第 1 章　项 目 综 述

学习目标

- 了解新闻客户端项目的功能与模块结构。
- 了解新闻客户端项目的界面交互效果。

新闻客户端项目是一个移动新闻客户端，包含体育、财经、娱乐、科技等多个新闻板块的新闻客户端，可以对普通新闻、图片新闻和视频新闻进行查看，要求通过网络接口，或者自己搭建的后台获取最新的新闻，并进行数据更新。本章将针对项目的整体功能进行简单介绍，并对软件整体效果进行演示。

1.1　项 目 分 析

【任务 1-1】项目名称

东仔移动新闻客户端。

【任务 1-2】项目概述

移动新闻客户端凭借其丰富的资讯资源、实时的信息推送和方便的社交互动被越来越多的用户认可。东仔移动新闻客户端利用网易新闻 API（应用程序接口）提供的新闻数据，呈现的文字、图片、视频等资讯信息量大、覆盖面广，能满足同学们日常的新闻咨询需求，整个项目覆盖了 Android 的主要知识点，培养了常见 Android 开源框架的开发技能，通过该项目的开发，能够帮助同学们早日达到 Android 开发工程师的标准。

【任务 1-3】开发环境

操作系统：Windows 系统。

开发工具：

- JDK8。
- Android Studio 2.2.2。

数据库：SQLite。

其他：新闻数据使用了网易新闻提供的新闻 API。

【任务 1-4】模块说明

移动新闻客户端主要分为 4 个功能模块，其中包含新闻纵横、图片中心、推荐视频、“我”，如图 1-1 所示。

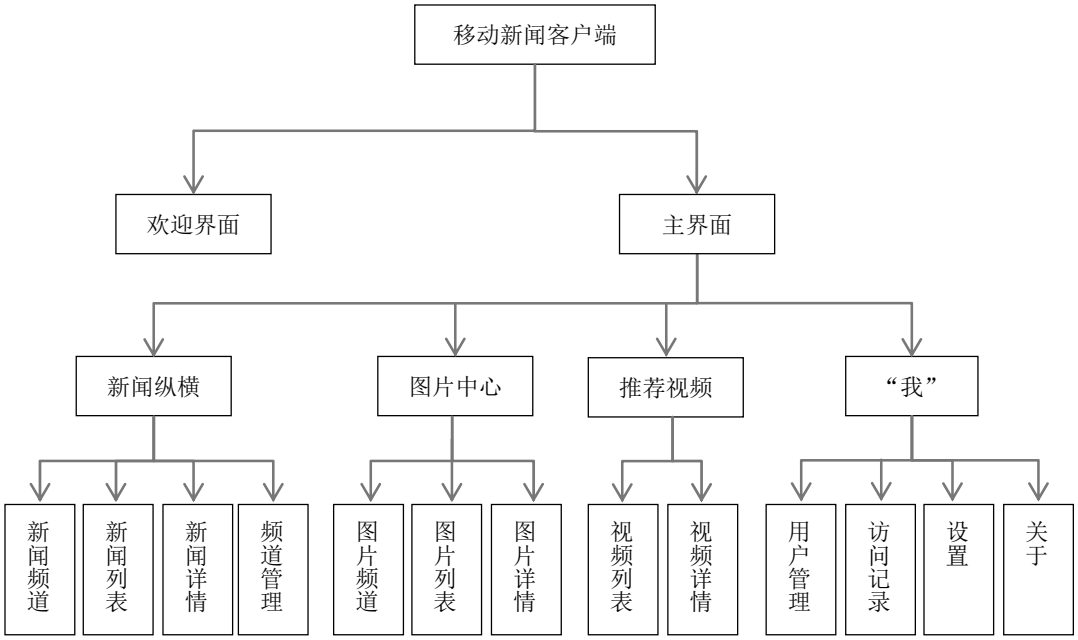

图 1-1　项目结构

从图 1-1 可以看出，移动新闻客户端主要分为两块：一个是欢迎界面；另一个是主界面。在欢迎界面中会显示程序的版本号以及功能提示等，在主界面中显示 4 个功能模块，每个功能模块还有多个具体的小功能。

1.2　效 果 展 示

【任务 1-5】欢迎界面和主界面

程序启动成功后，首先会在欢迎界面停留几秒然后进入主界面，主界面会默认加载新闻纵横 fragment，单击底部导航条的图片按钮、视频按钮或“我”的按钮可以切换到图片中心界面、推荐视频界面或“我”的界面，如图 1-2 所示。

