# 综合实验2 《小吃APP》页面架构实现

## 一、实验简介

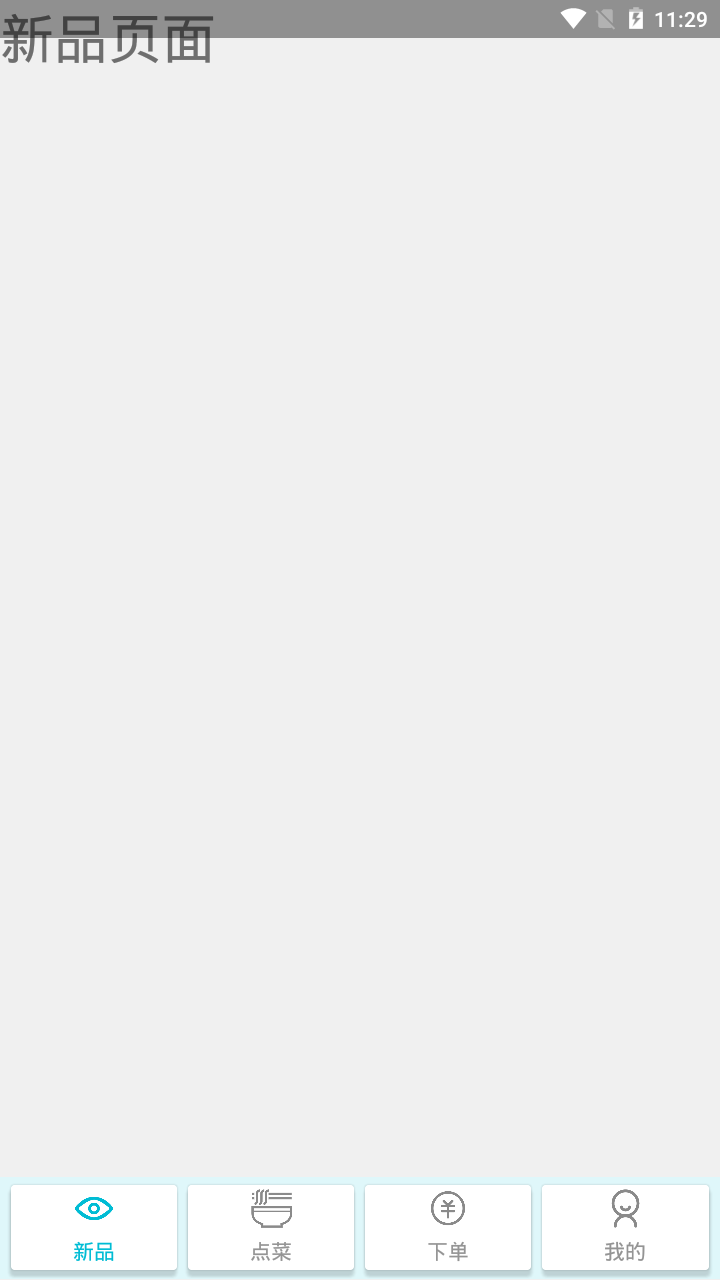
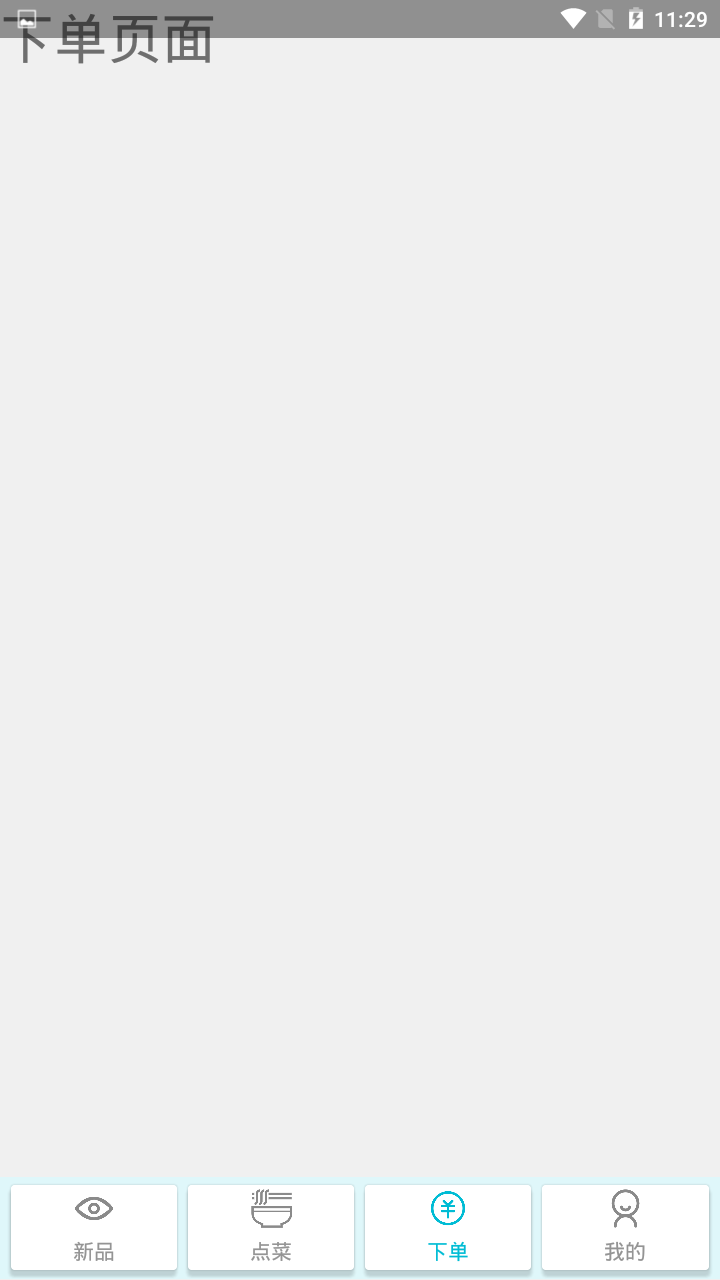
本次实验将在上次实验的界面基础上实现其功能逻辑，其主要利用Fragment+ViewPager2技术实现左右滑动屏幕翻页效果；实现点击底部4个导航选项切换或加载对应的页面；实现点击底部4个选项同时导航颜色做相应变化。

## 二、实验目标

* 掌握Fragment + ViewPager2作为页面框架技术
* 了解《小吃APP》的四个页面

## 实验操作步骤

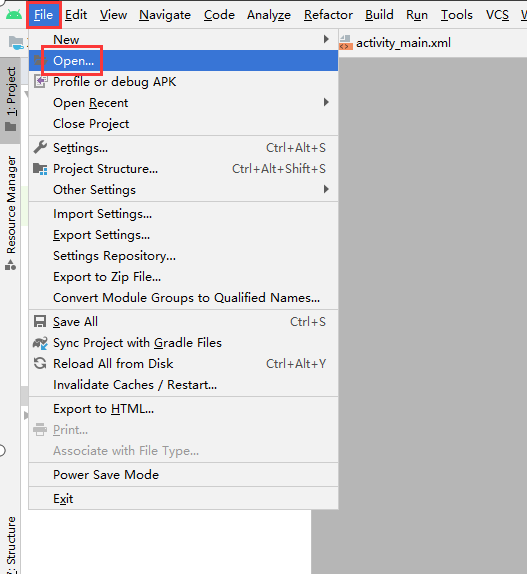
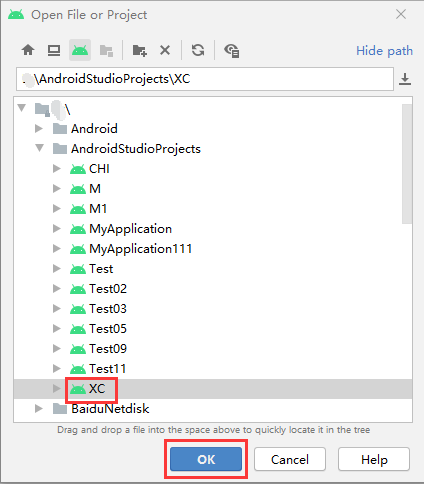
### 实现效果

