;
import cn.itcast.notepad.R;import
cn.itcast.notepad.adapter.NoteAdapter;import
cn.itcast.notepad.bean.Note;import cn.itcast.notepad.db.NoteDao;
public class MainActivity extends AppCompatActivity {
    private ListView lvNotes;
    private Button btnAdd;
    private NoteAdapter adapter;
    private NoteDao noteDao;
    private List<Note> noteList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // 初始化控件
        initView();
        // 初始化数据
        initData();
        // 绑定事件
        bindEvents();
    }
}
```

```

private void initView() {
    lvNotes = findViewById(R.id.lv_notes);
    btnAdd = findViewById(R.id.btn_add);
}

private void initData() {
    noteDao = new NoteDao(this);
    // 查询所有笔记
    noteList = noteDao.queryAllNotes();
    // 创建并设置适配器
    adapter = new NoteAdapter(this, noteList);
    lvNotes.setAdapter(adapter);
}

private void bindEvents() {
    // 添加按钮点击事件
    btnAdd.setOnClickListener(v -> {
        Intent intent = new Intent(MainActivity.this,
EditActivity.class);
        startActivity(intent);
    });

    // 列表项点击事件（修改笔记）
    lvNotes.setOnItemClickListener((parent, view, position,
id) -> {
        Note note = noteList.get(position);
        Intent intent = new Intent(MainActivity.this,
EditActivity.class);
        intent.putExtra("note_id", note.getId());
        startActivity(intent);
    });

    // 列表项长按事件（删除笔记）
    lvNotes.setOnItemLongClickListener((parent, view,
position, id) -> {
        Note note = noteList.get(position);
        showDeleteDialog(note.getId(), position);
        return true; // 消费事件，避免触发点击事件
    });
}

// 显示删除确认对话框
private void showDeleteDialog(int noteId, int position) {
    new AlertDialog.Builder(this)

```

```

        .setTitle("删除笔记")
        .setMessage("确定要删除这条笔记吗? ")
        .setPositiveButton("删除", (dialog, which) -> {
            // 执行删除操作
            int rows = noteDao.deleteNote(noteId);
            if (rows > 0) {
                // 删除成功, 更新列表
                noteList.remove(position);
                adapter.refreshData(noteList);
                Toast.makeText(this, "删除成功",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "删除失败",
Toast.LENGTH_SHORT).show();
            }
        })
        .setNegativeButton("取消", null)
        .show();
    }

    // 从编辑界面返回时刷新数据
    @Override
    protected void onResume() {
        super.onResume();
        noteList = noteDao.queryAllNotes();
        adapter.refreshData(noteList);
    }
}

```

四、归纳总结 (10 分钟)

1. 运行程序, 验证列表是否能正常显示 (初始为空)
2. 重点讲解 onResume 方法的作用: 从编辑界面返回时刷新数据
3. 分析长按删除的完整流程: 对话框确认 → 数据库删除 → 列表数据更新

第四课时: 编辑界面实现

一、复习巩固 (5 分钟)

1. 回顾 Activity 之间的数据传递方式
2. 提问: "如何区分添加和修改操作?" (通过传递的 ID 判断, -1 表示添加)

二、编辑界面布局 (activity_edit.xml)

```
xml
<?xml version="1.0" encoding="utf-8"?><LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- 标题栏 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="#3F51B5"
        android:gravity="center_vertical">

        <!-- 返回按钮 -->
        <Button
            android:id="@+id/btn_back"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:background="?android:selectableItemBackground"
            android:text="←"
            android:textColor="#FFFFFF"
            android:textSize="20sp" />

        <!-- 标题文本 -->
        <TextView
            android:id="@+id/tv_title"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:gravity="center"
            android:text="添加笔记"
            android:textColor="#FFFFFF"
            android:textSize="20sp"
            android:textStyle="bold" />
    </LinearLayout>

    <!-- 标题输入框 -->
    <EditText
        android:id="@+id/et_title"
        android:layout_width="match_parent"
```

	<pre> android:layout_height="wrap_content" android:hint="请输入标题" android:padding="12dp" android:textSize="16sp" /> <!-- 内容输入框 --> <EditText android:id="@+id/et_content" android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1" android:gravity="top" android:hint="请输入内容" android:inputType="textMultiLine" android:padding="12dp" android:textSize="14sp" /> <!-- 功能按钮 --> <LinearLayout android:layout_width="match_parent" android:layout_height="50dp" android:orientation="horizontal"> <Button android:id="@+id/btn_clear" android:layout_width="0dp" android:layout_height="match_parent" android:layout_weight="1" android:background="#F44336" android:text="清空" android:textColor="#FFFFFF" android:textSize="16sp" /> <Button android:id="@+id/btn_save" android:layout_width="0dp" android:layout_height="match_parent" android:layout_weight="1" android:background="#4CAF50" android:text="保存" android:textColor="#FFFFFF" android:textSize="16sp" /> </LinearLayout> </LinearLayout></pre>
--	---

三、编辑界面实现 (EditActivity.java)

```
java
运行
package cn.itcast.notepad.activity;
import android.os.Bundle;import android.view.View;import
android.widget.Button;import android.widget.EditText;import
android.widget.TextView;import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import cn.itcast.notepad.R;import
cn.itcast.notepad.bean.Note;import
cn.itcast.notepad.db.NoteDao;import
cn.itcast.notepad.utils.TimeUtils;
public class EditActivity extends AppCompatActivity {
    private EditText etTitle;
    private EditText etContent;
    private Button btnBack;
    private Button btnClear;
    private Button btnSave;
    private TextView tvTitle;

    private NoteDao noteDao;
    private int noteId = -1; // -1 表示添加模式

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit);

        // 初始化控件
        initView();
        // 初始化数据
        initData();
        // 绑定事件
        bindEvents();
    }

    private void initView() {
        etTitle = findViewById(R.id.et_title);
        etContent = findViewById(R.id.et_content);
        btnBack = findViewById(R.id.btn_back);
        btnClear = findViewById(R.id.btn_clear);
        btnSave = findViewById(R.id.btn_save);
    }
}
```

```

        tvTitle = findViewById(R.id.tv_title);
    }

    private void initData() {
        noteDao = new NoteDao(this);
        // 获取传递的笔记 ID
        noteId = getIntent().getIntExtra("note_id", -1);

        if (noteId != -1) {
            // 修改模式
            tvTitle.setText("修改笔记");
            // 查询笔记数据并显示
            Note note = noteDao.queryNoteById(noteId);
            if (note != null) {
                etTitle.setText(note.getTitle());
                etContent.setText(note.getContent());
            }
        } else {
            // 添加模式
            tvTitle.setText("添加笔记");
        }
    }

    private void bindEvents() {
        // 返回按钮
        btnBack.setOnClickListener(v -> finish());

        // 清空按钮
        btnClear.setOnClickListener(v -> {
            etTitle.setText("");
            etContent.setText("");
        });

        // 保存按钮
        btnSave.setOnClickListener(v -> saveNote());
    }

    // 保存笔记
    private void saveNote() {
        String title = etTitle.getText().toString().trim();
        String content = etContent.getText().toString().trim();

        // 验证标题
        if (title.isEmpty()) {

```

```

        Toast.makeText(this, "标题不能为空",
Toast.LENGTH_SHORT).show();
        return;
    }

    // 获取当前时间
    String time = TimeUtils.getCurrentTime();
    Note note = new Note(noteId, title, content, time);

    if (noteId == -1) {
        // 添加操作
        long id = noteDao.addNote(note);
        if (id != -1) {
            Toast.makeText(this, "添加成功",
Toast.LENGTH_SHORT).show();
            finish(); // 返回主界面
        } else {
            Toast.makeText(this, "添加失败",
Toast.LENGTH_SHORT).show();
        }
    } else {
        // 修改操作
        int rows = noteDao.updateNote(note);
        if (rows > 0) {
            Toast.makeText(this, "修改成功",
Toast.LENGTH_SHORT).show();
            finish(); // 返回主界面
        } else {
            Toast.makeText(this, "修改失败",
Toast.LENGTH_SHORT).show();
        }
    }
}
}}

```

四、功能测试（20 分钟）

1. 测试添加功能：
点击“添加笔记”→输入标题和内容→点击“保存”→主界面显示新笔记
2. 测试修改功能：
点击列表项→修改内容→点击“保存”→主界面显示更新后内容
3. 测试删除功能：
长按列表项→点击“删除”→笔记从列表中移除
4. 测试边界情况：

标题为空时点击保存→验证是否显示提示

五、归纳总结（10 分钟）

1. 强调添加和修改复用同一界面的优势：减少代码冗余
2. 分析数据传递流程：主界面→编辑界面（通过 Intent），编辑界面→主界面（通过 onResume 刷新）
3. 总结 ListView 相比 RecyclerView 的优势：实现简单，适合入门学习

第五、六课时：功能优化与教学总结、上机练习

一、功能优化（25 分钟）

1. 添加空列表提示

修改 MainActivity 的 initData 方法：

```
java
运行
private void initData() {
    noteDao = new NoteDao(this);
    noteList = noteDao.queryAllNotes();
    adapter = new NoteAdapter(this, noteList);
    lvNotes.setAdapter(adapter);

    // 添加空列表提示
    TextView emptyView = new TextView(this);
    emptyView.setText("暂无笔记，点击添加按钮创建");
    emptyView.setGravity(View.TEXT_ALIGNMENT_CENTER);
    emptyView.setTextSize(16);

    emptyView.setTextColor(getResources().getColor(R.color.gray_400)
);
    ((ViewGroup)lvNotes.getParent()).addView(emptyView);
    lvNotes.setEmptyView(emptyView);}

```

2. 优化列表项点击效果

创建 res/drawable/item_selector.xml：

