

Android 案例开发 实验报告

成绩	
----	--

实验名称 实验三 修改和删除我的支出数据并显示统计信息的设计与调试

专业班级 软件 172 班级学号 201715030208

姓 名 朱洪龙 实验日期 2019/03/29

(报告内容包括：实验题目、实验环境、实验步骤、实验结果、遇到的问题 and 解决办法、实验小结等。)

【实验题目】

1. 可以先实现我的支出中 `ListView` 内所有 `item` 数据的点击事件。
2. 设计一个 `Activity`，用来显示我的支出和收入数据（可以先判断传过来的数据类型再来显示），之后实现其修改和删除按钮的点击事件即可。
3. 实现数据管理，其中设置按钮点击事件来可视化支出和收入（通过统计图），余下的空间默认显示便签信息。
4. 剩下的系统管理（设置登录密码）、收支便签（添加一条文字信息）、帮助（显示一个 `HTML5` 的页面信息）和退出（一个 `finish()` 即可）都比较简单。

【实验环境】

OS: Windows 10

IDE: Android Studio 2.3.2

AVD: Nexus 5X API 23

minSdkVersion: 21 (android 5.0)

targetSdkVersion: 23 (android 6.0)

【实验步骤、程序调试过程中所有遇到的问题和解决办法】

设置点击事件：

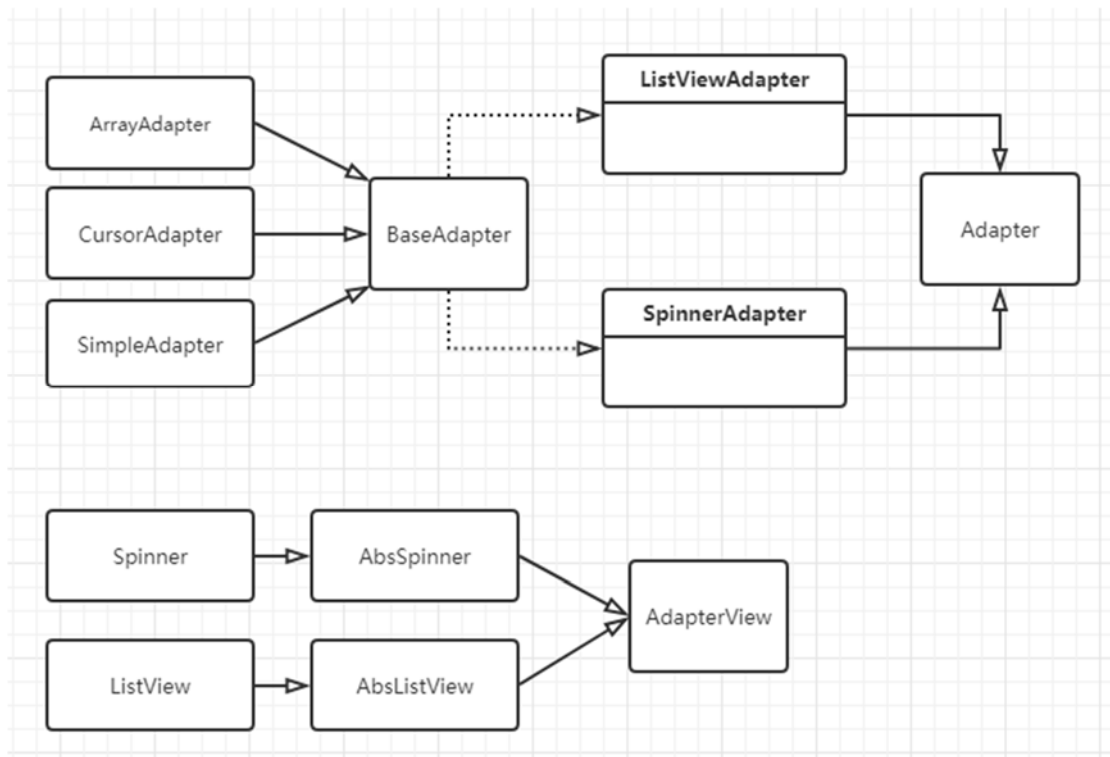
在收入信息中对显示的 `ListView` 设置 `setOnItemClickListener()` 点击事件，在方法 `onItemClick()` 中，先将该支出/收入信息中的编号截取出来（后面得通过它来查找到对应的数据并进行显示以供用户修改或删除），然后通过 `putExtra()` 方法将其与此操作的数据类型（支出还是收入）等信息一并带到另一个即将启动的 `Activity` 中。这里唯一需要注意的地方就是 `putExtra(String name, String value[])` 方法中的 `name` 参数，这个参数设为静态常量(`Flag`)，通过文档可知，后面的 `Activity` 若要获取刚才传入的数据的话，就得通过 `prefix`(前缀)来调用这个 `name`。(例如：`Outaccountinfo.Flag`)。

