

《Android程序设计》

实验报告

班级： 101011901

学号： 191027

姓名： 杨乃宸

实验四 来口家乡团圆菜 Ver1.015

上机一：考察知识点完成创建Fragment、在Activity中添加Fragment、实战演练—仿美团菜单）

形式：单独完成

题目：

请按照以下要求完成操作，要求如下：

（1）创建Fragment；

（2）实现在Activity中添加Fragment；

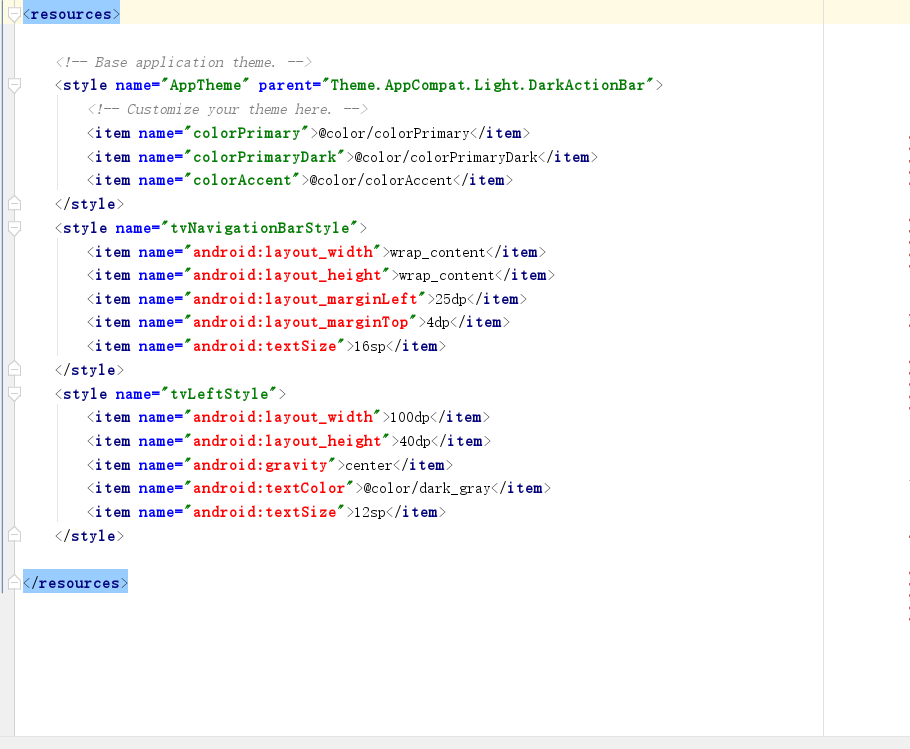
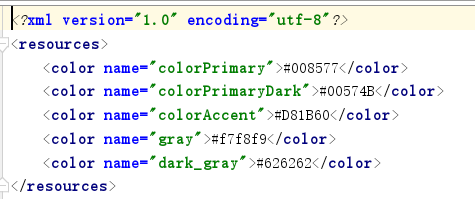
（3）实现一个仿美团菜单案例。

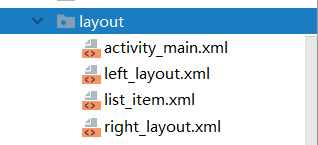
本小节我们通过仿美团外卖菜单的案例来学习如何在一个Activity中展示两个Fragment,并实现Activity与Fragment之间的通信功能。实现本案例的具体步骤如下所示：

搭建菜单界面布局  
 在仿美团外卖菜单的程序中只显示一个菜单界面，该界面主要包含3部分内容，分别为顶部的导航栏左侧的菜单栏和右侧的菜单列表。其中，顶部的导航栏中包含“点菜”“评价”“商家”和“好友拼单”文本信息，左侧的菜单栏中包含“推荐”和“进店必买”文本信息，右侧的菜单列表中包含菜品名称、菜品月售信息和菜品价格等信息。菜单界面如图所示