```
xml
<?xml version="1.0" encoding="utf-8"?><selector
xmlns:android="http://schemas.android.com/apk/res/android">

```

	<pre> <item android:drawable="@color/gray_200" android:state_pressed="true" /> <item android:drawable="@android:color/transparent" /></selector> </pre> <p>在 item_note.xml 的根布局添加：</p> <pre> xml android:background="@drawable/item_selector" </pre> <p>上机练习：使用程序模板完善代码，上机练习</p>
教学后记	<p>学生在实操中遇到问题时，多依赖教师逐一指导，效率较低（如多名学生询问“为何列表不显示”，原因均为未设置 ListView 的 Adapter）。后续可提前梳理“常见问题清单”（含问题现象、原因、解决方案），在上机前发放给学生，同时鼓励学生组建“互助小组”（基础好的学生帮助同伴），教师重点辅导小组内无法解决的难点，提升反馈效率。</p>

课题名称	第 7 章 内容提供者与内容观察者	计划课时	2 课时
教学引入	<p>在第 5 章数据存储中学习了 Android 数据持久化技术，包括文件存储、SharedPreferences 存储以及数据库存储，这些持久化技术所保存的数据都只能当前应用程序中访问。但在 Android 开发中，有时也会访问其他应用程序的数据。例如，使用支付宝转账时需要填写收款人的电话号码，此时就需要获取系统联系人的信息。为了实现这种跨程序共享数据的功能，Android 系统提供了一个组件 ContentProvider(内容提供者)。为了观察程序中数据的变化，Android 系统提供了一个内容观察者，本章将针对内容提供者与内容观察者进行详细地讲解。</p>		
教学目标	<ul style="list-style-type: none"> ● 使学生熟悉内容提供者的概述，能够阐述内容提供者的工作原理 ● 使学生掌握内容提供者的创建方式，能够独立创建内容提供者 ● 使学生掌握内容提供者的使用方式，能够使用内容提供者访问其他应用程序的数据 ● 使学生掌握内容观察者的使用方式，能够使用内容观察者观察其他程序的数据变化 		
教学重点	<ul style="list-style-type: none"> ● 创建内容提供者 ● 查询其他程序的数据 ● 什么是内容观察者 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——读取手机通讯录 ● 实战演练——监测数据变化 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	<p>讲解“实战演练——读取手机通讯录”时，重点强调权限申请的必要性：Android 系统将“读取通讯录”列为危险权限，需动态申请(Android 6.0+)，如同“进入他人房间需提前敲门”。通过对比“未申请权限直接崩溃”与“动态申请权限后正常读取”的效果，引导学生理解：用户的通讯录包含私人联系信息，属于核心隐私数据，开发者必须遵守系统权限机制，不可通过技术手段绕过权限校验，培养“尊重用户隐私、不越权访问数据”的职业伦理。</p>		
教学过程	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">(内容提供者概述、创建内容提供者、查询其他程序的数据、实战演练——读取手机通讯录)</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过需求引入的方式导入新课</p> <p>在第 5 章数据存储中学习了 Android 数据持久化技术，包括文件存储、SharedPreferences 存储以及数据库存储，这些持久化技术所保存的数据都只能当前应用程序中访问。但在 Android 开发中，有时也会访问其他应用程序的数据。例如，使用支付宝转账时需要填写收款人的电话号码，此时就需要获取系统联系人的信息。为了实现这种跨程序共享数据的功能，Android 系统提供了一个组件 ContentProvider(内容提供者)。为了观察程序中数据的变化，Android 系统提供了一个内容观察者，接下来将针对内容提供者与内容观察者进行详细</p>		

地讲解。

三、新课讲解

知识点 1-内容提供者概述

教师通过 PPT 讲解内容提供者概述。

- (1) ContentProvider 的工作原理。
- (2) 数据模型。
- (3) Uri。

知识点 2-创建内容提供者

教师通过 PPT 结合实际操作的方式讲解创建内容提供者。

- (1) 如何创建一个内容提供者。
 - 创建一个类继承抽象类 ContentProvider。
 - 重写该类中的 onCreate()、getType()、insert()、delete()、update()、query()方法。
- (2) 演示创建一个内容提供者 MyContentProvider。
 - 创建程序。
 - 创建 MyContentProvider。

知识点 3-查询其他程序的数据

教师通过 PPT 结合实际操作的方式讲解查询其他程序的数据。

- (1) 通过 parse()方法解析 Uri。
- (2) 通过 query()方法查询数据。
- (3) 通过 while()循环语句遍历查询到的数据。

知识点 4-实战演练——读取手机通讯录

教师通过 PPT 结合实际操作的方式讲解实战演练——读取手机通讯录。

- (1) 实现通讯录界面功能。
 - 创建程序。
 - 添加 recyclerview 库。
 - 添加界面控件。
 - 去掉默认标题栏。
 - 搭建通讯录列表条目界面布局。
- (2) 封装联系人信息实体类。
- (3) 编写通讯录列表的适配器。
- (4) 实现显示通讯录界面数据的功能。
 - 申请读取手机通讯录的权限。
 - 将数据显示到通讯录界面上。
- (5) 添加读取系统通讯录的权限。
- (6) 运行程序。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台 (<http://tch.ityx.com>) 布置本节课作业以及下节课的预习作业。

第二课时

(什么是内容观察者、实战演练——监测数据变化)

	<p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>七、通过需求引入的方式导入新课</p> <p>如果我们想要观察通过内容提供者操作的数据的变化，该如何进行观察呢？这时候就需要使用内容观察者，关于什么是内容观察者将在本节课给大家讲解，同时还会通过一个监测数据变化的案例讲解如何使用内容观察者。</p> <p>八、新课讲解</p> <p>知识点 1-什么是内容观察者</p> <p>教师通过 PPT 结合实际操作的方式讲解什么是内容观察者。</p> <p>(1) 内容观察者的作用。</p> <p>(2) 通过图例的方式讲解 ContentObserver 的工作原理。</p> <p>(3) 演示通过 ContentObserver 中的 onChange()方法观察特定的 Uri 代表的数据。</p> <ul style="list-style-type: none"> ● 创建内容观察者。 ● 注册内容观察者。 ● 取消注册内容观察者。 <p>知识点 2-实战演练——监测数据变化</p> <p>教师通过 PPT 结合实际操作的方式讲解实战演练——监测数据变化。</p> <p>(1) 通过操作数据库界面实现对数据库中的数据进行增、删、改、查操作。</p> <ul style="list-style-type: none"> ● 创建程序。 ● 导入界面图片。 ● 修改程序的主题样式。 ● 添加界面控件。 ● 创建数据库。 ● 创建内容提供者。 ● 实现操作数据库界面上的功能。 <p>(2) 创建监测数据库变化的程序。</p> <ul style="list-style-type: none"> ● 创建 MonitorData 程序。 ● 声明被监测的程序包名。 ● 运行程序。 <p>四、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p> <p>教师布置本节课作业以及下节课的预习作业。</p>
教学后记	<p>设计“隐私保护找茬”任务：提供一段“未校验权限就读取通讯录”的代码，让学生找出问题并修改（添加动态权限申请），在实践中理解“隐私保护的必要性”</p>

课题名称	第 8 章 广播机制	计划课时	2 课时
教学引入	在 Android 系统中，广播是一种运用在组件之间传递消息的机制，例如电池电量低时会发送一条提示广播。如果要接收并过滤广播中的消息，则需要使用 <code>BroadcastReceiver</code> （广播接收者），广播接收者是 Android 四大组件之一，通过广播接收者可以监听系统中的广播消息，实现在不同组件之间的通信，本章将针对广播机制进行详细讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生熟悉广播机制的概述，能够归纳广播机制的实现流程 ● 使学生熟悉广播接收者的概念，能够阐述什么是广播接收者 ● 使学生掌握广播接收者的创建方式，能够独立创建广播接收者 ● 使学生掌握自定义广播的方式，能够通过自定义广播实现饭堂小广播案例 ● 使学生掌握广播的类型，能够归纳有序广播与无序广播的工作流程 		
教学重点	<ul style="list-style-type: none"> ● 创建广播接收者 ● 自定义广播 ● 广播的类型 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——饭堂小广播 ● 实战演练——数鸭子 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	在讲解“有序广播与无序广播”时，类比“现实生活中的信息传递规则”——无序广播如同“校园广播通知放学”（所有班级同时收到，无先后顺序），有序广播如同“政府文件层层传达”（按级别高低依次接收，可拦截或修改内容）。通过“实战演练——数鸭子”案例，演示有序广播的优先级机制（优先级高的接收者先数鸭子，可拦截后续广播），引导学生理解：技术中的“顺序”与“优先级”对应现实中的“规则”与“职责”，开发者需严格遵循广播的传递规则（如有序广播的优先级配置），避免因“插队”或“违规拦截”导致信息传递混乱，培养“尊重规则、按秩序协作”的意识。		
教学过程	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">（广播机制的概述、什么是广播接收者、创建广播接收者）</p> <p>一、复习巩固 教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、创设情景，导入新课 在现实生活中，大多数人都会收听广播，例如出租车司机会收听实时路况的广播，来关注路面拥堵情况。同样在 Android 系统中内置了很多系统级别的广播，本节课我们将针对广播机制的概述、什么是广播接收者和创建广播接收者进行详细讲解。</p> <p>三、新课讲解</p> <p>知识点 1-广播机制的概述 教师通过 PPT 讲解广播机制的概述。</p> <p>（1）通过学校广播室发送广播，引出 Android 的广播机制。</p>		

(2) Android 中的广播机制的作用。

(3) 广播机制的实现流程。

(4) 广播的使用场景。

- 在同一个 APP 内部的同一组件内进行消息通信（单个或多个线程之间）。
- 在同一个 APP 内部的不同组件之间进行消息通信（单个进程）。
- 在同一个 APP 具有多个进程的不同组件之间进行消息通信。
- 在不同 APP 的组件之间进行消息通信。
- 在特定情况下，Android 系统与 APP 之间进行消息通信。

知识点 2-什么是广播接收者

教师通过 PPT 讲解什么是广播接收者。

(1) 广播接收者（BroadcastReceiver）组件的用途。

(2) 多个广播接收者接收广播的过程。

知识点 3-创建广播接收者

教师通过 PPT 结合实际操作的方式讲解创建广播接收者。

(1) 创建广播接收者的两种方式。

- 在应用程序中创建一个类继承 BroadcastReceiver 并重写 onReceive()方法来实现。
- 通过选中应用程序中的包，右击选择“New” → “Other” → “Broadcast Receiver”选项来创建。

(2) 注册广播接收者的两种方式。

- 动态注册。
- 静态注册。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第二课时

（自定义广播、实战演练——饭堂小广播、广播的类型、实战演练——数鸭子）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

八、通过需求引入的方式导入新课

当 Android 系统提供的广播不足以满足我们程序的需求时，我们可以自己定义一个广播实现程序中指定的功能。自定义完广播后，我们来想一下广播有什么类型？不同类型的广播能分别做什么事情？这些内容就是我们本节要讲解的内容，自定义广播和广播的类型，同时还通过自定义广播实现一个饭堂小广播的案例，通过有序广播实现一个数鸭子的案例。

九、新课讲解

知识点 1-自定义广播

教师通过 PPT 讲解自定义广播。

- (1) 什么是自定义广播。
- (2) 演示发送一个打招呼的自定义广播。
- (3) 通过图例演示自定义广播的发送与接收过程。

知识点 2-实战演练——饭堂小广播

教师通过 PPT 结合实际操作的方式讲解实战演练——饭堂小广播。

(1) 搭建饭堂小广播界面布局。

- 创建程序。
- 导入界面图片。
- 添加界面控件。
- 修改默认标题栏的名称。

(2) 实现饭堂小广播界面的功能。

- 初始化界面控件。
- 发送开饭消息的广播。
- 创建与注册广播接收者 MyBroadcastReceiver。
- 注销注册的广播接收者。

(3) 运行程序。

知识点 3-广播的类型

教师通过 PPT 讲解广播的类型。

- (1) 什么是无序广播。
- (2) 无序广播的工作流程。
- (3) 什么是有序广播。
- (4) 有序广播的工作流程。

知识点 4-实战演练——数鸭子

教师通过 PPT 结合实际操作的方式讲解实战演练——数鸭子。

(1) 搭建数鸭子界面布局。

- 创建程序。
- 导入界面图片。
- 创建 badge_bg.xml 文件。
- 定义序号控件的样式 badge_style。
- 定义图片控件的样式 duck_style。
- 添加界面控件。
- 修改默认标题栏的名称。

(2) 实现数鸭子界面的功能。

- 创建三个广播接收者。
- 动态注册创建的三个广播接收者。
- 实现发送有序广播的功能。
- 修改创建的三个广播接收者。

(3) 运行程序。

(4) 修改广播接收者的优先级。

(5) 拦截有序广播。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

教学后记	教材中对“系统广播的权限要求”讲解较简略，仅提及“部分系统广播需权限”，未详细列举常见系统广播的权限（如监听开机需 RECEIVE_BOOT_COMPLETED、监听短信需 RECEIVE_SMS）。导致学生在实战中频繁因“权限缺失”导致广播接收失败，需额外补充权限配置知识，占用课堂时间。后续教学需提前整理“常用系统广播 - 权限对应表”，并演示“权限申请与广播注册的联动流程”，降低学生理解门槛。
------	--

课题名称	第9章 服务	计划课时	2 课时
教学引入	<p>通常在程序中下载一些大文件时，程序突然退出，此时下载文件的任务会中断。为了避免出现下载任务中断的问题，我们可以使用 Android 系统提供的服务来下载大文件。服务是一个长期运行在后台的用户组件，没有用户界面。它除了可以在后台下载文件之外，还可以在后台执行很多任务，比如处理网络事务、播放音乐或者与一个内容提供者交互，本章将针对服务进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> ● 使学生熟悉服务的概述，能够归纳什么是服务 ● 使学生掌握服务的创建方式，能够独立创建一个服务 ● 使学生掌握服务的生命周期，能够使用服务生命周期中的方法 ● 使学生掌握服务的两种启动方式，能够实现服务的启动与关闭功能 ● 使学生掌握服务的通信，能够完成音乐播放器案例 		
教学重点	<ul style="list-style-type: none"> ● 服务的创建 ● 服务的生命周期 ● 调用 startService()方法启动服务 ● 调用 bindService()方法启动服务 ● 本地服务通信和远程服务通信 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——音乐播放器 		
教学方式	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>		
思政元素	<p>在讲解“服务的启动与停止”时，结合“资源节约”的价值观：服务作为后台长期运行的组件，若启动后不及时停止（如忘记调用 stopService()），会持续占用内存和电量，如同“离开房间未关灯”，造成资源浪费。通过演示“未停止服务的内存占用变化”（使用 Android Studio 的 Profiler 工具，展示服务持续运行时内存占用居高不下），引导学生理解：开发者有责任管理服务生命周期，避免因代码疏忽导致设备性能下降，培养“对用户设备负责、对资源消耗负责”的担当精神</p>		
教学过程	<p style="text-align: center;">第一课时 （服务概述、服务的创建、服务的生命周期）</p> <p>一、复习巩固 教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过直接引入的方式导入新课 前面我们已经学习过 Android 四大组件中的三个组件，分别是 Activity、BroadcastReceiver、ContentProvider，只剩下 Service（服务）还没有学习，本节课我们将针对服务概述、服务的创建和服务的生命周期进行详细讲解。</p> <p>三、新课讲解</p> <p>知识点 1-服务概述 教师通过 PPT 讲解服务概述。</p> <p>（1）什么是服务。</p> <p>（2）服务的应用场景。</p> <ul style="list-style-type: none"> ● 后台运行。 		

- 跨进程访问。

知识点 2-服务的创建

教师通过 PPT 结合实际操作的方式讲解服务的创建。

- (1) 选中程序包名，右击选择“New” → “Service” → “Service”选项，会弹出一个 Service 页面。
- (2) 在该页面输入服务的名称并单击“Finish”按钮即可创建服务。

知识点 3-服务的生命周期

教师通过 PPT 讲解服务的生命周期。

- (1) 服务的生命周期与启动服务的方式有关。
- (2) 通过图例展示使用不同方式启动服务的生命周期。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

第二课时

(调用 startService()方法启动服务、调用 bindService()方法启动服务)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

九、通过直接引入的方式导入新课

通过上一节课讲解的内容可知，启动服务有两种方式，分别是通过调用 startService()方法与 bindService()方法启动服务。本节课将针对这两种启动服务的方式进行详细讲解。

十、新课讲解

知识点 1-调用 startService()方法启动服务

教师通过 PPT 结合实际操作的方式讲解调用 startService()方法启动服务。

- (1) startService()方法的作用。
- (2) 通过一个开灯与关灯的案例演示如何通过调用 startService()方法与 stopService()方法来启动和关闭服务。
 - 搭建开灯与关灯界面布局。
 - 实现开灯与关灯界面的功能。
 - 运行程序。

知识点 2-调用 bindService()方法启动服务

教师通过 PPT 结合实际操作的方式讲解调用 bindService()方法启动服务。

- (1) bindService()方法的作用。
- (2) 通过一个绑定服务的案例来演示如何通过 bindService()方法与 unbindService()方法来绑定与解绑服务。
 - 搭建绑定服务界面布局。
 - 实现启动与关闭服务的功能。

第一步：创建服务类（定义后台任务逻辑）

新建 MusicService 类，继承 Service，重写服务生命周期方法，实现“启动时播放、销毁时停止”的逻辑：

