

中山大学数据科学与计算机学院本科生实验报告

(2017 年秋季学期)

课程名称：手机应用平台开发

任课教师：刘宁

年级	15352054	专业（方向）	软件工程（移动信息工程）
学号	15352054	姓名	陈远鹏
电话	18998634187	Email	2044757581@qq.com
开始日期	2017.10.20	完成日期	2017.10.24

一、 实验题目 Intent、Bundle 的使用以及 RecyclerView、ListView 的应用

二、 实现内容

【实验目的】

- 1.复习事件处理
- 2.学习Intent、Bundle在Activity跳转中的应用
- 3.学习RecyclerView、ListView以及各类适配器的用法

【实验内容】

本次实验模拟实现一个商品表，有两个界面，第一个界面用于呈现商品，如下所示：

E Enchated Forest

A Arla Milk

D Devondale Milk

K Kindle Oasis

W waitrose 早餐麦片

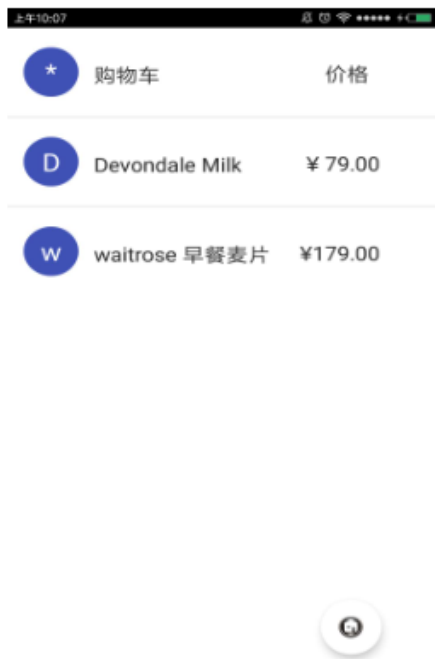
M Mcvitie's 饼干

F Ferrero Rocher



下面还有数据，就不截图了，数据在素材中有给出。

点击右下方的悬浮按钮可以切换到购物车：



上面两个列表点击任意一项后，可以看到详细的信息：

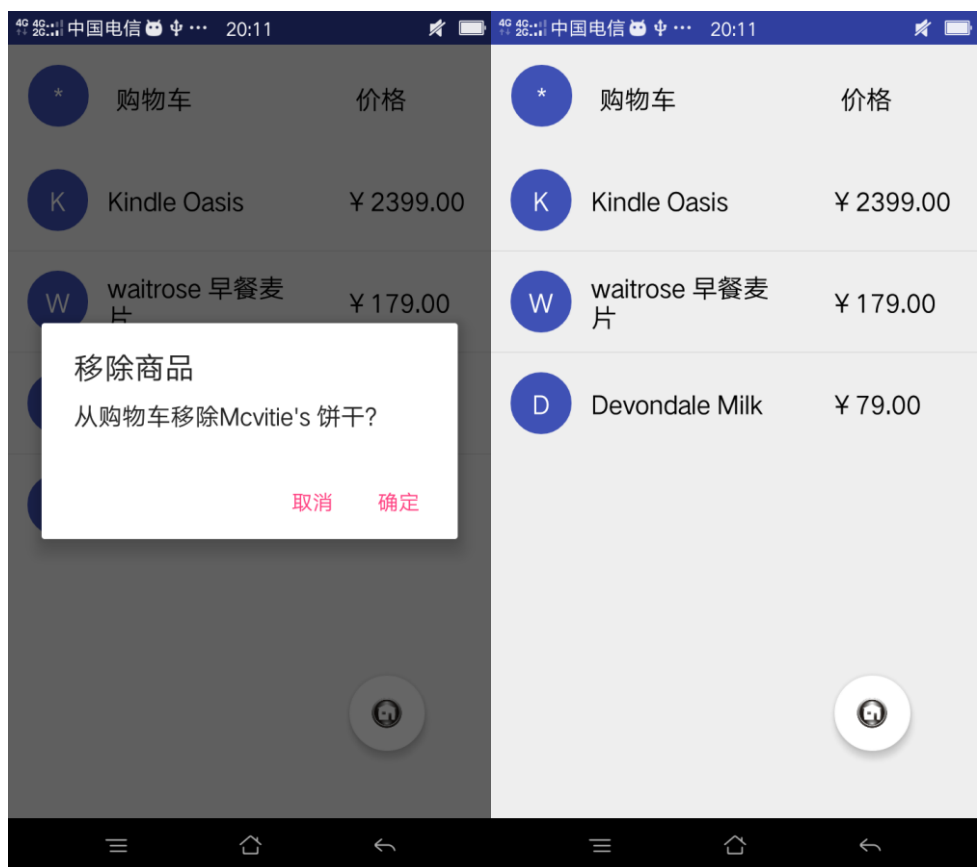
上面两个列表点击任意一项后，可以看到详细的信息：



三、 课堂实验结果

(1) 实验截图





(2) 实验步骤以及关键代码

1.xml 文件 (比较长的就不贴了):

