1月13日正式开始项目开发，之前虽然也断断续续做过一些准备性工作，如创建github的repository，但大都没有什么可以说的，故不写入此日志中。

1月13日

更名项目的包名为com.luoziyuan.notepal，更带有个人色彩。

学习金老师网站上的DBProgramMode事例，确定项目的基本结构。

直接进行打包测试，测试机的Android版本为4.4.2和7.0，均可正常安装运行。

使用ButterKnife框架简化控件使用。

简单的listview测试，如果activity是ConstraintLayout，listview需要Match Constraints，不然真机上会出现listview超出左右边界的问题。

SQLiteOpenHelper实现数据库创建和升级功能

DBSchema封装数据库相关信息，如数据库名，表名，字段名，为每个表定义一个静态类。其他代码全部通过这个类的静态字段使用这些信息，可以增强程序可读性。当数据库信息发生变化时，只需修改这个类即可。

1月15日

重组数据类，删除Item类，直接使用Note类作为基类，Exam，Homework，Affair作为派生类。

初步实现DBSchema、DataSource和DBHelper类。

可以实现数据库创建，访问，简单的数据插入，通过实机测试。

Android Studio自动把我的包名又加上了Android……

凡是使用activity的this作为context，必须要等到onCreate()方法中，否则会抛出NullPointerReference异常

sql语句要格外注意字符串，插入字符串数值时要用’’括起来

**数据库名，表名，字段名全部小写**

1月16日

新增view包，里面创建AddActivity类，用于进行添加操作。

实现Activity之间的通讯，可以通过AddActivity添加新记录。

实机测试，4.4.2版本上的Hint字体有问题，颜色和输入一样深。

7.0版本上点击输入框光标在Hint字符串中间。

下一步：对数据按日期排序，按日期降序显示；

数据绑定，实现动态刷新；

实现删除功能。

1月19日

Note类实现Comparable接口，按照日期进行排序。

实现添加新笔记后自动刷新listview。

一方面更新数据库，一方面更新listview，考虑到效率问题，不能每次更新数据库都重新读取一遍。采用直接向listview的数据集中插入新数据的方式，考虑到插入后的listview仍然要保持date降序排列，因此需要手动找出新记录应该插入的位置。同时，为了保证数据库中记录id和listview中记录id的一致性，放弃采用数据库自带的自增1功能，直接采用创建记录时的系统时间作为id。为了匹配System.currentTimeMillis()的long型返回值，数据库中该字段也设置为long型。（因未及时更新DataAdapter类中的代码造成了使用id删除记录时无效）

利用ButterKnife重构了一下DataAdapter类的代码。

OnItemLongClick必须是Boolean型。

添加ListView的OnItemClick事件（还未实现界面跳转）和OnItemLongClick事件（已实现单个删除功能）。

使用Bulder类创建消息框。

问题：当数据集为空时添加新记录不能动态刷新！

解决：手动向数据集添加记录时没有考虑数据集为空的情况，增加if语句即可。

通过实机测试。

1月20日

实现搜索功能，主界面最上面添加搜索框，listview能够随着搜索框内容动态刷新。

view包改名controller，Activity起的是controller的作用。

创建UpdateActivity，用于查看详情和更改。

实现查看详情和更改功能

优化显示，搜索框左侧添加放大镜图标，listview的contentText设置过长时省略。查看详情界面content从左上角开始显示。

问题：调用dataAdapter.notifyDataSetChanged()后listview不刷新

dataAdapter绑定的是List对象，如果把data赋以新的List对象，dataAdapter将不再起作用。

问题：maxEms不起作用。

设置singleline + ellipsize限制显示一行

问题：详细信息的content不自动换行。

设置inputType为textMultiLine

1月21日

问题：点击搜索框弹出输入法后，New按钮也跟着上去了……

在Manifest中MainActivity对应处加上android:windowSoftInputMode="adjustPan"

限制主题不能为空，添加/更改导致主题为空时，给出提示。

完成DBSchema类，含四个数据库的全部相关信息，更新DBHelper类。

创建Exam，Homework，Affair类，对应其他三种不同的记录类型。修改Note类，添加type字段，重载构造方法。

创建Builder，设置完后一定要show()！！！

选择记录类型的单选框不设置确定按钮：点一下就能完成的事为什么要点两下？

添加不同类型的记录——同一个Activity，加载不同布局。使用ButterKnife时添加@Nullable（字段）和@Optional（方法）。

使用DatePickerDialog和TimePickerDialog选择日期时间。

DatePickerDialog的月份是从0开始的！！！

listview显示不同样式的item，需要把DataAdapter的父类改为BaseAdapter（抽象类），重写一堆方法。已实现。

DataSource添加Exam的插入方法，所有类型的删除方法用同一个，使用switch(type)实现。

注意更新MainActivity的结果响应方法和AddActivity、UpdateActivity的相关代码。

使用Calendar类（java.util包）获取日期时间。

下一步：实现Exam的搜索、查看与更改；

实现Homework与Affair的CURD。

1月22日

脑子一热做了个压力测试，点击按钮插入2000条新记录。点了之后程序卡住，等了三分钟还没缓过来，直接手动kill掉了。重开发现已经插入了1000多条，显示正常，测试搜索性能良好，不过随便划了几下虚拟机就崩掉了。重开划了几下虚拟机又崩了。看了下应用信息，数据占用1.03MB。