```
java
运行
import android.app.Service;import android.content.Intent;import
android.os.IBinder;import android.widget.Toast;
// 音乐服务：模拟后台播放音乐 public class MusicService extends Service
{
    // 服务启动时调用：执行后台任务（如播放音乐）
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // 模拟播放音乐的逻辑（实际开发中可调用音频播放 API）
        Toast.makeText(this, "后台音乐开始播放~",
Toast.LENGTH_SHORT).show();
        // 打印日志，方便查看服务运行状态（Logcat 中筛选
“MusicService”）
        android.util.Log.d("MusicService", "服务启动，后台播放音乐");

        // 返回 START_STICKY：服务被系统意外杀死后，尝试重新创建（适
合音乐、下载等场景）
        return START_STICKY;
    }

    // 服务销毁时调用：释放资源（如停止音乐、关闭线程）
    @Override
    public void onDestroy() {
        super.onDestroy();
        // 模拟停止音乐的逻辑
        Toast.makeText(this, "后台音乐停止播放~",
Toast.LENGTH_SHORT).show();
        android.util.Log.d("MusicService", "服务销毁，停止播放音乐");
    }

    // 绑定服务时需实现（本实例为“启动服务”，暂返回 null）
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

第二步：注册服务（四大组件必做步骤）

在 AndroidManifest.xml 的<application>标签内注册服务（若不注册，服务无法启动）：

```
xml
<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/Theme.AppCompat.Light.DarkActionBar">

    <!-- 注册音乐服务：name 属性为服务类的完整路径（若在同一包下，可
    简写为“.MusicService”） -->
    <service android:name=".MusicService" />

    <!-- 主 Activity 注册（默认已有，无需修改） -->
    <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>
```

第三步：创建界面布局（触发服务启动 / 停止）

在 res/layout/activity_main.xml 中添加两个按钮，分别用于“启动服务”和“停止服务”：

```
xml
<?xml version="1.0" encoding="utf-8"?><LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">
```

```

<!-- 启动音乐服务按钮 -->
<Button
    android:id="@+id/btn_start_service"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="启动音乐服务"
    android:textSize="18sp"
    android:layout_marginBottom="20dp" />

<!-- 停止音乐服务按钮 -->
<Button
    android:id="@+id/btn_stop_service"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="停止音乐服务"
    android:textSize="18sp" />
</LinearLayout>

```

第四步：在 Activity 中控制服务（绑定按钮点击事件）

在 MainActivity 中通过 Intent 触发服务的启动与停止，实现按钮与服务交互：

```

java
运行
import android.content.Intent;import android.os.Bundle;import
android.view.View;import android.widget.Button;import
androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    // 声明服务相关的 Intent（用于启动/停止服务）
    private Intent musicServiceIntent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // 初始化 Intent：指定要操作的服务（MusicService）
        musicServiceIntent = new Intent(this, MusicService.class);

        // 1. 绑定“启动服务”按钮点击事件
        Button startBtn = findViewById(R.id.btn_start_service);
        startBtn.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            // 启动服务：调用系统 API，传入服务 Intent
            startService(musicServiceIntent);
        }
    });

    // 2. 绑定“停止服务”按钮点击事件
    Button stopBtn = findViewById(R.id.btn_stop_service);
    stopBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // 停止服务：调用系统 API，传入服务 Intent
            stopService(musicServiceIntent);
        }
    });
}
}

```

三、实例运行效果

1. **启动服务：**点击“启动音乐服务”按钮，弹出 Toast “后台音乐开始播放~”，Logcat 中显示“服务启动，后台播放音乐”，此时即使按 Home 键返回桌面（Activity 隐藏），服务仍在后台运行；
2. **停止服务：**点击“停止音乐服务”按钮，弹出 Toast “后台音乐停止播放~”，Logcat 中显示“服务销毁，停止播放音乐”，服务完全终止；
3. **意外场景：**若未点击“停止服务”直接关闭 app（从最近任务栏清除），服务会被系统保留一段时间（因返回 START_STICKY），若系统资源充足，会尝试重新启动服务。

四、关键说明

1. **服务生命周期：**启动服务的核心流程为 “startService() → onStartCommand()（执行任务）→ stopService()/stopSelf() → onDestroy()（释放资源）”；
2. **注册必做：**所有 Service 都必须在 AndroidManifest.xml 中注册，否则调用 startService() 会抛出 IllegalStateException；
3. **后台特性：**服务默认在主线程运行，若需执行耗时任务（如真实音乐播放、文件下载），需在 onStartCommand() 中开启子线程（如 new Thread {...}.start()），避免阻塞界面；
4. **资源释放：**启动服务后，务必通过 stopService() 或服务内部 stopSelf() 停止服务，否则服务会持续占用内存和电量，导致设备性能下降。

教学后记	仅通过文字描述和流程图，部分学生难以理解“startService()与bindService()的本质区别”，尤其是“绑定服务依赖组件生命周期”的特性。例如，学生常误以为“绑定启动的服务也能通过stopService()停止”，导致服务状态异常。后续教学可通过“动态演示”（如用两个按钮分别启动两种服务，销毁Activity观察服务是否存活），让学生直观感受差异，而非单纯记忆流程。

课题名称	第 10 章 Android 事件处理	计划课时	4 课时
教学引入	前面章节中介绍了 Android 中各种常用控件，它们组成了应用程序界面。通常情况下，一个 Android 应用程序中，用户与应用程序之间的交互是通过事件处理来完成的，因此我们通过处理这些事件就可以对界面上的控件进行交互，本章将针对 Android 事件处理进行详细讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生掌握基于回调机制的事件处理方法，能够处理按下、弹起、触摸等事件 ● 使学生掌握基于监听接口机制的事件处理方法，能够处理点击、长按等事件 ● 使学生掌握 Handler 消息机制原理，能够使用 Handler 进行线程间通信 		
教学重点	<ul style="list-style-type: none"> ● onKeyDown()方法 ● onKeyUp()方法 ● onTouchEvent()方法 ● onFocusChanged()方法 ● OnClickListener 接口 ● OnLongClickListener 接口 ● OnFocusChangeListener 接口 ● OnKeyListener 接口 ● OnTouchListener 接口 ● Handler 消息机制 		
教学难点	无		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	“onTouchEvent() 处理触摸事件”环节，讲解 MotionEvent 的 ACTION_DOWN/ACTION_MOVE/ACTION_UP 事件序列时，类比“生活中完成一件事的步骤”——按下（开始）→滑动（过程）→弹起（结束），缺一不可。通过案例演示“缺失 ACTION_UP 导致的 bug”（如触摸后控件状态无法恢复），让学生体会：事件处理需完整覆盖用户操作流程，避免因“遗漏步骤”影响用户体验，培养“考虑周全、不遗漏细节”的严谨态度。		

第一课时

(事件处理概述、基于回调机制的事件处理简介、onKeyDown()方法)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

在 Android 程序中，如果想要处理事件逻辑比较简单的 View 时，可以使用基于回调的事件处理机制进行处理，该事件主要在 View 特定的方法中进行处理。本节课将针对事件处理概述、基于回调机制的事件处理简介、onKeyDown()方法进行详细讲解。

三、新课讲解

知识点 1-事件处理概述

教师通过 PPT 讲解事件处理概述。

(1) Android 中的事件主要有两种。

- 键盘事件。
- 触摸事件。

(2) Android 平台提供了两种事件处理机制。

- 基于回调机制的事件处理。
- 基于监听接口的事件处理。

知识点 2-基于回调机制的事件处理简介

教师通过 PPT 讲解基于回调机制的事件处理简介。

- (1) 什么是基于回调机制的事件处理。
- (2) Android 系统的提供的处理事件的回调方法。

当用户与UI控件发生某个事件（如按下事件、滑动事件、双击事件）时，**程序会调用控件自己特定的方法处理该事件，这个处理过程就是基于回调机制的事件处理。**基于回调机制的事件处理包含处理物理按键事件和处理触摸事件。处理物理按键事件和处理触摸事件的回调方法的相关介绍如下表所示。

方法	说明
boolean onKeyDown(int keyCode,KeyEvent event)	当用户在该控件上按下某个按键时触发的方法
boolean onKeyUp(int keyCode,KeyEvent event)	当用户在该控件上松开某个按键时触发的方法
boolean onTouchEvent(MotionEvent event)	当用户在该控件上触发触摸事件时触发的方法
boolean onFocusChanged(MotionEvent event)	当用户在控件上的焦点发生改变时触发的方法

知识点 3-onKeyDown()方法

教师通过 PPT 结合实际操作的方式讲解 onKeyDown()方法。

- (1) onKeyDown()方法的作用。
- (2) 通过 onKeyDown()方法来屏蔽后退键以及在该方法中通过 Toast 来提示用户增加音量的信息。
 - 创建程序。
 - 编写界面交互代码。
 - 运行程序。

当 Android 设备上的物理按键被按下时，程序会回调 `onKeyDown()`方法，该方法是接口 `KeyEvent.Callback` 中的抽象方法。Android 程序中所有的 `View` 都实现了 `KeyEvent.Callback` 接口并重写了 `onKeyDown()`方法，该方法主要用于捕捉手机键盘被按下的事件，`onKeyDown()`方法的定义方式如下：

```
public boolean onKeyDown(int keyCode, KeyEvent event)
```

物理按键	KeyCode	说明
电源键	KEYCODE_POWER	启动或唤醒设备，将界面切换到锁定的屏幕
后退键	KEYCODE_BACK	返回到前一个界面
Home键	KEYCODE_HOME	返回到Home界面
增大音量键	KEYCODE_VOLUME_UP	增大当前上下文音量，如音乐播放器、手机铃声、通话音量等
减小音量键	KEYCODE_VOLUME_DOWN	减小当前上下文音量，如音乐播放器、手机铃声、通话音量等

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过布置本节课作业以及下节课的预习作业。

第二课时

(`onKeyUp()`方法、`onTouchEvent()`方法、`onFocusChanged()`方法)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

十、通过直接引入的方式导入新课

上一节课讲解了基于回调机制的事件处理方法 `onKeyDown()`，除了此方法之外，基于回调机制的事件处理方法还有 `onKeyUp()`、`onTouchEvent()` 和 `onFocusChanged()`，本节课将对这 3 个方法进行详细讲解。

十一、新课讲解

知识点 1-`onKeyUp()`方法

教师通过 PPT 讲解 `onKeyUp()`方法。

- (1) `onKeyUp()`方法的作用。
- (2) `onKeyUp()`方法的定义方式。

当 Android 设备上的物理按键弹起时，程序会回调 `onKeyUp()`方法，该方法同样是接口 `KeyEvent.Callback` 中的一个抽象方法，并且 Android 中所有 `View` 都实现了 `KeyEvent.Callback` 接口并重写了 `onKeyUp()`方法，该方法主要用于捕捉键盘按键弹起的事件，`onKeyUp()`方法的定义方式如下：

```
public boolean onKeyUp(int keyCode, KeyEvent event)
```

知识点 2-`onTouchEvent()`方法

教师通过 PPT 结合实际操作的方式讲解 `onTouchEvent()`方法。

- (1) onTouchEvent()方法的作用。
- (2) onTouchEvent()方法的定义方式。

onTouchEvent()方法是在 View 类中进行定义的，并且所有 View 的子类中全部重写了 onTouchEvent()方法，Android 程序可以通过该方法处理屏幕的触摸事件，onTouchEvent()方法的定义方式如下：

```
public boolean onTouchEvent(MotionEvent event)
```

- (3) onTouchEvent()方法处理的三种事件。

- 屏幕被按下。
- 屏幕弹起。
- 在屏幕中滑动。

屏幕被按下	当屏幕被按下时，会自动调用onTouchEvent()方法来处理事件，当该方法中的event.getAction()方法的值为MotionEvent.ACTION_DOWN时，表示屏幕被按下的事件。
--------------	---

屏幕弹起	表示手指或者触控笔离开屏幕时触发的事件，该事件需要onTouchEvent()方法来捕捉，当该方法中的event.getAction()方法的值为MotionEvent.ACTION_UP时，表示屏幕弹起的事件。
-------------	--

在屏幕中滑动	当手指或者触控笔在屏幕上滑动时触发的事件，该事件需要调用onTouchEvent()方法来处理。当event.getAction()方法的值为MotionEvent.ACTION_MOVE时，表示在屏幕中进行滑动的事件。
---------------	--

- (4) 通过案例讲解 onTouchEvent()方法的使用。在界面上显示了一个“Hello”按钮，通过该按钮被按下、弹起以及在按钮上进行移动的操作演示 onTouchEvent()方法的使用。