### 实现步骤

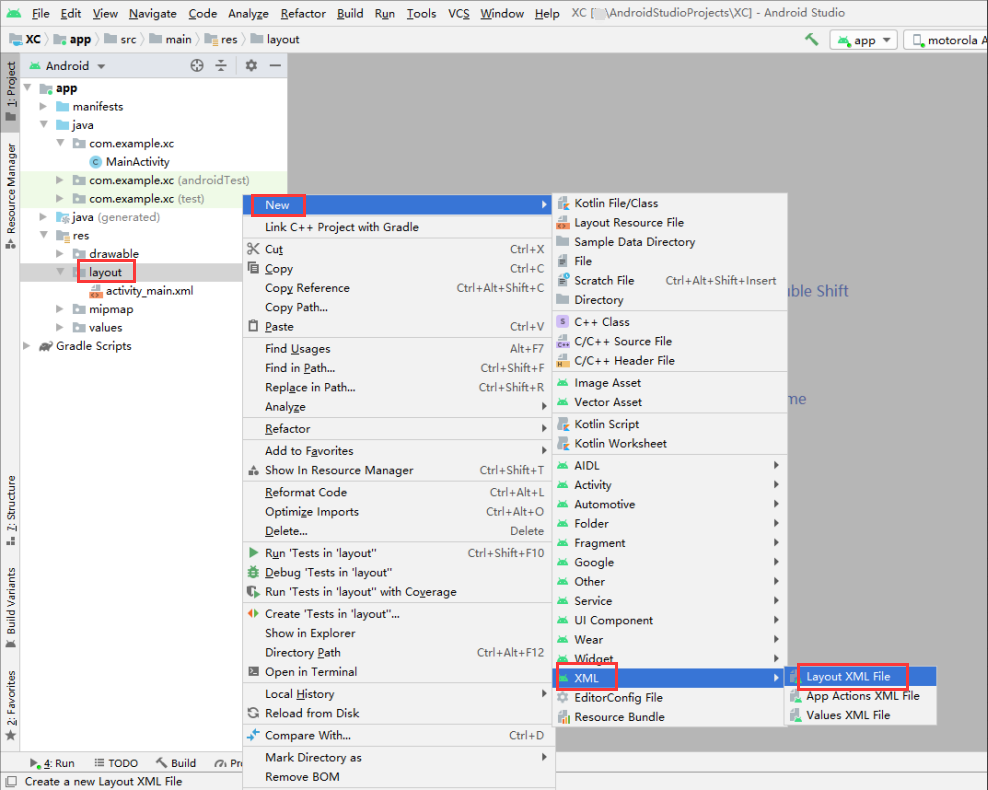
#### 第1步：打开综合实验1开发的【XC】项目；

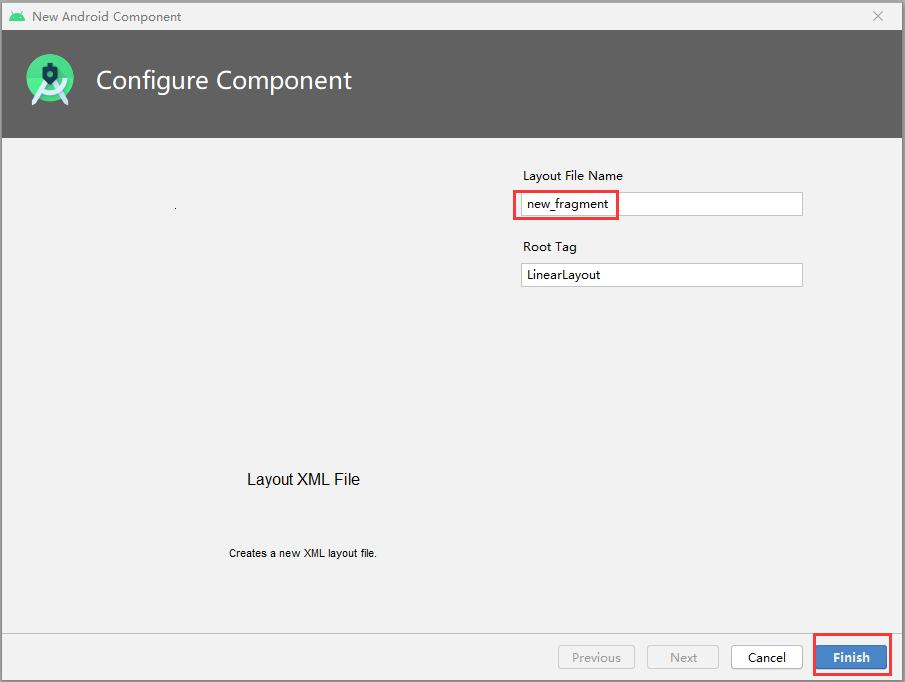
点击Android Studio工具左上角的【File】->【Open】->在弹出的界面里选择自己存放综合实验1项目【XC】的位置，选择【XC】，再点击【OK】即可。

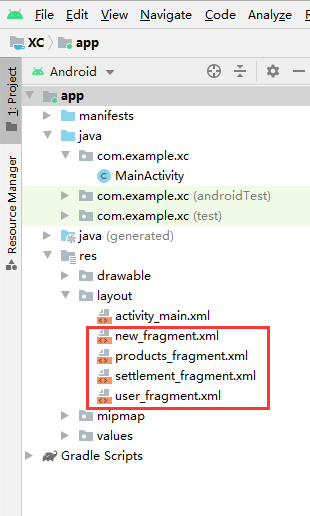
#### 第2步：创建四个选项布局文件

鼠标点击<app\res\layout\>目录，依次选择【New】->【XML】->【Layout XML File】->在弹出的窗口里填入布局文件名【new\_fragment】，再点击【Finish】，即会自动创建一个名为【new\_fragment.xml】的布局文件。





再以同样的方式创建3个名为：【products\_fragment.xml】、【settlement\_fragment.xml】、【user\_fragment.xml】的布局文件，创建完后的效果如下图所示，每个文件对应一个选项的界面。



#### 第3步：给布局文件添加TextView控件

以拖拽形式或手动编写形式插入下面TextView控件代码到上一步创建的四个布局文件里，并把android:text的属性值分别改为：新品页面、点菜页面、下单页面、我的页面。其效果为：让四个页面分别上显示对应文字。

<TextView

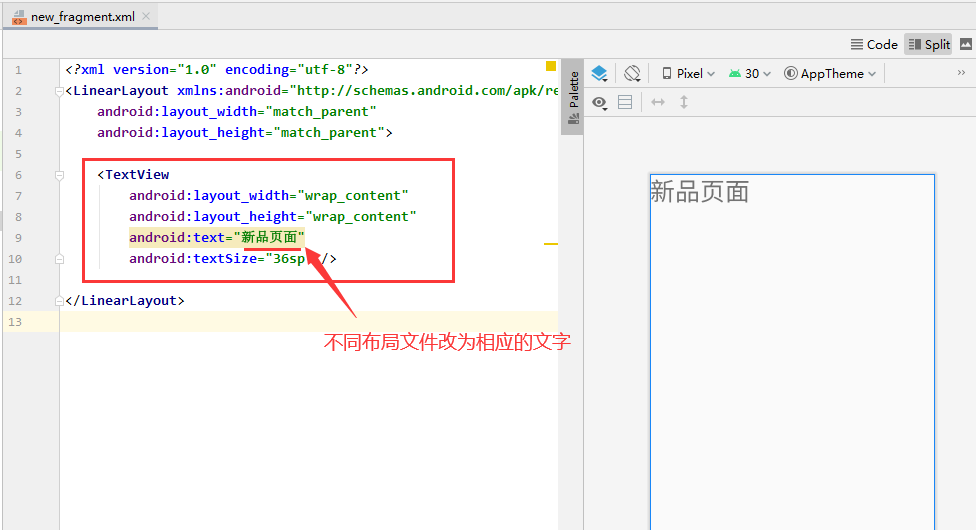
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="新品页面"

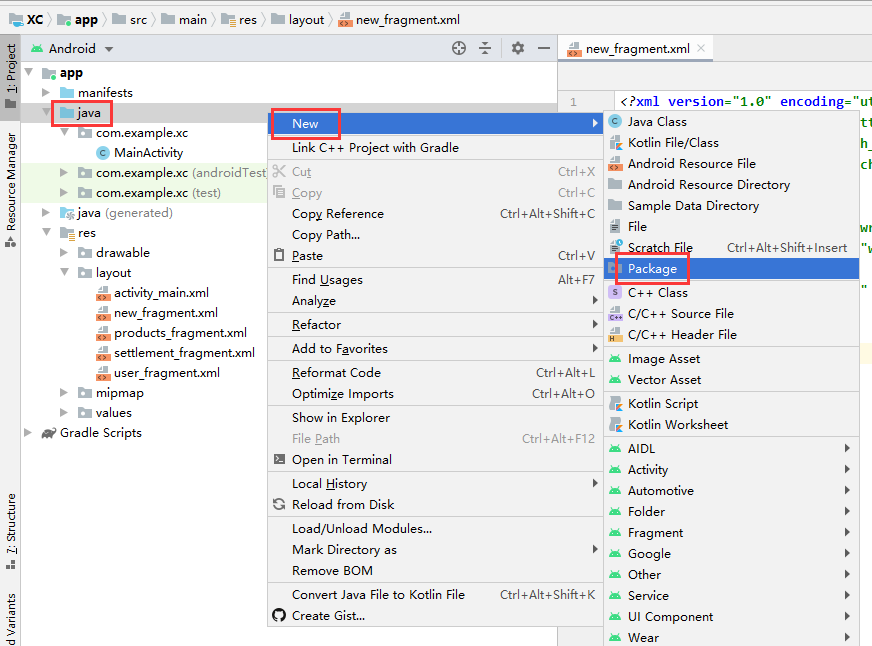
android:textSize="36sp" />

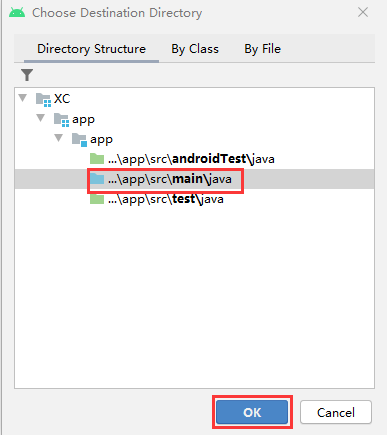
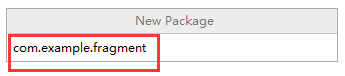
上述代码插入的具体位置和效果如下图所示



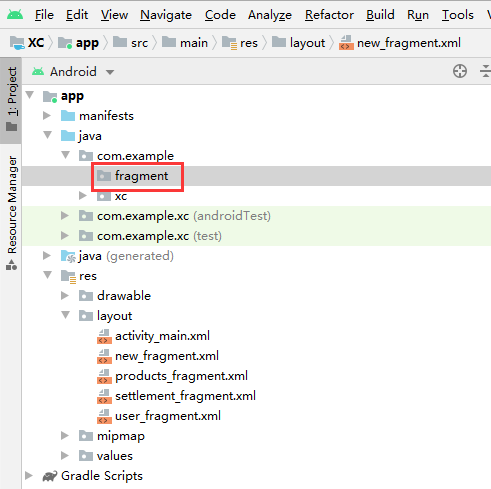
#### 第4步：新建fragment包

鼠标右键点击<app\java\>目录，依次选择【New】->【Packge】->在弹出的窗口里选择【...\app\src\main\java】，再点击【OK】->在弹出框里填入【com.example.fragment】，回车即可。



