|  |
| --- |
| **一、实验目的** |
| * 掌握RecyclerView中数据的增加删除 * 掌握RecyclerView添加ContextMenu |

|  |
| --- |
| **二、实验环境** |
| Android Studio |
| **三、实验内容**  上下文菜单是Android比较常用的功能，实现上下文菜单可以参照[RecyclerView添加ContextMenu的两种方案详解](https://blog.csdn.net/MakerCloud/article/details/82756355)   [紧凑型实现](https://stackoverflow.com/questions/51074871/my-recyclerview-contextmenu-doesnt-show)  本次实验我们在上个实验程序的基础上添加ContextMenu，包含增加、修改、删除等3个菜单项。点击添加菜单项，显示另一个Activity，Activity返回后就添加数据到主Activity的RecyclerView中，请用课堂上演示的ActivityResultLauncher实现。在实验报告中描述你实现的关键代码和演示效果。请把你完成的代码在同一个Gitee/GitHub项目上上传。我们的Adapater如果用数组保存数据时，由于Java里数组不能动态改变大小，因此不太方便。而采用模板类ArrayList能有效避免这个问题。Adapter的notifyDataSetChanged可告知RecyclerView刷新界面，此时是更新全部数据。注意RecyclerView缓存的项目可能有限，同时它还有一些局部刷新的功能，在使用时要尽量使用局部更新的函数。 比如 https://blog.melove.net/android-develop-recyclerview-notifydatasetchanged-pit/  尝试给主窗口的RecyclerView添加一个ContextMenu实现删除的功能。  完成实验后，调查ContextMenu实现的原理和过程，请分析我们给出的两个链接的三种实现ContextView的方法的区别和异同。为什么有时候要放在Activity中去生成上文菜单并响应？  区别：  方案一：从onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)方法传递进来的menuInfo中获取到相应的position信息。  方案二：在Adapter的ViewHolder中为每个itemView设置setOnLongClickListener监听，然后在长按监听回调中设置当前的position，为每个itemView设置setOnCreateContextMenuListener监听，通过上面记录的position来执行相应的动作。  方案三：在ViewHolder适配器类中实现这些方法  异同：  方案一和方案二都是在Activity或Fragment中处理ContextMenu相关逻辑，而方案三将ContextMenu相关逻辑封装在ViewHolder适配器类中。  方案一需要从menuInfo中获取position信息，而方案二和方案三都是在监听回调中记录position信息。  方案一和方案二都需要在每个ViewHolder中设置监听器，而方案三只需要在ViewHolder适配器类中实现相关方法。  方案二和方案三比方案一更灵活，能够根据不同的需求自定义ContextMenu的生成和处理逻辑。  有时候需要在Activity中生成上下文菜单并响应，是因为上下文菜单的处理逻辑可能和数据源以及其他UI组件有关联，需要Activity层面的操作。具体来说，可能存在以下情况：  1.上下文菜单的处理逻辑需要对数据源进行修改或删除。比如，在RecyclerView中长按某个item后弹出上下文菜单，其中包含“删除”选项，此时点击“删除”选项需要删除该item并在RecyclerView中更新UI。这种处理逻辑需要在Activity中实现，因为它需要对RecyclerView绑定的数据源进行修改。  2.上下文菜单的处理逻辑需要和其他UI组件交互。比如，在ListView中长按某个item后弹出上下文菜单，其中包含“分享到社交媒体”选项，此时点击“分享到社交媒体”选项需要调用系统分享功能，并将要分享的链接、图片等信息传递给分享功能。这种处理逻辑需要在Activity中实现，因为它需要与系统分享功能交互。  3.上下文菜单的生成和处理需要跨越多个Fragment或View，需要在Activity中协调。比如，在ViewPager中使用Fragment展示多个RecyclerView，每个RecyclerView都需要添加上下文菜单。此时，上下文菜单的生成和处理需要跨越多个RecyclerView及其所属的Fragment，并且需要在Activity中协调处理。  4.在这些情况下，将上下文菜单的生成和处理逻辑放在Activity中会更加方便和合理。同时，在实现上下文菜单相关逻辑时，还需注意避免Activity和其他UI组件之间的耦合过度。 |