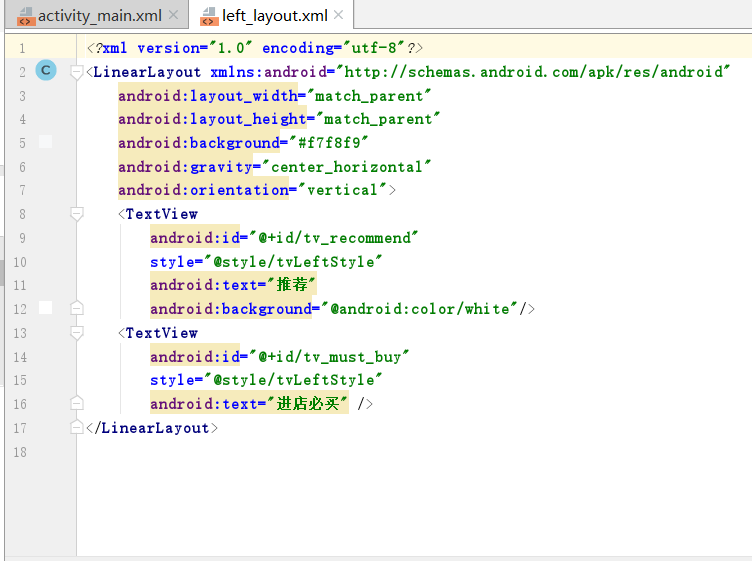


1. 创建程序  
   创建一个名为Menu的应用程序，指定包名为cn.itcast..menu。  
    (2)导入界面图片  
    将菜单界面需要的图片导入程序中创建的drawable-hdpi文件夹中。  
    (3)创健文本样式  
   由于菜单界面的顶部导航栏中“点菜”“评价”和“商家”文本信息的样式是一样的，所以需要将这些相同的样式代码抽取出来单独放在创建的样式tvNavigationBarStyle中。菜单界面中左侧菜单栏中的“推荐”和“进店必买”文本信息的样式也是一样的，也需要将这些样式代码抽取出来单独放在创建  
   中。在程序的res/values/styles.xml文件中，具体代码如下：

  
(4)添加浅灰色与深灰色颜色值  
由于菜单界面顶部导航栏的背景与左侧菜单栏的背景都是浅灰色的，同时顶部导航栏中的文字“评价”和“商家”是深灰色的，左侧菜单栏中的文字“推荐”与“进店必买”也是深灰色的，所以需要在程序的res/values/colors.xml文件中分别添加一个名为gray和dark\_gray的颜色，这2个颜色的值分别为（浅灰色）  
和（深灰色），具体代码如下：  
  
(5)放置界面控件  
布局文件布局代码参考如下:



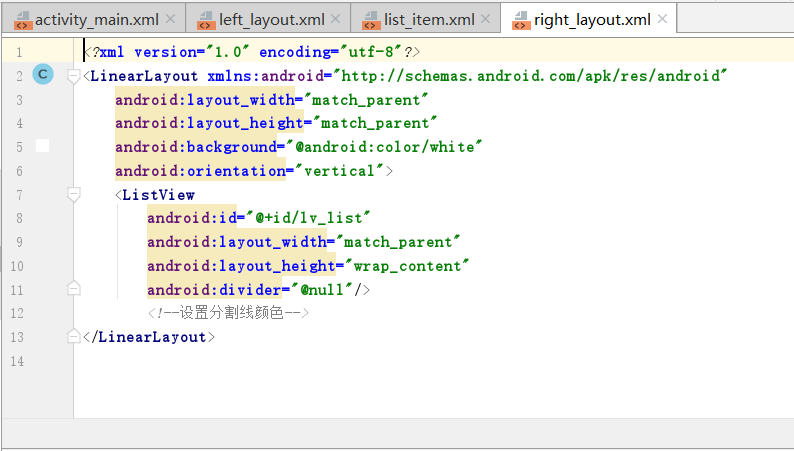
1.   
   （6）搭建左侧菜单栏界面布局  
   在Menu程序的res/layout文件夹中创建一个left\_layout.xml文件，在该文件中通过代码来搭建左侧菜单栏界面布局。在left\_layout..xml文件中，放置2个TextView控件分别用于显示左侧菜单栏中的“推荐”与“进店必买”文本信息。