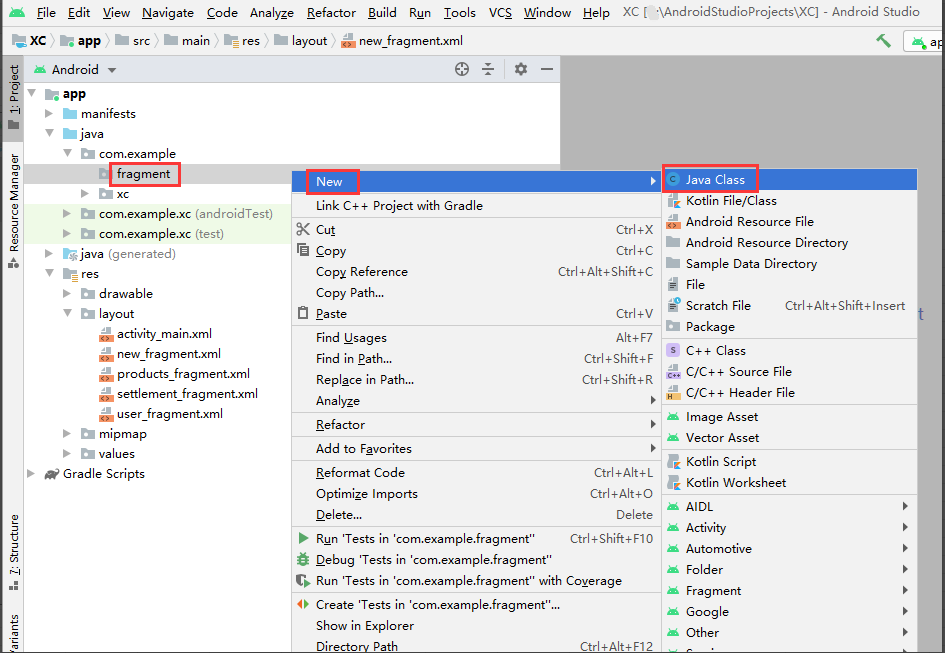
 

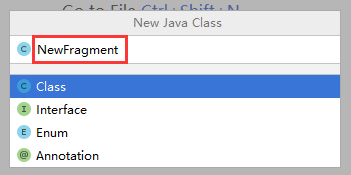
创建成功后的效果如下图所示



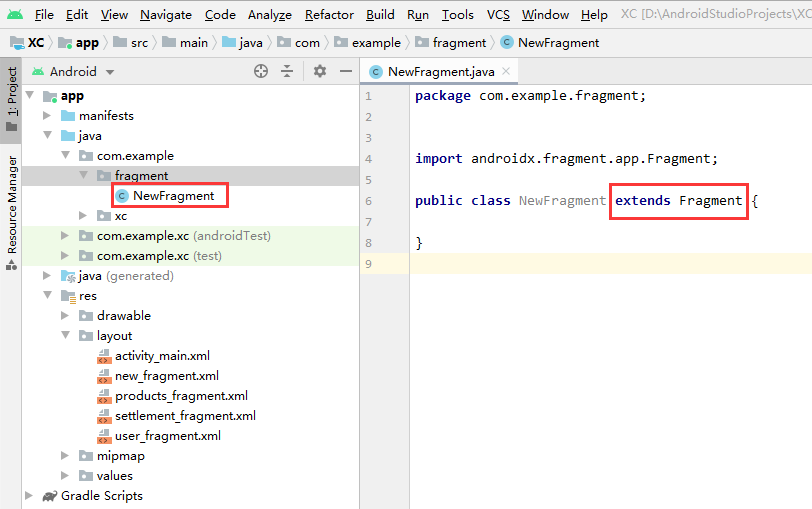
#### 第5步：创建四个选项Fragment类

鼠标右键点击上一步新建的【fragment】目录，依次选择【New】->【Java Class】->在弹出框里填入类名【NewFragment】，回车即可。





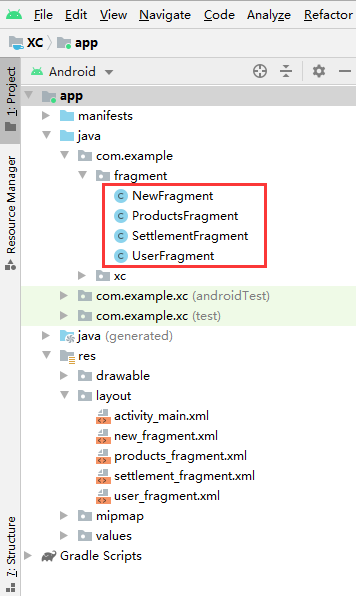
打开的创建的【NewFragment】类，让其继承Fragment类，如下图所示。



上述代码在编写的时候需对Fragment类导包，需要导入

import androidx.fragment.app.Fragment;

然后再重复上面的步骤，创建3个类名为：【ProductsFragment】、【SettlementFragment】、【UserFragment】的Fragment类，创建四个Fragment类的作用为：用于编写四个选项的功能逻辑代码.创建后的效果如下图所示。



#### 第6步：实现选项Fragment类和布局文件关联

在上一步创建的四个Fragment类里，插入如下四段代码，其作用为：关联第2步创建的四个选项布局文件。

【NewFragment】类里要插入的如下

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view=inflater.inflate(R.layout.new\_fragment,container,false); //关联布局文件

return view;

}

【ProducstFragment】类里要插入的代码如下

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view=inflater.inflate(R.layout.products\_fragment,container,false); //关联布局文件

return view;

}

【SettlementFragment】类里要插入的代码如下

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view=inflater.inflate(R.layout.settlement\_fragment,container,false); //关联布局文件

return view;

}

【UserFragment】类里要插入的代码如下

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view=inflater.inflate(R.layout.user\_fragment,container,false); //关联布局文件

return view;

}

上述代码插入的具体位置如下图所示



上述代码在编写的时候需要导入下面的包

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import com.example.xc.R;

#### 第7步：重写onResume方法

把下面的代码插入到四个Fragment类里，其作用是：每次显示选项页面时，如果要刷新数据，可以在对应Fragment类的onResume()方法里编写。

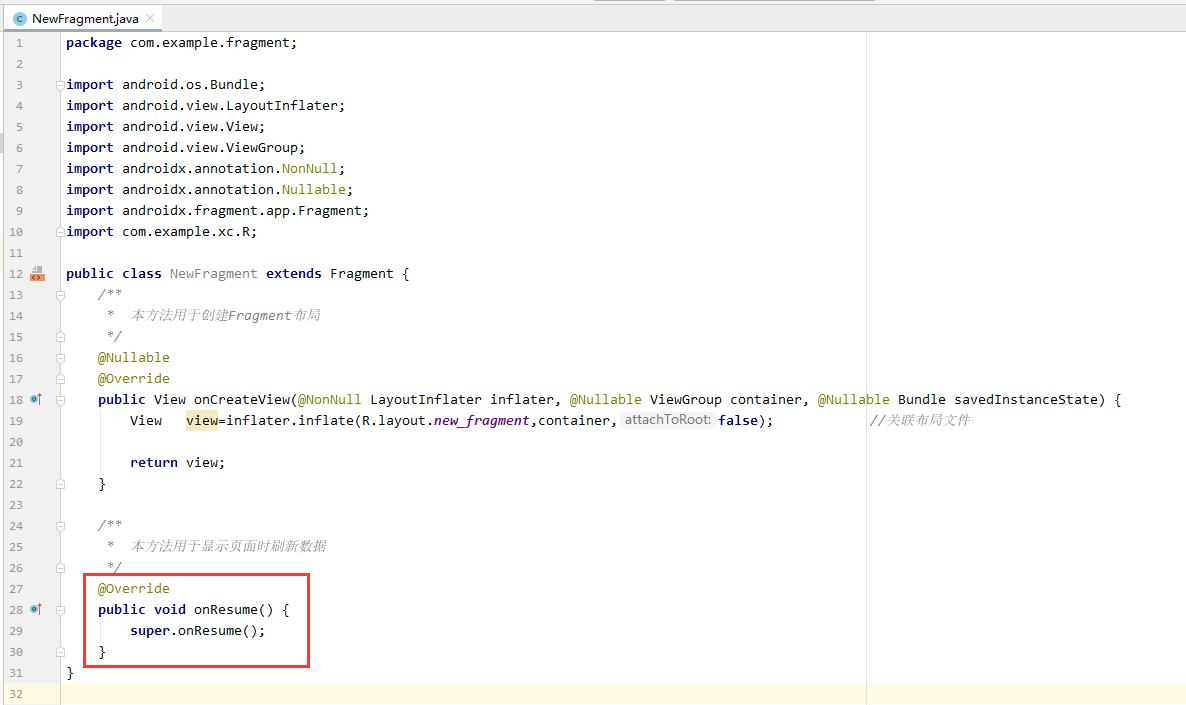
@Override

public void onResume() {

super.onResume();

