## 2023年夏季Java小学期大作业

## 实验报告

朱泓舟 2022010804

1. **代码结构**

**1.1．App整体框架：**

* + 1. **.App概况：**本App的整体结构为一个Activity配合多个Fragment。App运行自始至终，此Activity一直运行，而界面均使用Fragment实现。App包含四大主界面，分别为新闻主界面、搜索主界面、历史界面和收藏界面。
    2. .**package概述**：数据类型包datatype，包括News、NewsList两个Java类，用以定义新闻对象；数据库包database，包括DatabaseHelper类，用以实现数据库，以便实现历史和收藏两大功能；主界面ui包，包括五个子package，分别为favorites（实现收藏主界面）、history(实现历史记录主界面)、news(实现新闻主页主界面)、search（实现搜索主界面），以及具体实现新闻列表、新闻详情的newsfragment包。此外，urladpter包用于从给定的网络接口获取数据。
    3. **.四大主界面的调用关系概览**：

**Mainactivity：**App启动后，Mainactivity创建底部导航栏和和用于盛装fragment的容器“fragmentContainer”。根据底部导航栏的点击情况，决定创建哪一种主界面的fragment，并将之置于fragmentContainer中。初始默认进入新闻主界面。

**新闻主界面：**包含分类（Category）栏目及选择分类的功能。在分类确定后，调用newsfragment包的类，显示新闻列表，并赋予点击进入详情页、记录历史记录等功能。

**搜索主界面：**提供搜索服务，当用户确定搜索索引后，调用newsfragment包的类，显示搜索结果列表。

**历史记录与收藏主界面：**这两大主界面均从Databasehelper提供的数据库读取信息，在确定新闻信息后，调用newsfragment包的相关类，来显示新闻列表。

* 1. **各部分代码作用：**

我们以package为单位对各个Java类进行说明。

**database包中Databasehelper类：**内置SQLite数据库的数据库类。采用单例模式，全局所有fragment共享一个此类实例。内含创建、插入、删除、查找功能，用于存储历史记录和收藏记录。

**datatype包**：News类：新闻类，四大主界面的核心数据类型。NewsList类：一个中介，目的是：在从网络接口获取新闻数据时，本项目采用了Gson库来实现字符串直接转成对象，然而由于返回结果只有data这一项是真正的新闻，所以无法直接转成News类，需要NewsList来作为一个中转。

**urladapter包**：GetUrl类：从网络接口获取新闻。包含设置接口参数功能、发起网络请求并获取返回值的功能。注：返回值是一个json格式的字符串。JsonParser类：将字符串转换成News类。NewsFetchr类：封装好上述两类，是urladapter包内唯一的对外接口，作用为接收接口参数，并返回News的ArrayList。

**ui包：**news包内：NewsCategory是主体界面，由于显示类别（category），并与 ChoiceAdapter关联，以显示具体的新闻列表；CategoryChoice用于选择类别。Search包内SearchFragment用于用户输入搜索索引并执行搜索。Newsfragment包内，NewsListFragment和NewsAdapter配合，用于具体显示新闻列表，DetailFragment用于显示新闻详情。favorites和history包中FavoritesFragment和HistoryFragment两类：从数据库读取存储的新闻，并显示历史/收藏列表。

1. **具体实现**

仍按照1.2的顺序进行介绍。

**2.0. Mainactivity：**两大部分：fragmentContainer和bottomNavigationView.前者用以盛装主界面，后者则用于四大主界面的导航。

**难点：**①如何实现四大主界面之间的切换(解决：底部导航)②如何保证切出新闻主界面fragment再切回后，能恢复原先的category列表（解决：在这个fragment与Mainactivity之间通过接口和bundle传参）。

**亮点：**“在安卓开发过程中，使用单个Activity和多个Fragment能够更好地管理和组织界面逻辑。” 单个Activity配合多个Fragment相比多个Activity的策略更具优势。因此，此项目采取前者。

**2.1. database包中Databasehelper类**：单例模式&&SQLite&&线程锁。具体而言：此类继承自SQLite辅助类SQLiteOpenHelper数据库共有两张table，分别为历史/收藏，在此类中实现了向它们嵌入/删除的方法。同时，为保证历史/收藏记录能在各大界面互相调用时保持一致，全部fragment应共有一个数据库。为此，设置静态私有变量instance，实现单例模式。与此同时，为避免线程错误，采用synchronized关键字，实现互斥。