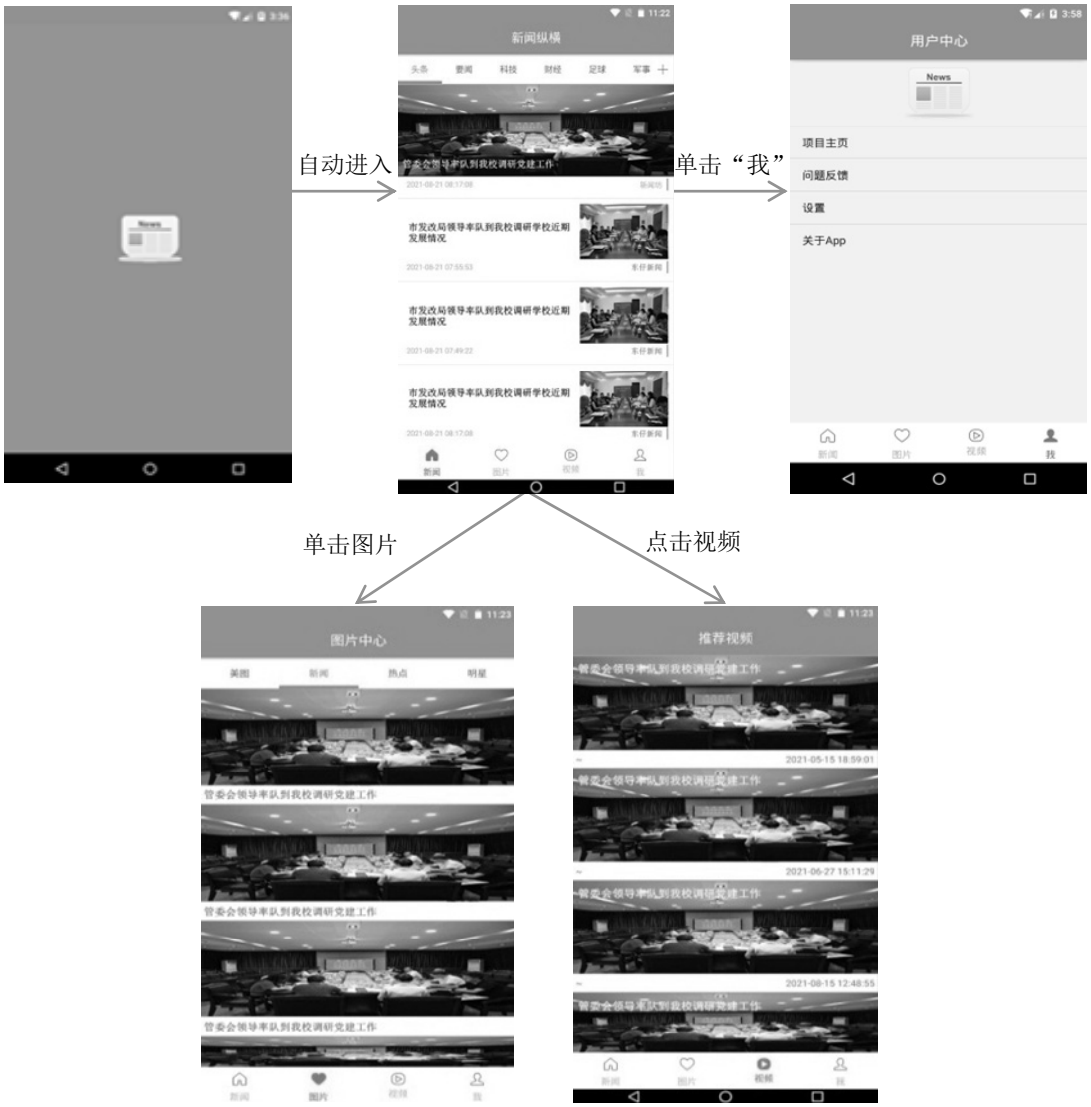

图 1-2　欢迎界面、主界面

【任务 1-6】新闻界面

新闻界面包含新闻列表界面、普通新闻详情页和图片新闻详情页，新闻分为多个频道，单击新闻列表界面上方的新闻频道导航条的某个频道时，会进入该频道新闻列表界面，单击某一条新闻则可以根据新闻类型（普通或图片）在普通新闻详情页或图片新闻详情页中查看新闻的具体内容，滚动新闻频道导航条可以查看更多的新闻频道，单击新闻频道导航条最右侧的加号会进入新闻频道管理页面，如图 1-3 所示。