}

上述代码插入的具体位置如下图所示



#### 第8步：定义initFragment方法

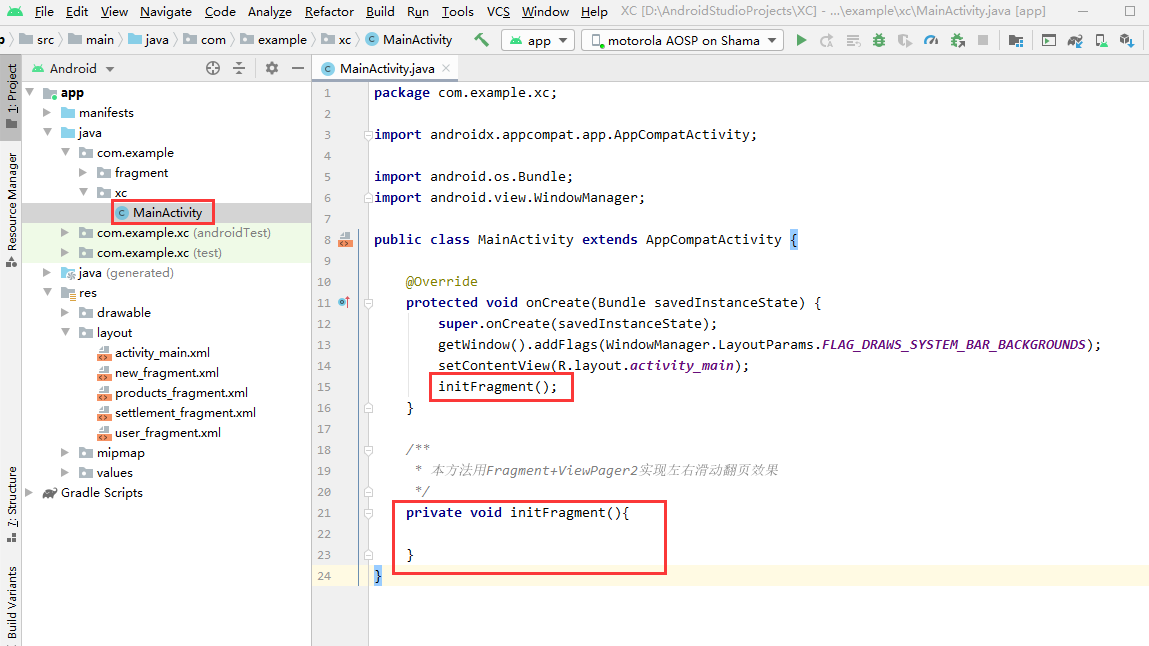
在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面的两段代码，其作用为：定义名为initFragment的方法，并调用，用于编写Fragment+ViewPager2左右滑动翻页效果的代码。

initFragment();

private void initFragment(){

}

上述代码插入的具体位置如下图所示

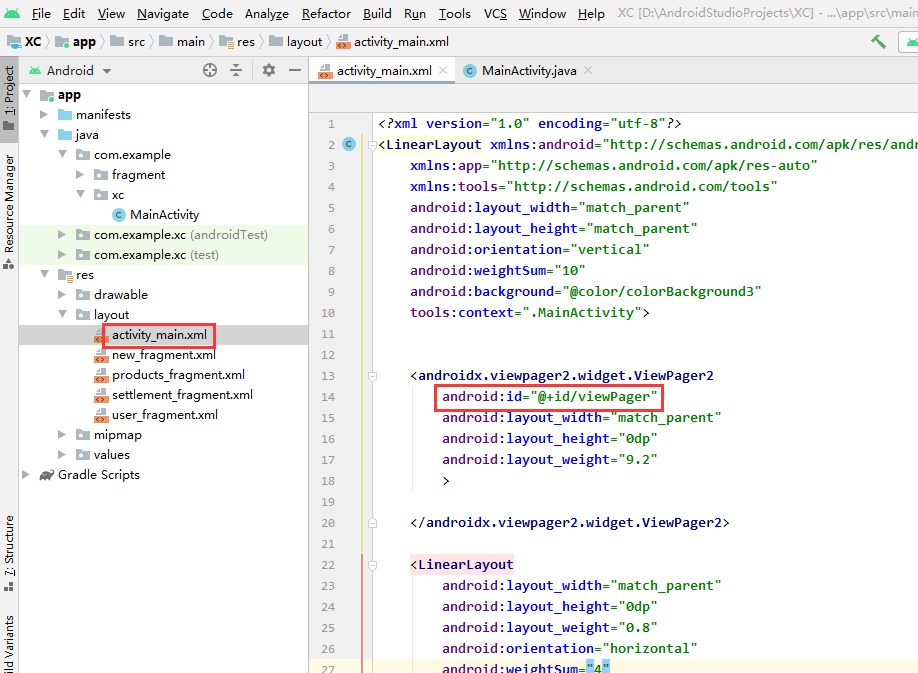


#### 第9步：用Fragment+ViewPager2实现滑动翻页

先在<app\res\layout\>目录下的【activity\_main.xml】布局文件里添加下面的代码，其作用为：给ViewPager2控件添加android:id属性。

android:id="@+id/viewPager"

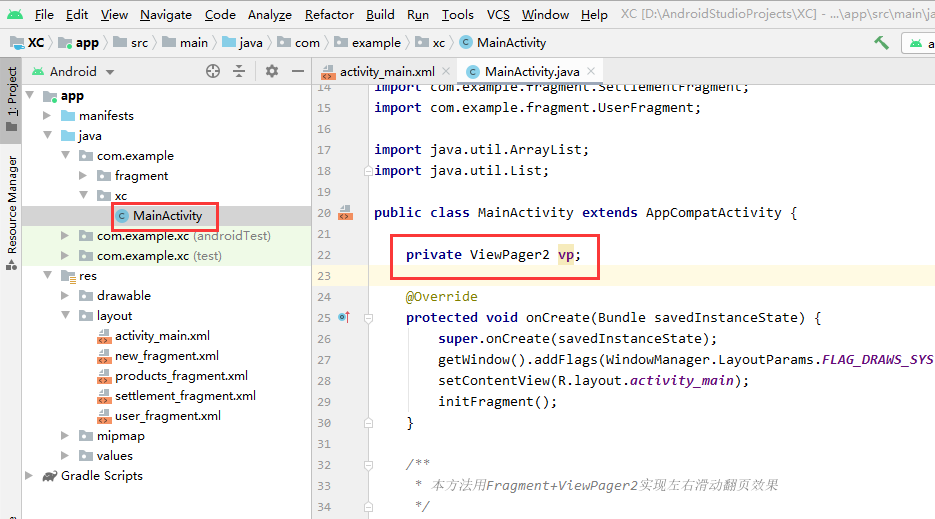
上述代码添加的具体位置如下图所示



然后在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面代码，其作用是定义变量ViewPager2 vp。

private ViewPager2 vp;

上述代码插入的具体位置如下图所示



上述代码在编写的时候需对ViewPager2类导包，需要导入

import androidx.viewpager2.widget.ViewPager2;

最后在第8步定义的initFragment方法里插入下面代码，其效果是：用Fragment+ViewPager2技术实现左右滑动屏幕翻页效果。

//创建Fragment集合

final List<Fragment> list=new ArrayList<>();

list.add(new NewFragment());

list.add(new ProductsFragment());

list.add(new SettlementFragment());

list.add(new UserFragment());

vp=findViewById(R.id.viewPager); //获得ViewPager2控件

//设置预加载的Fragment页面数量，可防止流式布局StaggeredGrid数组越界错误。

vp.setOffscreenPageLimit(list.size() - 1); //设置适配器

FragmentStateAdapter adapter=new FragmentStateAdapter(MainActivity.this) {

@NonNull

@Override

public Fragment createFragment(int position) {

return list.get(position);

}

@Override

public int getItemCount() {

return list.size();

}

};

vp.setAdapter(adapter); //把适配器添加给ViewPager2

上述代码在编写的时候需要导入下面的包

import java.util.ArrayList;

import java.util.List;

import com.example.fragment.NewFragment;

import com.example.fragment.ProductsFragment;

import com.example.fragment.SettlementFragment;

import com.example.fragment.UserFragment;

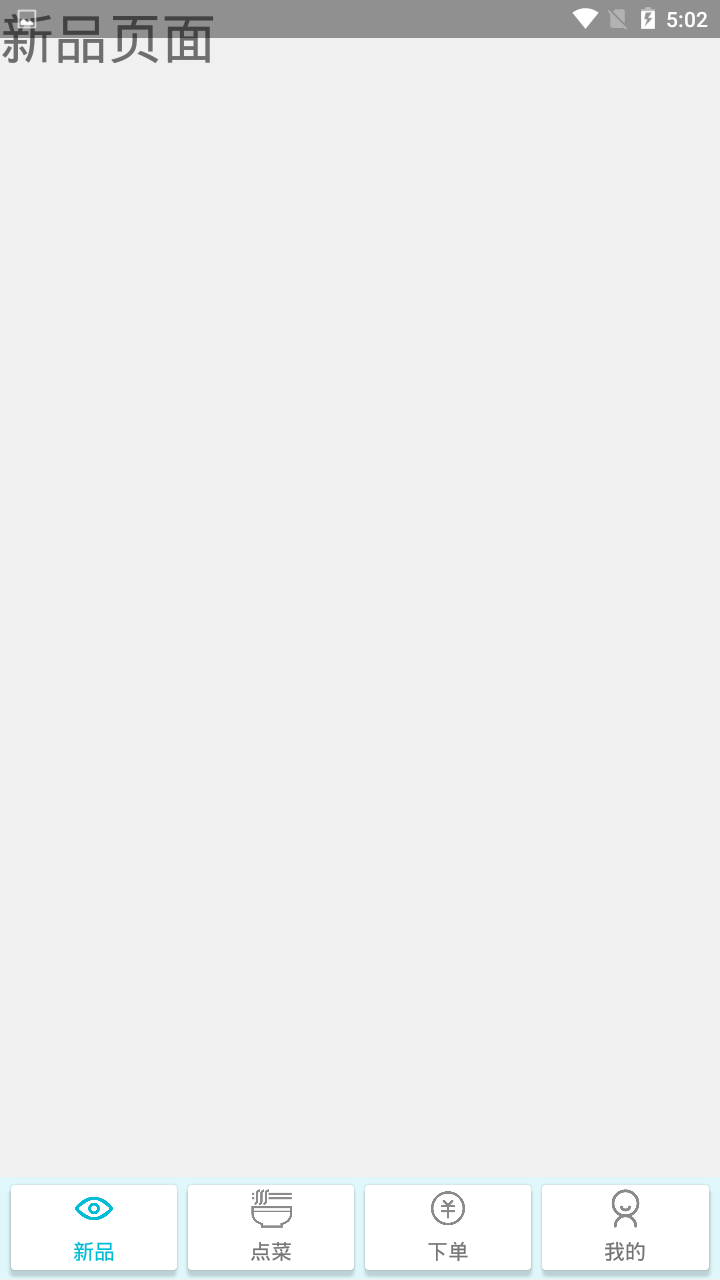
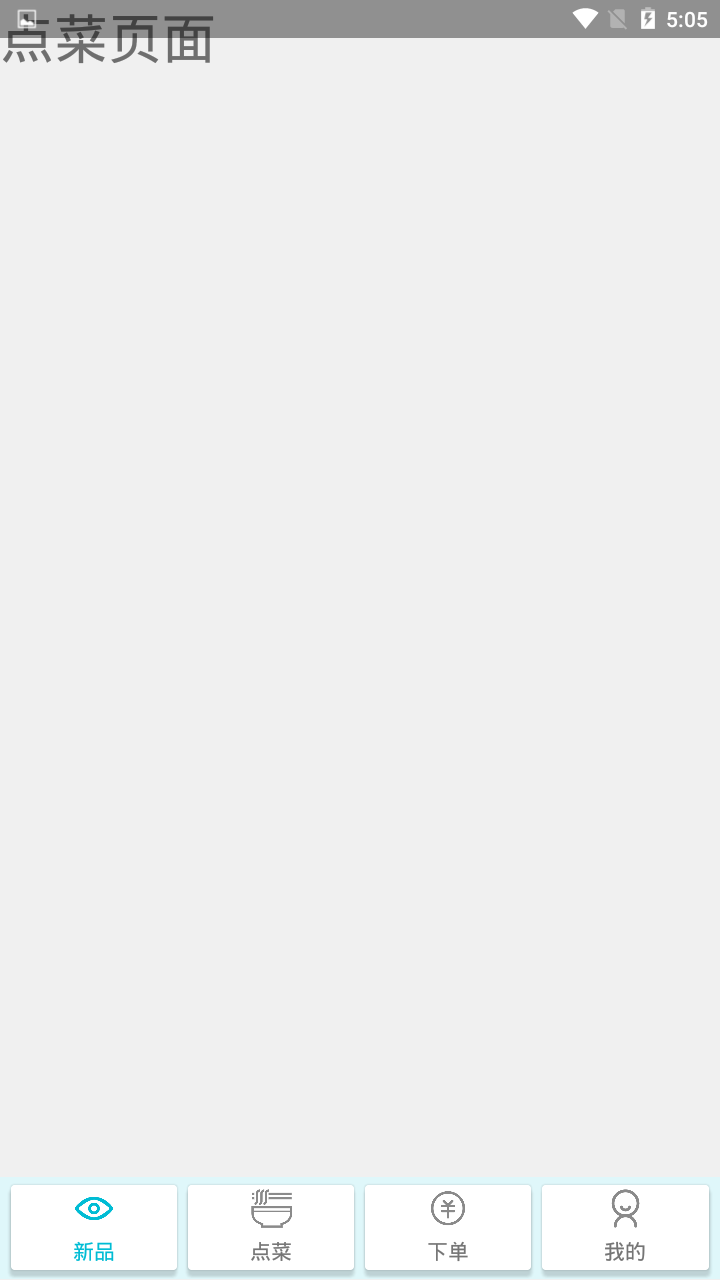
import androidx.fragment.app.Fragment;