**难点：**①各大fragment之间需要使用同一数据库②数据库读写需要线程互斥。，此两处的设计相对困难。

**亮点：**使用单例模式和自带的synchronized关键字，轻量级地解决上述问题（而非将数据库实例作为参数在各大fragment之间暴力传递）。

**2.2 datatype包**：普通的两个Java类，根据网络接口返回值实现即可

**2.3. urladapter包**

**①GetUrl类：**包含两部分内容：设置接口参数和发起网络请求。前者甚易不提，后者使用第三方库OkHttp3实现。

**难点**：发起网络请求需要在子线程进行，且子线程不能直接修改UI界面。（此问题算作这部分的难点，具体解决策略则详见2.4.①（新闻列表）的“难点部分”）。

**亮点：**采用OkHttp3库而非Java自带的网络库，提升性能，且免于复杂的IO机制（比如，若不使用OkHttp3，则需要手动逐行读取网络请求返回值，而OkHttp3这直接返回整个字符串）。

**②JsonParser类**：使用Gson库，将字符串直接转换成NewsList实例。

**亮点：**使用Gson库直接获得实例本身就是此处的亮点。

**③NewsFetcher类：**封装好上述两类，接受外部传来的接口参数，输出获得的新闻。

**2.4 ui包：**这是此项目的核心部分。由于其较为复杂，故此处不再按照package顺序，而是以功能为索引进行介绍。

**①分类（Category）设置：**在NewsCategory类中采取TabLayout+ViewPager2的模式，Tablayout负责显示Categories,ViewPager通过配合CategoryAdapter，实现点击Tab即可进入对应的Fragment。至于如何实现“增删Category”，则有另一个Fragment即CategoryChoice负责，两个Fragment之间通过bundle传参，以实现Category参数的同步化。

**难点：**如何实现Category参数的同步化？解决：通过bundle传参。

**亮点：**增删Category动画效果较为美观。

**②搜索设置：**SearchFragment只负责面向用户获取搜索索引，采用单选框RadioGroup、EditView实现与用户的交互。

**亮点：**（也是①②两项共有的亮点）注意到，所谓“按照Category检索新闻”与Search本质上均为设置网络接口的参数，所有这两个界面本质功能是完全相同的。既然如此，新闻主界面和搜索主界面的设置，就都是先获得用户输入，尔后将此作为参数，传递给一个新创建的NewsListFragment，显示新闻/搜索列表由它负责。这无疑提高了各大类的解耦合性，符合OOP的编程思想。

**③新闻/历史/收藏列表：**三个类都具有新闻列表功能：NewsListFragment、HistoryFragment、FavoritesFragment。事实上，本应该将这三类抽象成一个基类，但由于前期工作量较大，重构成本较高，故未实现。具体而言，NewsListFragment是根据传入的bundle中参数，设置NewsFetcher参数，并获取News实例，而HistoryFragment、FavoritesFragment则是从数据库直接读取，这是三者的差别所在。三者共性：将News实例可视化。此处，整体架构方面采用了Recyclerview，并辅以NewsAdapter这一Recyclerview所必需的适配器。细节处理方面：“上拉加载，下拉刷新”采用了第三方库SmartRefreshLayout，其中上拉加载是将接口参数page+1，下拉刷新则是重新进行网络请求；列表中的文字、图像、视频显示，均在NewsAdapter实现，分别采用了TextView、Glide、VideoViewer；历史记录的灰名，则通过设置NewsAdapter的onItemClickListener接口，在点击后将被点击条目的新闻参数嵌入数据库；而在NewsAdapter中，再读取所有为历史记录的标题，只要与当前标题气合，就设为灰色。