①activity_main.xml //商品表布局文件, 包含一个 RecyclerView 和一个 FloatingActionButton

```
<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout android:id="@+id/main_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <android.support.v7.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:background="@drawable/background"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </android.support.v7.widget.RecyclerView>
    <android.support.design.widget.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/shoplist"
        app:backgroundTint="#FFFFFF"
        android:id="@+id/floatButton"
        android:layout_gravity="end|bottom"
        android:layout_margin="50sp"
    >
    </android.support.design.widget.FloatingActionButton>
</android.support.design.widget.CoordinatorLayout>
```

②item1.xml//商品表每一行的布局

```

<?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:layout_margin="15sp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cycle"
        android:textSize="17sp"
        android:textColor="#FFFFFF"
        android:textAlignment="center"
        android:layout_margin="10sp"
        android:background="@drawable/shape"
        android:gravity="center_vertical"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="17sp"
        android:id="@+id/goodname"
        android:textColor="#000000"
        android:layout_gravity="center_vertical"/>
</LinearLayout>

```

③item2.xml//购物车每一行的布局

```

<?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:layout_margin="15sp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cycle2"
        android:textSize="17sp"
        android:textColor="#FFFFFF"
        android:textAlignment="center"
        android:layout_margin="13sp"
        android:background="@drawable/shape"
        android:gravity="center_vertical"/>
    <TextView
        android:layout_width="140dp"
        android:layout_height="wrap_content"
        android:textSize="17sp"
        android:id="@+id/goodname2"
        android:textColor="#000000"
        android:layout_gravity="center_vertical"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="17sp"

```

```

        android:id="@+id/goodname2"
        android:textColor="#000000"
        android:layout_gravity="center_vertical"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="17sp"
    android:id="@+id/goodprice2"
    android:layout_marginLeft="40dp"
    android:textColor="#000000"
    android:layout_gravity="center_vertical"/>
</LinearLayout>

```

2..Java 代码实现：

1、首先我们要实现一个商品表(MainActivity.xml)，用 RecyclerView 实现，里面的每一项由自己定义的一个 CommonAdapter 填充;我们先把每一项的信息用自己构建的一个类 information 来“绑在一起”，下面是 **information.class**:

```

public class information implements Serializable {

    public information(int imageId, String name, String info1, String info2, String price) {
        this.imageId=imageId;
        this.name=name;
        this.price=price;
        this.info1=info1;
        this.info2=info2;
    }

    public int getImageId() { return imageId; }

    public void setImageId(int imageId) { this.imageId = imageId; }
    public String getName() { return name; }

    public void setName(String name) { this.name = name; }
    public String getInfo1() { return info1; }

    public void setInfo1(String weight) { this.info1 = weight; }
    public String getInfo2() { return info2; }

    public void setInfo2(String weight) { this.info2 = weight; }
    public String getPrice() { return price; }

    public void setPrice(String price) { this.price = price; }
    public void setPrice (String price) { this.price = price; }

    private int imageId;
    private String name;
    private String info1;
    private String info2;
    private String price;
}