创建 `InfoManage`(继承自 `Activity`)并设计其布局文件：

布局文件跟新增支出/收入类似（在支出和收入布局 `TextView` 中不同的地方有标题和地点/付款方，这些地方可以不设置，因为在 `InfoManage` 中会根据传过来的类型进行设置），不在赘述。在 `InfoManage` 中，获取 9 个组件（相同的有 7 个，还有两个是地点/付款方），然后获取传过来的数据，用 `getStringArray()` 获取。接着判断支出/收入，对应其类型设置其特殊的 `TextView`，然后根据传过来的 `id` 值调用 `DAO` 数据库访问对象中的 `find()` 方法返回一个收入/支出模板对象，将所有数据设置到 `EditText` 中，**难点**在于设置其中的一个下拉列表中的数据。因为收入和支出的类型不同，所以得先判断后再设置下拉列表中的 `array` 数据。根据判断结果，通过 `ArrayAdapter` 调用 `createFromResource()` 返回一个字符队列适配器，其所需参数为 `Context context`(`this` 即可)、`int textArrayResId`(在 `value` 目录中定义的字符资源中设置的 `id`(其通过转换为 `R` 类中的 `int` 型来进行索引))、`int textViewId`(设置显示数据的视图类型)，最后再通过 `Spinner` 对象调用 `setAdapter()` 来设置。接着通过 `setPrompt()` 设置其显示的类型(`setPrompt()` 中的参数为 `CharSequence`，是一个接口，而模板返回的是一个 `String` 类型，这是典型的接口回调技术，由于 `String` 类实现了这个 `CharSequence` 接口中的方法)，最后就是实现修改和删除按钮的点击事件了。两个按钮都得判断收入/支出类型，修改就是建立个数据模板，将编辑框中的数据

设置到模板中（别忘了模板中还有个 id），然后调用 DAO 中的 update 方法并弹出提示信息即可（Spinner 中的数据通过 `getSelectedItem()` 方法获得）；删除就更简洁了，有了 id 值就可直接判断收支类型，调用 DAO 中的 delete 方法即可。

拓展：



实现数据管理

新建一个继承自 `Activity` 的 `Showinfo` 类，其中在顶部水平设置三个按钮，通过点

击分别可以显示收入、支出和便签信息，其中收入和支出通过一个 `intent` 进入另一个 `Activity` 进行直方图的显示；而便签则为默认的信息显示在下方空白处（通过一个 `ListView` 组件）。三个按钮代表三个不同的信息类型，所以可以将显示信息作为一个方法封装起来作为 `ShowInfo(int intType)` 方法使用，只需传入一个资源参数。在 `ShowInfo` 方法中，通过 `Switch` 来对传入的类型分别进行处理。收入/支出类型时，建立 `Intent` 对象并传入此类型（收入/支出）；若为便签，则通过 `DAO` 对象查找所有的数据返回一个 `List` 集合，然后遍历这个集合，格式化每条数据，然后新建适配器并进行设置。对便签中的数据项设置点击事件，将此便签的 `id` 通过 `intent` 传递到 `FlagManage` 中，`FlagManage` 与 `InfoManage` 类似。

这里的**难点**主要是设计直方图并将数据显示出来，在 `eclipse` 中设计好的直方图搬到 `AS` 这里来显示时，效果差强人意（同比缩小了好几倍，文字略显模糊，难道是 `API` 版本问题？还是 `IDE` 问题？），因此需要修改数据，但是这些数据基本都是联系的，一个一个修改太麻烦了，所以得找到这些数据最终的依赖项，没错，就是横竖的坐标轴的数据。在 `model` 包中创建一个 `Point` 类来描述一个点的位置，然后在图表中将表示横竖轴的起点和终点都换成 `Point` 来表述，由此，其它可依靠此数据进行适量增减常量即可，一些像字体大小、画笔大小等设为常量，修改因此更加简便。