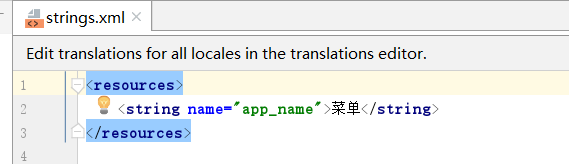
（7）搭建菜单列表界面的条目布局  
在Menu程序的res/layout文件夹中创建一个list\_item.xml文件，在该文件中通过代码来搭建菜单列表的条目布局。在list\_item.xml文件中，放置1个ImageView控件用于显示菜品的图片；放置3个TextView控件分别用于显示菜品名称、菜品月售信息和菜品价格。



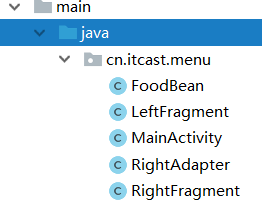
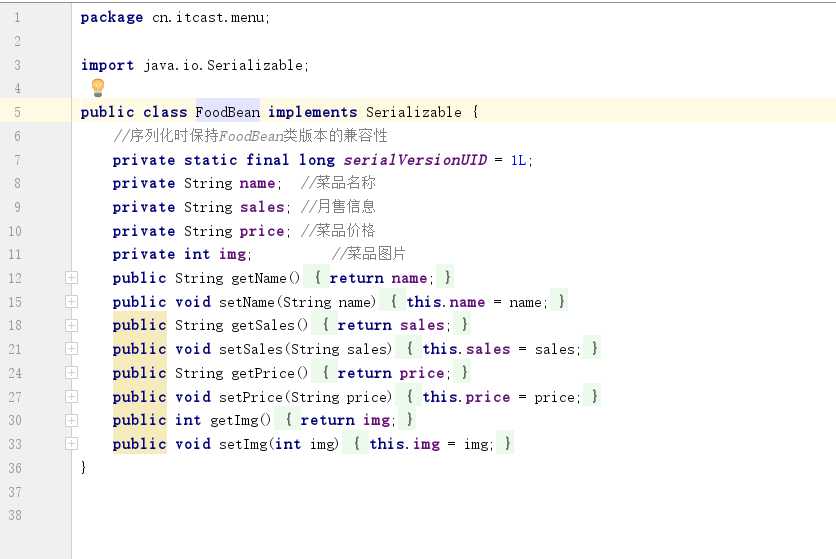
（8）搭建右侧菜单列表界面布局  
在Menu程序的res/layout文件夹中创建一个right\_layout.xml文件，在该文件中通过代码来搭建右侧菜单列表界面布局。在right\_layout.xml文件中，放置1个ListView控件用于显示菜单列表信息。