```

接下来构建 CommonAdapter 的实例，并把它填充到 RecyclerView 里面；

/* 为每一项数据创建一个对象，并添加在List中 */

```
final List<information> Infos = new ArrayList<information>() {{
    add(new information(R.drawable.enchantedforest, "Enchanted Forest", "作者", "Johanna Basford", "¥ 5.00"));
    add(new information(R.drawable.arla, "Arla Milk", "产地", "德国", "¥ 59.00"));
    add(new information(R.drawable.devondale, "Devondale Milk", "产地", "澳大利亚", "¥ 79.00"));
    add(new information(R.drawable.kindle, "Kindle Oasis", "版本", "Johanna Basford", "¥ 2399.00"));
    add(new information(R.drawable.waitrose, "waitrose 早餐麦片", "重量", "2Kg", "¥ 179.00"));
    add(new information(R.drawable.mcvitie, "Mcvitie's 饼干", "产地", "英国", "¥ 14.90"));
    add(new information(R.drawable.ferrero, "Ferrero Rocher", "重量", "300g", "¥ 132.59"));
    add(new information(R.drawable.maltesers, "Maltesers", "重量", "118g", "¥ 141.43"));
    add(new information(R.drawable.lindt, "Lindt", "重量", "249g", "¥ 139.43"));
    add(new information(R.drawable.borggreve, "Borggreve", "重量", "640g", "¥ 28.90"));
}};

add(new information(R.drawable.borggreve, "Borggreve", "重量", "640g", "¥ 28.90"));
}};
final List<item> itemList = new ArrayList<>(10);
List<item2> shoppingList = new ArrayList<>(10);
String[] primeletter={"E", "A", "D", "K", "W", "M", "F", "M", "L", "B"};
String[] things={"Enchanted Forest", "Arla Milk", "Devondale Milk", "Kindle Oasis", "waitrose 早餐麦片", "Mcvitie's 饼干",
    "Ferrero Rocher", "Maltesers", "Lindt", "Borggreve"};
String[] prices={"¥ 5.00", "¥ 59.00", "¥ 79.00", "¥ 2399.00", "¥ 179.00", "¥ 14.90", "¥ 132.59", "¥ 141.43", "¥ 139.43", "¥ 28.90"};

for(int i=0;i<10;i++)
{
    item temp = new item(things[i], primeletter[i]);
    itemList.add(temp);
}

final CommonAdapter commonAdapter = new CommonAdapter(this, R.layout.item, itemList) {
    @Override
    public void convert(ViewHolder holder, item item) {
        TextView name = holder.getView(R.id.goodname);
        name.setText(item.getName());
        TextView first = holder.getView(R.id.cycle);
        first.setText(item.getPrime());
    }
};
```

为 commonAdapter 设置监听器和点击事件，长按的时候删除列表项，点击的时候转到商品详情的 activity，用 intent 实现；

```
commonAdapter.setOnItemClickListener(new CommonAdapter.OnItemClickListener() {
    @Override
    public Boolean onLongClick(int position) {
        int i=position;
        String[] number={"1", "2", "3", "4", "5", "6", "7", "8", "9", "10"};
        Toast.makeText(MainActivity.this, "移除第"+number[i]+"个商品", Toast.LENGTH_SHORT).show();
        itemList.remove(position);
        commonAdapter.notifyDataSetChanged();
        return true;
    }

    @Override
    public void onClick(int position){
        Intent intent = new Intent(MainActivity.this, good.class);
        information temp = Infos.get(position);
        intent.putExtra("Info", temp);
        startActivity(intent);
    }
});
```

设置浮动按钮的点击事件——转到购物车页面；

```
final FloatingActionButton floatingActionButton=(FloatingActionButton)findViewById(R.id.floatButton);
floatingActionButton.setOnClickListener((v) -> {
    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
    startActivity(intent);
});
```

2、接下来我们看**购物车 activity(SecondActivity.xml)**的实现，这个界面由 listview 组成，我用了自己定义的 mylistadapter 来填充，首先在外面定义一个静态的 list 存储放入购物车的商品，方便删除和增加；

```
static List<item2> shoplist=new ArrayList<~>(100);
```

接下来填充 listView:

```
for (int i=0;i<shoplist.size();i++){
    data.add(shoplist.get(i));
}
ListView listView=(ListView)findViewById(R.id.lv2);
final information p = (information) getIntent().getSerializableExtra("Info2");
final myListAdapter mListAdapter = new myListAdapter(SecondActivity.this, data);
listView.setAdapter(mListAdapter);
```

设置点击事件:长按会弹出对话框；


```

final information p = (information) getIntent().getSerializableExtra("Info2");
final myListAdapter mListAdapter = new myListAdapter(SecondActivity.this, data);
listView.setAdapter(mListAdapter);
listView.setOnItemLongClickListener((parent, view, position, id) -> {
    AlertDialog.Builder message = new AlertDialog.Builder(SecondActivity.this);
    message.setTitle("移除商品");
    message.setMessage("从购物车移除" + shopList.get(position).getName()+"?");
    message.setPositiveButton("确定", (dialogInterface, i) -> {
        shopList.remove(position);
        data.remove(position);
        mListAdapter.notifyDataSetChanged();
    });
    message.setNegativeButton("取消", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialogInterface, int i) {
        }
    });
    message.create().show();
    return true;
});