图 1-3　新闻界面

【任务 1-7】图片界面

图片界面包含图片列表界面、图片详情页，图片分为多个频道，单击图片列表界面上方的图片频道导航条的某个频道时，会进入该频道图片列表界面，单击某一条图片则可以在图片详情页中查看图片的具体内容（可左右滚动查看图片集中的多张图片），如图 1-4 所示。

图 1-4　图片界面

【任务 1-8】视频界面

视频界面包含图片列表界面、图片详情页，视频不分频道，单击某一条视频则可以在视频详情页中播放视频的具体内容（可通过控制条控制视频的播放），如图 1-5 所示。

图 1-5　视频界面

【任务 1-9】“我”的界面

“我”的界面包含功能列表界面、单击用户图片可以进行用户的登录、注册、显示我的资料，单击设置可以在设置界面中进行清除缓存以及设置新闻字体大小，单击用户管理

可以进行修改密码、设置密保问题，单击关于 App 可以检查代码是否有更新及查看城市天气，如图 1-6 所示。

图 1-6　“我”的界面

1.3 本 章 小 结

本章整体介绍了东仔移动新闻客户端项目的功能、模块以及项目效果，读者只需在头脑中有个简单的了解即可，在接下来的章节中，会一一实现这些功能模块以及界面的设计。

1.4 习 题

1．Android 程序的真正入口是什么？

2．JSON 数据与 XML 数据各有哪些优缺点？

第 2 章　欢迎界面和主程序

学习目标

- 掌握欢迎界面的开发，能够独立制作欢迎界面。
- 掌握底部导航栏的开发，能够实现导航功能。

2.1　欢 迎 界 面

任务综述

在每个应用程序中欢迎界面都是必不可少的，它的主要作用是展示产品 Logo、检查程序完整性、检查程序的版本更新、加载广告页、做一些初始化操作等。本节将针对欢迎界面开发进行详细讲解。

【知识点】

- 布局文件的创建与设计。
- RelativeLavout 布局、TextView 控件。
- Timer 与 TimerTask。

【技能点】

- 实现 Android 项目的创建。
- 通过 Timer 实现界面延迟跳转。
- 通过 PackageManager 获取程序版本号。

图 2-1　欢迎界面

【任务 2-1】欢迎界面

【任务分析】

移动新闻客户端项目的欢迎界面效果如图 2-1 所示。

【任务实施】

（1）创建项目

首先创建一个工程，将其命名为 DGPTNetNews，指定包名为 com.dgpt.newnews。

（2）导入界面图片

将项目的 icon 图标 app-icon.png 导入 mipmap 文件夹的 mipmap-hdpi 中。mipmap 文件

夹通常用于存放应用程序的启动图标，它会根据不同手机分辨率对图标进行优化，其他图片资源要放到 drawable 文件夹中。将图片复制到 mipmap 文件夹时会弹出一个对话框，显示 mipmap-hdpi、mipmap-mdpi、mipmap-xhdpi、mipmap-xxhdpi、mipmap-xxxhdpi 5 个文件夹，按照分辨率不同选择合适的文件夹存放图片即可。

（3）创建欢迎界面

在程序中选中 com.dgpt.newnews 包，在该包下创建一个 activity 包，然后在 activity 包中创建一个 Empty Activity 类，名为 SplashActivity，并将布局文件名指定为 activity-splash.xml。具体代码请扫描下方二维码。

2-1-1

（4）修改清单文件

每个应用程序都会有属于自己的 icon 图标，同样新闻客户端项目也会使用自己的 icon 图标，因此需要在 AndroidManifest.xml 的<application>标签中修改 icon 属性、引入东仔移动新闻客户端图标，具体代码如下：

```
1  android:icon="@mipmap/app_icon"
```

项目创建后所有界面需要使用自定义的蓝色标题栏，因此需要在<application>标签中修改 theme 属性，去掉程序默认的标题栏，具体代码如下：

```
1  android:theme="@style/Theme.AppCompat.NoActionBar"
```

东仔移动新闻客户端启动时，首先进入的是欢迎界面 SplashActivity 而不是系统默认的 MainActivity，因此需要将欢迎界面指定为程序默认启动界面。在配置文件中将 MainActivity 的<intent-filter>标签以及标签内的所有内容剪切至 SplashActivity 所在<activity>标签中，具体代码如下：

```
1   <activity android:name="cn.dgpt.netnews.activity.SplashActivity">
2   <intent-filter>
3   <action android:name="android.intent.action.MAIN" />
4   <category android:name="android.intent.category.LAUNCHER" />
5   </intent-filter>
6   </activity>
```