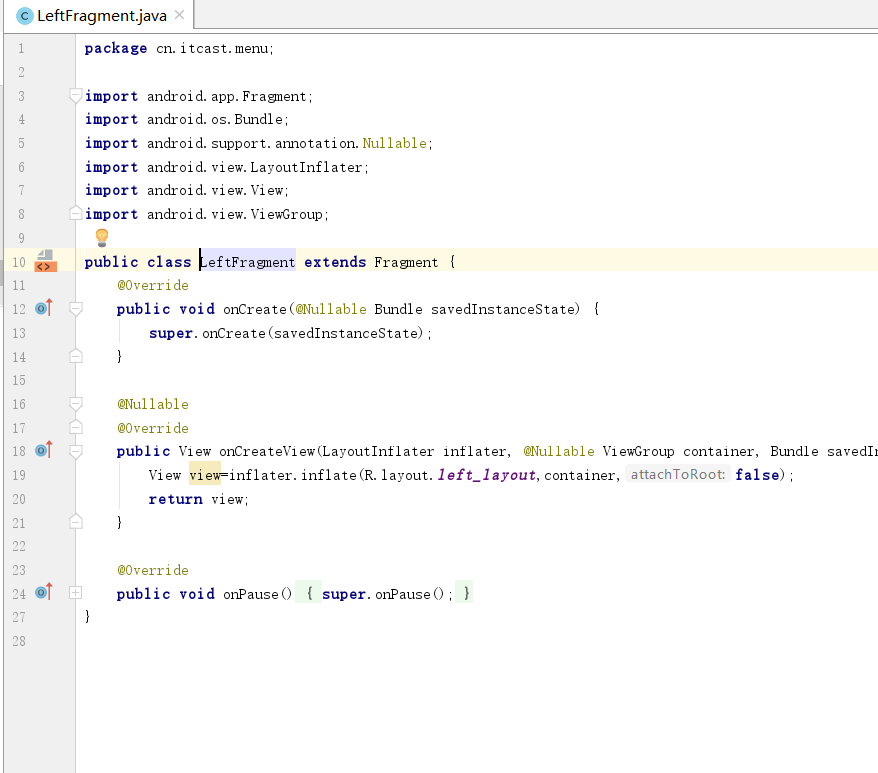


(9)修改默认标题栏的名称  
由于仿美团外卖菜单程序的名称为“Mem”,所以程序中默认标题栏的名称就为“Memu”。为了将默认标题栏的名称修改为“菜单”，我们需要修改res\_values文件夹中的strings.xml文件，在该文件中找到属性name的值为app\_name的标签，将该标签中的值设置为“菜单”。