```

为 listView 每个项设置监听器，点击后转到相应的商品详情页；

```

listView.setOnItemClickListener((parent, view, position, id) -> {
    Intent intent = new Intent(SecondActivity.this, good.class);
    intent.putExtra("Info", p);
    startActivity(intent);
});

```

点击浮动按钮转到商品表；

```

final FloatingActionButton floatingActionButton=(FloatingActionButton)findViewById(R.id.floatButton);
floatingActionButton.setOnClickListener((v) -> {
    Intent intent = new Intent(SecondActivity.this, MainActivity.class);
    startActivity(intent);
});

```

3、最后讲一下商品详情页(good.xml)的实现:

接收从商品表发来的 intent,提供页面的图片 ID，各种字符串等资源；

```

final information p = (information) getIntent().getSerializableExtra("Info");

```

把得到的数据包拿走相应部分对页面的每个组成部分进行设置；

```

TextView info1 = (TextView) findViewById(R.id.info1);
info1.setText(p.getInfo1());
TextView info2 = (TextView) findViewById(R.id.info2);
info2.setText(p.getInfo2());
TextView price = (TextView) findViewById(R.id.price);
price.setText(p.getPrice());

TextView name = (TextView) findViewById(R.id.Name);
name.setText(p.getname());

ImageView img = (ImageView) findViewById(R.id.img);
img.setImageResource(p.getImageId());

```

星星的切换时使用 tag，每个 View 都可以设置 tag，通过 tag 可以用来判断该 View 现在的状态。在初始化的时候，将 tag 设置为 0，标记此时为空心星星，如果星星被点击了，并且 tag 为 0，那么就把图片换为实心的星星，然后设置 tag 为 1；如果 tag 为 1，那么就把图片换为空心的星星，然后设置 tag 为 0。建议在 java 文件中给需要的 view 设置 tag。

```

/* 星星的切换 */
final Button star = (Button) findViewById(R.id.star);
star.setOnClickListener((v) -> {
    if (!tag) {
        star.setBackground(getDrawable(R.drawable.full_star));
        tag = true;
    } else {
        star.setBackground(getDrawable(R.drawable.empty_star));
        tag = false;
    }
});

```

购物车图标点击弹出 Toast 信息，并且把数据包传给购物车,shoplist 加入该项商品的信息（用 item 打包）（实现商品加入购物车功能）

```

shop.setOnClickListener((v) -> {
    String name=p.getname();
    String price=p.getPrice();
    int a=0;
    String prime=name.substring(a, a);
    item2 temp=new item2(name, prime, price);
    shoplist.add(temp);
    Toast.makeText(good.this, "商品已添加到购物车", Toast.LENGTH_SHORT).show();
    /////
    Intent intent = new Intent(good.this, SecondActivity.class);
    intent.putExtra("Info2", p);
    startActivity(intent);
    /////
});

```

（3）实验遇到困难以及解决思路

1、在写 RecyclerView 的 Adapter 的时候，一直不知道 `mRecyclerView.s` 是来自哪里的东西，加上去老报错，而且构造函数的变量也没有出现过，正当迷茫的时候，构造函数这次词点醒了我，就像我们写 c++ 代码的时候，我们是在 class 里面声明了成员变量后，才有对应的成员函数的。加上成员变量就没有问题了，中间 ppt 有个地方有点问题，改了几个传进的参数就好了；

2、在设计页面的时候背景不知道为啥是黑色的，一直到验收我都是加上了白色的背景，知道验收的时候美女师姐 TA 帮我把 style 改了之后才显示正常；

3、这次还是会出现闪退的情况，根据上次的经验肯定是代码哪里写崩了，我上次是变量声明没有放在 setContentView 后面，没想到这次还是这样，经过这次教训挺深刻了；

4、在实现 listView 的时候，我刚开始什么都没有显示，很郁闷，明明有这个元件却看不见，后来百度才明白 listview 默认背景色为黑色，字体为白色，然后我把字体和背景改了之后，能正常显示了，但是没有下划线，网上是用 view 做的，但是用了之后发现竟然闪退，后来百度搜了一下，原来我用的而是 ArrayAdapter 填充的，作为最简单的 Adapter，在布局文件只能容纳一个 TextView，然后我把它换成了只有一个 TextView 的布局，能正常跑了，验证了我的猜想。但是下划线怎么办？然后我换成了 SimpleAdapter，发现无论用什么布局，线都显示不对，然后我换成了 Button，竟然成功了，安卓真是个奇幻的东西.....