**四、实验及分析**

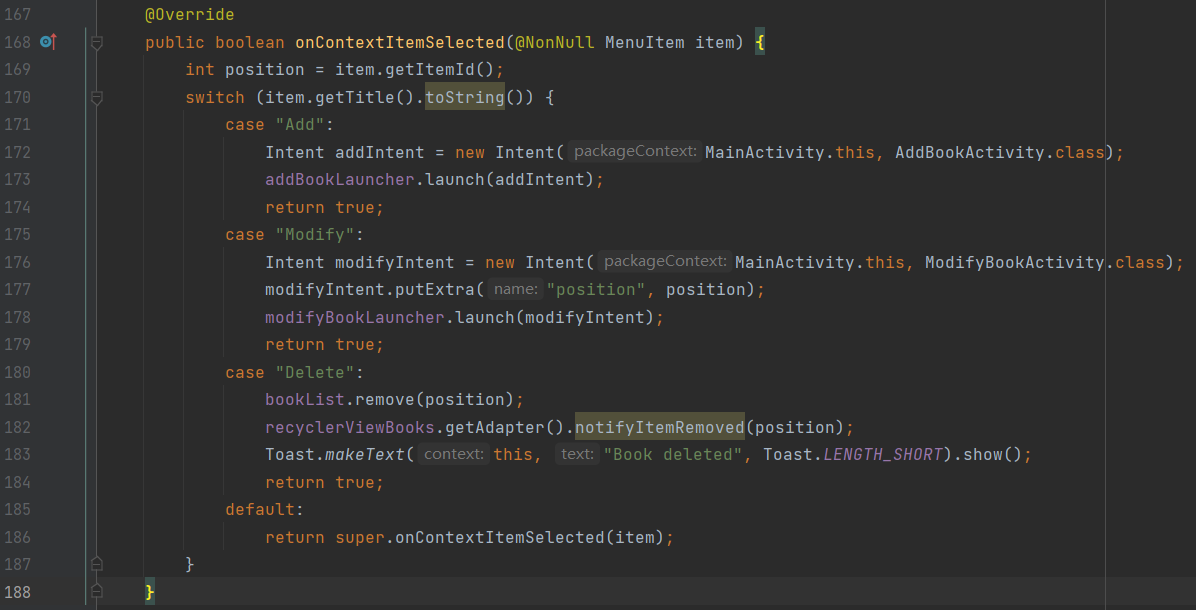
实现添加菜单项并使用ActivityResultLauncher的关键代码如下所示：