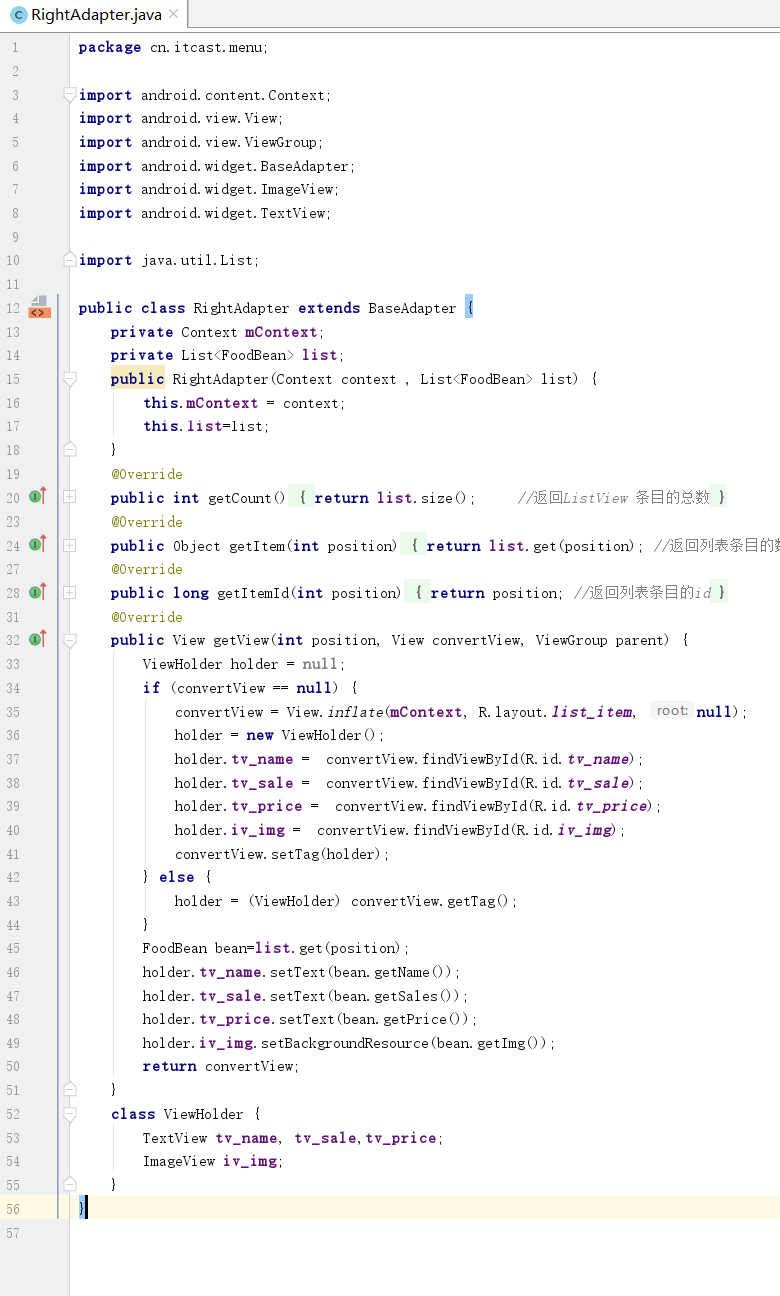


1. 实现菜单界面功能  
   在仿美团外实菜单程序中，当点击菜单界面左侧桌单档中的“推撑成“进店必买”文本时，界面右侧会显示对应的菜单列表，每个菜单列表都是由多个菜单条目组成的，每个菜单条目中需要显示菜品的图片、菜品的月售信息和菜品的价格。实现菜单界面功能的具体步骤如下所示。