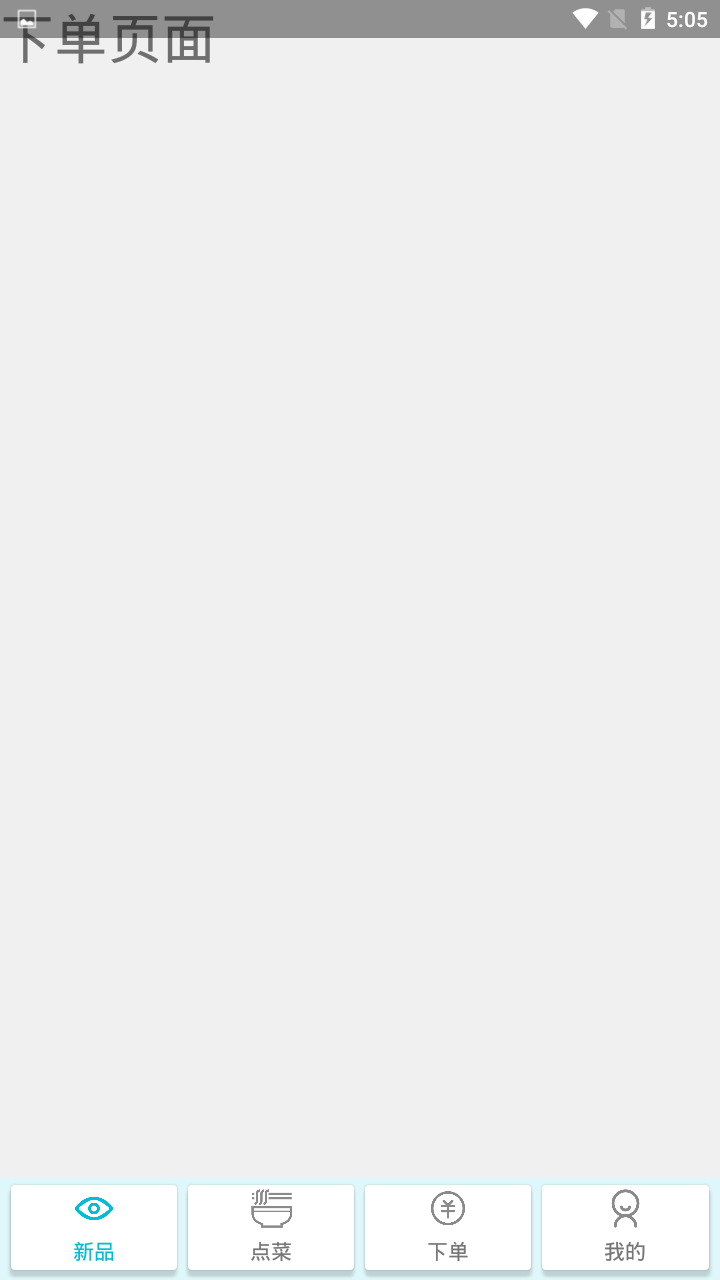
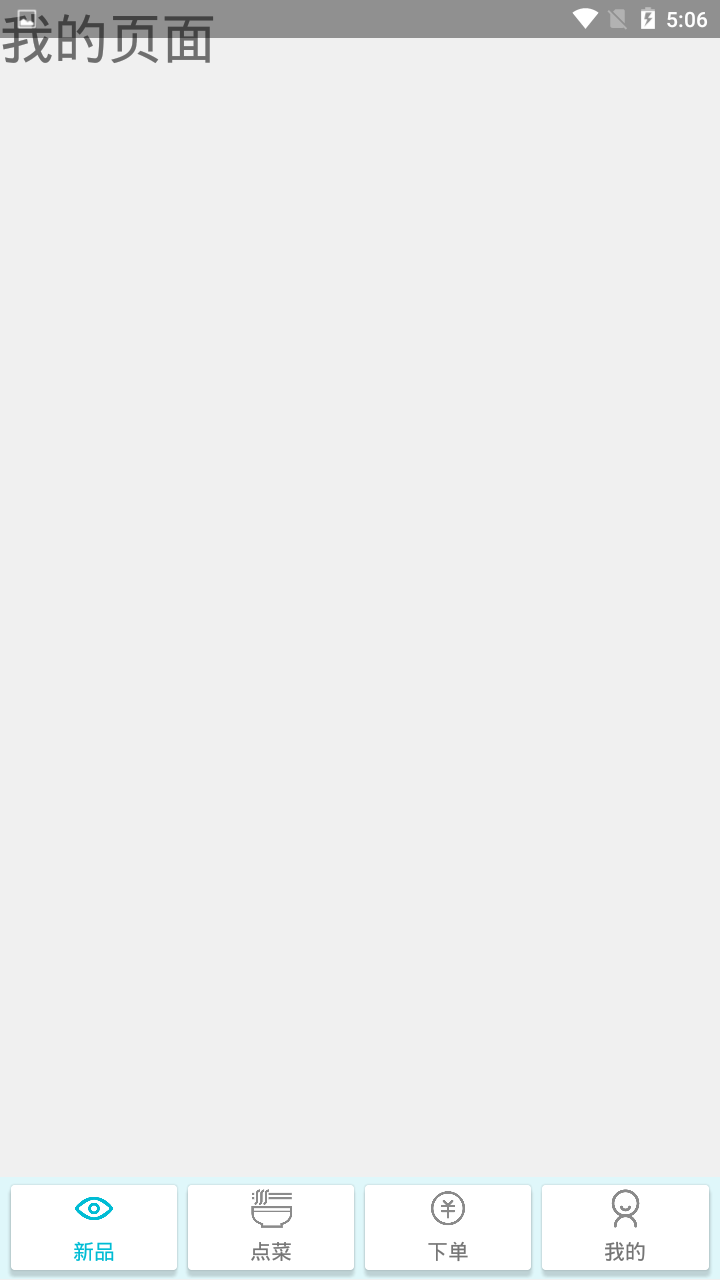
import androidx.viewpager2.adapter.FragmentStateAdapter;

import androidx.annotation.NonNull;

#### 第10步：运行测试

上面所有的步骤都正确实现后，就可以生成apk文件运行测试了。具体步骤参考实验1的第2大步和第3大步，把App安装到手机后的效果如下图所示，如果左右滑动屏幕，能依次显示新品页面、点菜页面、下单页面和我的页面表示完成。