1.在onCreate()方法中注册ActivityResultLauncher，并实现回调函数。



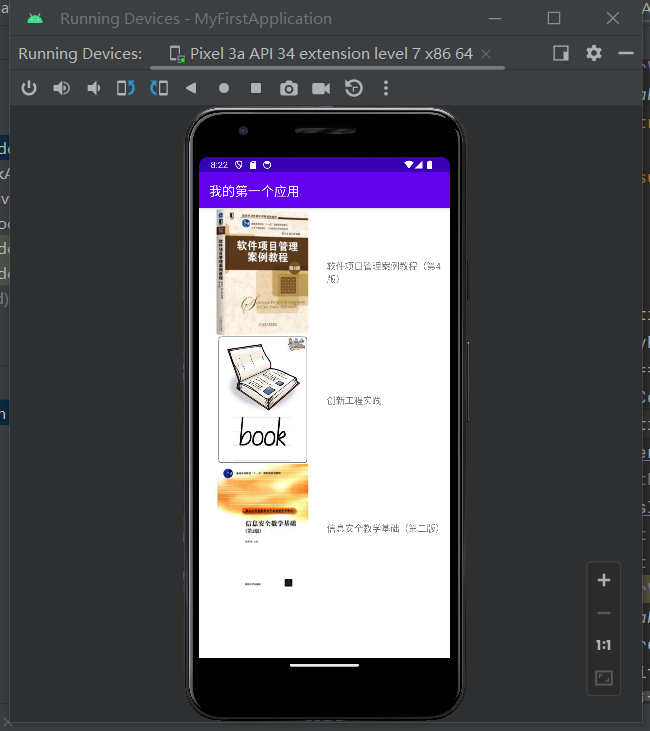


2.在onCreateContextMenu()方法中为每个菜单项设置点击事件。

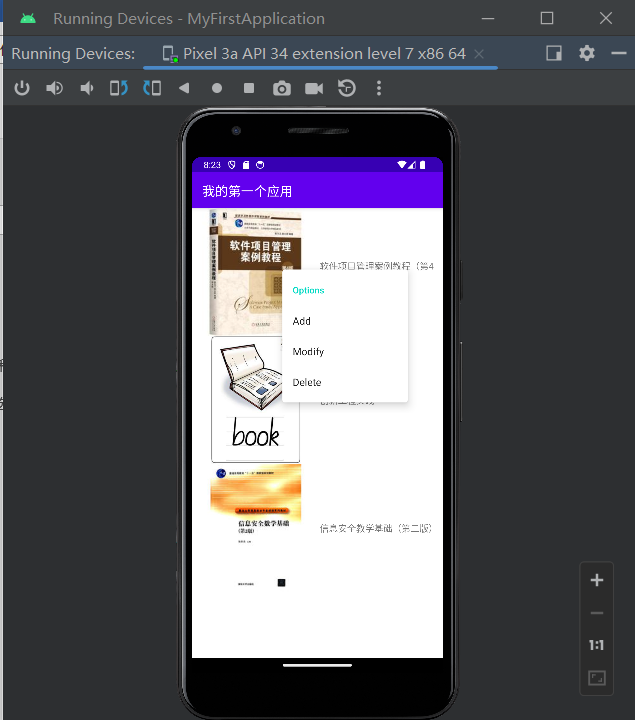


演示效果如下：

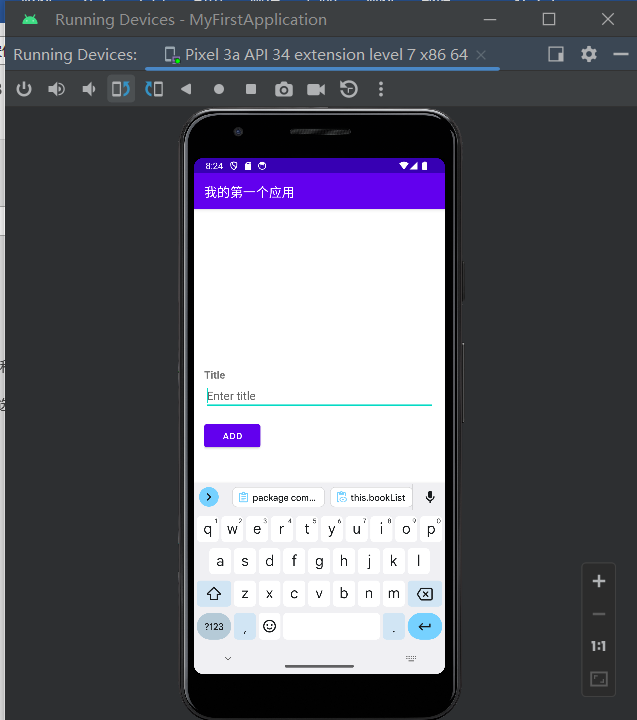
1.初始页面。



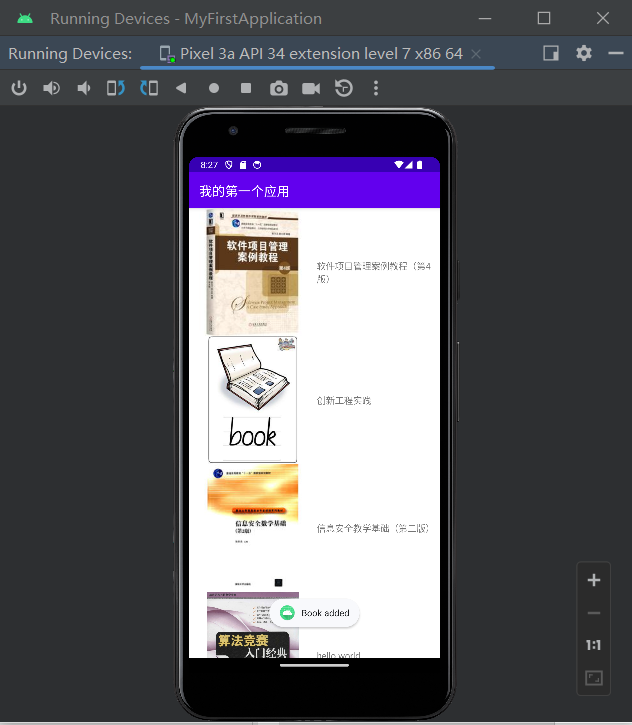
2.菜单项。



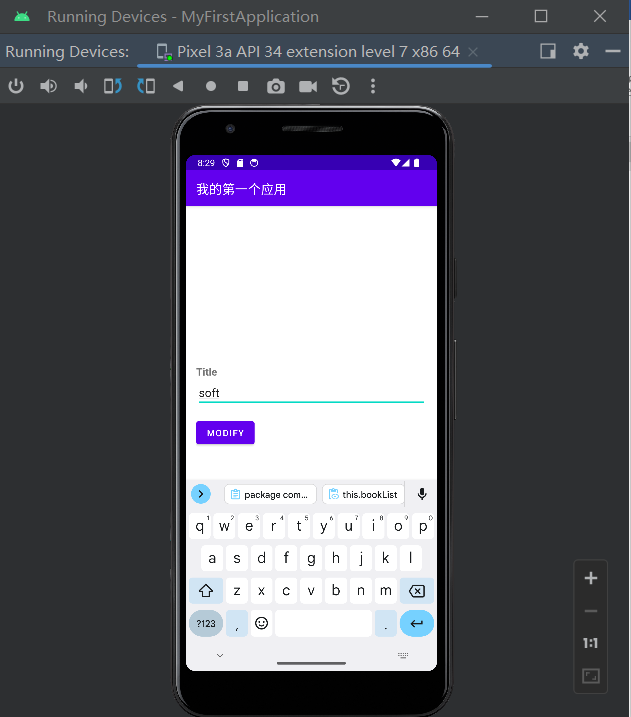
3.Add页面。



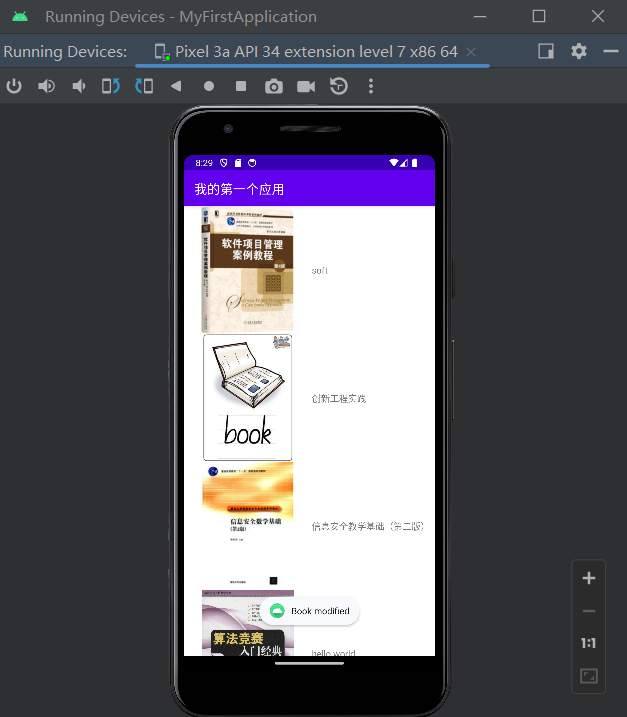