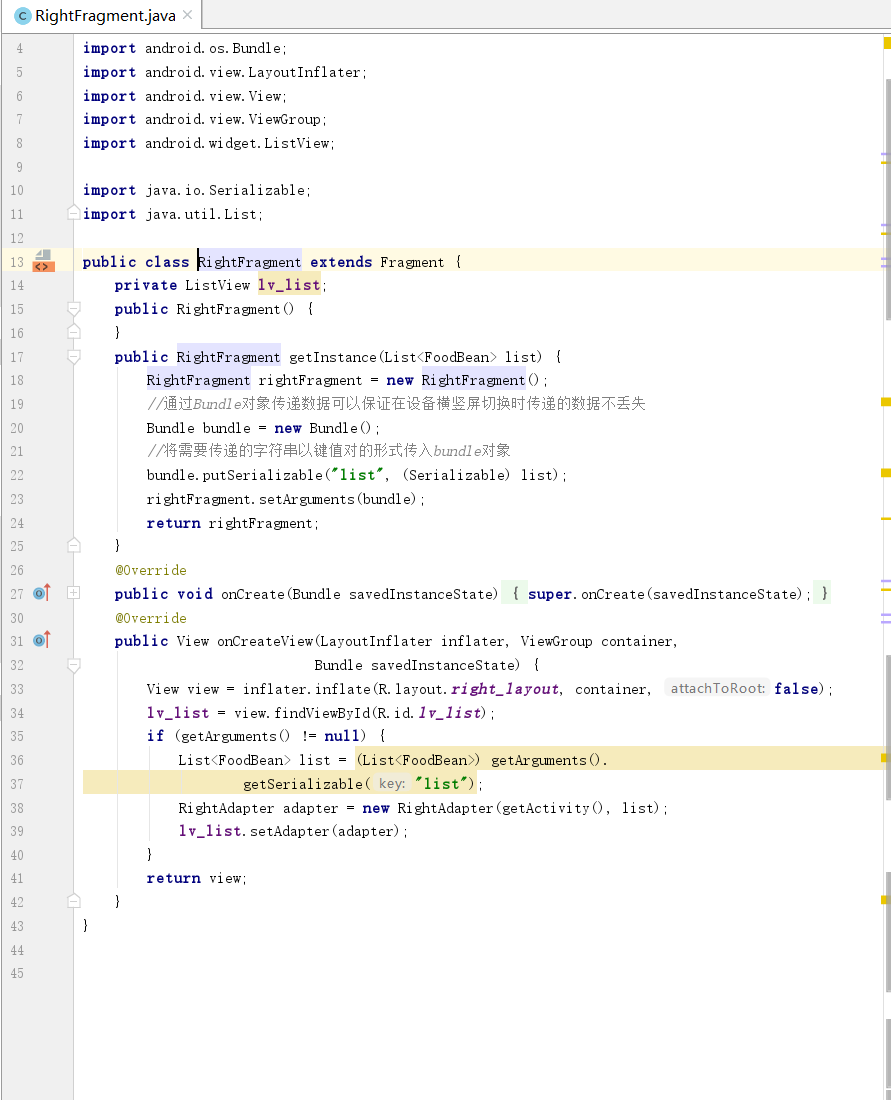
  
(1)封装菜品信息的实体类  
  
(2)加载左侧菜单栏界面布局



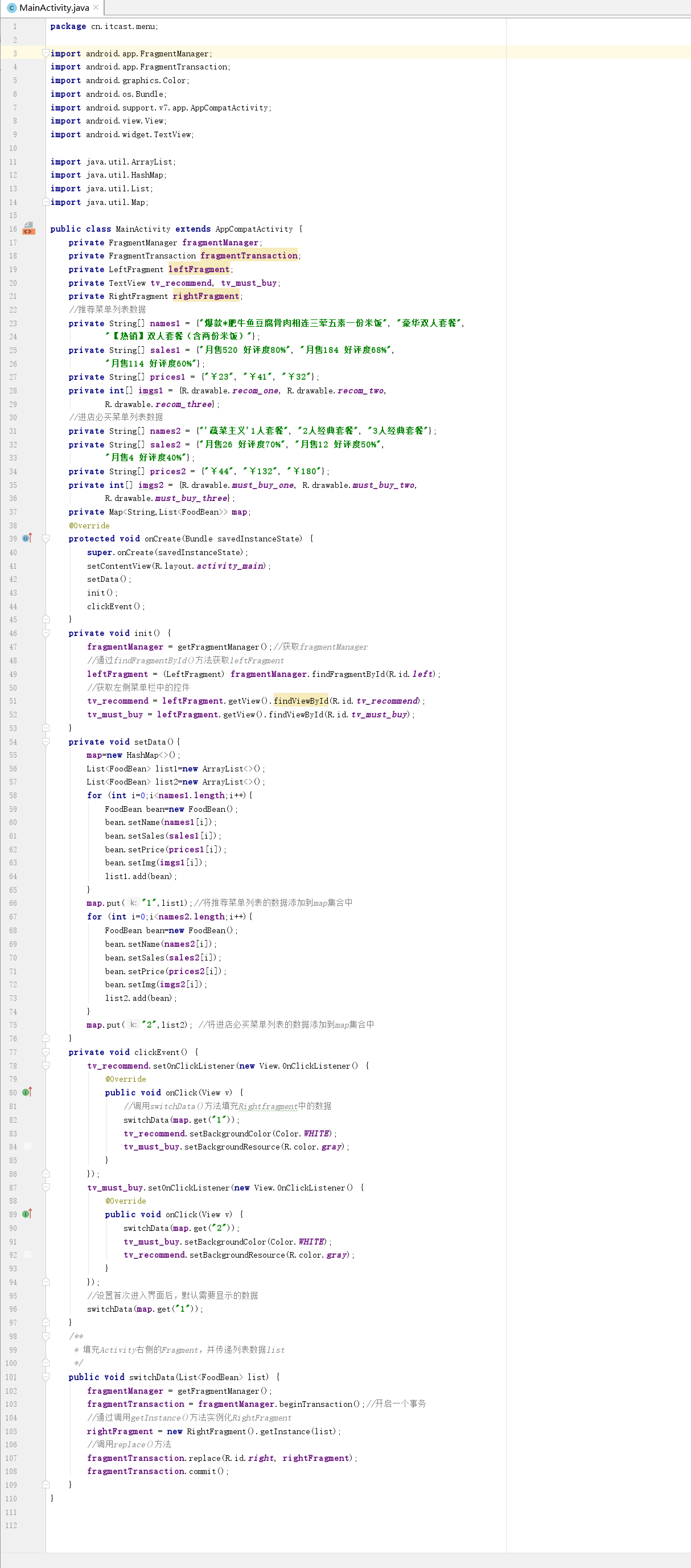
1. 编写右侧菜单列表的数据适配器  
   由于菜单界面中的右侧菜单列表是用ListView控件展示的，所以我们需要创建一个数据适配器RightAdapter对ListView控件进行数据适配。