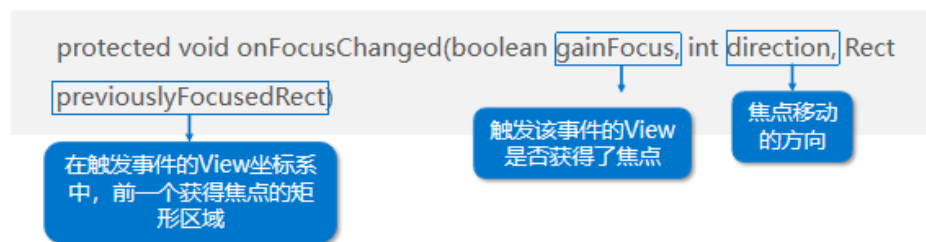
- 创建程序。
- 编写逻辑代码。
- 运行程序。

知识点 3-onFocusChanged()方法

教师通过 PPT 讲解 onFocusChanged()方法。

- (1) onFocusChanged()方法的作用。
- (2) onFocusChanged()方法的定义方式。

onFocusChanged()方法是焦点改变的回调方法，只能在 View 中重写。当某个控件重写了该方法后，焦点发生变化时，会自动调用该方法来处理焦点改变的事件，其定义方式如下：



四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过学习通布置本节课作业。

第三课时

（基于监听接口机制的事件处理简介、OnClickListener 接口、OnLongClickListener 接口）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

上一节课介绍了如何通过回调机制进行事件处理，一般来说，基于回调的事件处理可用于处理一些具有通用性的事件，基于回调的事件处理代码会显得比较简洁。但对于某些特定的事件，无法使用基于回调的事件处理时，只能采用基于监听的事件处理，本节课将介绍基于监听接口机制的事件处理简介、OnClickListener 接口和 OnLongClickListener 接口的内容。

三、新课讲解

知识点 1-基于监听接口机制的事件处理简介

教师通过 PPT 讲解基于监听接口机制的事件处理简介。

(1) 在事件监听的处理模型中主要涉及三个对象。

- EventSource。
- Event。
- EventListener。



(2) 通过图例展示基于监听接口机制的事件处理流程。

(3) 开发一个基于监听事件的处理模型的大致步骤。

- 获取普通界面组件（事件源），即被监听的对象。
- 实现事件监听器类，该监听器类是一个特殊的 Java 类，即该类必须实现一个 XxxListener 接口。
- 事件源调用 setXxxListener()方法，将事件监听器对象注册给普通组件（事件源）。

(4) View 类中的内部接口。

知识点 2-OnClickListener 接口

教师通过 PPT 结合实际操作的方式讲解 OnClickListener 接口。

- (1) OnClickListener 接口的作用。
- (2) OnClickListener 接口接口中需要实现的方法的定义方式。
- (3) 通过案例讲解 OnClickListener 接口与该接口中需要实现的 onClick() 方法。

// 通过匿名内部类来实现 OnClickListener 接口

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Toast.makeText(MainActivity.this, "您点击了按钮",  
Toast.LENGTH_SHORT).show();  
    }  
});
```

- (4) 实现界面上按钮的点击事件的具体步骤。

- 创建项目。
- 添加界面控件。
- 编写界面交互代码。
- 运行程序。

知识点 3-OnLongClickListener 接口

教师通过 PPT 结合实际操作的方式讲解 OnLongClickListener 接口。

- (1) OnLongClickListener 接口的作用。
- (2) OnLongClickListener 接口中需要实现的方法的定义方式。
- (3) 在 OnClickListener 接口知识点中的案例基础之上，对 ClickListener 程序中的“按钮”进行长按事件的处理。将程序中的 MainActivity 实现 OnLongClickListener 接口，并重写该接口中的 onLongClick()方法。

```

public class MainActivity extends AppCompatActivity implements View.OnLongClickListener {
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        .....
        button.setOnLongClickListener(this);    // 注册监听
    }
    @Override
    public boolean onLongClick(View v) {
        if (v==button) {    // 判断长时间按下的控件是否是button
            Toast.makeText(MainActivity.this, "您长时间按下了按钮",Toast.LENGTH_SHORT).show();
        }
        .....
    }
}

```

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师布置本节课作业。

第四课时

（OnFocusChangeListener 接口、OnKeyListener 接口、OnTouchListener 接口、Handler 消息机制）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

基于监听接口机制的事件处理，除了上一节课讲解的 OnClickListener 接口与 OnLongClickListener 接口之外，还有 OnFocusChangeListener 接口、OnKeyListener 接口和 OnTouchListener 接口，本节课将针对这 3 个接口与 Handler 消息机制进行详细讲解。

三、新课讲解

知识点 1-OnFocusChangeListener 接口

教师通过 PPT 结合实际操作的方式讲解 OnFocusChangeListener 接口。

(1) OnFocusChangeListener 接口的作用。

(2) OnFocusChangeListener 接口中的回调方法 onFocusChange() 的定义方式。

OnFocusChangeListener 接口用于处理控件焦点发生改变的事件，如果对某个控件注册了该监听器，则当该控件失去焦点或者获得焦点时都会触发 OnFocusChangeListener 接口中的回调方法 onFocusChange()，该回调方法的定义方式如下：

```
public void onFocusChange(View v, boolean hasFocus)
```

(3) 通过一个选择动物的案例来介绍 OnFocusChangeListener 接口的使用。

- 创建程序。
- 导入界面图片。
- 添加界面控件。
- 编写界面交互代码。
- 运行程序。



知识点 2-OnKeyListener 接口

教师通过 PPT 结合实际操作的方式讲解 OnKeyListener 接口。

- (1) OnKeyListener 接口的作用。
- (2) OnKeyListener 接口的定义方式。

OnKeyListener 接口是对设备的键盘进行监听的接口，如果对界面上某个 View 注册了该监听器，则当该 View 获得焦点并触发了键盘事件时，程序便会回调该接口中的 onKey()方法。该方法的定义方式如下：

```
public boolean onKey(View v, int keyCode, KeyEvent keyEvent)
```

- (3) 通过一个案例来演示如何使用 OnKeyListener 接口与该接口中的 onKey()方法。
 - 创建程序。
 - 导入图片。
 - 编写界面交互代码。
 - 运行程序。

知识点 3-OnTouchListener 接口

教师通过 PPT 讲解 OnTouchListener 接口。

- (1) OnTouchListener 接口的作用。
- (2) OnTouchListener 接口定义方式。

	<p>OnTouchListener 接口是处理手机屏幕的触摸事件需要监听的接口，当在 View 的范围内进行按下、抬起或滑动等动作时都会触发该事件并实现该接口中的 onTouch()方法，该方法的定义方式如下：</p> <pre>public boolean onTouch (View V, MotionEvent motionEvent)</pre> <p>知识点 4-Handler 消息机制 教师通过 PPT 讲解 Handler 消息机制。</p> <p>(1) 什么是 Handler 消息机制。</p> <p>(2) Handler 消息机制主要包括的四个关键对象。</p> <ul style="list-style-type: none"> ● Message。 ● Handler。 ● MessageQueue。 ● Looper。 <p>Handler 是一种异步回调机制，主要负责与子线程进行通信。 Handler 机制主要包括四个关键对象： Message: 是线程之间传递的消息，它可以在内部携带少量的信息，用于在不同线程之间交换数据。 Handler: 是处理者的意思，它主要用于发送消息和处理消息。 MessageQueue: 是消息队列的意思，它主要用来存放通过 Handler 发送的消息。 Looper: 是每个线程中的 MessageQueue 的管家。调用 Looper 的 loop()方法后，就会进入到一个无限循环中。</p> <p>(3) 通过图例梳理整个 Handler 消息处理流程。</p> <p>四、归纳总结 教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p>
教学后记	<p>针对事件传递：制作 “触摸事件传递流程图解”，标注 “父控件拦截” “子控件消费” 等关键节点；设计 “拦截事件” 小实验（如在 LinearLayout 中重写 onInterceptTouchEvent，观察子按钮是否接收事件），让学生亲手验证传递规则。</p>

课题名称	第 11 章 网络编程	计划课时	8 课时
教学引入	<p>在移动互联网时代，手机联网实现信息互通是最基本的功能体验。例如，在上下班的途中或旅行时，只要有时间人们就会拿出手机上网，通过手机接收新资讯、搜索网络资源。Android 作为智能手机市场中主流的操作系统，它的强大离不开其对网络功能的支持。Android 系统提供了多种实现网络通信的方式。接下来，我们从最基础的 HTTP 协议开始，到 Android 中原生的 HttpURLConnection、WebView 控件的使用以及网络数据的解析进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> ● 使学生了解 HTTP 的简介，能够说出什么是 HTTP ● 使学生掌握 HttpURLConnection 访问网络的方式，能够使用 GET 与 POST 方式提交请求数据 ● 使学生掌握 WebView 控件的使用，能够使用 WebView 控件加载不同的网页 ● 使学生掌握 JSON 数据的解析方式，能够通过不同的方式解析 JSON 数据 		
教学重点	<ul style="list-style-type: none"> ● 使用 HttpURLConnection 访问网络 ● 使用 WebView 浏览网页 ● 使用 WebView 执行 HTML 代码 ● 设置 WebView 支持 JavaScript ● JSON 解析 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——商品砍价界面 		
教学方式	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>		
思政元素	<p>在讲解 “HttpURLConnection 的超时设置” 时，类比 “生活中的等待规则” ——setConnectTimeout(5000)（5 秒连接超时）如同 “打电话时 5 秒内无人接听则挂断”，避免无限等待浪费资源。通过演示 “无超时设置导致的程序卡死”（如网络断开时 getInputStream() 无限阻塞），让学生理解：网络请求必须设置超时，且需在 onDestroy() 中取消请求（如关闭 InputStream、断开连接），避免内存泄漏或界面无响应，培养 “主动处理异常、不忽视边界情况” 的责任意识。</p> <p>通过对比 “字段名不匹配导致的解析空值” 与 “规范映射后的正确解析”，让学生体会：跨端数据交互依赖 “格式约定”，开发者需秉持 “严谨负责” 的态度，避免因字段名疏忽导致数据解析失败，培养 “对协作方负责、对数据准确性负责” 的职业习惯。</p>		
教学过程	<p style="text-align: center;">第一、二课时</p> <p style="text-align: center;">（HTTP 简介、使用 HttpURLConnection 访问网络）</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过直接引入的方式导入新课</p> <p>Android 对 HTTP 通信提供了很好的支持，通过标准的 Java 类 HttpURLConnection 便可实现基于 URL 的请求及响应功能。HttpURLConnection 继承自 URLConnection 类，它可以发送和接收任何类型和长度的数据，也可以</p>		

设置请求方式、超时时间。本节课将针对 HTTP 协议与 URLConnection 访问网络进行详细讲解。

三、新课讲解

知识点 1-HTTP 简介

教师通过 PPT 讲解 HTTP 简介。

(1) 什么是 HTTP。

- HTTP 请求。
- HTTP 响应。

(2) 通过手机端访问服务器端的图例展示 HTTP 请求与响应的通信过程。

Android 对 HTTP 通信提供了很好的支持，通过标准的 Java 类 `HttpURLConnection` 便可实现基于 URL 的请求及响应功能。`HttpURLConnection` 继承自 `URLConnection` 类，它可以发送和接收任何类型和长度的数据，也可以设置请求方式、超时时间。本节将针对 HTTP 协议与 `URLConnection` 访问网络进行详细讲解。

HTTP (Hypertext Transfer Protocol) 即超文本传输协议，它规定了浏览器和服务器之间相互通信的规则。

HTTP 协议是一种请求/响应式的协议。

当客户端与服务器端建立连接后，向服务器端发送的请求，被称作 HTTP 请求。当服务器端接收到请求后会做出响应，称为 HTTP 响应。

知识点 2-使用 `HttpURLConnection` 访问网络

教师通过 PPT 的方式讲解使用 `HttpURLConnection` 访问网络。

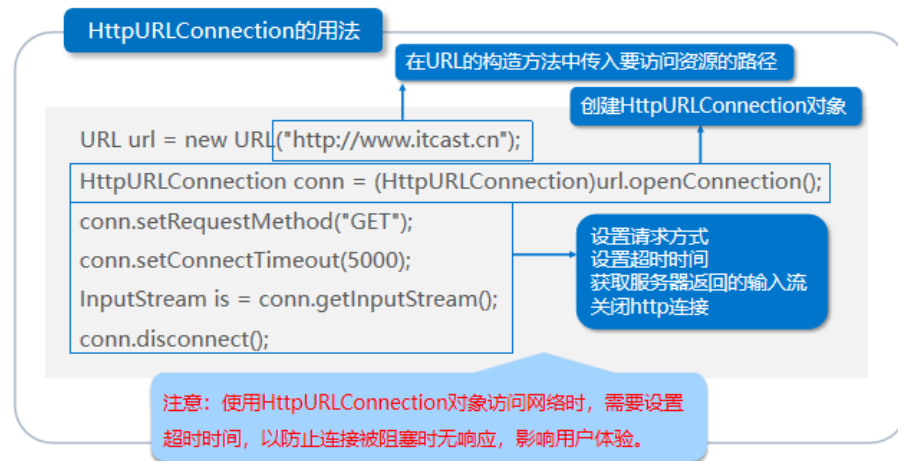
(1) `HttpURLConnection` 的作用。

(2) 手机端与服务器端建立连接并获取服务器返回数据的示例代码。

- GET 方式提交数据。
- POST 方式提交数据。



使用手机客户端访问百度网页时，会发送一个 HTTP 请求，当服务器端接收到这个请求后，会做出响应并将百度页面（数据）返回给客户端浏览器，这个请求和响应的过程实际上就是 HTTP 通信的过程。

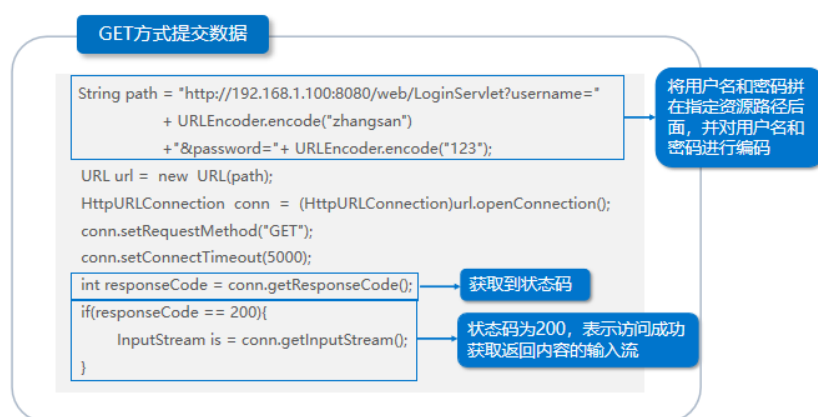


1.GET 方式提交数据

GET 方式是以实体的方式得到由请求 URL 所指向的资源信息，它向服务器提交的参数跟在请求 URL 后面。使用 GET 方式访问网络 URL 的内容一般要小于 1KB。

2.POST 方式提交数据

使用 POST 请求方式提交数据时，提交的数据是以键值对的形式封装在请求实体中，用户通过浏览器无法看到发送的请求数据，因此 POST 方式要比 GET 方式相对安全。



POST方式提交数据