【任务 2-2】欢迎界面逻辑代码

【任务分析】

欢迎界面主要展示产品 Logo 和版本信息，通常会在该界面停留一段时间之后自动跳转至其他界面，因此，需要在逻辑代码中设置欢迎界面暂停几秒（本项目为 3s）后再跳转，

并获取程序的版本号。

【任务实施】

（1）获取版本号

在 SplashActivity 中创建 init()方法，在该方法中获取 TextView 控件，通过 PackageManager（包管理器）获取程序版本号（版本号是 build.gradle 文件中的 versionName 值），并将其显示在 TextView 控件上。

（2）让界面延迟跳转

在 init()方法中使用 Timer 以及 TimerTask 类设置欢迎界面延迟 3s 再跳转到主界面（MainActivity 所对应的界面，此界面目前为空白页面）。具体代码请扫描下方二维码。

2-2-1

其中，第 23、24 行代码首先通过 PackageManager 的 getPackageInfo()方法获取 PackageInfo 对象，然后通过该对象的 versionName 属性获取程序的版本号，最后通过 setText()方法将获取到的版本号设置到 TextView 控件。具体代码如下：

```
22          //获取程序包信息
23      PackageInfo info= getPackageManager().getPackageInfo
(getPackageName(), 0);
24          tv_version.setText("V"+info.versionName);
```

其中，第 32～40 行代码的作用是让程序在欢迎界面停留 3s 后跳转。在此段代码中主要用到两个类，分别为 Timer 类和 TimerTask 类。其中，Timer 类是 JDK（Java 开发工具包）中提供的一个定时器工具，使用时会在主线程之外开启一个单独的线程执行指定任务，任务可以执行一次或多次；TimerTask 类是一个实现了 Runnable 接口的抽象类，同时代表一个可以被 Timer 执行的任务，因此跳转到主界面的任务代码写在 TimerTask 的 run()方法中。Timer 的 schedule()方法是任务调度方法，在 3s 之后调度 TimerTask 执行跳转操作，实现延迟跳转功能。具体代码如下：

```
32      TimerTask task=new TimerTask() {
33          @Override
34          public void run() {
35              Intent intent=new Intent(SplashActivity.this,MainActivity.
class);
36              startActivity(intent);
37              SplashActivity.this.finish();
38          }
39      };
40      timer.schedule(task, 3000);//设置这个task在延迟3s之后自动执行
```

2.2　主　程　序

任务综述

在每个应用程序中主程序都是必不可少的，它的主要作用是作为每个子功能的统一入口，作为每个二级功能的容器，在主程序中通过底部导航条可以进入不同的二级功能。本节将针对主程序开发进行详细讲解。

【知识点】

- 掌握底部导航栏的开发。
- FrameLayout 布局、LinearLayout 布局。
- FragmentTabHost 的使用。

【技能点】

- 自定义 FragmentTabHost。
- 使用 FragmentTabHost 的步骤。
- 初始化底部标签栏。

【任务 2-3】主界面布局

【任务分析】

移动新闻客户端项目的主界面效果如图 2-2 所示。

图 2-2　主界面

【任务实施】

（1）设置自定义 FragmentTabHost

Android 自带 FragmentTabHost 控件每次切换 Fragment，都会执行 Fragment 的 onCreateView 和 onDestroyView 方法，每次切换都会创建和销毁 Fragment 实例，这样每次切换回来时不能保持上次浏览的位置。为了让 FragmentTabHost 切换 Fragment 时保持原有浏览状态，需要修改原生的 FragmentTabHost。具体代码请扫描下方二维码。

2-3-1

根据 FragmentTabHost 的源代码可知，Fragment 是使用 detach 和 attach 切换的，只要把切换方式换为 hide 和 show 就行了（FragmentTabHost 的源代码中，不止 doTabChanged 方法中有

Fragment 的切换，需要把切换方式全部换了，因此更换了第 157、158、181、182、241、242、251、252 行代码）。具体代码如下：

```
157 //            ft.detach(info.fragment);
158               ft.hide(info.fragment);
181 //            ft.detach(tab.fragment);
182               ft.hide(tab.fragment);
241 //            ft.detach(mLastTab.fragment);
242               ft.hide(mLastTab.fragment);
251 //            ft.attach(newTab.fragment);
252               ft.show(newTab.fragment);
```

