“小米便签”精读笔记

项目组长：计科2101 张文睿

负责任务：

精读笔记整合与格式调整

精读以下功能：

新建文件夹

查看文件夹

删除文件夹

修改文件夹名称

增加提醒

删除提醒

组员1：计科2102 吴锦洋

精读分析以下功能：

导出文本

进入清单模式

发送到桌面

自动保存便签

分享

组员2：大数据2102 吴佳瑞

精读以下功能：

新建便签

删除便签

移动便签

同步

组员3：大数据2102 凌国栩

精读以下功能：

修改字体大小

修改背景颜色

关键词搜索

识别电话号码

**目录**

[**1. 功能一：新建便签 8**](#_Toc148900479)

[**1.1. 功能介绍 8**](#_Toc148900480)

[**1.2. 业务请求与响应过程 8**](#_Toc148900481)

[**1.3. 类的作用 9**](#_Toc148900482)

[**1.3.1. 界面层相关的类 9**](#_Toc148900483)

[**1.3.2. 业务层相关的类 12**](#_Toc148900484)

[**1.3.3 模型层相关的类 12**](#_Toc148900485)

[**1.3.4 数据层相关的类 14**](#_Toc148900486)

[**2. 功能二：删除便签 15**](#_Toc148900487)

[**2.1. 功能介绍 15**](#_Toc148900488)

[**2.2. 业务请求与响应过程 15**](#_Toc148900489)

[**2.3. 类的作用 16**](#_Toc148900490)

[**2.3.1. 界面层相关的类 16**](#_Toc148900491)

[**2.3.2. 业务层相关的类 19**](#_Toc148900492)

[**2.2.3. 模型层相关的类 20**](#_Toc148900493)

[**2.2.4. 数据层相关的类 22**](#_Toc148900494)

[**3. 功能三：移动便签 23**](#_Toc148900495)

[**3.1. 功能介绍 23**](#_Toc148900496)

[**3.2. 业务请求与响应过程 23**](#_Toc148900497)

[**3.3. 类的作用 24**](#_Toc148900498)

[**3.3.1. 界面层相关的类 24**](#_Toc148900499)

[**3.3.2. 业务层相关的类 25**](#_Toc148900500)

[**3.3.3 模型层相关的类 27**](#_Toc148900501)

[**3.3.4 数据层相关的类 29**](#_Toc148900502)

[**4. 功能四：新建文件夹 30**](#_Toc148900503)

[**4.1. 功能介绍 30**](#_Toc148900504)

[**4.2. 业务请求与响应过程 30**](#_Toc148900505)

[**4.3. 类的作用 31**](#_Toc148900506)

[**4.3.1. 界面层相关的类 31**](#_Toc148900507)

[**4.3.2. 业务层相关的类 36**](#_Toc148900508)

[**4.3.3. 模型层相关的类 38**](#_Toc148900509)

[**4.3.4. 数据层相关的类 41**](#_Toc148900510)

[**5. 功能五：查看文件夹 43**](#_Toc148900511)

[**5.1. 功能介绍 43**](#_Toc148900512)

[**5.2. 业务请求与响应过程 43**](#_Toc148900513)

[**5.3. 类的作用 44**](#_Toc148900514)

[**5.3.1. 界面层相关的类 44**](#_Toc148900515)

[**5.3.2. 业务层相关的类 46**](#_Toc148900516)

[**5.3.3模型层相关的类 51**](#_Toc148900517)

[**5.3.4 数据层相关的类 52**](#_Toc148900518)

[**6. 功能六：删除文件夹 54**](#_Toc148900519)

[**6.1. 功能介绍 54**](#_Toc148900520)

[**6.2. 业务请求与响应过程 54**](#_Toc148900521)

[**6.3. 类的作用 55**](#_Toc148900522)

[**6.3.3. 模型层相关的类 62**](#_Toc148900523)

[**6.3.4. 数据层相关的类 63**](#_Toc148900524)

[**7. 功能七：修改文件夹名称 65**](#_Toc148900525)

[**7.1. 功能介绍 65**](#_Toc148900526)

[**7.2. 业务请求与响应过程 65**](#_Toc148900527)

[**7.3. 类的作用 66**](#_Toc148900528)

[**7.3.1. 界面层相关的类 66**](#_Toc148900529)

[**7.3.2. 业务层相关的类 71**](#_Toc148900530)

[**7.3.3. 模型层相关的类 73**](#_Toc148900531)

[**7.3.4. 数据层相关的类 76**](#_Toc148900532)

[**8. 功能八：导出便签 77**](#_Toc148900533)

[**8.1. 功能介绍 77**](#_Toc148900534)

[**8.2. 业务请求与响应过程 78**](#_Toc148900535)

[**8.3. 类的作用 79**](#_Toc148900536)

[**8.3.1. 界面层相关的类 79**](#_Toc148900537)

[**8.3.2. 业务层相关的类 84**](#_Toc148900538)

[**8.3.3. 模型层相关的类 86**](#_Toc148900539)

[**8.3.4. 数据层相关的类 91**](#_Toc148900540)

[**9. 功能九：关键字查询 93**](#_Toc148900541)

[**9.1. 功能介绍 93**](#_Toc148900542)

[**9.2. 业务请求与响应过程 93**](#_Toc148900543)

[**9.3. 类的作用 93**](#_Toc148900544)

[**9.3.1. 界面层相关的类 93**](#_Toc148900545)

[**9.3.2. 业务层/模型相关的类：Activity 95**](#_Toc148900546)

[**9.3.3.数据层相关的类 95**](#_Toc148900547)

[**10. 功能十：修改字体大小 96**](#_Toc148900548)

[**10.1. 功能介绍 96**](#_Toc148900549)

[**10.2. 业务请求与响应过程 97**](#_Toc148900550)

[**10.3. 类的作用 97**](#_Toc148900551)

[**10.3.1. 界面层/业务层相关的类 97**](#_Toc148900552)

[**10.3.2. 模型层相关的类 99**](#_Toc148900553)