#### 第11步：定义initClick方法

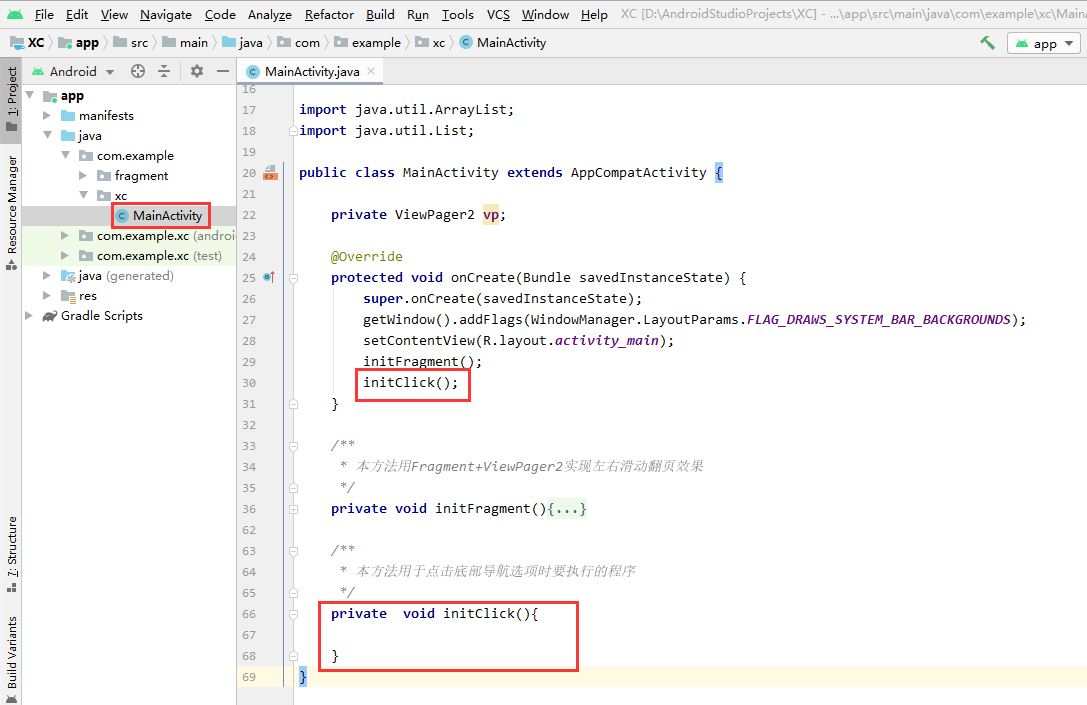
在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面的两段代码，其作用为：定义名为initClick的方法，并调用，用于编写点击底部导航选项时要执行的程序。

initClick();

private void initClick(){

}

上述代码插入的具体位置如下图所示



#### 第12步：获取ImageView控件和TextView控件对象

先在<app\res\layout\>目录下的【activity\_main.xml】布局文件里插入下面的属性代码，其作用是：给每个ImageView控件和TextView控件添加相应的android:id属性。

新品选项控件要插入的代码为

android:id="@+id/imageHome"

android:id="@+id/textHome"

点菜选项控件要插入的代码为

android:id="@+id/imageProduct"

android:id="@+id/textProduct"

下单选项控件要插入的代码为

android:id="@+id/imageSettlement"

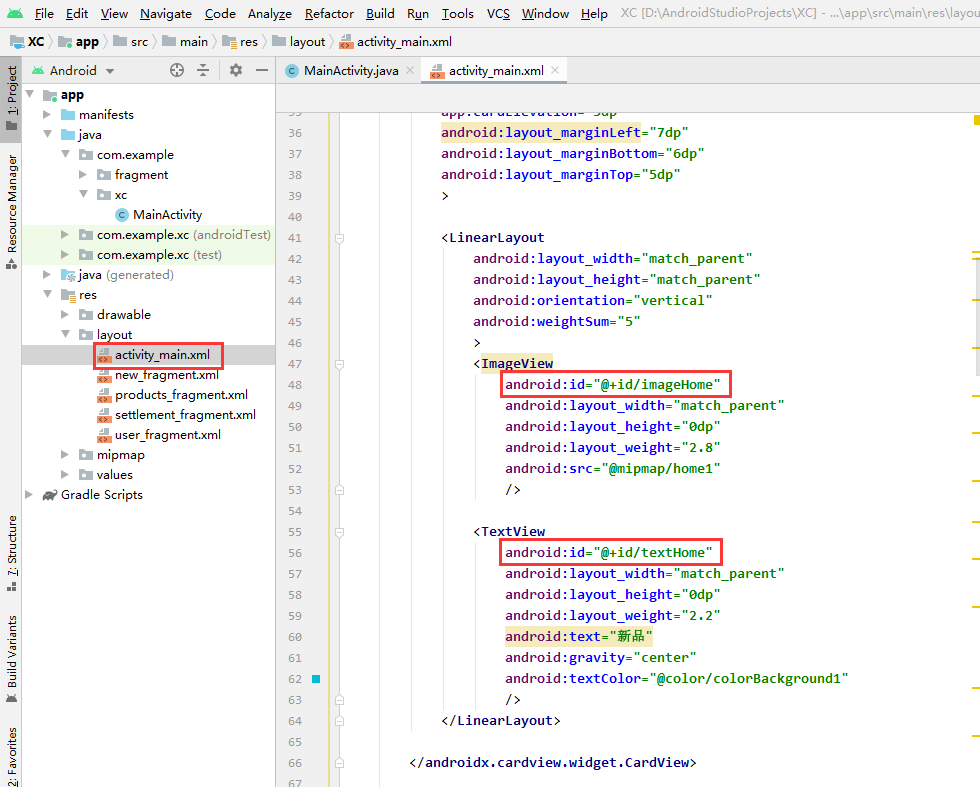
android:id="@+id/textSettlement"

我的选项控件要插入的代码为

android:id="@+id/imageUser"

android:id="@+id/textUser"

上述代码插入的具体位置如下图所示



然后在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面的代码，其作用是:定义相应ImageView控件和TextView控件变量。

public static ImageView homeiv;

public static TextView hometv;

public static ImageView productiv;

public static TextView producttv;

public static ImageView settlementiv;

public static TextView settlementtv;

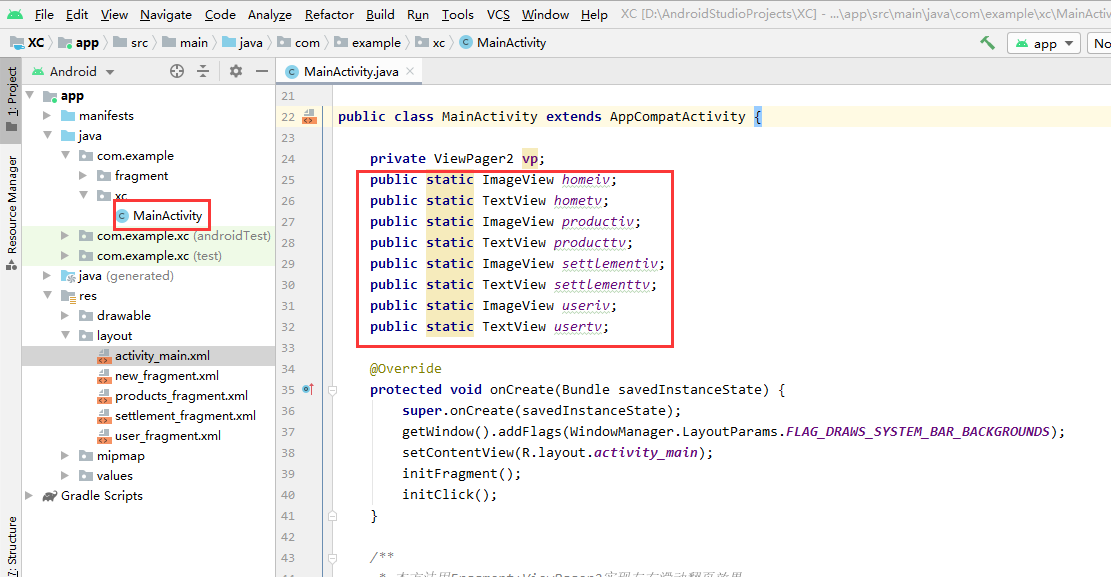
public static ImageView useriv;

public static TextView usertv;

**检测条件：**

检测<app\java\com.example.xc\>路径下的【MainActivity.java】文件里是否有【initClick】关键字；

上述代码插入的具体位置如下图所示



上述代码在编写的时候需对ImageView类和TextView类导包，需要导入

import android.widget.ImageView;

import android.widget.TextView;

最后在第11步定义的initClick()方法里添加下面的代码，其作用是：获取所有ImageView控件和TextView控件对象。

homeiv=findViewById(R.id.imageHome);

hometv=findViewById(R.id.textHome);

productiv=findViewById(R.id.imageProduct);

producttv=findViewById(R.id.textProduct);

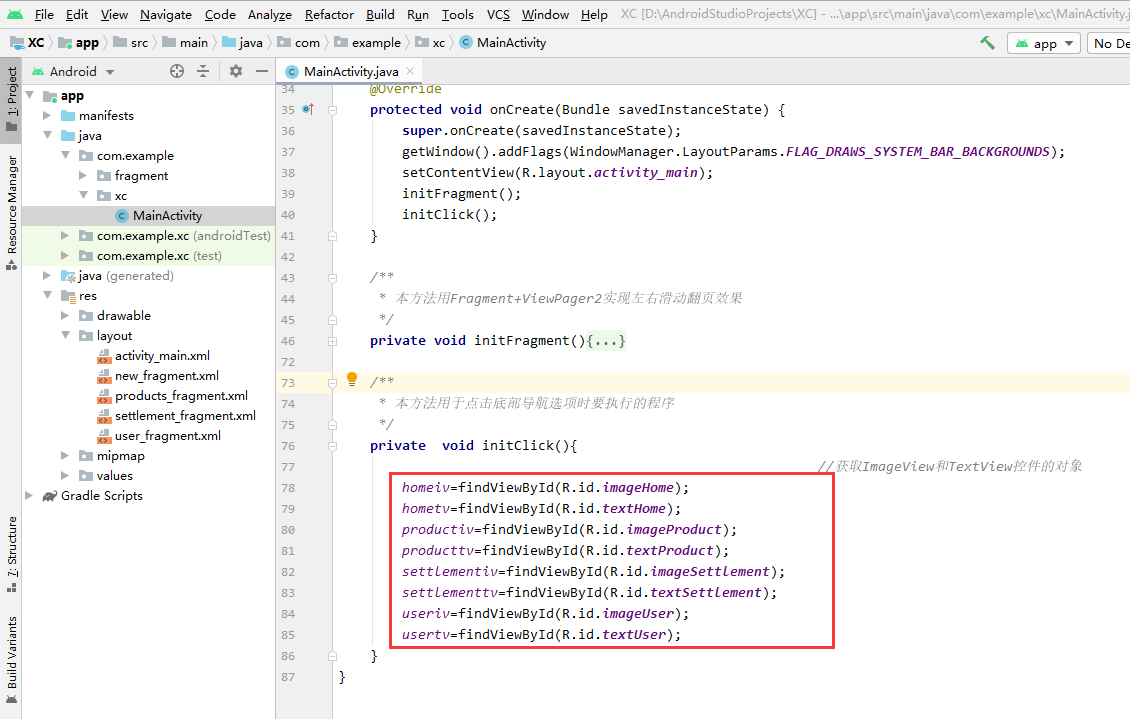
settlementiv=findViewById(R.id.imageSettlement);

settlementtv=findViewById(R.id.textSettlement);

useriv=findViewById(R.id.imageUser);

usertv=findViewById(R.id.textUser);