（2）导入自定义控件 FragmentTabHost

FragmentTabHost 包含两个内容：TabWidget 和 FrameLayout。标准的 FragmentTabHost 是 TabWidget 在上面，FrameLayout 在下面。如果想要制作 TabWidget 在下面的效果，可以在布局文件中手动添加一个 FrameLayout，并将 FragmentTabHost 中的 FrameLayout 的宽高设为 0 即可。不能省略掉 FragmentTabHost 中的 FrameLayout。具体代码请扫描下方二维码。

2-3-2

【注意】FragmentTabHost、TabWidget、FrameLayout 的 id 必须使用系统的 id 来为各组件指定 id 属性，否则将出现异常，FragmentTabHost 的 id 是 android:id/tabhost，TabWidget 的 id 是 android:id/tabs，FrameLayout 的 id 是 android:id/tabcontent，不能使用自定义的 id 即+id/。FrameLayout 用于放置导航条，TabWidget 被 Fragment 覆盖，在本例中我们没有使用 TabWidget，使用自己在第 10～16 行定义的 FrameLayout 放置 Fragment。在 FragmentTabHost 这一个布局中包裹着 FrameLayout（也可以换成其他布局，主要作用是存放 Fragment）和 TabWidget 控件。具体代码如下：

```
10    <FrameLayout
11        android:id="@+id/realtabcontent"
12        android:layout_width="fill_parent"
13        android:layout_height="0dip"
14        android:layout_weight="1"
15        android:background="@color/bg_color"
16        />
```

【任务 2-4】主程序逻辑代码

【任务分析】

主程序主要实现的是底部导航条，底部导航条使用任务 2-3 自定义的 FragmentTabHost，主要包含 3 个步骤。首先初始化 FragmentTabHost，其次新建 TabSpec 并将 TabSpec 添加进

FragmentTabHost，最后做一些基本设置，实现单击标签就会显示到对应的 Fragment，效果如图 2-3 所示。

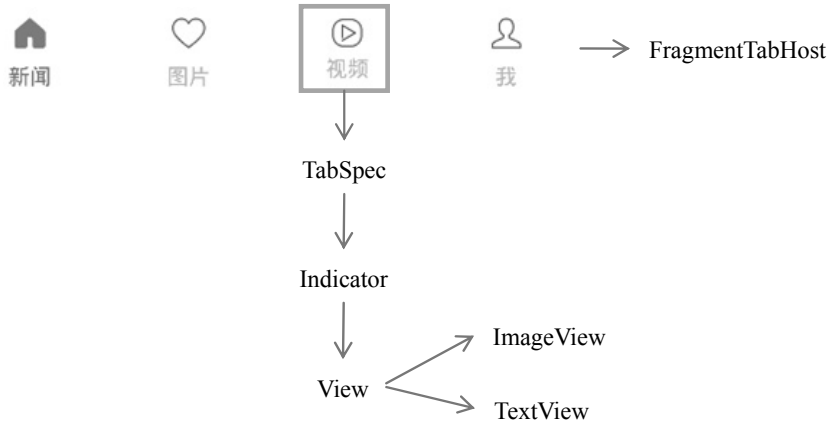

图 2-3　FragmentTabHost 结构图

【任务实施】

（1）初始化 FragmentTabHost

初始化 FragmentTabHost 主要在布局中找到 id 为 android.R.id.tabhost 的控件就是 FragmentTabHost；然后传入 3 个参数，第一个是上下文内容 Context，第二个是 FragmentManager，第三个是放置 fragment 的容器的 id，注意容器必须是 FrameLayout 类型，因为内部定义的类型就是这个，内部会根据 id 来进行初始化。

```
1  mTabHost = (FragmentTabHost)findViewById(android.R.id.tabhost);
2  MTabHost.setup(this, getSupportFragmentManager(), R.id.realtabcontent);
```