(4)加载右侧菜单列表界面布局，通过创建Fragment来显示右侧菜单列表的信  
息。



1. 实现显示菜单的效果  
   由于仿美团外卖菜单的界面中需要显示推荐菜单列表与进店必买菜单列表信息，所以我们需要在MainActivit中定义init方法、setData方法、clickEven方法和switchDat方法，这些方法分别用于初始化界面控件、设置界面数据、实现界面控件的点击事件和切换右侧菜单列表的数据。

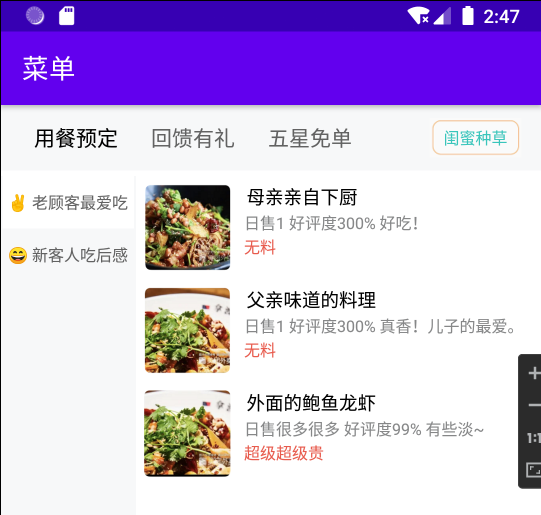


## 程序源代码

1. package cn.itcast.menu;  
     
   import androidx.appcompat.app.AppCompatActivity;  
   import androidx.fragment.app.FragmentManager;  
   import androidx.fragment.app.FragmentTransaction;  
     
   import android.graphics.Color;  
   import android.os.Bundle;  
   import android.view.View;  
   import android.widget.TextView;  
     
   import java.util.ArrayList;  
   import java.util.HashMap;  
   import java.util.List;  
   import java.util.Map;  
     
   public class MainActivity extends AppCompatActivity {  
    private FragmentManager fragmentManager;  
    private FragmentTransaction fragmentTransaction;  
    private LeftFragment leftFragment;  
    private TextView tv\_recommend, tv\_must\_buy;  
    private RightFragment rightFragment;  
    private String[] names1 = {"饭α 一家三口", "菜β 其乐融融", "肉γ 铁锅炖自己"};  
    private String[] sales1 = {"月售xxx 好评度xx%", "月售yyy 好评度yy%", "月售zzz 好评度zz%"};  
    private String[] prices1 = {"￥12", "$23", "￥34"};  
    private int[] imgs1 = {R.drawable.*recom\_one*, R.drawable.*recom\_two*, R.drawable.*recom\_three*};  
    private String[] names2 = {"母亲亲自下厨", "父亲味道的料理", "外面的鲍鱼龙虾"};  
    private String[] sales2 = {"日售1 好评度300% 好吃！", "日售1 好评度300% 真香！儿子的最爱。", "日售很多很多 好评度99% 有些淡~ "};  
    private String[] prices2 = {"无料", "无料", "超级超级贵"};  
    private int[] imgs2 = {R.drawable.*must\_buy\_one*, R.drawable.*must\_buy\_two*, R.drawable.*must\_buy\_three*};  
    private Map<String, List<FoodBean>> map;  
     
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.*activity\_main*);  
    setData();  
    init();  
    clickEvent();  
    }  
     
    private void init() {  
    fragmentManager = getSupportFragmentManager();  
    leftFragment = (LeftFragment) fragmentManager.findFragmentById(R.id.*left*);  
    tv\_recommend = leftFragment.getView().findViewById(R.id.*tv\_recommend*);  
    tv\_must\_buy = leftFragment.getView().findViewById(R.id.*tv\_must\_buy*);  
    }  
     