[**10.3.3. 数据层相关的类 100**](#_Toc148900554)

[**11. 功能十一：修改背景颜色 100**](#_Toc148900555)

[**11.1. 功能介绍 100**](#_Toc148900556)

[**11.2. 业务请求与响应过程 101**](#_Toc148900557)

[**11.3. 类的作用 101**](#_Toc148900558)

[**11.3.1. 界面层/业务层相关的类 101**](#_Toc148900559)

[**11.3.2. 模型层相关的类 103**](#_Toc148900560)

[**11.3.3. 数据层相关的类 104**](#_Toc148900561)

[**12. 功能十二：清单模式 105**](#_Toc148900562)

[**12.1. 功能介绍 105**](#_Toc148900563)

[**12.2. 业务请求与响应过程 105**](#_Toc148900564)

[**12.3. 类的作用 106**](#_Toc148900565)

[**12.3.1. 界面层相关的类 106**](#_Toc148900566)

[**12.3.2. 模型层相关的类 112**](#_Toc148900567)

[**12.3.3. 数据层相关的类 117**](#_Toc148900568)

[**13. 功能十三：发送到桌面 119**](#_Toc148900569)

[**13.1. 功能介绍 119**](#_Toc148900570)

[**13.2. 业务请求与响应过程 119**](#_Toc148900571)

[**13.3. 类的作用 120**](#_Toc148900572)

[**13.3.1. 界面层相关的类 120**](#_Toc148900573)

[**13.3.2. 模型层相关的类 121**](#_Toc148900574)

[**13.3.3. 数据层相关的类 124**](#_Toc148900575)

[**14. 功能十四：自动保存便签 125**](#_Toc148900576)

[**14.1. 功能介绍 125**](#_Toc148900577)

[**14.2. 业务请求与响应过程 126**](#_Toc148900578)

[**14.3. 类的作用 127**](#_Toc148900579)

[**14.3.1. 界面层相关的类 127**](#_Toc148900580)

[**14.3.2. 模型层相关的类 128**](#_Toc148900581)

[**14.3.3. 数据层相关的类 131**](#_Toc148900582)

[**15. 功能十五：识别电话号码 132**](#_Toc148900583)

[**15.1. 功能介绍 132**](#_Toc148900584)

[**15.2. 业务请求与响应过程 132**](#_Toc148900585)

[**15.3. 类的作用 133**](#_Toc148900586)

[**15.3.1. 界面层/业务层相关的类 133**](#_Toc148900587)

[**15.3.2. 模型层相关的类 135**](#_Toc148900588)

[**15.3.3. 数据层相关的类 137**](#_Toc148900589)

[**16. 功能十六：同步 138**](#_Toc148900590)

[**16.1. 功能介绍 138**](#_Toc148900591)

[**16.2. 业务请求与响应过程 138**](#_Toc148900592)

[**16.3. 类的作用 139**](#_Toc148900593)

[**16.3.1. 界面层相关的类 139**](#_Toc148900594)

[**16.3.2. 业务层相关的类 141**](#_Toc148900595)

[**16.3.3. 模型层相关的类 142**](#_Toc148900596)

[**16.3.4. 数据层相关的类 144**](#_Toc148900597)

[**17. 功能十七：分享便签 145**](#_Toc148900598)

[**17.1．功能介绍 145**](#_Toc148900599)

[**17.2．业务请求与响应过程 146**](#_Toc148900600)

[**17.3. 类的作用 147**](#_Toc148900601)

[**17.3.1. 界面层相关的类 147**](#_Toc148900602)

[**17.3.2. 业务层相关的类 153**](#_Toc148900603)

[**17.3.3. 模型层相关的类 153**](#_Toc148900604)

[**17.3.4.数据层相关的类 156**](#_Toc148900605)

[**18. 功能十八：添加提醒 158**](#_Toc148900606)

[**18.1. 功能介绍 158**](#_Toc148900607)

[**18.2. 业务请求与响应过程 158**](#_Toc148900608)

[**18.3. 类的作用 160**](#_Toc148900609)

[**18.3.1. 界面层相关的类 160**](#_Toc148900610)

[**18.3.2. 业务层相关的类 162**](#_Toc148900611)

[**18.3.3. 模型层相关的类 164**](#_Toc148900612)

[**18.3.4. 数据层相关的类 166**](#_Toc148900613)

[**19. 功能十九：关闭提醒 168**](#_Toc148900614)

[**19.1. 功能介绍 168**](#_Toc148900615)

[**19.2. 业务请求与响应过程 168**](#_Toc148900616)

[**19.3. 类的作用 169**](#_Toc148900617)

[**19.3.1. 界面层相关的类 169**](#_Toc148900618)

[**19.3.2. 业务层相关的类 171**](#_Toc148900619)

[**19.3.3. 模型层相关的类 173**](#_Toc148900620)

[**19.3.4. 数据层相关的类 175**](#_Toc148900621)

[**20. 收获和问题 177**](#_Toc148900622)

[**20.1. 收获 177**](#_Toc148900623)

[**20.2. 存在的问题 177**](#_Toc148900624)