（2）新建 TabSpec 添加到 FragmentTabHost

因为底部 Tab 由一张图片和 tab 名称组成，所以首先需要定义每个底部标签的实体类和布局文件。具体代码请扫描下方二维码。

2-4-1　　2-4-2

每个 Tab 类包含名称、图标以及单击后要显示的 Fragment。

每个 Tab 标签包含一个 ImageView 显示图标和一个 TextView 显示名称。

在定义了 Tab 标签的实体类和标签后，在程序中需要初始化标签数据，主程序分为 4 个模块，分别是新闻纵横、图片中心、推荐视频和“我”。具体代码如下：

```
1  private List<BottomTab> mBottomTabs = new ArrayList<>(5);
2  //新闻标签
3  BottomTab bottomTab_news = new
```

```
4  BottomTab(NewsFragment.class,R.string.news_fragment,R.drawable.select_
icon_news);
5  //图片标签
6  BottomTab bottomTab_photo = new
7  BottomTab(PhotoFragment.class,R.string.photo_fragment,R.drawable.select_
icon_photo);
8  //视频标签
9  BottomTab bottomTab_video = new
10 BottomTab(VideoFragment.class,R.string.video_fragment,R.drawable.select_
icon_video);
11  //“我”标签
12  BottomTab bottomTab_about = new
13  BottomTab(MyFragment.class,R.string.about_fragment,R.drawable.select_
icon_about);
14  mBottomTabs.add(bottomTab_news);
15  mBottomTabs.add(bottomTab_photo);
16  mBottomTabs.add(bottomTab_video);
17  mBottomTabs.add(bottomTab_about);
```

每个 Tab 标签的图标我们使用了 android 的选择器，能够根据不同的选定状态来定义不同的实现效果，具体来讲有如下状态。

- android:drawable：单纯地放一个 drawable 资源。
- android:state_pressed：是否按下，如一个按钮触摸或者单击动作。
- android:state_focused：是否取得焦点，比如用户选择了一个文本框等。
- android:state_hovered：光标是否悬停，通常与 focused state 相同，它是 Android 4.0 的新特性。
- android:state_selected：是否被选中，它与 focused state 并不完全一样，如一个 ListView 的 item 被选中的时候，该 item 里面的各个子组件也就被选中了。
- android:state_checkable：是否能被 check。如 RadioButton 是可以被 check 的。
- android:state_checked：是否被 check 了，如一个 RadioButton 被 check 了。
- android:state_enabled：是否能够接受触摸或者单击事件，也就是说是否可用。具体代码如下：

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <selector xmlns:android="http://schemas.android.com/apk/res/android">
3  <!--非焦点状态-->
4  <item android:drawable="@drawable/b_newhome_tabbar"
5     android:state_focused="false" android:state_pressed="false"
6     android:state_ selected="false"/>
7  <item android:drawable="@drawable/b_newhome_tabbar_press"
8     android:state_focused="false" android:state_pressed="false"
9     android:state_selected="true"/>
10    <!--焦点状态-->
11  <item android:drawable="@drawable/b_newhome_tabbar_focus"
12    android:state_focused="true" android:state_pressed="false"
13    android:state_selected="false"/>
```

```
<item android:drawable="@drawable/b_newhome_tabbar_focus"
  android:state_focused="true" android:state_pressed="false"
  android:state_selected="true"/>
  <!--按下状态-->
<item android:drawable="@drawable/b_newhome_tabbar_press"
  android:state_pressed="true" android:state_selected="true"/>
<item android:drawable="@drawable/b_newhome_tabbar_press"
  android:state_pressed="true"/>
</selector>
```

在初始化标签数据后，需要设置底部 Tab 的图片和文字，因此写了一个类似于适配器的方法，把初始化的数据绑定到布局的相应控件上。具体代码如下：

```
private View buildIndicator(BottomTab bottomTab){
 View view = mInflater.inflate(R.layout.tab_indicator, null);//取得布局实例
 ImageView img = (ImageView) view.findViewById(R.id.icon_tab);//取得布局对象
 TextView text = (TextView) view.findViewById(R.id.txt_indicator);
 img.setBackgroundResource(bottomTab.getIcon()); //设置图标
 text.setText(bottomTab.getTitle());             //设置标题
 return  view;
}
```

在做完上述准备活动后，可以新建 TabSpec，并将 TabSpec 添加进 FragmentTabHost。具体代码如下：

```
for (BottomTab bottomTab : mBottomTabs){
  TabHost.TabSpec tabSpec = mTabHost.newTabSpec(getString(bottomTab.getTitle()));
  //对 Tab 按钮添加标记和图片
  tabSpec.setIndicator(buildIndicator(bottomTab));
  //将 Tab 按钮添加进 Tab 选项卡中，添加 Fragment
  mTabHost.addTab(tabSpec, bottomTab.getFragment(),null);

```

（3）其他一些设置

将 TabSpec 添加进 FragmentTabHost 后可以设置分割线以及默认显示那个 fragment，具体代码如下：