4.Add后。



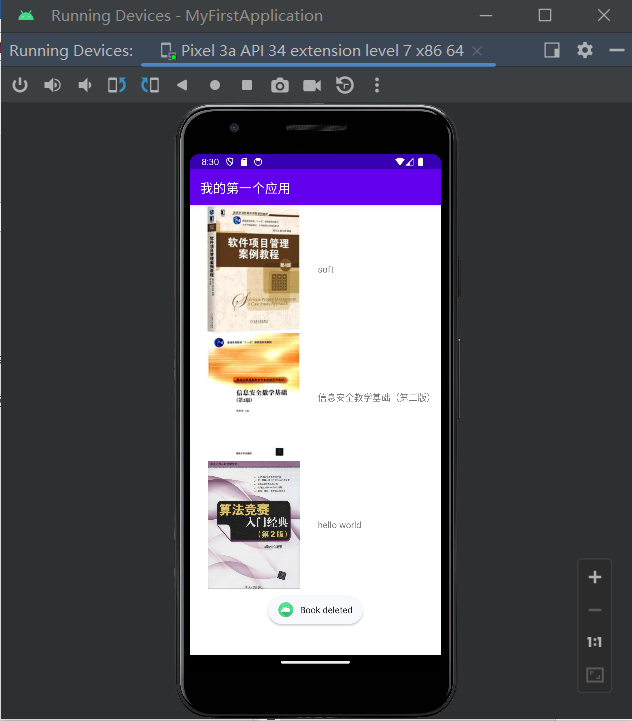
5.Modify界面。



6.Modify后。



7.Delete后。



**五、实验总结**

本次实验的主要目标是在现有的实验程序的基础上添加ContextMenu，包含增加、修改和删除等三个菜单项，并使用ActivityResultLauncher实现跳转到另一个Activity并返回数据后更新主Activity的RecyclerView。

在实验中，我首先注册了两个ActivityResultLauncher，分别用于处理添加图书和修改图书的操作。然后，在主Activity的RecyclerView中为每个列表项添加了长按弹出ContextMenu的功能。通过在创建ContextMenu时添加菜单项，并设置点击事件，实现了对应操作的跳转和数据更新。