1. **功能一：新建便签**
   1. **功能介绍**

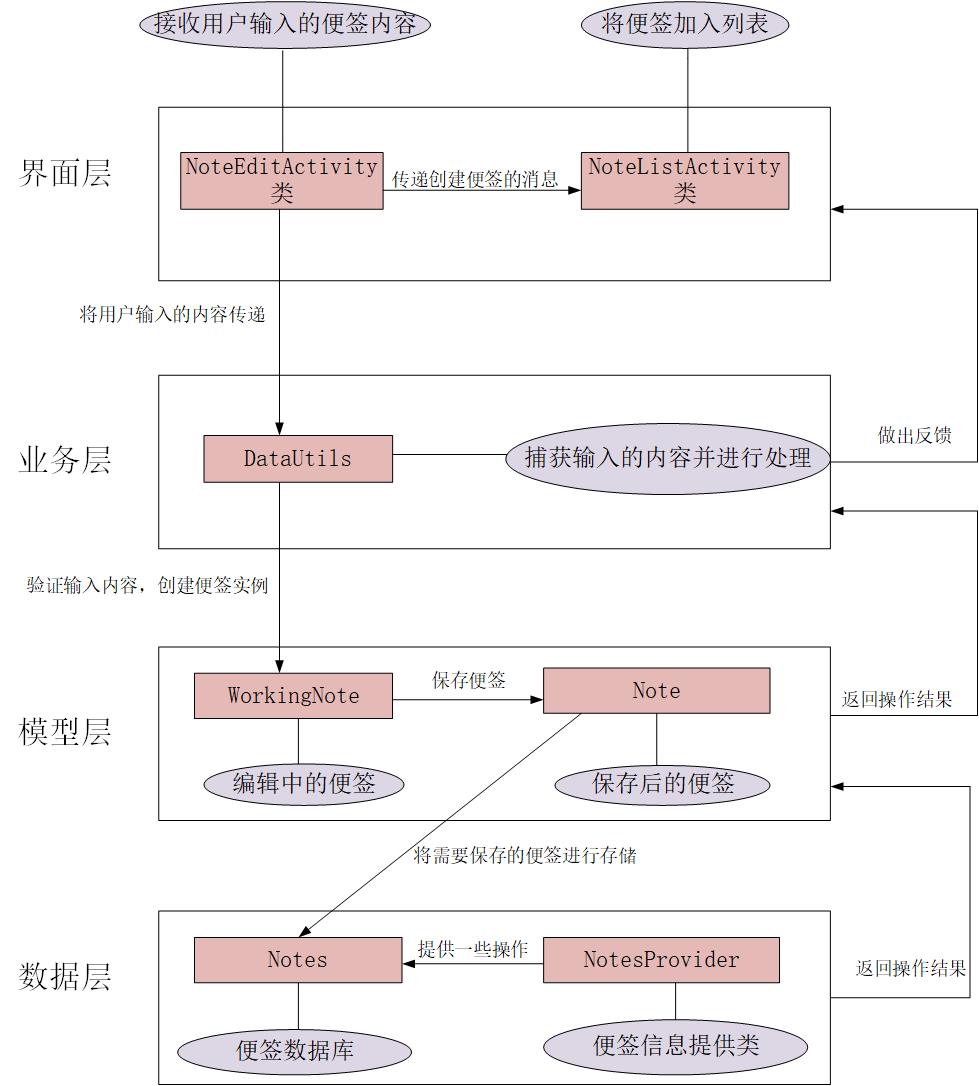
**功能概述：**新建一条便签并存储到本地数据库

**操作步骤：**在主列表界面点击NewNote按钮，进入便签编辑界面，在编辑界面输入文本，右上角可以更换便签背景颜色；按ctrl+m弹出上下文菜单，点击上下文菜单选项可对便签设置闹钟提醒，更改字体大小。编辑完成后按返回键即可保存便签至本地数据库

* 1. **业务请求与响应过程**

在新建便签的过程中，四个层级各自的作用大致如下：

1. **界面层**：该层管理与用户交互的界面。对于小米便签来说，界面层包含创建新便签的入口，以及用于输入便签内容的文本框等UI元素。用户在这里进行操作，如点击新建按钮，输入便签内容。
2. **业务层**：该层处理应用的核心业务逻辑。在这种情况下，业务逻辑可能包括：捕获用户输入的便签内容，验证输入（例如，检查输入是否为空），然后创建新的便签实例。
3. **模型层**：模型层通常包含了应用的数据模型定义。关于小米便签，一个便签实例的模型可能包括：便签的ID, 便签的内容, 创建时间，最后修改时间等等。
4. **数据层**：该层负责管理数据的存储和检索。在创建新的便签时，一份该便签的复制品会存储在本地数据库（如果需要离线访问的话）。同时，应用还可能会将新的便签同步到云服务器，使得用户可以在其他设备上访问这条便签。



* 1. **类的作用**
     1. **界面层相关的类**

**1.3.1.1、类名：NoteEditActivity**

1. **类的作用：**

这个类可以显示便签的内容，允许用户编辑便签，包含一个文本编辑框，使用户可以输入、编辑和格式化便签的文本内容，保存便签的修改，提供界面交互，处理用户的交互操作

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 用于管理小米便签中编辑界面的头部视图，用于优化列表或视图的性能，它持有对子视图的引用，以便在需要时快速访问。 |
| mNoteEditor; | 用于接收用户输入的文本，以便创建新的便签或编辑现有的便签内容 |
| mNoteEditorPanel; | 用于显示编辑便签界面的面板视图 |
| mWorkingNote; | 包含便签的相关信息 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate | super.onCreate  (savedInstanceState)：  调用父类的 onCreate 方法以执行基本的活动创建操作。  this.setContentView  (R.layout.note\_edit)：  设置当前活动的布局为 note\_edit.xml，这将决定界面的呈现样式。 | 用于在活动（Activity）创建时进行初始化 | 检查是否存在已保存的活动状态和通过 getIntent() 获取到的意图。如果没有保存的状态并且初始化活动状态失败，那么调用 finish() 方法结束当前活动。 |
| onSaveInstanceState | mWorkingNote：便签保存在数据库中的形式  outState：用于存放已经被删除的便签名称 | 用于保存当前活动的状态，以便在需要时进行恢复。用于保存正在编辑的便签的相关信息。 | 通过检查 mWorkingNote 是否已经存在于数据库中来决定是否需要保存便签。  如果 mWorkingNote 还不存在于数据库中，可能意味着这是一条新的便签（此时没有便签ID），因此会调用 saveNote() 方法将该便签保存到数据库中，以生成一个便签ID。然后，将便签的ID（mWorkingNote.getNoteId()）保存到 outState 中，以便在稍后的状态恢复操作中使用。 |
| onClick | View v：需要进行鼠标点击操作的视图  Id：视图名称 | 用于处理用户在界面中点击操作的事件。 | 根据用户点击位置（R.id.btn\_set\_bg\_color）  （sBgSelectorBtnsMap）  （sFontSizeBtnsMap）  的区别，进行相应的判断和响应 |
| clearSettingState | mNoteBgColorSelector：  与背景颜色设置相关的视图  mFontSizeSelector：  与字体大小设置相关的视图 | 用于处理用户对背景颜色和字体大小的设置 | 通过检查 mNoteBgColorSelector 和 mFontSizeSelector 这两个视图的可见性，用于取消设置状态，根据当前界面上的设置选择器的可见性，来决定是否隐藏这些选择器，并返回相应的布尔值以指示是否取消了设置状态 |
| onOptionsItemSelected | MenuItem：菜单选项 | 根据被选中的菜单项的ID，执行相应的操作 | 这段代码是一个菜单项选择的回调方法，根据被选中的菜单项的ID，执行相应的操作，具体到新建便签的这一操作时，当菜单项的ID为menu\_new\_note时，调用createNewNote()方法。 |
| createNewNote | Intent ：创建一个新的 Intent 实例，用于启动 NoteEditActivity | 保存当前正在编辑的便签 | 检查用户是否有对当前便签进行了修改，如果有，会将修改后的便签内容保存到数据库中。 |