```
URL url = new URL("http://192.168.1.100:8080/web/LoginServlet");
URLConnection conn = (URLConnection) url.openConnection();
conn.setConnectTimeout(5000);
conn.setRequestMethod("POST");
String data = "username=" + URLEncoder.encode("zhagsan")
    + "&password=" + URLEncoder.encode("123");
conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
conn.setRequestProperty("Content-Length", data.length() + "");
conn.setDoOutput(true);
OutputStream os = conn.getOutputStream();
os.write(data.getBytes());
int code = conn.getResponseCode();
if (code == 200) {
    InputStream is = conn.getInputStream();
}
```

准备数据并给参数进行编码

设置请求头数据提交方式以及提交数据的长度，这里是以form表单的方式提交

以流的形式将数据写到服务器上

在实际开发中，手机端与服务器端进行交互的过程中避免不了要提交中文到服务器，这时就会出现中文乱码的情况。无论是 GET 方式还是 POST 方式提交参数时都要给参数进行编码，编码方式必须与服务器解码方式一致。同样，在获取服务器返回的中文字符时，也需要用指定格式进行解码。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师布置本节课作业以及下节课的预习作业。

第三、四课时

（使用 WebView 浏览网页、使用 WebView 执行 HTML 代码、设置 WebView 支持 JavaScript）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

十一、通过需求引入的方式导入新课

Android 系统默认提供了内置浏览器，该浏览器使用了开源的 WebKit 引擎，WebKit 不仅能搜索网址、查看电子邮件，还可以播放视频。在 Android 程序中，如果想要使用该内置浏览器，则需要通过 WebView 控件来实现，该控件不仅可以指定 URL，还可以加载并执行 HTML 代码，同时还支持 JavaScript。本节课将对使用 WebView 浏览网页、使用 WebView 执行 HTML 代码和设置 WebView 支持 JavaScript 进行详细讲解。

十二、新课讲解

知识点 1-使用 WebView 浏览网页

教师通过 PPT 结合实际操作的方式讲解使用 WebView 浏览网页。

(1) WebView 控件的作用。

(2) 在 XML 布局文件中添加 1 个 WebView 控件的具体代码。

(3) WebView 控件的常用方法。

在 Android 程序中，WebView 控件可以在 XML 布局文件中使用<WebView>标签来添加，也可以在 Java 文件中通过 new 关键字来创建。

采用在 XML 布局文件中添加<WebView>标签的形式，具体代码如下。

<WebView

```
android:id="@+id/webView"  
android:layout_width="match_parent"  
android:layout_height="match_parent" />
```

WebView控件的常用方法如下表所示。

方法名称	功能描述
loadUrl(String url)	用于加载指定URL对应的网页
loadData(String data, String mimeType, String encoding)	用于将指定的字符串数据加载到浏览器中
loadDataWithBaseURL(String baseUrl, String data, String mimeType, String encoding, String historyUrl)	基于URL加载指定的数据
capturePicture()	用于创建当前屏幕的快照
goBack()	用于执行后退操作，相当于浏览器上后退按钮的功能
goForward()	用于执行前进操作，相当于浏览器上前进按钮的功能
stopLoading()	用于停止加载当前页面
reload()	用于刷新当前页面

(4) 通过案例演示如何使用 WebView 控件加载网页。

- 创建程序。
- 添加界面控件。
- 编写界面交互代码。

知识点 2-使用 WebView 执行 HTML 代码

教师通过 PPT 结合实际操作的方式讲解使用 WebView 执行 HTML 代码。

- (1) loadDataWithBaseURL()方法的作用。
- (2) loadDataWithBaseURL()方法的定义方式。
- (3) 通过案例演示如何使用 WebView 控件加载 HTML 代码。
 - 创建程序。
 - 实现 WebView 控件加载 HTML 代码的功能。

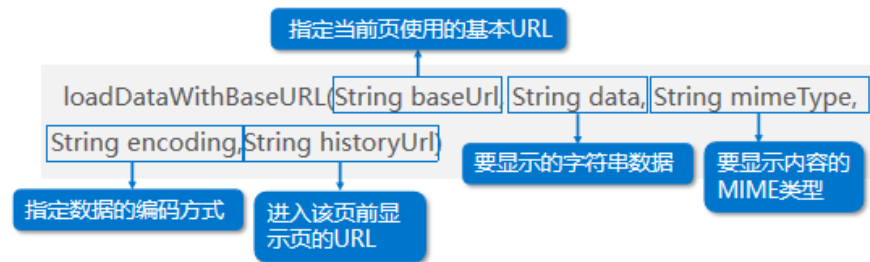
在 MainActivity 中实现 WebView 控件浏览网页的功能，通过 WebView 控件的 loadUrl()方法来加载指定的网页，主要代码如下。

```
WebView webView=findViewById(R.id.webView);  
webView.loadUrl("https://www.itheima.com/");
```

需要在清单文件（AndroidManifest.xml）的<manifest>标签中添加允许访问网络资源的权限。

```
<uses-permission android:name="android.permission.INTERNET"/>
```

WebView 类提供了 loadData()和 loadDataWithBaseUrl()方法加载 HTML 代码。当使用 loadData()方法来加载带中文的 HTML 内容时会产生乱码，但是使用 loadDataWithBaseUrl()方法就不会出现这种情况。loadDataWithBaseUrl()方法的定义方式如下：



知识点 3-设置 WebView 支持 JavaScript

教师通过 PPT 结合实际操作的方式讲解设置 WebView 支持 JavaScript。

- (1) setJavaScriptEnabled()方法的作用。
- (2) 设置 WebView 控件支持 JavaScript 代码。
 - 获取 WebView 控件的 WebSettings 对象，然后调用该对象的 setJavaScriptEnabled()方法让 WebView 控件支持 JavaScript 代码。
 - 使用 WebView 控件的 setWebChromeClient()方法实现加载网页中提示框的效果。
- (3) 通过案例演示如何使用 WebView 控件支持一个带有 JavaScript 代码的网页。
 - 创建程序。
 - 导入带有 JavaScript 代码的文件。
 - 添加界面控件。
 - 创建背景选择器。
 - 实现 WebView 控件加载 JavaScript 代码的功能。
 - 运行程序。

为了解决 WebView 控件在默认情况下不支持 JavaScript 代码的问题，我们需要通过 setJavaScriptEnabled()方法来设置 WebView 控件，使其可以支持 JavaScript 代码。

```
WebSettings settings= webview.getSettings(); // 获取WebSettings对象
settings.setJavaScriptEnabled(true);
webview.setWebChromeClient(new WebChromeClient());
```

设置JavaScript可用

使WebView控件显示带有JavaScript代码的提示框

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

第五、六课时

(JSON 数据、JSON 解析、实战演练——商品砍价界面)

一、复习巩固

知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

Android 应用界面上的数据信息大部分都是通过网络请求从服务器上获取到的，获取到的数据类型常见的就是 JSON 类型，JSON 是一种新的数据格式，这种格式的数据不可以直接显示到程序的界面上，需要将该数据解析为一个集合或对象的形式才可以显示到界面上，本节课将对 JSON 数据、JSON 解析和实战演练——商品砍价界面进行详细讲解。

三、新课讲解

知识点 1-JSON 数据

教师通过 PPT 讲解 JSON 数据。

- (1) 什么是 JSON。
- (2) JSON 数据的两种表示结构。
 - 对象结构。
 - 数组结构。

(1) JSON 即 JavaScript Object Notation (JavaScript 对象表示法)，是一种轻量级的数据交换格式。

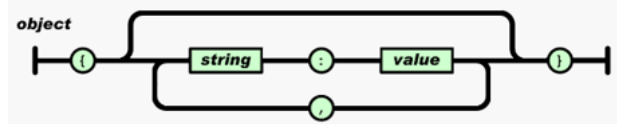
(2) JSON 可以传输 String、Number、Boolean 类型的数据，也可以传输数组或者 Object 对象。

(3) JSON 文件的扩展名为.json。

(4) JSON 分为 JSON 对象和 JSON 数组两种数据结构。

对象结构的JSON数据

以 “ { ” 开始, 以 “ } ” 结束。中间部分由以 “ , ” 分隔的键值对构成, 注意键和值之间以 “ : ” 分隔。



对象结构的JSON数据的基本语法格式如下:

```
{
  key1:value1,
  key2:value2,
  .....
}
```

键key必须为String类型, 值value可以是String、Number、Object、Array等数据类型。

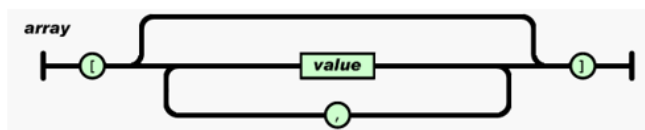
对象结构的JSON数据

例如, 一个address对象包含城市、街道、邮编等信息, JSON的表示形式如下:

```
{"city":"Beijing", "street":"Xisanqi", "postcode":100096}
```

数组结构的JSON数据

以 “[” 开始, 以 “] ” 结束。中间部分由0个或多个以 “ , ” 分隔的对象 (value) 的列表组成。其存储形式如下图所示。



数组结构的JSON数据的语法格式如下:

```
[
  value1,
  value2,
  .....
]
```

值value可以是String、Number、Boolean、null等数据类型。

数组结构的JSON数据

如果一个数组结构的JSON数据中包含了String、Number、Boolean、null类型的具体数据，JSON的表示形式如下：

```
["abc",12345,false,null]
```

数组结构的JSON数据

如果一个数组结构的JSON数据中包含了两个对象结构的数据，JSON的表示形式如下：

```
[  
  {  
    "name":"LiLi",  
    "city":"Beijing"  
  },  
  {  
    "name":"LiLei",  
    "city":"Shanghai"  
  }  
]
```

数组结构的JSON数据

假设上一页中的JSON数据的每个对象中还包含一个hobby信息，其value值是一个数组，则JSON的表示形式如下：

```
[  
  {  
    "name":"LiLi",  
    "city":"Beijing",  
    "hobby":["篮球","乒乓球","听音乐"]  
  },  
  {  
    "name":"LiLei",  
    "city":"Shanghai",  
    "hobby":["看电影","羽毛球","游泳"]  
  }  
]
```

注意：使用JSON存储单个数据（如“abc”），一定使用数组形式，因为对象结构必须是由“key:value”的形式构成。

知识点 2-JSON 解析

教师通过 PPT 的方式讲解 JSON 解析。

- (1) 使用 JSONObject 与 JSONArray 类解析 JSON 数据。
- (2) 使用 Gson 库解析 JSON 数据。

如果想要将 JSON 文件中的数据显示到 Android 程序的界面上，则首先需要将 JSON 数据解析出来。假设现在有两条 JSON 数据，其中 json1 是对象结构的数据，json2 是数组结构的数据，示例代码如下：

```
json1:
{"name":"zhangsan", "age":27, "married":true}
json2:
[{"name":"lisi", "age": 25}, {"name":"Jason", "age":20}]
```

如果要解析上一页中两种格式的 JSON 数据，解析方式有两种，具体介绍如下：

1. 使用 JSONObject 与 JSONArray 类解析 JSON 数据

为了解析 JSON 数据，Android SDK 为开发者提供了 org.json 包，该包存放了解析 JSON 数据的类，JSONObject 类用于解析对象结构的 JSON 数据，JSONArray 类用于解析数组结构的 JSON 数据，这两个类解析 JSON 数据的具体代码如下所示。

- (1) 使用 JSONObject 类解析对象结构的 JSON 数据，示例代码如下：

```
JSONObject jsonObj = new JSONObject(json1);
String name = jsonObj.optString("name");
int age = jsonObj.optInt("age");
boolean married = jsonObj.optBoolean("married");
```

- 2) 使用 JSONArray 类解析数组结构的 JSON 数据，示例代码如下：