上述代码插入的具体位置如下图所示



#### 第13步：给选项添加点击事件

先在<app\res\layout\>目录下的【activity\_main.xml】布局文件里所有CardView控件标签里的LinearLayout控件标签内分别插入下面的属性代码，其作用是：给选项的LinearLayout控件添加相应的android:id属性。

【新品】LinearLayout控件插入

android:id="@+id/btnHome"

【点菜】LinearLayout控件插入

android:id="@+id/btnProduct"

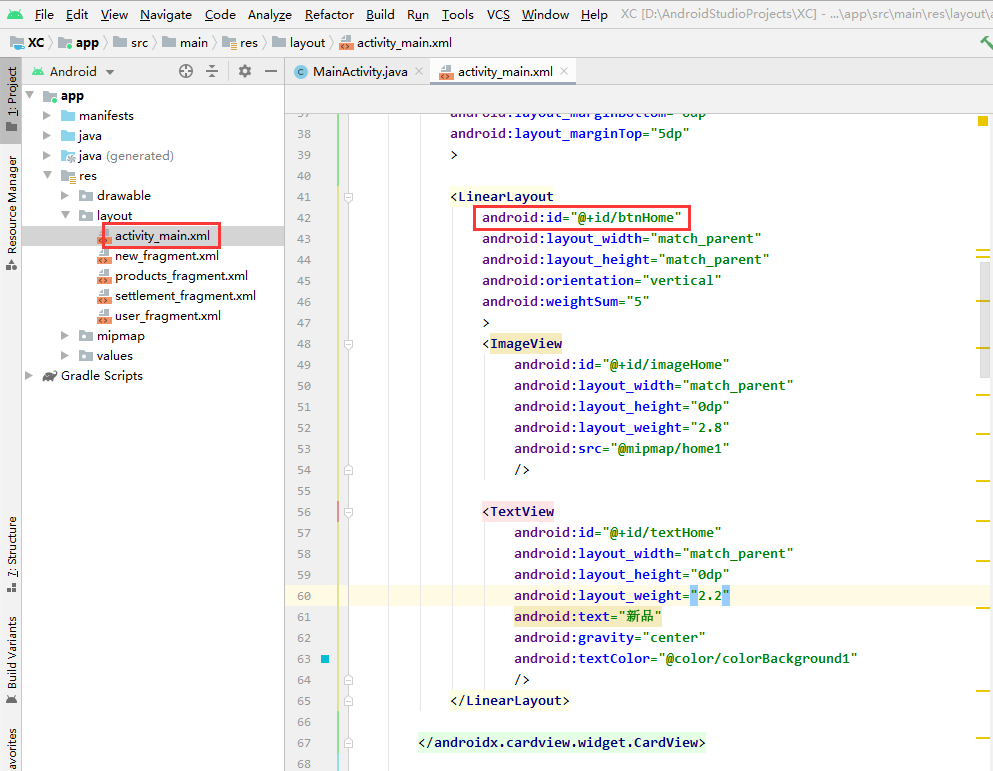
【下单】LinearLayout控件插入

android:id="@+id/btnSettlement"

【我的】LinearLayout控件插入

android:id="@+id/btnUser"

上述代码插入的具体位置如下图所示



然后在<app\java\com.example.xc\>路径下的【MainActivity】类里接着上一步的代码后面添加下面的代码，其作用是：给每个选项的LinearLayout控件添加点击事件。

//给新品选项添加点击事件

LinearLayout btnhome=findViewById(R.id.btnHome);

btnhome.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

}

});

//给点菜选项添加添加事件

LinearLayout btnproduct=findViewById(R.id.btnProduct);

btnproduct.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

}

});

//给下单选项添加点击事件

LinearLayout btnsettlement=findViewById(R.id.btnSettlement);

btnsettlement.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

}

});

//给我的选项添加点击事件

LinearLayout btnuser=findViewById(R.id.btnUser);

btnuser.setOnClickListener(new View.OnClickListener() {

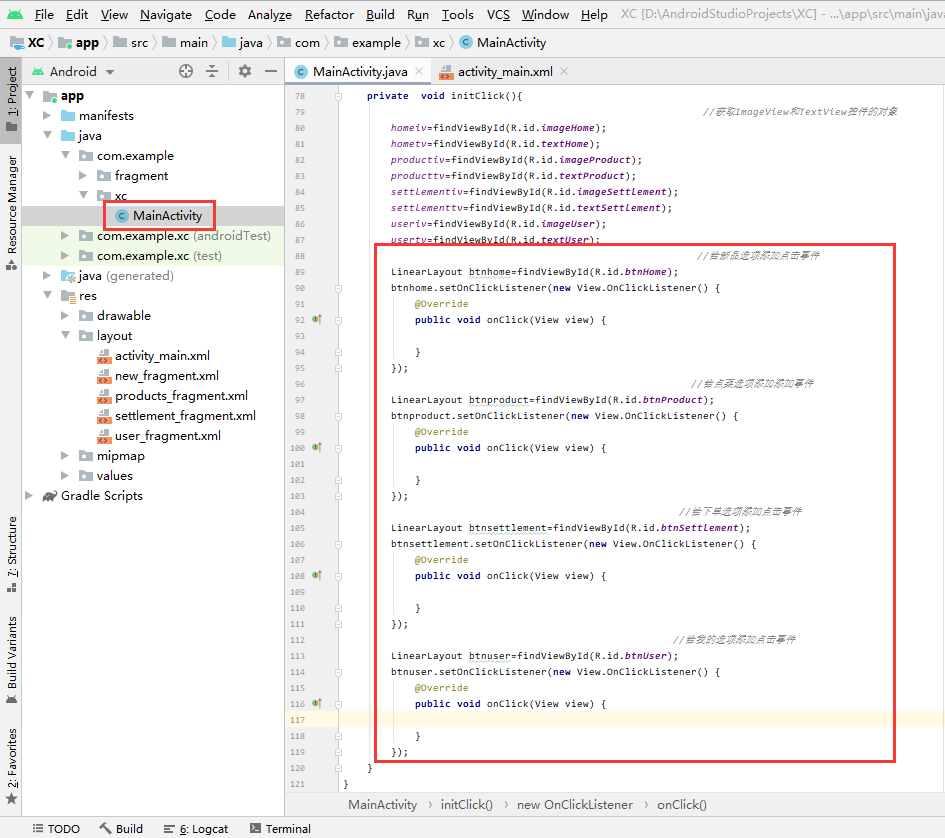
@Override

public void onClick(View view) {

}

});

上述代码添加的具体位置如下图所示



上述代码在编写的时候需对LinearLayout类和View类导包，需要导入

import android.widget.LinearLayout;

import android.view.View;

#### 第14步：实现点击选项打开对应页面

在上面的4个点击事件代码里分别插入下面代码，其效果是：点击底部导航选项，打开对应界面。

vp.setCurrentItem(0,false);

vp.setCurrentItem(1,false);

vp.setCurrentItem(2,false);

vp.setCurrentItem(3,false);

上述代码插入的具体位置如下图所示



其中【setCurrentItem(1,false)】方法用于打开集合里对应的Fragment对象页面，前面的参数表示要打开的对象下标；后面的参数有2个值：true和false，true表示打开时显示滑动效果，false表示打开时禁止滑动效果。

#### 第15步：封装底部导航颜色变化代码

在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面四个方法的代码，其效果为：封装选中选项时，各选项的颜色做相应变化的程序，选中的呈现青蓝色，没有选中的呈现淡灰色。

/\*\*

\* 选中新品选项时的颜色

\*/

public static void initHomeBtnColor(){

homeiv.setImageResource(R.drawable.home1);

hometv.setTextColor(Color.rgb(0,188,212));

productiv.setImageResource(R.drawable.product2);

producttv.setTextColor(Color.rgb(148,148,148));

settlementiv.setImageResource(R.drawable.settlement2);

settlementtv.setTextColor(Color.rgb(148,148,148));

useriv.setImageResource(R.drawable.user2);

usertv.setTextColor(Color.rgb(148,148,148));

}

/\*\*

\* 选中点菜选项时的颜色

\*/

public static void initProductBtnColor(){

homeiv.setImageResource(R.drawable.home2);

hometv.setTextColor(Color.rgb(148,148,148));

productiv.setImageResource(R.drawable.product1);

producttv.setTextColor(Color.rgb(0,188,212));

settlementiv.setImageResource(R.drawable.settlement2);

settlementtv.setTextColor(Color.rgb(148,148,148));

useriv.setImageResource(R.drawable.user2);

usertv.setTextColor(Color.rgb(148,148,148));

}

/\*\*

\* 选中下单选项时的颜色

\*/

public static void initSettlementBtnColor(){

homeiv.setImageResource(R.drawable.home2);

hometv.setTextColor(Color.rgb(148,148,148));

productiv.setImageResource(R.drawable.product2);

producttv.setTextColor(Color.rgb(148,148,148));

settlementiv.setImageResource(R.drawable.settlement1);

settlementtv.setTextColor(Color.rgb(0,188,212));

useriv.setImageResource(R.drawable.user2);

usertv.setTextColor(Color.rgb(148,148,148));