**1.3.1.2类名：NotesListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 用于管理便签列表展示的适配器（Adapter）实例。 |
| mNotesListView; | 用于显示便签列表的 ListView 对象。 |
| mAddNewNote; | 用于增加新的note对象 |
| BackgroundQueryHandler | 用于在后台执行异步查询操作 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| createNewNote | Intent：通过创建一个新的 Intent 对象，将 NoteEditActivity 类作为目标 Activity； | 用于创建新的便签或编辑已有的便签 | createNewNote() 方法被调用时，会在当前的 NotesListActivity 中打开 NoteEditActivity 的界面，用于创建新的便签或编辑已有的便签。 |
| startAsyncNotesListQuery | mBackground  QueryHandler：异步查询类对象，负责异步线程访问数据库取出相应的便签数据 | 对便签进行按照日期的排序 | 在后台异步查询便签列表数据，并根据一些条件进行筛选，以获取与当前文件夹相关的便签数据，并按照类型和修改日期降序进行排序。 |

* + 1. **业务层相关的类**

**1.3.2.1类名：**DataUtils

1. **类的作用：**

便签数据处理工具类，封装如查找、移动、删除数据等操作，是一种用于处理数据相关操作的实用工具类

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |

* + 1. **模型层相关的类**

**1.3.3.1、类名：Note**

1. **类的作用：**单个便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteDiffValues | 存储便签的差异值 |
| mNoteData | 存储便签数据 |
| TAG | private static final String TAG = "Note"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getNewNoteId | ContentValues：用于存储便签的各个属性和值  Context：上下文参数  folderId：文件夹名称 | 获取新便签的ID | 从数据库中获取一个新的便签 ID。创建一个对象，用于存储便签的各个属性和值。获取当前时间作为新便签的创建时间。  将各种属性和值添加到对象中，返回新便签的 ID。 |
| Note | ContentValues：用于存储便签的各个属性和值 | Note 类的构造函数。它用于初始化一个 Note 对象。 | 这是一个无参数的构造函数，用于创建 Note 对象。 |
| getNewNoteId | Context：上下文参数  folderId：文件夹名称 | 在数据库中创建一个新的便签，并返回该便签的ID | 创建一个待插入数据库的对象，用于设置新便签的各个属性，指定便签所属的文件夹，获取插入后自动生成的 noteId |
| setNoteValue | Key：需要被设置的属性  Value：应该设置到key中的值 | 设置便签的某个属性的值，并对相关的属性进行相应的更新。 | 通过参数 key 和 value，将指定属性的新值添加到 mNoteDiffValues 中 |

**1.3.3.2类名：WorkingNote**

1. **类的作用：**正在编辑的便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNote | Note for the working note |
| mNoteId | Note Id |
| mNoteSettingStatusListener | 用于监听便签设置变化的监听器 |
| DATA\_PROJECTION | 一个字符串类型的数组，其中包含了一些列名或字段名作为查询数据的投影 |
| NOTE\_PROJECTION | 包含了一组用于查询便签数据的列名 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| WorkingNote | WorkingNote的一些参数需要 | 私有构造方法 | 私有构造方法 |
| createEmptyNote | Context：上下文参数  folderId：文件夹名称  widgetId：插件id  widgetType：插件类型  defaultBgColorId：  用于颜色设置 | 创建一个空的便签，并进行一些初始设置 | 创建了一个 WorkingNote 类的实例，这行代码设置了便签的背景颜色为默认颜色，设置了便签的小部件ID。  设置了，便签的小部件类型最后，该方法返回创建的 WorkingNote 实例。 |
| saveNote | Context：上下文参数  Id：需要的id参数 | 一个同步保存便签的方法，并且在保存之前进行一些检查和操作 | 便签内容的验证和检查，以决定是否要执行保存操作。检查便签是否已经存在于数据库中。如果便签已经存在，则不需要创建新的便签。  获取一个新的便签ID。根据给出的参数，使用上下文和文件夹ID来为新便签生成一个唯一的ID。  。 |
| isWorthSaving | mIsDeleted：标记了删除操作  existInDatabase TextUtils.isEmpty  ：判断了当前便签是否存在于数据库中并且内容是否为空  existInDatabase()  mNote.isLocalModified()  判断当前便签是否在数据库中存在，并且是否有本地修改 | 用于判断是否值得保存，这可以和naveNote联动 | 检查是否标记了删除操作，如果已经删除，则可能认为不值得保存。  结合判断了当前便签是否存在于数据库中并且内容是否为空。如果便签既不在数据库中也没有内容，则可能认为不值得保存。  判断当前便签是否在数据库中存在，并且是否有本地修改。如果便签在数据库中存在但没有本地修改，则可能认为不值得保存。 |