```
JSONArray jsonArray = new JSONArray(json2);
for(int i = 0; i < jsonArray.length(); i++) {
    JSONObject jsonObj = jsonArray.getJSONObject(i);
    String name = jsonObj.optString("name");
    int age = jsonObj.optInt("age");
}
```

前面两个类解析 JSON 数据时用到了 optString()方法、optInt()方法、optBoolean()方法，这些方法在解析数据时是安全的，如果对应的字段不存在，这些方法会有默认返回值，程序不会报错。

2. 使用 Gson 库解析 JSON 数据

使用 Gson 库之前必须创建 JSON 数据对应的实体类，实体类中的成员名称必须与 JSON 数据中的 key 值一致，在解析 JSON 数据的示例代码中，数据 json1 对应的实体类以 Person1 为例，数据 json2 对应的实体类以 Person2 为例，实体类 Person1 的具体代码如下所示。

- (1) 使用 Gson 库解析对象结构的 JSON 数据，示例代码如下：

```
Gson gson = new Gson();
Person1 person1 = gson.fromJson(json1, Person1.class);
```

(2) 使用 Gson 库解析数组结构的 JSON 数据，示例代码如下：

```
Gson gson = new Gson();  
Type listType = new TypeToken<List<Person2>>().getType();  
List<Person2> person2 = gson.fromJson(json2, listType);
```

知识点 3-实战演练——商品砍价界面

教师通过 PPT 结合实际操作的方式讲解实战演练——商品砍价界面。

(1) 搭建砍价界面布局。

- 创建程序。
- 导入界面图片。
- 添加 recyclerview 库。
- 添加界面控件。
- 搭建商品列表的条目布局。
- 修改默认标题栏的名称。

(2) 实现砍价界面的功能。

- 添加 okhttp 库。
- 添加 gson 库。
- 添加 picasso 库。
- 商品信息数据的准备。
- 封装商品信息的实体类。
- 编写商品列表的适配器。
- 实现显示商品列表数据的功能。

(3) 设置网络权限。

(4) 运行程序。

第七、八课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：(考察知识点为使用 WebView 浏览网页、使用 WebView 执行 HTML 代码、设置 WebView 支持 JavaScript)

形式：单独完成

题目：

请按照以下要求完成操作，要求如下：

- (16) 实现使用 WebView 控件加载网页的案例。
- (17) 实现使用 WebView 控件加载 HTML 代码。
- (18) 实现使用 WebView 控件支持带有 JavaScript 代码的网页的案例。

上机：(考察知识点为实战演练——商品砍价界面) /或新闻消息界面

形式：单独完成

题目：

实现显示商品数据的砍价界面的案例，要求如下：

- (10) 创建商品砍价界面程序 Hagggle。

	<p>(11) 搭建砍价界面布局。</p> <p>(12) 通过 JSON 解析将商品数据显示到砍价界面上。</p> <p>(13) 实现砍价界面的功能。</p>
教学后记	<p>对“子线程发起请求、UI 线程更新界面”的完整流程讲解较简略，仅演示基础代码，未深入分析“Handler 的正确使用”“AsyncTask 的替代方案”（如 Coroutine、Retrofit）。导致学生在作业中频繁出现“主线程网络请求”“子线程更新 UI”等错误。后续需提供“网络请求工具类模板”，明确“子线程请求→Handler 传递消息→UI 线程更新”的步骤，并通过 StrictMode 演示主线程网络请求的崩溃效果，强化线程规则记忆。</p>

课题名称	第 13 章 图形图像处理	计划课时	4 课时
教学引入	图形图像在 Android 应用中会经常用到，如一些程序的图标、界面的美化等都离不开图形图像。Android 系统对图形图像的处理非常强大，不仅提供了一些绘图类，还提供了 Matrix 类为图像添加特效，同时还提供了补间动画、逐帧动画和属性动画。接下来本章将针对 Android 常用的绘图类、图形图像特效以及动画进行讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生掌握常用绘图类的使用，能够绘制不同的图形 ● 使学生掌握如何使用 Matrix 类，能够实现为图像添加特效的功能 ● 使学生掌握动画的使用，能够实现补间动画、逐帧动画和属性动画的效果 		
教学重点	<ul style="list-style-type: none"> ● Bitmap 类 ● BitmapFactory 类 ● Paint 类 ● Canvas 类 ● 为图像添加特效 ● 补间动画 ● 逐帧动画 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——绘制小狗 ● 实战演练——飞舞的蝴蝶和鸟 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	讲解“Paint 类的样式设置”时，类比“工匠制作工艺品的严谨态度”——Paint.setAntiAlias(true)（抗锯齿）如同“工匠打磨器物边缘，使其光滑无毛刺”；Paint.setStrokeCap(Paint.Cap.ROUND)（圆角线条）如同“工匠处理线条端点，使其圆润美观”。通过演示“无抗锯齿的锯齿状线条”与“有抗锯齿的平滑线条”、“方角端点”与“圆角端点”的视觉差异，让学生理解：绘图细节的精准控制是“让作品更精致”的关键，培养“精益求精、注重细节”的工匠精神。		
教学过程	<p style="text-align: center;">第一课时 （Bitmap 类、BitmapFactory 类、Paint 类）</p> <p>一、复习巩固 教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过直接引入的方式导入新课 Android 中常用的绘图类有 Bitmap 类、BitmapFactory 类、Paint 类以及 Canvas 类，通过对这几个类的使用可以分别实现创建位图、将指定资源解析为位图、创建画笔、绘制画布等功能。本节课先对 Bitmap 类、BitmapFactory 类、Paint 类进行详细讲解。</p> <p>三、新课讲解 知识点 1-Bitmap 类 教师通过 PPT 讲解 Bitmap 类。 (1) Bitmap 类的作用。</p>		

(2) Bitmap 类提供的静态方法。

(3) 通过示例代码演示如何创建一个 Bitmap 类的对象。

Bitmap 类是 Android 中用于表示位图图像的类，它可以理解为内存中的一幅图像。它负责存储图像的像素信息，是 Android 中处理图像的基础。我们在应用中显示的图片、绘制的图形最终都会以 Bitmap 的形式存在，它可以从资源文件、文件、网络等多种来源创建。



知识点 2-BitmapFactory 类

教师通过 PPT 讲解 BitmapFactory 类。

(1) BitmapFactory 类的作用。

(2) BitmapFactory 类提供的常用方法。

BitmapFactory类的常用方法

方法名称	功能描述
decodeFile(String pathName)	将指定路径的文件解码为位图
decodeStream(InputStream is)	将指定输入流解码为位图
decodeResource(Resources res, int id)	将给定的资源id解析为位图

Bitmap bitmap =

```
BitmapFactory.decodeResource(this.getResources(),R.drawable.icon);
```

通过decodeResource()方法将drawable文件夹中的icon.png图片资源解码为位图

知识点 3-Paint 类

教师通过 PPT 讲解 Paint 类。

(1) Paint 类的作用。

(2) Paint 类的常用方法。

方法名称	功能描述
setARGB(int a, int r, int g, int b)	设置颜色，各参数值均为0~255之间的整数，几个参数分别用于表示透明度、红色、绿色和蓝色的值
setColor(int color)	设置画笔颜色，参数值可以是Color.RED（红色）、Color.parseColor("#000000")（黑色）等
setAlpha(int a)	设置绘制图形的透明度，参数值的范围为0~255，数值越小图形越透明
setAntiAlias(boolean aa)	设置画笔是否使用抗锯齿功能
setTextAlign(Align align)	设置绘制文本时的文字对齐方式。参数值为Align.CENTER、Align.LEFT、Align.RIGHT，分别表示居中、左对齐、右对齐
setTextSize(float textSize)	设置绘制文本时的文字大小
setFakeBoldText(boolean fakeBoldText)	设置绘制文本时文字是否为粗体
setDither(boolean dither)	指定是否使用图像抖动，如果使用会使图像颜色更加平滑、饱满、清晰
setShadowLayer(float radius, float dx, float dy, int color)	设置阴影。radius表示阴影的角度，dx和dy表示阴影在x轴和y轴上的距离，color表示阴影的颜色
setXfermode(Xfermode xfermode)	设置图像的混合模式

```
Paint paint = new Paint();
paint.setColor(Color.RED);
```

指定画笔颜色为红色

第二课时

（Canvas 类、实战演练——绘制小狗、为图像添加特效）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

十二、通过直接引入的方式导入新课

上一节课讲解了常用的绘图类 Bitmap、BitmapFactory 和 Paint，本节课将讲解常用的绘图类 Canvas、实战演练——绘制小狗和为图像添加特效的内容。

十三、新课讲解

知识点 1-Canvas 类

教师通过 PPT 讲解 Canvas 类。

- (1) Canvas 类的作用。
- (2) Canvas 类的常用方法。

方法名称	功能描述
drawRect(Rect r, Paint paint)	使用画笔绘制矩形
drawOval(RectF oval, Paint paint)	使用画笔绘制椭圆形
drawCircle(float cx, float cy, float radius, Paint paint)	使用画笔在指定位置画出指定半径的圆
drawLine(float startX, float startY, float stopX, float stopY, Paint paint)	使用画笔在指定位置画线
drawRoundRect(RectF rect, float rx, float ry, Paint paint)	使用画笔绘制指定圆角矩形，其中rx表示X轴圆角半径，ry表示Y轴圆角半径

在View的onDraw()方法中使用画笔类Paint在画布上绘制矩形。

```
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    Paint paint = new Paint();
    paint.setColor(Color.RED);
    Rect rect = new Rect(40,40,200,100);
    canvas.drawRect(rect,paint);
}
```

创建画笔
构建矩形对象并为其指定位置、宽高
调用Canvas中绘制矩形的方法

知识点 2-实战演练——绘制小狗

教师通过 PPT 结合实际操作的方式讲解实战演练——绘制小狗。

- (1) 创建程序。
- (2) 导入界面图片。
- (3) 自定义绘制小狗的 View。
- (4) 添加界面控件。
- (5) 修改默认标题栏的名称。

知识点 3-为图像添加特效

教师通过 PPT 结合实际操作的方式讲解为图像添加特效。

- (1) Matrix 类的作用。
- (2) Matrix 类中实现平移、旋转、缩放和倾斜特效的方法。
 - 实现平移特效的方法。
 - 实现旋转特效的方法。
 - 实现缩放特效的方法。
 - 实现倾斜特效的方法。

特效	方法名称	功能描述
平移	setTranslate(float dx,float dy)	指定图像在X、Y轴移动dx和dy的距离
	preTranslate(float dx, float dy)	使用前乘的方式计算在X、Y轴平移的距离。
	postTranslate(float dx,float dy)	使用后乘的方式计算在X、Y轴平移的距离。
旋转	setRotate(float degrees)	指定图片旋转degrees度
	preRotate(float degrees)	使用前乘的方式指定图片旋转degrees度
	postRotate(float degrees, float px, float py)	使用后乘的方式控制图像以参数px和py为轴心旋转degrees度
缩放	setScale(float sx, float sy)	指定图像在X轴和Y轴的缩放比例为sx和sy
	preScale(float sx, float sy)	使用前乘的方式计算图像在X轴和Y轴的缩放比例
	postScale(float sx, float sy)	使用后乘的方式计算图像在X轴和Y轴的缩放比例
倾斜	setSkew(float kx, float ky)	指定图像在X、Y轴的倾斜值
	preScale(float kx, float ky)	使用前乘的方式设置图像在X、Y轴的倾斜值
	postScale(float kx, float ky)	使用后乘的方式设置图像在X、Y轴的倾斜值

	<p>(3) 通过案例演示如何使用 Matrix 类为图片添加向右下角移动的特效。</p> <ul style="list-style-type: none"> ● 创建程序。 ● 导入界面图片。 ● 创建 TranslateView 类。 ● 引用 TranslateView 类。 <p>(4) 运行结果。</p> <p>四、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p> <p>教师通过学习通布置本节课作业。</p> <p style="text-align: center;">第三、四课时 (上机练习)</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机：（考察知识点为实战演练——绘制小狗、为图像添加特效）</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照以下要求完成操作，要求如下：</p> <p>(19) 通过常用的绘图类实现绘制小狗的案例。</p> <p>(20) 通过 Matrix 类实现为图像添加特效的案例。</p>
<p>教学后记</p>	<p>绘制图形在应用开发中应用比较广泛，可添加一些 APP 常用的应用场景进行实现和讲解</p>

课题名称	第 14 章 多媒体应用开发	计划课时	4 课时
教学引入	<p>随着互联网的迅速发展，手机已经成为人们日常生活中必不可少的设备，设备里面的多媒体资源想必是很多人的兴趣所在。多媒体资源一般包括音频、视频等，针对不同的多媒体资源 Android 系统提供了不同的类对其进行操作。接下来，本章将针对多媒体应用中的音频和视频操作进行讲解。</p>		
教学目标	<ul style="list-style-type: none"> ● 使学生掌握 MediaPlayer 类与 SoundPool 类的使用，能够实现播放音频文件的功能 ● 使学生掌握 VideoView 控件的使用，能够实现播放视频文件的功能 ● 使学生掌握 MediaPlayer 类与 SurfaceView 控件的使用，能够实现播放视频文件的功能 		
教学重点	<ul style="list-style-type: none"> ● 使用 MediaPlayer 类播放音频 ● 使用 SoundPool 类播放音频 ● 使用 VideoView 控件播放视频 ● 使用 MediaPlayer 类和 SurfaceView 控件播放视频 		
教学难点	<ul style="list-style-type: none"> ● 实战演练——弹钢琴 ● 实战演练——VideoView 视频播放器 ● 实战演练——SurfaceView 视频播放器 		
教学方式	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>		
思政元素	<p>在讲解“MediaPlayer 类的资源释放”时，类比“生活中的设备断电习惯”——MediaPlayer 在播放音频后若不调用 release() 方法（如同“离开房间未关闭电器电源”），会导致音频资源占用（如扬声器、解码器），甚至引发内存泄漏。通过这种实践，培养学生“珍惜系统资源、避免浪费”的环保思维，理解“多媒体组件的资源释放不仅影响应用性能，更是对设备硬件的负责任使用”。</p>		
教学过程	<p style="text-align: center;">第一、二课时</p> <p style="text-align: center;">（使用 MediaPlayer 类播放音频、使用 SoundPool 类播放音频、实战演练——弹钢琴）</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过需求引入的方式导入新课</p> <p>随着手机硬件的不断提升，手机已经成为人们日常生活中必不可少的设备，设备里面的多媒体资源想必是很多人的兴趣所在，多媒体资源一般包括音频和视频。为了让读者更好地学习多媒体资源，本节课我们将针对 MediaPlayer 类播放音频、SoundPool 类播放音频和实战演练——弹钢琴进行详细讲解。</p> <p>三、新课讲解</p> <p>知识点 1-使用 MediaPlayer 类播放音频</p> <p>教师通过 PPT 结合实际操作的方式讲解使用 MediaPlayer 类播放音频。</p> <ol style="list-style-type: none"> （1）MediaPlayer 类的作用。 （2）MediaPlayer 类的常用方法。 （3）通过案例演示使用 MediaPlayer 类播放音频。 		

- 实例化 MediaPlayer 类。
- 设置数据源。
- 播放音频文件。
- 暂停播放。
- 重新播放。
- 停止播放。

实例化MediaPlayer类