使用String的substring(start,end)获取子串（左含右不含），使用Integer的parseInt(String)把String转Int。

实现Exam的改查功能。

打开软件时listview定位到今日的事务

问题：插入新记录到数据集的顺序不对，始终插入到最后。

add(**index**,element)！！！

使用Refactor重命名布局文件：

activity\_add->add\_note,

activity\_addex->add\_exam,

activity\_update->update\_node,

activity\_updatex->update\_exam,

控件ID还是得手动改……

真机上的DatePickerDialog和TimePickerDialog都只是个简单的框（4.4和7.0都是）……虚拟机上格外高端，忽悠人？

Homework类新增deadline字段，基类的date字段用于存放创建日期。

重命名布局文件：

listview->listview\_note

listviewex->listview\_exam

每种类型的记录有各自的显示、添加、查看布局。

更新DataSource的两个get方法，从四个数据表中获取四种记录类型。

在布局设计页面中设置各个Text的内容，增加可读性。

封装各个类的get方法，增加可读性。

考虑到更改功能使用删除+插入的方式实现的，故把所有数据类改为只读类。

完善DataAdapter类，每种类型一个ViewHolder类。

Homework类型是按createDate排序还是按deadline排序？

先用createDate排序便于测试。

重构MainActivity中添加和更改功能后续操作的代码。

1月23日

使用//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分割线划分代码区，将同一文件汇中不同模块的代码隔开，增加可读性

实现Homework类型记录的添加删除，实现Affair类型记录的添加删除

实现Homework类型和Affair类型的查看更改

问题：添加一个日期比现有记录都小的记录时，没有插入到数据集中

修改插入功能的代码

基本功能已齐！

问题：4.4手机DatePickerDialog和TimePickerDialog没有取消……

点击退回键日期文本仍然会改变，但是再次点击显示改前的，即使点确定更改一个日期再点开还是显示改前的……

1月24日

目标：添加AddActivity和UpdateActivity的ActionBar；

添加Note类型的content框改为从文字左上角开始显示，更符合一般人的习惯；

完善DBHelper类，保证数据库更新时数据不丢失；

ActionBar->在Manifest文件中设置Activity的theme，但是Studio已经默认了整个Application的theme为AppTheme

直接往Activity里添加相关东西（如label）就好

巨坑！！！

label死都不显示，因为AddActivity和UpdateActivity继承了Activity类，改为AppCompatActivity就好了。

完善DBHelper类，在onUpdate方法中，删除旧表之前把所有记录读取保存，创建新表后再插入，保证数据库更新时数据不丢失，在onCreate方法中添加代码，创建数据库同时插入引导用信息

如果只是清除应用数据，并不会删除数据库，只有删除应用才会跟着删除数据库

解决4.4版本上DatePickerDialog没有取消按钮：

单纯添加一个取消按钮并不能解决问题，因为即使点击取消按钮，onDateSet方法依然会被回调。也就是说，点击“取消”、点击手机返回键、点击空白区域 这3种操作的结果是一样的！

解决：不使用OnDateSetListener，而直接手动设置“完成”按钮和“取消”按钮的事件。

TimePickerDialog的getTimePicker还是TestAPI！！！先暂时放弃。

解决DatePickerDiaolog总是显示第一次设置的日期的问题。

下一步：在MainActivity中完善ActionBar，增加说明和清空记录。

更换桌面图标。

ActionBar的menu按键就是不！显！示！

去TMD Cast Button！！！用普通的Item即可！！！

添加about菜单按钮

1月25日

问题：菜单把图标遮住了

解决方案：在style中设置<item name="overlapAnchor">false</item>注意不要带android！！！（要求API21）

添加清空记录功能，DataSource类相应添加deleteAll()方法

优化listview的UI，增加iconText，dateText改为主题蓝色

搜索功能添加对date、deadline、time和place的搜索（除了id以外都在搜索范围内）

给textview添加边框：在drawable中创建xml文件，里面设置shape，然后设置textview的background为该文件

问题：查看Exam类型和Affair类型的记录，点击确定后即使不修改也会触发更新

检测“点击设定日期/时间/deadline”的位置不对，提前即可

问题：7.0以下版本无法通过font-family使用自定义字体

下载相应字体文件放在res/font下，使用typeface在加载textview时手动设置。注意getAsset()是context的方法

问题：添加完assets就tm无法编译？？？

Error:java.util.concurrent.ExecutionException: java.util.concurrent.ExecutionException: com.android.tools.aapt2.Aapt2Exception: AAPT2 error: check logs for details

个破字体，差点害的整个项目玩儿完

把icon全部换成图标算了……

使用ImageView，可以让图片自动缩放

res中所有资源文件名统统小写（layout,font,drawable……）

不要随便删除res中的文件

4.4测试ok，但是about还是盖住Menu……

森维发现的一个bug：创建Homework时不设deadline，查看时再设会崩……

已解决

4.4突然显示正常（颜色、hint）……

1月27日

添加最后一项功能，筛选分类（主界面中）

主界面添加selectButton，与原有的addButton合并到一个LinearLayout中

相应地修改DataSource类，把get方法细化为每种类型有一个无参数的get和一个带参数的get，原来的get方法改名为getAll

筛选对话框应为复选框

Note类增加静态方法getTypeCount，返回记录类型个数

筛选分类已实现

设置所有listview项的themeText为singleLine，ellipsize默认就是end

给搜索框添加一个删除按钮，点击后清空

添加一个ImageView，与searchText合并入一个ConstraintLayout

问题:在searchText中输入时，ImageView时大时小……

把ConstraintLayout的高度设置为定值即可

4.4上cursive字体也显示不出来……不过默认的那个效果还可以

修改Note类的compareTo方法，对于Homework类型的记录使用deadline进行排序。相应修改DataSource的insertHomework方法，使用deadline比较，进而插入到数据集中的正确位置。重写Homework类的getDate()方法，返回deadline，便于前述方法使用。