* + 1. **数据层相关的类**

**1.3.4.1、类名：Notes**

1. **类的作用：**

便签数据库，用于记录便签相关属性和数据

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| DataConstants | 用于指示一些常量 |
| NoteColumns、DataColumns | 用于定义一些final类型的值 |
| TextNote、CallNote | 对DataColumns进行接口的实现 |

此类由于是data类，用于保存和记录数据，没有相关的方法

* + - 1. **类名：NotesProvider**

1. **类的作用：**便签信息提供类
2. **类的精读：**

继承了 ContentProvider （Android 中用于提供数据访问接口的基类，可以从其他应用程序中访问和共享数据。）类

用于处理小米便签数据的内容提供器。它可能提供了一些方法来实现对便签数据的增删改查操作，以及与其他应用程序进行数据共享和访问的功能。

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| insert | URI：唯一标识符  ContentValues：目录 | 用于插入数据到数据库中 | 据提供的 uri，它将区分是插入便签数据还是插入其他数据。 |
| getType | URI：唯一标识符 | 判断给定的 Uri 是属于什么类型的数据 | 根据给定的 Uri 获取其类型，并返回一个字符串表示 |

1. **功能二：删除便签**
   1. **功能介绍**

**功能概述：**将一条便签从本地数据库中删除

**操作步骤：**在便签列表界面长按某一便签，会进入选择界面，选中想要删除的便签，点击下方垃圾桶即可进行删除，能够对便签集进行有效管理

* 1. **业务请求与响应过程**

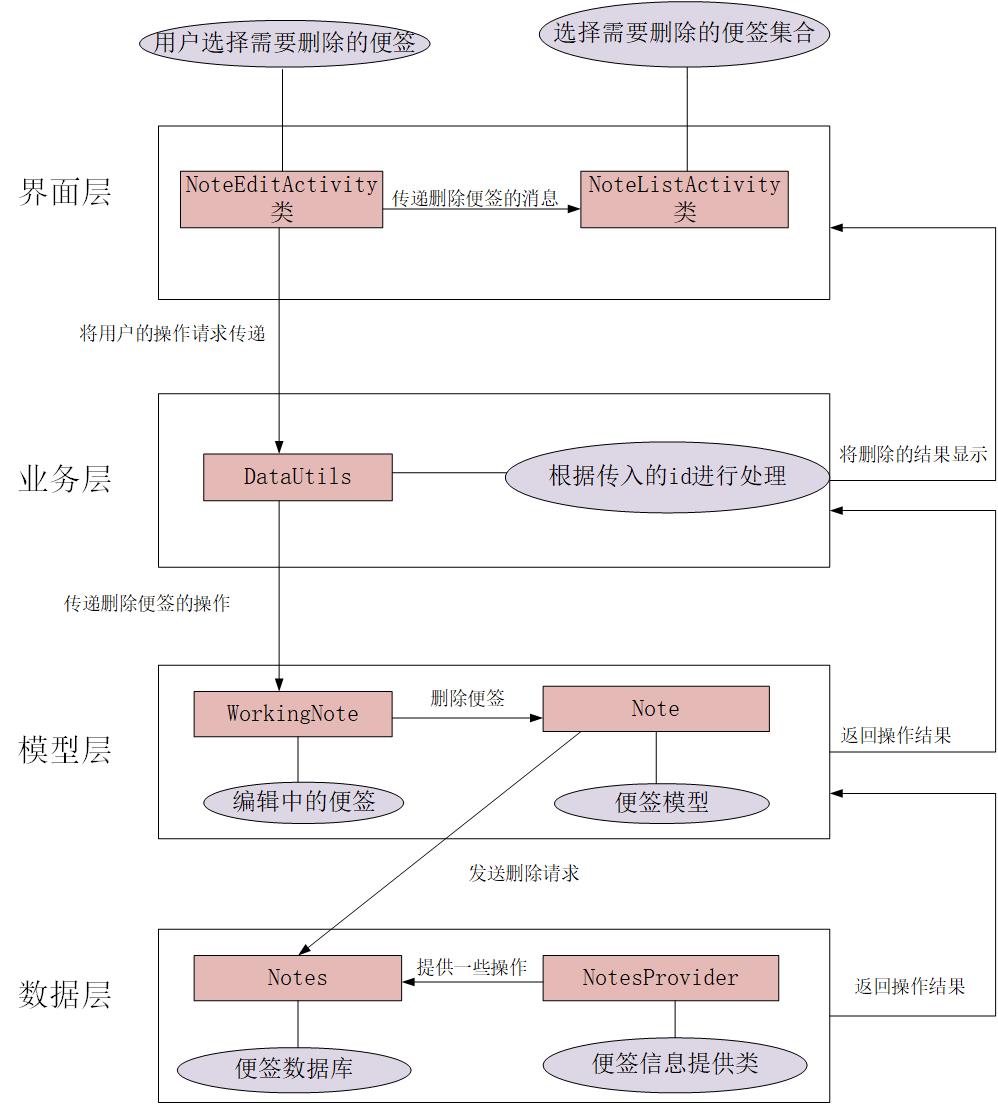
在删除便签的过程中，四个层级各自的作用大致如下：

**1. 界面层**：用户在小米便签应用的界面上选择要删除的便签，并点击删除按钮或选择删除选项时，触发界面层的事件处理。

**2. 业务层**：界面层接收到用户操作后，将请求传递给业务层。业务层负责处理业务逻辑，包括验证用户操作的合法性、构造请求对象、调用模型层接口等。

**3. 模型层**：业务层将删除便签的请求传递给模型层。模型层表示应用程序的核心组成部分，处理与数据相关的操作。它负责与数据层进行交互，执行删除便签的操作，并将结果返回给业务层。

**4. 数据层**：模型层通过调用数据层的接口，向数据库或存储系统发送删除便签的请求。数据层负责实际的数据操作，包括获取、保存、更新和删除数据等。在删除便签的过程中，数据层会根据请求中的标识符或其他信息，从数据库或存储系统中删除相应的便签数据，并返回操作的结果给模型层。