```
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
```

MediaPlayer接收的音频类型有很多种，其中有四种较为常用：

- AudioManager.STREAM_MUSIC (音乐)
- AudioManager.STREAM_RING (响铃)
- AudioManager.STREAM_ALARM (闹钟)
- AudioManager.STREAM_NOTIFICATION (提示音)

设置数据源

设置数据源有三种方式，分别是设置播放应用自带的音频文件、设置播放SD卡中的音频文件、设置播放网络音频文件。

```
mediaPlayer = MediaPlayer.create(this, R.raw.xxx);
mediaPlayer.setDataSource("SD卡中的音频文件路径");
mediaPlayer.setDataSource("http://www.xxx.mp3");
```

需要注意的是，播放网络中的音频文件时，需要在清单文件中添加访问网络的权限。
`<uses-permission android:name="android.permission.INTERNET"/>`

(1) 播放小音频文件

```
mediaPlayer.prepare();
mediaPlayer.start();
```

prepare()方法是同步操作，在主线程中执行，它会对音频文件进行解码，当prepare()执行完成之后才会向下执行。

(2) 播放大音频文件

```
mediaPlayer.prepareAsync();  
mediaPlayer.setOnPreparedListener(new OnPreparedListener(){  
    public void onPrepared(MediaPlayer player){  
        mediaPlayer.start();  
    }  
})
```

prepareAsync()是子线程中执行的异步操作，不影响主线程。但是如果音频文件没解码完毕就执行start()方法会播放失败。因此要监听音频是否准备好的监听器OnPreparedListener。音频解码完成时会执行onPreparedListener()中的onPrepared()方法，在该方法中执行播放音乐的操作。

暂停播放

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.pause();  
}
```

暂停播放之前要先判断MediaPlayer对象是否存在，并且是否正在播放音乐。

重新播放

播放状态下进行重播：

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.seekTo(0);  
    return;  
}
```

暂停状态下进行重播：

```
if(mediaPlayer!=null){  
    mediaPlayer.seekTo(0);  
    mediaPlayer.start();  
}
```

停止播放

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.stop();  
    mediaPlayer.release();  
    mediaPlayer = null;  
}
```

停止播放

调用release()方法将占用的资源释放并将MediaPlayer置为空

知识点 2-使用 SoundPool 类播放音频

教师通过 PPT 结合实际操作的方式讲解使用 SoundPool 类播放音频。

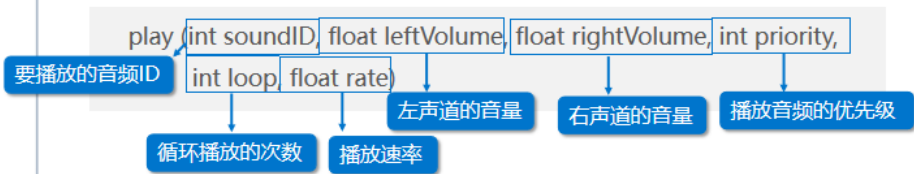
- (1) SoundPool 类的作用。
- (2) SoundPool 类设置音频的常用方法。
- (3) 通过案例演示如何通过 SoundPool 类播放音频。
 - 创建 SoundPool 类的对象。
 - 加载音频文件。
 - 播放音频。

SoundPool 即音频池，可以同时播放多个短小的音频，而且占用的资源比较少，它适合在应用程序中播放按键音或者消息提示音等。SoundPool 类中设置音频的常用方法如下表所示。

方法名称	功能描述
load()	加载音频文件
play()	播放音频
pause(int streamID)	根据加载的资源ID暂停播放音频
resume(int streamID)	根据加载的资源ID继续播放暂停的音频
stop(int streamID)	根据加载的资源ID停止音频的播放
unload(int soundID)	从音频池中卸载音频资源ID为soundID的资源
release()	释放音频池资源

播放音频

调用SoundPool对象的play()方法可播放指定的音频，具体如下：



播放raw文件夹中的sound.wav音频文件的示例代码如下：

```
soundpool.play(soundpool.load(MainActivity.this, R.raw.sound, 1), 1, 1, 0, 0, 1);
```

知识点 3-实战演练——弹钢琴

教师通过 PPT 结合实际操作的方式讲解案例——弹钢琴。

- (1) 创建程序。
- (2) 导入音频文件。
- (3) 导入界面图片。
- (4) 添加界面控件。
- (5) 创建背景选择器。
- (6) 设置钢琴界面横屏显示。
- (7) 编写界面交互代码。
- (8) 运行结果。

第三、四课时

(使用 **VideoView** 控件播放视频、实战演练——**VideoView** 视频播放器、使用 **MediaPlayer** 类和 **SurfaceView** 控件播放视频、实战演练——**SurfaceView** 视频播放器)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

十三、通过需求引入的方式导入新课

上节课讲解了 **MediaPlayer** 类与 **SoundPool** 类播放音频与弹钢琴案例，我们学会播放音频后，那么视频该如何播放呢？此时 **Android** 系统给提供了 **VideoView** 控件、**MediaPlayer** 类和 **SurfaceView** 控件来播放视频，本节课我们将针对使用 **VideoView** 控件播放视频、实战演练——**VideoView** 视频播放器、使用 **MediaPlayer** 类和 **SurfaceView** 控件播放视频、实战演练——**SurfaceView** 视频播放器进行详细讲解。

十四、新课讲解

知识点 1-使用 VideoView 控件播放视频

教师通过 PPT 结合实际操作的方式讲解使用 VideoView 控件播放视频。

- (1) VideoView 控件的作用。
- (2) VideoView 控件的常用方法。
- (3) 通过案例讲解如何通过 VideoView 控件播放视频。
 - 在布局文件中添加 VideoView 控件。
 - 视频的播放。
 - 为 VideoView 控件添加控制器。

1.在布局文件中添加VideoView控件

```
<VideoView  
    android:id="@+id/videoview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

2.视频的播放

```
VideoView videoView = (VideoView) findViewById(R.id.videoview);  
videoView.setVideoPath("mnt/sdcard/apple.avi");  
videoView.setVideoURI(Uri.parse("http://www.xxx.avi"));  
videoView.start();
```

播放本地视频

加载网络视频

3. 为VideoView控件添加控制器

MediaController控制器中包含媒体播放器 (MediaPlayer) 中的一些典型按钮, 如播放/暂停 (Play/ Pause)、倒带 (Rewind)、快进 (Fast Forward) 与进度滑动器 (progress slider) 等。VideoView控件能够绑定媒体播放器 (MediaController), 从而使播放状态和控件中显示的图像同步, 示例代码如下。

```
MediaController controller = new MediaController(context);  
videoView.setMediaController(controller);
```

绑定控制器

知识点 2-实战演练——VideoView 视频播放器

教师通过 PPT 结合实际操作的方式讲解实战演练——VideoView 视频播放器。

- (1) 创建程序。
- (2) 导入视频文件。
- (3) 添加界面控件。
- (4) 实现播放视频功能。
- (5) 运行结果。

	<p>知识点 3-使用 MediaPlayer 类和 SurfaceView 控件播放视频 教师通过 PPT 结合实际操作的方式讲解使用 MediaPlayer 类和 SurfaceView 控件播放视频。</p> <p>(1) MediaPlayer 类和 SurfaceView 控件的作用。</p> <p>(2) 使用 MediaPlayer 类和 SurfaceView 控件实现视频播放器的过程。</p> <ul style="list-style-type: none"> ● 在布局中添加 SurfaceView 控件。 ● 获取界面显示容器并设置类型。 ● 回调 addCallback()方法。 ● 播放视频。 <p>知识点 4-实战演练——SurfaceView 视频播放器 教师通过 PPT 结合实际操作的方式讲解实战演练——SurfaceView 视频播放器。</p> <p>(1) 创建程序。</p> <p>(2) 添加资源文件。</p> <p>(3) 添加界面控件。</p> <p>(4) 设置播放视频界面横屏显示。</p> <p>(5) 实现播放视频功能。</p> <ul style="list-style-type: none"> ● 初始化界面控件。 ● 监听 Surface 创建、改变和销毁时的状态。 ● 实现 SeekBar 进度条的监听事件。 ● 显示或隐藏视频播放界面的进度条、播放或暂停图标。 <p>(6) 运行程序。</p> <p>四、归纳总结</p> <p>教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。</p> <p style="text-align: center;">(上机练习)</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机：（考察知识点为实战演练——弹钢琴、实战演练——VideoView 视频播放器）</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照以下要求完成操作，要求如下：</p> <p>(21) 完成弹钢琴案例。</p> <p>(22) 完成 VideoView 视频播放器案例。</p>
教学后记	<p>MediaPlayer 的“初始化→准备→播放→暂停→停止”状态流程抽象，仅通过文字描述和流程图，部分学生难以理解“状态间的切换规则”。例如，学生常混淆“prepare()（同步）”与“prepareAsync()（异步）”的使用场景，导致播放时机错误。后续教学可通过“状态切换动画”（如用不同颜色标注可切换状态）或“分步调试”（在每个状态方法中打印 Log），让抽象流程可视化。</p>

课题名称	第 11 章 实践项目——天气预报	计划学时	6 学时
内容分析	讲解安卓小项目的构建过程和实现过程；		
教学目标	掌握安卓小项目的开发过程		
教学重点	天气预报应用的开发过程		
教学难点	代码编写		
教学方式	教师采用课堂讲授为主，使用教学 PPT 辅助授课。		
教学过程	<p>第一课时</p> <p>1. 项目介绍和需求分析</p> <ul style="list-style-type: none"> ○ 介绍天气预报项目的背景和意义，如天气预报对人们日常生活、出行等的重要性。 ○ 分析天气预报应用的基本功能需求，如获取实时天气数据、显示天气信息（包括温度、天气状况、风力等）、提供未来几天的天气预报、展示天气图标等。 ○ 讨论项目的技术需求，如网络请求（获取天气数据的接口和方式）、数据解析（如何将获取到的数据解析成可用的格式）、界面设计（布局、颜色、字体等）。 ○ 思政融入点：引导学生关注天气变化对人们生活的影响，培养学生的环保意识和对自然的敬畏之心。例如，提醒学生天气变化对农业、交通等领域的影响，以及我们应该如何保护环境，减少气候变化对我们的影响。 <p>2. Android 开发环境搭建</p> <ul style="list-style-type: none"> ○ 介绍 Android Studio 开发工具的安装步骤和注意事项，包括下载安装包、安装过程中的选项设置等。 ○ 讲解如何配置 Android 开发环境，如安装 Android SDK、设置环境变量等。 ○ 演示创建一个新的 Android 项目的过程，包括项目名称、包名、最低 SDK 版本等的设置。 ○ 学生实际操作，搭建自己的开发环境并创建项目，教师巡视指导，帮助学生解决遇到的问题。 ○ 思政融入点：在开发环境搭建过程中，培养学生的耐心和细心，让学生养成 		

良好的编程习惯。例如，提醒学生在安装和配置过程中要仔细阅读提示信息，按照步骤进行操作，避免出现错误。同时，鼓励学生遇到问题时不要轻易放弃，要积极寻找解决问题的方法。

第二课时

1. 网络请求和数据解析

- 介绍 Android 中网络请求的常用方式，如使用 `URLConnection` 或 `OkHttp` 库。讲解它们的优缺点和适用场景。
- 详细讲解如何发送 `GET` 或 `POST` 请求获取天气数据，包括请求的 `URL` 格式、参数设置等。
- 介绍数据解析的方法，如使用 `JSON` 解析库。讲解 `JSON` 数据的格式和解析原理。
- 演示如何将获取到的 `JSON` 数据进行解析和存储，提取出需要的天气信息。
- **思政融入点：**强调网络安全和数据合法性的重要性，引导学生遵守相关法律法规，不获取和使用非法数据。例如，提醒学生在获取数据时要注意数据的来源是否可靠，避免使用来路不明的数据。同时，教育学生要尊重他人的知识产权，不随意窃取他人的数据。

2. 代码实现和调试

- 学生根据教师的演示，开始编写网络请求和数据解析的代码。教师提供示例代码，学生在此基础上进行修改和完善。
- 教师巡视指导，帮助学生解决遇到的问题，如网络请求失败、数据解析错误等。
- 讲解代码调试的方法和技巧，如使用 `Logcat` 输出日志信息，查看程序的运行状态和错误信息。
- 学生进行代码调试，确保网络请求和数据解析功能的正确性，能够成功获取并解析天气数据。
- **思政融入点：**在代码实现和调试过程中，培养学生的严谨态度和责任感，让学生明白代码质量的重要性。例如，提醒学生在编写代码时要认真细致，避免出现粗心大意的错误。同时，鼓励学生对自己的代码进行严格的测试和调试，确保代码的稳定性和可靠性。