}

/\*\*

\* 选中我的选项时的颜色

\*/

public static void initUserBtnColor(){

homeiv.setImageResource(R.drawable.home2);

hometv.setTextColor(Color.rgb(148,148,148));

productiv.setImageResource(R.drawable.product2);

producttv.setTextColor(Color.rgb(148,148,148));

settlementiv.setImageResource(R.drawable.settlement2);

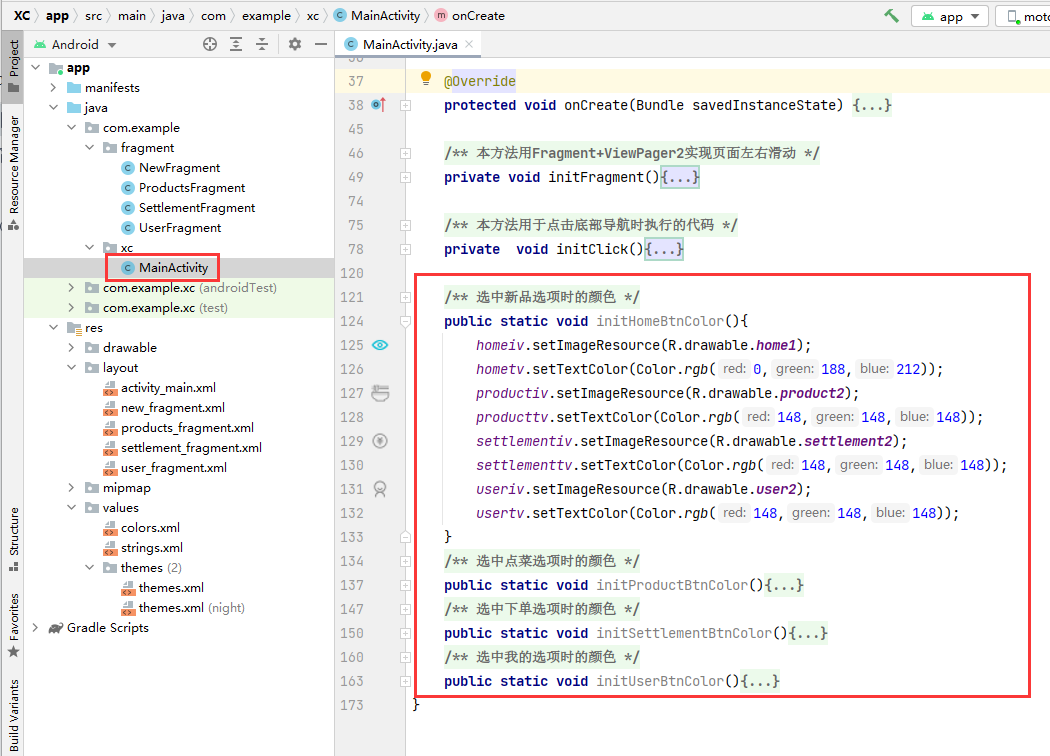
settlementtv.setTextColor(Color.rgb(148,148,148));

useriv.setImageResource(R.drawable.user1);

usertv.setTextColor(Color.rgb(0,188,212));

}

上述代码插入的具体位置如下图所示



上述代码在编写的时候需对Color类导包，需要导入

import android.graphics.Color;

#### 第16步：实现点击选项导航颜色变化

在第13步点击事件代码里分别插入下面的四段代码，调用上一步实现的方法。其效果为：点击底部导航选项时，选中的选项颜色为青蓝色，未选中的选项颜色为淡灰色。

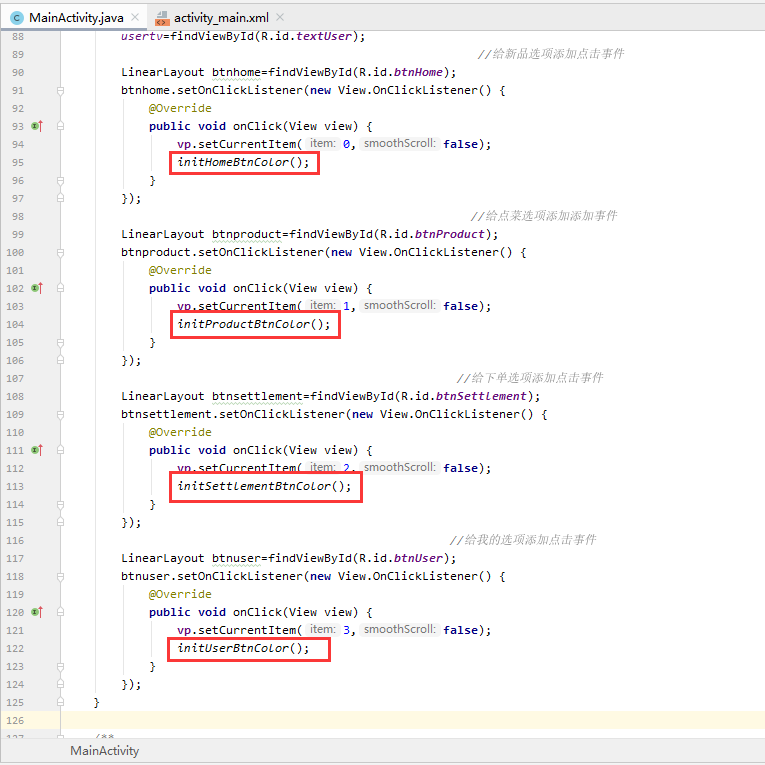
initHomeBtnColor();

initProductBtnColor();

initSettlementBtnColor();

initUserBtnColor();

上述代码插入的具体位置如下图所示



#### 第17步：实现滑动页面选项颜色变化

在<app\java\com.example.fragment\>目录下面的四个Fragment类里的onResume方法里分别插入下面的代码，其效果为：左右滑动页面时，底部导航选项颜色相应变化，滑到的页面其底部导航选项颜色为青蓝色，其他的选项为淡灰色。

NewFragment类里的onResume方法要插入的代码为

MainActivity.initHomeBtnColor();

ProductsFragment类里的onResume方法要插入的代码为

MainActivity.initProductBtnColor();

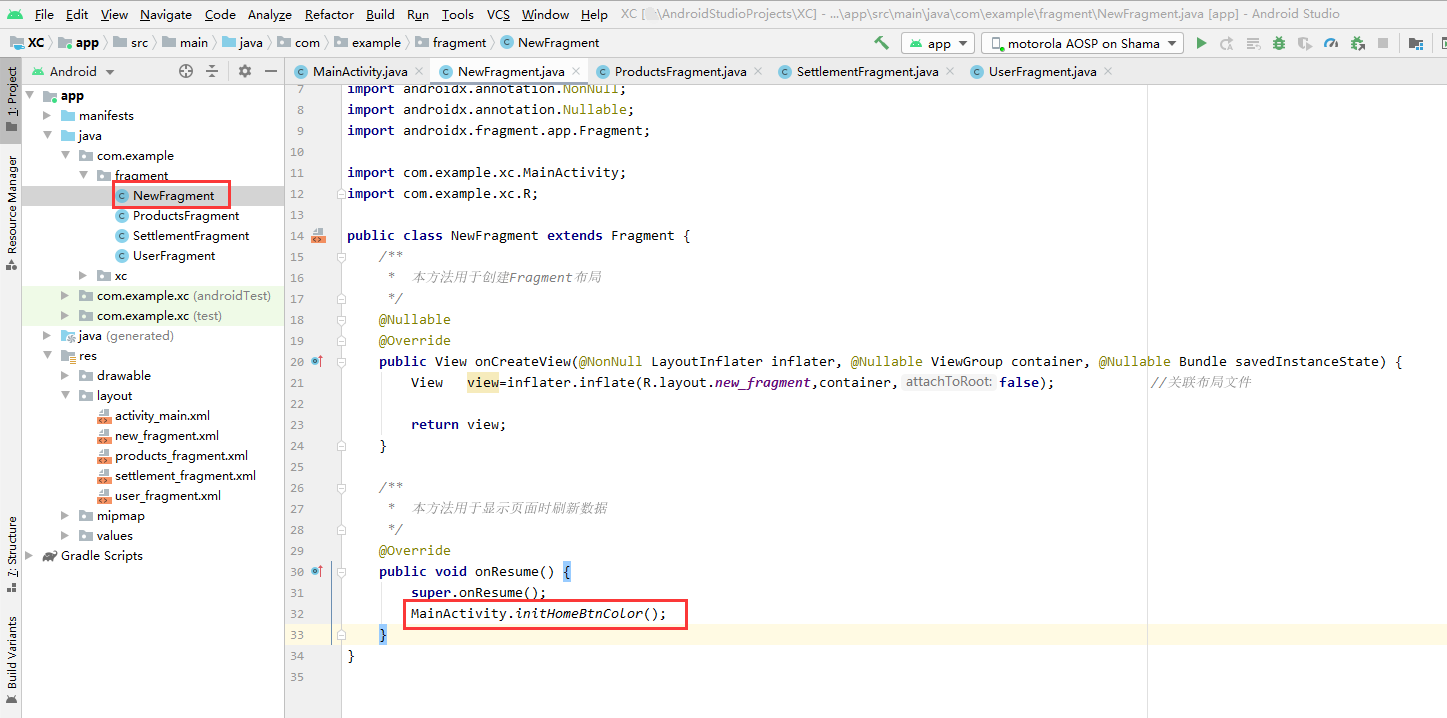
SettlementFragment类里的onResume方法要插入的代码为

MainActivity.initSettlementBtnColor();

UserFragment类里的onResume方法要插入的代码为

MainActivity.initUserBtnColor();

上述代码插入的具体位置如下图所示



上述代码在编写的时候需对MainActivityr类导包，需要导入

import com.example.xc.MainActivity;

#### 第18步：运行测试

上面所有的步骤都正确实现后，就可以生成apk文件运行测试了。具体步骤参考实验1的第2大步和第3大步，把App安装到手机后的效果如下图所示，点击选项或左右滑动页面，底部导航选项颜色会做相应变化。