* 1. **类的作用**
     1. **界面层相关的类**

**2.2.1.1、类名：NoteEditActivity**

1. **类的作用：**

这个类可以显示便签的内容，允许用户编辑便签，包含一个文本编辑框，使用户可以输入、编辑和格式化便签的文本内容，保存便签的修改，提供界面交互，处理用户的交互操作

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 用于管理小米便签中编辑界面的头部视图，用于优化列表或视图的性能，它持有对子视图的引用，以便在需要时快速访问。 |
| mNoteEditor; | 用于接收用户输入的文本，以便创建新的便签或编辑现有的便签内容 |
| mNoteEditorPanel; | 用于显示编辑便签界面的面板视图 |
| mWorkingNote; | 包含便签的相关信息 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate | super.onCreate  (savedInstanceState)：  调用父类的 onCreate 方法以执行基本的活动创建操作。  this.setContentView  (R.layout.note\_edit)：  设置当前活动的布局为 note\_edit.xml，这将决定界面的呈现样式。 | 用于在活动（Activity）创建时进行初始化 | 检查是否存在已保存的活动状态和通过 getIntent() 获取到的意图。如果没有保存的状态并且初始化活动状态失败，那么调用 finish() 方法结束当前活动。 |
| onSaveInstanceState | mWorkingNote：便签保存在数据库中的形式  outState：用于存放已经被删除的便签名称 | 用于保存当前活动的状态，以便在需要时进行恢复。用于保存正在编辑的便签的相关信息。 | 通过检查 mWorkingNote 是否已经存在于数据库中来决定是否需要保存便签。  如果 mWorkingNote 还不存在于数据库中，可能意味着这是一条新的便签（此时没有便签ID），因此会调用 saveNote() 方法将该便签保存到数据库中，以生成一个便签ID。然后，将便签的ID（mWorkingNote.getNoteId()）保存到 outState 中，以便在稍后的状态恢复操作中使用。 |
| onClick | View v：需要进行鼠标点击操作的视图  Id：视图名称 | 用于处理用户在界面中点击操作的事件。 | 根据用户点击位置（R.id.btn\_set\_bg\_color）  （sBgSelectorBtnsMap）  （sFontSizeBtnsMap）  的区别，进行相应的判断和响应 |
| dispatchTouchEvent | MotionEvent：当前事件 | 用于处理触摸事件的分发 | 在函数中，会根据条件判断来决定是否隐藏一些视图，并返回一个布尔值来指示是否消费了该事件。 |
| onOptionsItemSelected | MenuItem：菜单选项 | 根据被选中的菜单项的ID，执行相应的操作 | 这段代码是一个菜单项选择的回调方法，根据被选中的菜单项的ID，执行相应的操作，具体到删除便签的这一操作时，当菜单项的ID为menu\_delete\_note时，创建一个确认删除提示对话框，并在用户点击确认时调用deleteCurrentNote()方法进行删除操作，然后结束当前活动。 |
| deleteCurrentNote | mWorkingNote：便签模型，为正在被操作的便签 | 删除当前的便签 | 首先，它检查mWorkingNote表示的便签是否存在于数据库中，通过调用existInDatabase()方法进行检查。如果便签存在于数据库中，它会创建一个 HashSet 对象 ids，用于存储要删除的便签的唯一标识符。如果便签成功从数据库中删除或移动，将会调用 mWorkingNote.markDeleted(true) 方法，将 mWorkingNote 对象标记为已删除 |

**2.2.1.2、类名：NotesListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 用于管理便签列表展示的适配器（Adapter）实例。 |
| mNotesListView; | 用于显示便签列表的 ListView 对象。 |
| mAddNewNote; | 用于增加新的note对象 |
| BackgroundQueryHandler | 用于在后台执行异步查询操作 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| startAsyncNotesListQuery | mBackground  QueryHandler：异步查询类对象，负责异步线程访问数据库取出相应的便签数据 | 对便签进行按照日期的排序 | 在后台异步查询便签列表数据，并根据一些条件进行筛选，以获取与当前文件夹相关的便签数据，并按照类型和修改日期降序进行排序。 |
| batchDelete | AsyncTask：判断同步  mNotesListAdapter：用于选中便签集合  doInBackground 方法最后返回 widgets 对象。 | 批量删除便签 | 该异步任务首先通过 mNotesListAdapter 获取选中的便签的集合 widgets。  如果当前不处于同步模式（!isSyncMode()），则直接调用 DataUtils.batchDeleteNotes() 方法来删除选中的便签。  如果当前处于同步模式，将选中的便签移动到垃圾箱文件夹，调用 DataUtils.batchMoveToFolder() 方法。 |

* + 1. **业务层相关的类**
       1. **类名：**DataUtils

1. **类的作用：**

便签数据处理工具类，封装如查找、移动、删除数据等操作，是一种用于处理数据相关操作的实用工具类

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |

### 2.2.3. 模型层相关的类

**2.2.3.1、类名：Note**

1. **类的作用：**单个便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteDiffValues | 存储便签的差异值 |
| mNoteData | 存储便签数据 |
| TAG | private static final String TAG = "Note"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getNewNoteId | ContentValues：用于存储便签的各个属性和值  Context：上下文参数  folderId：文件夹名称 | 获取新便签的ID | 从数据库中获取一个新的便签 ID。创建一个对象，用于存储便签的各个属性和值。获取当前时间作为新便签的创建时间。  将各种属性和值添加到对象中，返回新便签的ID。 |
| Note | ContentValues：用于存储便签的各个属性和值 | Note 类的构造函数。它用于初始化一个 Note 对象。 | 这是一个无参数的构造函数，用于创建 Note 对象。 |
| getNewNoteId | Context：上下文参数  folderId：文件夹名称 | 在数据库中创建一个新的便签，并返回该便签的ID | 创建一个待插入数据库的对象，用于设置新便签的各个属性，指定便签所属的文件夹，获取插入后自动生成的 noteId |
| setNoteValue | Key：需要被设置的属性  Value：应该设置到key中的值 | 设置便签的某个属性的值，并对相关的属性进行相应的更新。 | 通过参数 key 和 value，将指定属性的新值添加到 mNoteDiffValues 中 |