```
String str = "https://www.mxnzp.com/api/weather/forecast/" +  
            cityname +  
            "  
            市?app_id=ejnkusldmgsnq6pa&app_secret=TGptMUI4elZOcWdhRmQrSVVMUXRmZz09";  
  
OkHttpClient okHttpClient = new OkHttpClient();//创建单例  
  
Request request = new Request.Builder()//创建请求
```

	<pre>.url(str) .build(); Response response = okHttpClient.newCall(request).execute();//执行请求 String mContent = response.body().string();//得到返回响应，注意 response.body().string() 只能调用一次！ JSONObject js1 = new JSONObject(mContent); JSONObject data = js1.optJSONObject("data"); JSONArray weatherarray = data.getJSONArray("forecasts"); // 这里天气预报数据有多组，可以用一个循环来进行数据读取 ArrayList<Weather> weatherlist = new ArrayList<>(); for(int i=0;i<weatherarray.length();i++){ Weather weather = new Weather(); JSONObject weatherjson = weatherarray.getJSONObject(i); weather.setDate(weatherjson.optString("date")); weather.setDayOfWeek(weatherjson.optString("dayOfWeek")); weather.setDayWeather(weatherjson.optString("dayWeather")); weather.setNightWeather(weatherjson.optString("nightWeather")); weather.setDayTemp(weatherjson.optString("dayTemp")); weather.setNightTemp(weatherjson.optString("nightTemp")); weather.setDayWindDirection(weatherjson.optString("dayWindDirection")); weather.setNightWindDirection(weatherjson.optString("nightWindDirection")); weather.setDayWindPower(weatherjson.optString("dayWindPower"));</pre>
--	--

```
        weather.setNightWindPower(weatherjson.optString("nightWindPower"));

        weatherlist.add(weather);
    }

    weatherCanvas.weatherlist =weatherlist;

    weatherCanvas.post(new Runnable() {

        @Override

        public void run() {

            weatherCanvas.invalidate();

        }

    });

    String temp1 = weatherlist.get(0).getDayTemp();

    String weather1 = weatherlist.get(0).getDayWeather();

    String power = weatherlist.get(0).getDayWindPower();

    String dr = weatherlist.get(0).getDayWindDirection();
```

第三课时

1. 界面设计

- 介绍天气预报应用的界面设计原则，如简洁、美观、易用等。强调界面设计要符合用户的使用习惯和审美需求。
- 展示一些优秀的天气预报应用界面，让学生学习借鉴，分析它们的布局、颜色搭配、图标设计等优点。
- 讲解 Android 中界面布局的常用方式，如使用 XML 布局文件或代码动态布局。介绍它们的特点和适用场景。
- 演示如何设计天气预报应用的界面，包括使用哪些控件来显示天气信息、图

标、温度等。展示如何进行布局的调整和优化，以达到最佳的显示效果。

- **思政融入点：**引导学生注重用户体验，设计出友好、美观的界面，提高用户满意度。例如，提醒学生在设计界面时要考虑用户的需求和感受，尽量使界面简洁明了，易于操作。同时，鼓励学生发挥创意，设计出具有个性和特色的界面。

2. 界面实现和优化

- 学生根据教师的演示，开始编写界面布局的代码。使用 XML 布局文件或代码动态布局来实现天气预报应用的界面。
- 教师巡视指导，帮助学生解决遇到的问题，如控件的摆放位置、颜色的搭配、字体的大小等。
- 讲解界面优化的方法，如提高界面响应速度、减少内存占用等。介绍一些优化技巧，如使用合适的图片格式、避免内存泄漏等。
- 学生进行界面优化，确保应用的流畅性和稳定性。对界面进行调整和改进，使其更加美观和易用。
- **思政融入点：**在界面实现和优化过程中，培养学生的创新意识和审美能力，让学生不断追求卓越。例如，鼓励学生尝试不同的布局方式和颜色搭配，寻找最佳的设计方案。同时，提醒学生要注重细节，不断完善界面的每一个部分。

```
public class MainActivityWeatherInfo extends AppCompatActivity {

    TextView tv_cityname, tv_temp, tv_weather, tv_windpower, tv_winddirection;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_weather_info);

        tv_cityname = findViewById(R.id.s_cityname);
        tv_temp = findViewById(R.id.s_temp);
        tv_weather = findViewById(R.id.s_weather);
        tv_windpower = findViewById(R.id.s_windPower);
        tv_winddirection = findViewById(R.id.s_windDirection);

        Intent intent = getIntent();
        String cityname = intent.getStringExtra("cityname");
        String temp = intent.getStringExtra("temp");
        String weather = intent.getStringExtra("weather");
        String windpower = intent.getStringExtra("windpower");
        String winddirection = intent.getStringExtra("winddirection");

        tv_cityname.setText(cityname);
        tv_temp.setText(temp);
        tv_weather.setText(weather);
        tv_winddirection.setText(winddirection);
        tv_windpower.setText(windpower);
    }
}
```

第四课时

1. 功能集成和测试

- 将网络请求、数据解析和界面设计进行集成，实现天气预报应用的完整功能。讲解如何将各个模块的代码整合在一起，使其能够协同工作。
- 讲解功能测试的方法，如手动测试、自动化测试等。介绍如何使用测试工具来检测应用的功能是否正常。
- 学生进行功能测试，确保应用的各项功能正常，如能够准确获取天气数据、显示天气信息、更新天气预报等。
- 教师巡视指导，帮助学生解决测试中发现的问题，如数据显示异常、界面卡顿等。
- **思政融入点：**在功能集成和测试过程中，培养学生的团队协作精神和沟通能力，让学生学会相互配合，共同完成项目。例如，提醒学生在集成过程中要与小组成员密切合作，及时沟通解决问题。同时，鼓励学生认真对待测试工作，确保应用的质量。

2. 问题解决和优化

- 学生提出在项目开发过程中遇到的问题，教师进行集中解答和指导。针对常见问题，如网络请求失败、数据解析错误、界面显示异常等，提供解决方案。
- 讨论如何进一步优化应用的性能和用户体验，如提高数据加载速度、优化界面布局、增加交互功能等。
- 学生根据讨论结果，对应用进行优化和改进。对代码进行优化，提高程序的运行效率；对界面进行调整，提高用户体验。
- **思政融入点：**在问题解决和优化过程中，培养学生的解决问题的能力 and 创新精神，让学生学会不断挑战自我，追求更高的目标。例如，鼓励学生积极思考，寻找创新的解决方案，不断提升应用的质量和性能。

第五课时

1. 拓展功能实现

- 介绍一些天气预报应用的拓展功能，如空气质量监测、天气预警推送等。讲解这些功能的实现原理和技术要点。
- 演示如何在天气预报应用中添加空气质量监测功能，如获取空气质量数据、显示空气质量指数等。
- 学生根据教师的演示，开始实现空气质量监测功能，编写相应的代码。
- **思政融入点：**在拓展功能实现过程中，培养学生的创新意识和探索精神，让学生敢于尝试新的技术和方法，不断开拓进取。例如，鼓励学生积极探索新的功能和技术，尝试将其应用到项目中，提升应用的竞争力。

2. 功能完善和优化

- 学生继续完善天气预报应用的其他功能，如优化天气数据的更新机制、增加天气图表展示等。
- 教师巡视指导，帮助学生解决遇到的问题，如数据更新不及时、图表显示异

常等。

- 讲解如何进一步优化应用的性能，如减少内存占用、提高响应速度等。
- 学生进行性能优化，确保应用在各种设备上都能流畅运行。
- **思政融入点：**在功能完善和优化过程中，培养学生的责任心和敬业精神，让学生认真对待每一个细节，努力打造高质量的作品。例如，提醒学生要关注用户的需求和反馈，不断改进和完善应用的功能和性能，提高用户满意度。

```
public class WeatherCanvas extends View {  
    ArrayList<Weather> weatherlist;  
  
    public WeatherCanvas(Context context, @Nullable AttributeSet attrs) {  
        super(context, attrs);  
    }  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        Paint mypaint = new Paint();  
  
        // 这块画布接下来是要画温度的变化情况  
        mypaint.setColor(Color.BLUE);  
        mypaint.setTextSize(20);  
  
        //  
        // canvas.drawCircle(100,100,80,mypaint);  
        //  
        // canvas.drawCircle(300,100,80,mypaint);  
  
        float step = 150;  
        float y_step = 15;  
        float start_x = 20;  
        float topy = 600;  
        if(weatherlist != null) {  
            // 画温度的折线图 连线  
  
            Log.i("test",weatherlist.get(0).getDayTemp().substring(0,weatherlist.get(0).getDayTemp().length()-1));  
  
            float temp0 = Float.valueOf(weatherlist.get(0).getDayTemp().substring(0,weatherlist.get(0).getDayTemp().length()-1));  
            float temp1 = Float.valueOf(weatherlist.get(1).getDayTemp().substring(0,weatherlist.get(1).getDayTemp().length()-1));  
            float temp2 = Float.valueOf(weatherlist.get(2).getDayTemp().substring(0,weatherlist.get(2).getDayTemp().length()-1));  
            float temp3 = Float.valueOf(weatherlist.get(3).getDayTemp().substring(0,weatherlist.get(3).getDayTemp().length()-1));  
            // float temp4 = Float.valueOf(weatherlist.get(4).getDayTemp().substring(0,weatherlist.get(4).getDayTemp().length()-1));  
            // float temp5 = Float.valueOf(weatherlist.get(5).getDayTemp().substring(0,weatherlist.get(5).getDayTemp().length()-1));  
            // float temp6 = Float.valueOf(weatherlist.get(6).getDayTemp().substring(0,weatherlist.get(6).getDayTemp().length()-1));  
  
            canvas.drawLine(start_x, topy-temp0*y_step, start_x+step, topy-temp1*y_step, mypaint);  
            canvas.drawLine(start_x+step, topy-temp1*y_step, start_x+step*2, topy-temp2*y_step, mypaint);  
            canvas.drawLine(start_x+step*2, topy-temp2*y_step, start_x+step*3, topy-temp3*y_step, mypaint);  
  
            canvas.drawText(temp0 + "°", start_x, topy-temp0*y_step-20, mypaint);  
            canvas.drawText(temp1 + "°", start_x+step, topy-temp1*y_step-20, mypaint);  
        }  
    }  
}
```

第六课时

1. 界面美化和交互优化

- 介绍一些界面美化的方法，如使用自定义图标、调整颜色搭配、增加动画效果等。
- 演示如何优化天气预报应用的交互体验，如增加手势操作、优化界面切换效果等。
- 学生根据教师的演示，开始进行界面美化和交互优化的工作，提升应用的美观度和易用性。
- **思政融入点：**在界面美化和交互优化过程中，培养学生的审美能力和用户意识，让学生学会从用户的角度出发，设计出更加友好和舒适的界面。例如，鼓励学生注重细节，追求完美，打造出具有吸引力的界面。

2. 兼容性测试和修复

	<ul style="list-style-type: none"> ○ 讲解兼容性测试的重要性，以及如何进行兼容性测试，确保应用在不同型号的设备上都能正常运行。 ○ 学生进行兼容性测试，发现并解决应用在不同设备上可能出现的问题，如界面显示异常、功能失效等。 ○ 教师巡视指导，帮助学生分析和解决兼容性问题，提供相应的解决方案。 ○ 思政融入点：在兼容性测试和修复过程中，培养学生的耐心和细心，让学生学会认真对待每一个问题，确保应用的质量和稳定性。例如，提醒学生要考虑到不同用户的设备情况，尽可能地提高应用的兼容性，让更多的用户能够正常使用。
<p style="text-align: center;">教 学 后 记</p>	<p>针对线程安全：设计“子线程直接操作 UI”与“通过 post 切换线程”的对比实验，让学生亲手触发 <code>CalledFromWrongThreadException</code>，再通过 post 解决问题，加深对规则的记忆。</p>