    private void setData() {  
    map = new HashMap<>();  
    List<FoodBean> list1 = new ArrayList<>();  
    List<FoodBean> list2 = new ArrayList<>();  
    for (int i = 0; i < names1.length;  
    i++) {  
    FoodBean bean = new FoodBean();  
    bean.setName(names1[i]);  
    bean.setSale(sales1[i]);  
    bean.setPrice(prices1[i]);  
    bean.setImg(imgs1[i]);  
    list1.add(bean);  
    }  
    map.put("1", list2);  
    for (int i = 0; i < names2.length;  
    i++) {  
    FoodBean bean = new FoodBean();  
    bean.setName(names2[i]);  
    bean.setSale(sales2[i]);  
    bean.setPrice(prices2[i]);  
    bean.setImg(imgs2[i]);  
    list2.add(bean);  
    }  
    map.put("2", list2);  
    }  
     
    private void clickEvent() {  
    tv\_recommend.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
    switchData(map.get("1"));  
    tv\_recommend.setBackgroundColor(Color.*WHITE*);  
    tv\_must\_buy.setBackgroundResource(R.color.*gray*);  
    }  
    });  
    tv\_recommend.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
    switchData(map.get("2"));  
    tv\_must\_buy.setBackgroundColor(Color.*WHITE*);  
    tv\_recommend.setBackgroundResource(R.color.*gray*);  
    }  
    });  
    switchData(map.get("1"));  
    }  
     
    public void switchData(List<FoodBean> list) {  
    fragmentManager = getSupportFragmentManager();  
    fragmentTransaction = fragmentManager.beginTransaction();  
    rightFragment = new RightFragment().getInstance(list);  
    fragmentTransaction.replace(R.id.*right*, rightFragment);  
    fragmentTransaction.commit();  
    }  
   }



## 二、 运行结果界面



## 三、遇到的问题总结

改bug时没注意记 对比了各厂的云虚拟机 想起来的说说：

重装了两次AS和三次雷电，等清清内存我再两手准备多下俩备用模拟器

!! 校园网有时候连虚拟机要设置一下代理配置原生UA，不然会出现很多复杂要求的状况 卡在94%和广告的UI上打不开了 例如：远程主机强迫关闭了一个现有的连接。(qemu-system-i386.exe:37316):GLib-ERROR \*\*

c:\users\joshuaduong\desktop\emu-prebuilts-x64\src\glib-2.38

2\glib\gmem.c:175:failed to allocate 69632 bytes

?? 用了<fragment> 视图就没了

?? 有些自动补全在我不知情情况下进行，调试发现一口气把爆红全修好了

!! 颜色值的#千万不能忘

?? Layout有height常有width（一般来说）

?? @Override 有时候后补上和按顺序先写产生的效果（或者说背后进行的机制）不一样

!! 导包不同之后的操作就都有点区别了

比如说新版提倡的本地androidx关键词系列前缀下的new的fm用的get了SupportFragmentManager()的方法 不是FragmentManager

!! Java和c++的表达式有些不同 只能是new、赋值(返回值)/构造、自增自减和调用。没返回值赋值语句不能再次算作表达式

AS修改运行成功过但没完全运行所有想要效果的项目时闪退后可能计算机引起关机、我搞了个word自动保存的加载项

改hosts的时候生成的新程序，虚拟机检测可能加载不出来，要重开AS，清理内存。关闭任务管理的adb相关线程。后续使用必要的程序已经build（要先试好gradle版本）好了，关掉后开启虚拟机也不用再次重启AS了

Win内置的一些东西仅仅是能用，为了方便虚拟机运行，我进BIOS开了VT。甚至为它更新了我好久都不想动的Win10全套N卡 A卡驱动

安卓容错度高，也就意味着肯定能弄出一个东西来，软件有问题每次跑都不怕。但这也就说明你想要的如果出错了比起其它开发不会更加明确的报错，没能力改出来就只能不要了.有利有弊吧。

我什么也没做，这是报告已经写好重新打开检查代码时自动出的问题。无语子了姐妹们，虽然不知道我又好像改变了什么，又好使了

Android resource linking failed

遇到困难别退缩，推荐老师好文

[(110条消息) Android resource linking failed （全网最详细，最全面的解决方案）\_ScottePerk的博客-CSDN博客\_android资源链接失败](https://blog.csdn.net/ScottePerk/article/details/124405230)