直方图显示支出或收入信息，因此这里有两个判断的地方，一个是取到需要显示在横轴下的一组收入或支出的类型，另一个是通过 `DAO` 取出对应的数据。整个过程就是在 `Activity` 主视图上显示一个自定义的 `View`，因此可以创建一个内部 `View` 类，然后在 `onCreate()` 中通过 `addView (View child)` 方法加入进来即可。建立一个 `MyView`，调用 `View` 默认的构造方法，重写 `onDrag()` 方法。在 `onDrag()` 中调用默认的已写好的 `onDrag(Canvas canvas)`，其已为我们初始化好了画板，我们只需通过画笔渲染即可。创建画笔，设置笔触宽度，抗锯齿，笔触颜色，然后画出横竖坐标轴；**难点**在柱形图这里，首先设置画笔为填充样式，首先获得所有金额数据（一个浮点型数组对象），然后获得其中的最大值。因为柱形图中代表收支类型的数据的高度都得有个参照物，也就是设置个比例。至于为何不是取最小值，显然，这个柱形图得约束不得超过横竖坐标轴的表示范围，所以得用其中的最大值，将其约束在范围内。至于这个约束方法，得先获得 `x`、`y` 轴所能表示的范围的数值（`x` 只需获得第一个柱形图的起点 `x` 位置，`y` 只需设置其终点，即不能超过的位置即可，然后设置一个 `height`，用来表示所能表示的柱形图的最适高度（这个高度为其它柱形的参照（**实体**），而最大金额是其它金额的参照（**抽象**）），因此，其中的**难点**主要在于如何将实体与抽象联系起来。画柱形图用到 `drawRect()` 方法其中的参数为起点 `x`，起点 `y`，对角 `x`，对角 `y`（根据对角线确立一个矩形）。上面已经有了起点 `x`，只需按照每个柱形图的位置增加一定的间隔和 `i` 个柱形图宽度即可，主要是这个起点 `y` 的确立。实体与抽象的联系，`y-height/max*money[i]`，

这个数组中的最大值就是 `max`，即整个最大值为 `y-height`，这个就是最高的柱形图的起点 `y` 值了。

【实验结果】

D	03-19 09:04:07.130	1274	1274	com.livejq.activity	AndroidRuntime	Shutting down VM
W	03-19 09:04:07.130	1274	1274	com.livejq.activity	dalvikvm	threadid=1: thread exiting with uncaught exception (group=0x40e71930)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	FATAL EXCEPTION: main
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	java.lang.RuntimeException: Unable to instantiate activity ComponentInfo{com.livejq.activity/com.livejq.activity.InfoManage}: java.lang.NullPointerException
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	on
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2106)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2230)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.app.ActivityThread.access\$400(ActivityThread.java:141)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:1234)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.os.Handler.dispatchMessage(Handler.java:99)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.os.Looper.loop(Looper.java:137)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.app.ActivityThread.main(ActivityThread.java:5041)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at java.lang.reflect.Method.invokeNative(Native Method)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at java.lang.reflect.Method.invoke(Method.java:511)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:793)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:560)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at dalvik.system.NativeStart.main(Native Method)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	Caused by: java.lang.NullPointerException
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.content.ContextWrapper.openOrCreateDatabase(ContextWrapper.java:229)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.database.sqlite.SQLiteOpenHelper.getWritableDatabase(SQLiteOpenHelper.java:224)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at android.database.sqlite.SQLiteOpenHelper.getWritableDatabase(SQLiteOpenHelper.java:164)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at com.livejq.dao.OutaccountDAO.<init>(OutaccountDAO.java:21)
E	03-19 09:04:07.170	1274	1274	com.livejq.activity	AndroidRuntime	at com.livejq.activity.InfoManage.<init>(InfoManage.java:39)

在我的支出中点击其中的数据项时跳转到 `InfoManage` 中显示详细信息，在 `InfoManage` 中发生了如上的空指针错误，经老师提醒，仔细查看后发现：`InfoManage` 调用 `DAO` 时，是在声明属性的地方直接实例化一个对象，然后返回作为其中一个属性来使用。但是 `Activity` 的机制并非想象的如此执行，它是先调用 `onCreate()` 方法来初始化，而属性是后面才进行初始化和空间的分配的，然而在 `onCreate()` 中却调用了还没实例化的空对象（对于 `InfoManage` 是属性）所引起的空指针异常的情况：

Android is starting...