5、感觉这一次我深深的体验到了布局的深奥，我刚开始用了 RelativeLayout，发现如果里面嵌套其他布局的话，会整个乱掉，如果用 Constraint 布局的话，如果里面再嵌套一个的话，会发生很多不可预测的现象，例如闪退、图片无论设置权重多大都是一大块的现象，最后我是用 LinearLayout 里面嵌套两个相对布局，完美实现了。中途还发现商品详情页的图片会出现图片不同，导致在页面上看到的大小和比例也不一样，明明写了 height 是 match_parent 了呀？奇了怪，后来发现把 width 设为 match_parent 竟然好了，不应该是 wrap-content 吗？有点迷，不过总算是解决了，再次证明了安卓真是个奇幻的东西.....

四、 实验思考及感想

这次实验是自从上了移动应用开发这门课，安装了 android studio 后上的第一门课，是我第一次编写手机应用（虽然只是初级界面），是我第一次接触 xml 这种编程语言并第一次投入实践。做完之后，感觉大体上和我们大一的时候有一门课叫做移动信息工程基础有点像，只不过在那门课上，老师没有教我们编程语言之类的知识，而是用 apple 网站上的一个编写应用的在线工具来按照自己想法设计一个手机应用，那个说实话挺简单的，但没有这门课学的实用。毕竟有很多东西要把它的一行代码都吃透弄懂才能按照自己的想法来设计实现。感觉在这次实验中，学到了很多新的在课上没有讲过但很实用的东西，通过自己从一点一点从无到有地实现，真的觉得这次实验难度挺大的，而且中途觉得没问题的代码跑出来就是意想不到的错误，特别是布局这一块，需要调很久。这次实验的难点应该就是 activity 之间怎么通过 intent 实现切换和数据传递，还有适配器的自定义和运用。

在这次实验过程我碰到了不少的问题，但都通过自己努力和 TA 同学的帮助下解决了，比如刚开始编写界面的时候因为不熟悉软件，写的时候是一行一行代码写的，速度很慢，而且在实验过程中也是经常因为遇到些小挫折（代码有地方写错或是怎么），是非常耗费时间的。后来看到同学那种简便的操作，我才意识到之前方法的效率之低，原来可以把模块拖到界面，然后代码会自动生成，而不用人为输入；在写 RecyclerView 的 Adapter 的时候，一直不知道 `mRecyclerView.smoothScrollToPosition()` 是来自哪里的东西，加上去老报错，而且构造函数的变量也没有出现过，正当迷茫的时候，构造函数这次词点醒了我，就像我们写 c++ 代码的时候，我们是在 class 里面声明了成员变量后，才有对应的成员函数的。加上成员变量就没有问题了，中间 ppt 有个地方有点问题，改了几个传进的参数就好了；在设计页面的时候背景不知道为啥是黑色的，一直到验收我都是加上了白色的背景，知道验收的时候美女师姐 TA 帮我把 style 改了之后才显示正常；然而这次还是会出现闪退的情况，根据上次的经验肯定是代码哪里写崩了，我上次是变量声明没有放在 setContentView 后面，没想到这次还是这样，经过这次教训挺深刻了；在实现 listView 的时候，我刚开始什么都没有显示，很郁闷，明明有这个元件却看不见，后来百度才明白 listView 默认背景色为黑色，字体为白色，然后我把字体和背景改了之后，能正常显示了，但是没有下划线，网上是用 view 做的，但是用了之后发现竟然闪退，后来百度搜了一下，原来我用的而是 ArrayAdapter 填充的，作为最简单的 Adapter，在布局文件只能容纳一个 TextView，然后我把它换成了只有一个 TextView 的布局，能正常跑了，验证了我的猜想。但是下划线怎么办？然后我换成了 SimpleAdapter，发现无论用什么布局，线都显示不对，然后我换成了 Button，竟然成功了，安卓真是 个奇幻的东西.....

感觉这一次我深深的体验到了布局的深奥，我刚开始用了 RelativeLayout，发现如果里面嵌套其他布局的话，会整个乱掉，如果用 Constraint 布局的话，如果里面再嵌套一个的话，会发生很多不可预测的现象，例如闪退、图片无论设置权重多大都是一大块的现象，最后我是用 LinearLayout 里面嵌套两个相对布局，完美实现了。中途还发现商品详情页的图片会出现图片不同，导致在页面上看到的大小和比例也不一样，明明写了 height 是 match_parent 了呀？奇了怪，后来发现把 width 设为 match_parent 竟然好了，不应该是 wrap-content 吗？有点迷，不过总算是解决了，再次证明了安卓真是 个奇幻的东西.....