```
mTabHost.getTabWidget().setShowDividers(LinearLayout.SHOW_DIVIDER_NONE);
                                                    //设置没有分割线
mTabHost.setCurrentTab(0);                          //设置默认显示第一个
```

最后给出完整的主程序逻辑代码供大家参考。具体代码请扫描下方二维码。

2-4-3

2.3　常用工具类

任务综述

在日常软件开发中会使用工具类，常用工具类其实就是对于 String、Collection、I/O 等常用类的功能的扩展。比如 I/O 读写文件。其实大多数时候用户希望有一个文件路径，然后调用方法就可直接得到文件内容（字符串或者字节数组形式）。

如果没有工具类，那么每个读文件的地方都会有一段重复的代码。所以，用户肯定会将重复的部分提取出来。那么，提取出来放哪儿呀？要知道这个功能可是在任何类都能调用的。因此，开发出一些常用的工具类存放各种功能，以方便应用程序开发，提高开发效率，减轻程序员的工作量。本节将针对常用工具类开发进行详细讲解。

【知识点】

- 掌握底部导航栏的开发。
- FrameLayout 布局、LinearLayout 布局。
- FragmentTabHost 的使用。

【技能点】

- 自定义 FragmentTabHost。
- 使用 FragmentTabHost 的步骤。
- 初始化底部标签栏。

【任务 2-5】日志工具类

【任务分析】

安卓开发离不开记录 log 日志，因此封装了一份简单的日志工具类，具有设置日志总开关，是否写入文件，日志过滤器和自定义标签，锁定打印 log 的类、函数名及行号，初始化可以使用 init 函数也可以使用建造者模式等功能。

【任务实施】

Android 日志打印类 LogUtils，能够定位到类名、方法名以及出现错误的行数并保存日志文件，还能根据日志级别决定输出显示哪些信息。具体代码请扫描下方二维码。

2-5-1

【任务 2-6】Toast 工具类

【任务分析】

Android 项目中常用来提醒用户某个事件发生了的一种可视化组件就是 Toast，添加 Toast 的方法也很简单，使用 Toast.makeText()方法来设置显示的内容以及时间，切记使用 Toast.show()方法将 Tosat 显示出来。相信大部分同学在使用系统 Toast 的时候，都感觉不太尽如人意，因为系统 Toast 显示的位置比较固定，并且字体颜色等会跟随系统版本变化。

【任务实施】

Toast 在项目开发中主要问题有：Toast 在 activity 销毁后，还在显示；当 Toast 响应单击事件时，如果用户连续单击，就会导致多个 Toast 排队等待依次显示。具体代码请扫描下方二维码。

2-6-1

第 46～51 行，可以解决当 Toast 响应单击事件时，如果用户连续单击，就会导致多个 Toast 排队等待依次显示的问题，源代码中每次调用 makeText()方法都是通过 handler 发送异步任务来调用远程通知服务显示通知的，这样自然就造成了排队显示的现象。由此可想，如果在显示期间只修改同一个 Toast 对象的显示内容，这样显示的都是最后一次修改的内容效果，刚好 Tosat 也为用户提供了 Toast.setText(massage)方法来修改显示的内容，等待问题迎刃而解。具体代码如下：

```
46        if (toast == null) {
47            toast = Toast.makeText(context, spannableString, duration);
48        } else {
49            toast.setText(spannableString);
50            toast.setDuration(duration);
51        }
```

第 62～67 行，可以在 UI（用户界面）隐藏或者销毁前取消 Toast 显示，解决 Toast 在 activity 销毁后还在显示的问题。具体代码如下：

```
62    public static void cancel() {
63        if (toast != null) {
64            toast.cancel();
65            toast = null;
66        }
67    }
```

最后，我们还自定义了 Toast 显示的弹出框的形状样式，文件命名为 toast_frame_

style.xml。具体代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="1000dp"/>
    <solid android:color="@color/colorPrimaryDark"/>
    <stroke
        android:width="0.5dp"
        android:color="@color/colorAccent"/>
    <padding
         android:top="10dp"
         android:bottom="10dp"
         android:left="10dp"
         android:right="10dp"/>
</shape>
```

【任务 2-7】String 工具类

【任务分析】

StringUtils 是对 Java 中 String 类的增强和补充，简化开发。在代码中有时候要经常进行比较，确定一些条件满不满足，例如是否是空，是否是 null，等等，这时就需要一个功能强大的 String 工具类。

【任务实施】

自定义 String 工具类主要包含判空、取空格、校验字符串、判断两字符串是否相等、null 转为长度为 0 的字符串、返回字符串长度、日期格式、时间处理等需要在应用中用到的方法。具体代码请扫描下方二维码。