* + - 1. **类名：WorkingNote**

1. **类的作用：**正在编辑的便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNote | Note for the working note |
| mNoteId | Note Id |
| mNoteSettingStatusListener | 用于监听便签设置变化的监听器 |
| DATA\_PROJECTION | 一个字符串类型的数组，其中包含了一些列名或字段名作为查询数据的投影 |
| NOTE\_PROJECTION | 包含了一组用于查询便签数据的列名 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| WorkingNote | WorkingNote的一些参数需要 | 私有构造方法 | 私有构造方法 |
| createEmptyNote | Context：上下文参数  folderId：文件夹名称  widgetId：插件id  widgetType：插件类型  defaultBgColorId：  用于颜色设置 | 创建一个空的便签，并进行一些初始设置 | 创建了一个 WorkingNote 类的实例，这行代码设置了便签的背景颜色为默认颜色，设置了便签的小部件ID。  设置了，便签的小部件类型最后，该方法返回创建的 WorkingNote 实例。 |
| markDeleted | Mark：一个布尔值 | 用于设置便签的删除状态 | markDeleted 方法用于标记便签的删除状态。通过将布尔值 mark 赋值给成员变量 mIsDeleted，可以设置便签的删除状态。 |
| isWorthSaving | mIsDeleted：标记了删除操作  existInDatabase TextUtils.isEmpty  ：判断了当前便签是否存在于数据库中并且内容是否为空  existInDatabase()  mNote.isLocalModified()  判断当前便签是否在数据库中存在，并且是否有本地修改 | 用于判断是否值得保存，这可以和naveNote联动 | 检查是否标记了删除操作，如果已经删除，则可能认为不值得保存。  结合判断了当前便签是否存在于数据库中并且内容是否为空。如果便签既不在数据库中也没有内容，则可能认为不值得保存。  判断当前便签是否在数据库中存在，并且是否有本地修改。如果便签在数据库中存在但没有本地修改，则可能认为不值得保存。 |

* + 1. **数据层相关的类**

**2.2.4.1、类名：Notes**

1. **类的作用：**

便签数据库，用于记录便签相关属性和数据

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| DataConstants | 用于指示一些常量 |
| NoteColumns、DataColumns | 用于定义一些final类型的值 |
| TextNote、CallNote | 对DataColumns进行接口的实现 |

此类由于是data类，用于保存和记录数据，没有相关的方法

* + - 1. **类名：NotesProvider**

1. **类的作用：**便签信息提供类
2. **类的精读：**

继承了 ContentProvider （Android 中用于提供数据访问接口的基类，可以从其他应用程序中访问和共享数据。）类

用于处理小米便签数据的内容提供器。它可能提供了一些方法来实现对便签数据的增删改查操作，以及与其他应用程序进行数据共享和访问的功能。

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| insert | URI：唯一标识符  ContentValues：目录 | 用于插入数据到数据库中 | 据提供的 uri，它将区分是插入便签数据还是插入其他数据。 |
| getType | URI：唯一标识符 | 判断给定的 Uri 是属于什么类型的数据 | 根据给定的 Uri 获取其类型，并返回一个字符串表示 |
| delete | URI：唯一标识符  deleteData:布尔值  selection：删除选择内容 | 一个删除便签或数据的方法 | 通过匹配 Uri 来确定要删除的是便签还是数据。根据不同的 Uri 类型，采取不同的删除操作。  在删除便签时，根据传入的选择条件和参数执行删除操作。将删除影响的行数保存到 count 变量中。 |