Starting apps.



LTE 2:28

数据管理

支出汇总

收入汇总

便签信息

1|This is test

Unfortunately, 个人理财通 has stopped.

OK





LTE 4:17

数据管理

便签管理

请输入便签，最多输入200字

This is test Code

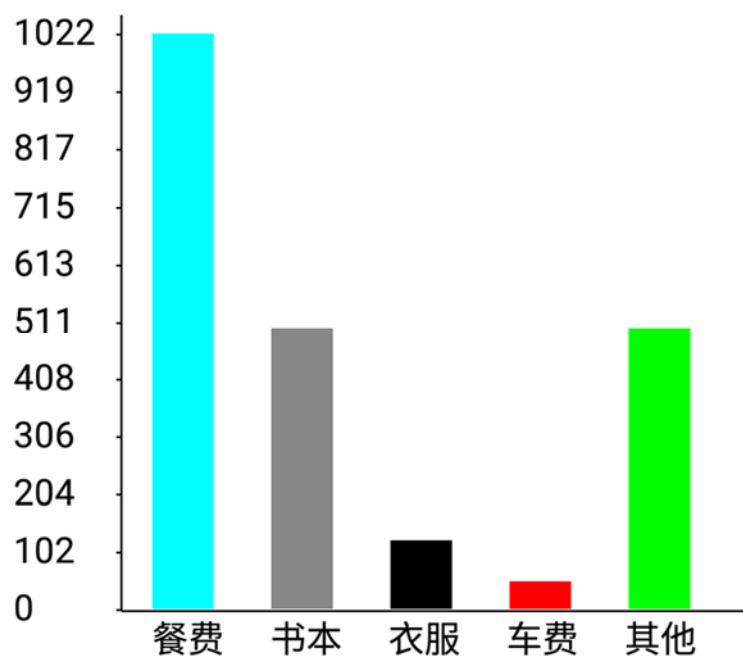
修改

删除



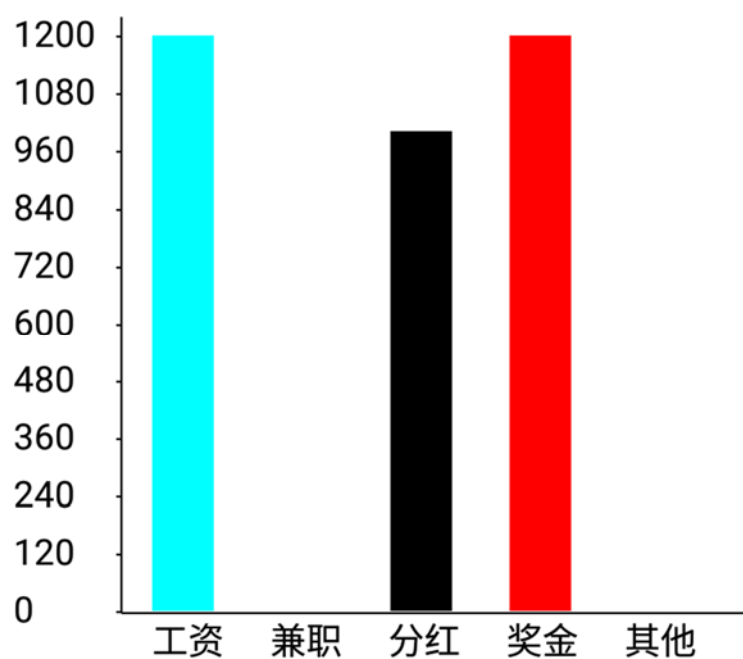
统计图

个人理财通的支出统计图



统计图

个人理财通的收入统计图





LTE 12:03

数据管理

支出汇总

收入汇总

便签信息

1|This is test

2|zhl 208





LTE 12:04

收支管理

支出管理

金 额: 12.0

时 间: 2019-3-29

类 别: 餐费 ▼

地 点: supermarket

备 注: test

【数据】 修改成功!

修改

删除



上午11:29

4G

个人理财通



新增支出



新增收入



我的支出



我的收入



数据管理



系统设置



收支便签



帮助



退出

【实验小结】

现在开发的 app(android 6.0)安装到我们现在的手机出现不适配问题(全面屏 android 8.1.0), 下面出现一部分“黑屏”现象, 如上,可能跟 API 版本有关, 有待探究....