2-7-1

【任务 2-8】PrefUtils 工具类

【任务分析】

PrefUtils 是对 Android 中对 SharePreference 的封装，使用起来简单方便。在代码中主要进行对不同类型数据在 SharePreference 的按 key 存取。

【任务实施】

自定义 PrefUtils 工具类主要包括 getBoolean(Context ctx, String preferencesName, String key, **boolean** defValue)、setBoolean(Context ctx, String preferencesName, String key, **boolean**

value)等方法，方法的参数基本一致，降低了开发难度。具体代码请扫描下方二维码。

2-8-1

【任务 2-9】ListDataSave 工具类

【任务分析】

存储 List 数据到本地的方式有很多种，对于不想用 sqlite 或者其他方式，又或是数据量很少的话，不妨可以试下用 SharedPreferences 保存。由于 SharedPreferences 只能保存 Map 型的数据，因此必须要做其他转换才行。

用于保存各种 List 数据，最常见的莫过于 ListView、Gridviw 中的数据，支持类型有 List<String>、List<Map<String,Object>>、List<JavaBean>。

【任务实施】

ListDataSave 主要是用 Gson 把 List 转换成 JSON 类型，再利用 SharedPreferences 保存的，还可以把存在 SharedPreferences 数据取出，通过 JSON 解析转换为 List 对象数组。具体代码请扫描下方二维码。

2-9-1

【任务 2-10】ThreadManager 工具类

【任务分析】

ThreadManager 工具类主要用于管理线程池。线程池是预先创建线程的一种技术。线程池在还没有任务到来之前，创建一定数量的线程，加入空闲队列中，然后对这些资源进行复用，减少频繁创建和销毁对象的次数。

【任务实施】

在东仔移动新闻客户端中，经常频繁创建和销毁线程，耗资源、耗时间，而使用线程池可以节约资源、节约时间，同时有的线程执行任务的时间甚至比创建和销毁线程的时间还短。所以 ThreadManager 工具类使用较为频繁。具体代码请扫描下方二维码。

2-10-1

【任务 2-11】其他工具类

【任务分析】

其他工具类是一些较小的为方便开发而设计的一些自定义类，主要包括单位转换、文件写入和读取、MD5 加密解密、网络的判断、UI 线程处理的一些基本工具。

【任务实施】

（1）DensityUtils 工具类

DensityUtils 工具类是屏幕 px 与 dp 转换工具，能够根据手机的分辨率从 dp 单位转换成 px（像素），或 px（像素）单位转换成 dp，还能获取屏幕宽度与取屏幕高度，单位为 px（像素）。具体代码请扫描下方二维码。

2-11-1

（2）CommonUtils 工具类

CommonUtils 工具类的主要功能有随机返回一种颜色，获得当前日期以及获取 values 中的各种资源。具体代码请扫描下方二维码。

2-11-2

（3）IOUtils 工具类

在开发过程中，经常遇到从流中解析数据，或者把数据写入流中，或者输入流转换为输出流，而且最后还要进行流的关闭，原始 JDK 自带的方法写起来太复杂，还要注意各种异常，IOUtils 工具类可以让用户快速、方便地操作流来读写文件。具体代码请扫描下方二维码。

2-11-3

（4）NetWorkUtil 工具类

网络操作是很多操作的基础，因此专门写一个工具类来判断网络是否连接，Wi-Fi 是否打开，Wi-Fi 是否连接，GPS 是否打开，获取连接网络类型（3G/4G/Wi-Fi，包含运营商信

息）。具体代码请扫描下方二维码。

2-11-4

（5）UIUtils 工具类

UIUtils 工具类主要是处理和界面显示层有关的公共方法，含 Context 获取，资源获取，判断是否运行在主线程，运行在主线程修改 UI。具体代码请扫描下方二维码。

2-11-5

（6）MD5Encoder 工具类

MD5Encoder 工具类是 MD5 加密工具类，主要包含对字符串进行 MD5 加密，将文件名转为 MD5，以便于写入磁盘，字节数据转换为十六进制。具体代码请扫描下方二维码。

2-11-6

2.4　本 章 小 结

本章主要讲解了东仔移动新闻客户端项目的欢迎界面、主程序以及常用工具类。这 3 个功能模块是本项目最简单、最基础的部分，因此首先讲解，以便读者熟悉项目的开发流程以及开发步骤，方便后续学习。

2.5　习　　题

1．Android 中有几种数据存储方式？每种方式有哪些特点？

2．为什么要对 ListView 控件进行优化？以及如何优化？