**难点：**此处核心难点即为网络请求时线程机制。注意到安卓UI主线程不允许网络请求，且只要5秒钟无响应就出现ANR问题。因此，不仅不能在主线程进行网络请求，而且不能在主线程等到子线程结束再继续运行。这导致一个问题：如何获取子线程返回的News数据，并“当且仅当”子线程结束后再设置UI?解决方案：在子线程通过Handler.post方法切换到主线程。

**亮点：**在图片处理过程中，有些图片网络链接“无权访问”，不能加载。为此，如何保证“仅当图片能正常加载”才显示ImageView？如果不能实现这点，就会导致大量非法链接引入空白占位图片，极不美观。解决方法：在布局xml中设置ImageView属性为GONE，不可见；利用Glide的addListener机制，设置仅当onResourceReady才让ImageView的属性变成VISIBLE。

**④新闻详情：**DeatailFragment类。此类采用ScrollView实现滚动界面，内容呈现方式与NewsAdapter相同。

**难点：**如何在新闻列表中实现点击进入详情页？解决方案：在NewsAdapter里设置onItemClickListener接口，并在新闻列表的三个类里将此接口设为具有切换fragment的功能。

**亮点：**解决上述难点即为亮点。

1. **总结与心得**

此项目花费时间从第一周周四至验收，共计三周。其中，我第一周的前几天（周四至周日）用于完成“原始积累”。这一过程中，我深感知识“从无到有”的不易：面对纯自学的环境（当时还没讲到安卓），从何入手来了解安卓的开发方式？具体而言，安卓开发的基本代码框架是什么？怎么能运行起一个简单的程序？要实现我的需求，需要哪些API?这些API有哪些参数、如何使用？为解决上述“入门”问题，我采用了如下方式：读参考书（优点在于系统地入门，缺点在于大量参考书没有示例代码、基本指导，讲述的多为“xml文件是什么”“四大组件的生命周期”，虽然这确实是基础性的内容，但很难说有助于迅速上手运行一个自己的程序）；查找Android Developers官网（优点在于权威详细、有时效性，缺点在于此处多为“怎么用”的解说，而我却先要解决为了实现功能而需要“用什么”的问题；而且，也显然没有所谓“手把手的入门操作”）；查找Stack Overflow等社区（优点在于“接地气”、有详细指导、代码范例，且能指出一些官网所为指出的bug及其原因、注意事项，缺点在于不系统）；研读小规模的安卓开源代码（当然，内容上与本次作业无关）+询问LLM某些代码是干什么的。在入门后，通过“面向LLM(提出需求，问它应该用哪个API，再到官网上学习，而非让它直接给出示例代码，因为其正确性堪忧)”“面向Google”“面向Android Developers”的学习方式，我逐步完成了项目的开发。

上述原始积累的过程是我小学期的最大收获。虽然将来不一定从事安卓开发，但在短时间内，从无到有地入门一个比较复杂的开发系统，此中种种困难与解决方案，无疑极大地提高了我自学与解决问题的能力。

当然，我的这一项目在架构上仍有遗憾：在项目完成后，我逐渐醒悟，目前的ui包下的分类的合理性尚可改进：将FavoritesFragment、HistoryFragment、NewsListFragment抽象成一个基类，作为后端的适配器，能够进一步提高代码的解耦合性与可读性。虽然现在由于时间精力所限，未能完成此部分的重构，但这一教训加深了我对代码逻辑性、封装性的理解：设计完善合理的工程，需要在动手操作前，初步论证好整体架构是什么样、为什么是这样，想好合理的结构、调用关系，而动手写代码时，某部分为何是这样的写法/为什么这么分类/为什么在某一个包里，都必须清晰明确。

至此，十分感谢许斌老师、各位助教，许老师详细深入的讲解帮助我建立了完善的Java基础知识结构，而助教们的答疑解惑、辛苦付出，也正是我能完成此项目的必要条件。

对于此课程的个人建议主要是，可否将安卓开发参考书提前发布？有鉴于市面上安卓教程良莠不齐，而我们的参考书又是老师优中选优、精挑细选而来，那么如能在布置大作业当日发布，或许有助于大家安卓方面的入门。