1. **功能三：移动便签**
   1. **功能介绍**

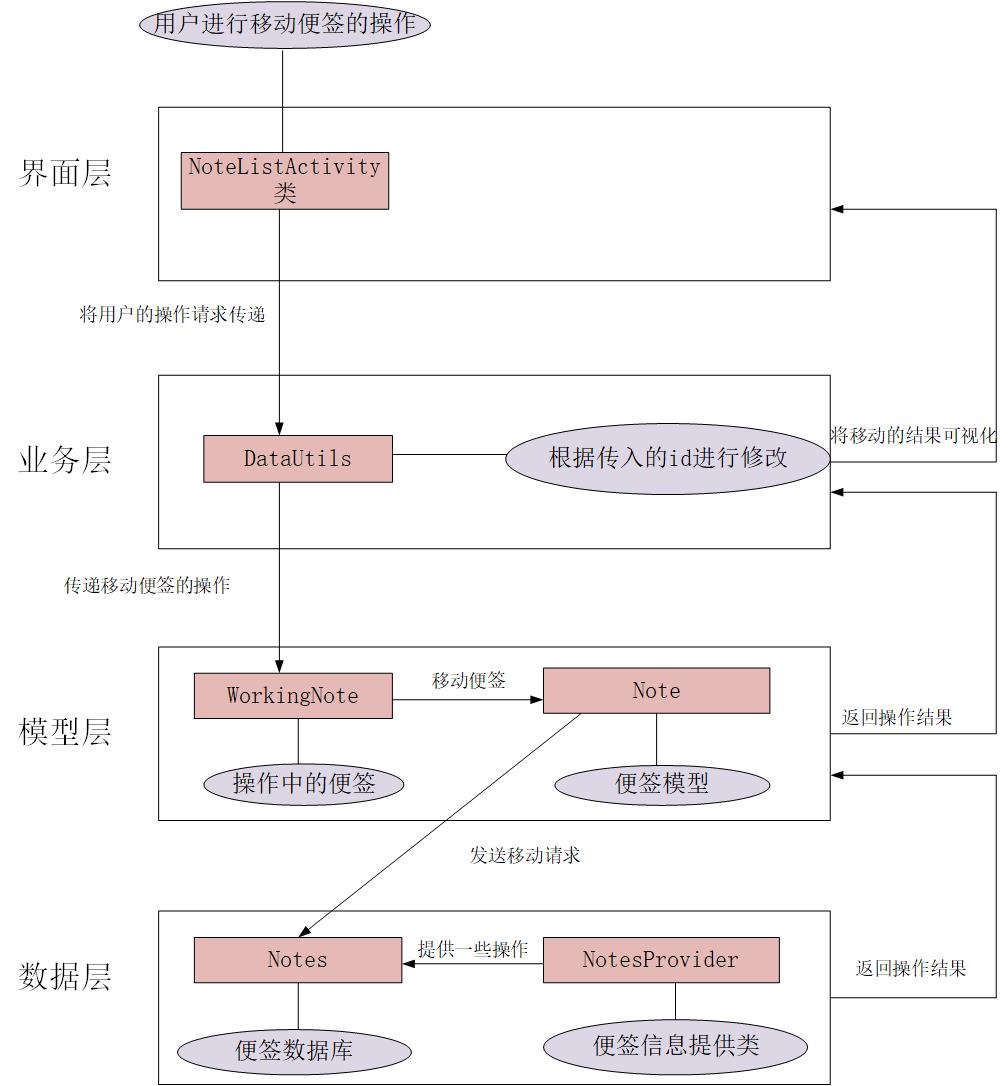
**功能概述：**将便签进行移动操作，方便管理

**操作步骤：**在便签列表界面长按某一便签，进入选择界面，选中想要移动的标签，点击左下方移动按钮，选择目标文件夹，就可以将快速创建的便签归类以进行有效管理

* 1. **业务请求与响应过程**

在移动便签的过程中，四个层级各自的作用大致如下：

1. **界面层**：界面层是用户与移动便签应用进行交互的部分。它包括应用的用户界面、用户输入的处理和展示等。用户通过界面层发送操作请求，如创建新的便签、移动便签位置等。
2. **业务层**：业务层处理来自界面层的请求，并进行相应的业务逻辑处理。它负责验证用户操作的合法性、调用合适的模型进行数据处理等。在移动便签中，业务层会根据用户请求，调用模型层的接口进行数据处理。
3. **模型层**：模型层是处理数据的核心部分。它包含处理数据的逻辑和相关的操作方法。在移动便签中，模型层负责创建、更新、删除便签数据等。它可以对数据进行操作，以满足业务层的需求。
4. **数据层**：数据层负责存储和管理移动便签应用的数据。它包括数据库、文件系统或其他数据存储机制。数据层提供了对数据的读取和存储操作，以满足模型层的需求。移动便签的数据层负责将数据存储到服务器端，并处理请求中的数据读取和更新操作。



* 1. **类的作用**
     1. **界面层相关的类**
        1. **类名：NotesListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 用于管理便签列表展示的适配器（Adapter）实例。 |
| mNotesListView; | 用于显示便签列表的 ListView 对象。 |
| mAddNewNote; | 用于增加新的note对象 |
| BackgroundQueryHandler | 用于在后台执行异步查询操作 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| startAsyncNotesListQuery | mBackground  QueryHandler：异步查询类对象，负责异步线程访问数据库取出相应的便签数据 | 对便签进行按照日期的排序 | 在后台异步查询便签列表数据，并根据一些条件进行筛选，以获取与当前文件夹相关的便签数据，并按照类型和修改日期降序进行排序。 |
| openFolder | mCurrentFolderId：当前文件夹的唯一标识符 | 打开一个文件夹，并根据传入的参数对应的数据来执行相应的操作。 | 于打开一个文件夹，并根据传入的参数 NoteItemData 对应的数据来执行相应的操作 |

* + 1. **业务层相关的类**
       1. **类名：**DataUtils

1. **类的作用：**

便签数据处理工具类，封装如查找、移动、删除数据等操作，是一种用于处理数据相关操作的实用工具类

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

* + 1. **模型层相关的类**

**3.3.3.1、类名：Note**

1. **类的作用：**单个便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteDiffValues | 存储便签的差异值 |
| mNoteData | 存储便签数据 |
| TAG | private static final String TAG = "Note"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getNewNoteId | ContentValues：用于存储便签的各个属性和值  Context：上下文参数  folderId：文件夹名称 | 获取新便签的ID | 从数据库中获取一个新的便签 ID。创建一个对象，用于存储便签的各个属性和值。获取当前时间作为新便签的创建时间。  将各种属性和值添加到对象中，返回新便签的 ID。 |
| Note | ContentValues：用于存储便签的各个属性和值 | Note 类的构造函数。它用于初始化一个 Note 对象。 | 这是一个无参数的构造函数，用于创建 Note 对象。 |
| getNewNoteId | Context：上下文参数  folderId：文件夹名称 | 在数据库中创建一个新的便签，并返回该便签的ID | 创建一个待插入数据库的对象，用于设置新便签的各个属性，指定便签所属的文件夹，获取插入后自动生成的 noteId |
| setNoteValue | Key：需要被设置的属性  Value：应该设置到key中的值 | 设置便签的某个属性的值，并对相关的属性进行相应的更新。 | 通过参数 key 和 value，将指定属性的新值添加到 mNoteDiffValues 中 |

* + - 1. **类名：WorkingNote**

1. **类的作用：**正在编辑的便签的模型
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNote | Note for the working note |
| mNoteId | Note Id |
| mNoteSettingStatusListener | 用于监听便签设置变化的监听器 |
| DATA\_PROJECTION | 一个字符串类型的数组，其中包含了一些列名或字段名作为查询数据的投影 |
| NOTE\_PROJECTION | 包含了一组用于查询便签数据的列名 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| WorkingNote | WorkingNote的一些参数需要 | 私有构造方法 | 私有构造方法 |
| createEmptyNote | Context：上下文参数  folderId：文件夹名称  widgetId：插件id  widgetType：插件类型  defaultBgColorId：  用于颜色设置 | 创建一个空的便签，并进行一些初始设置 | 创建了一个 WorkingNote 类的实例，这行代码设置了便签的背景颜色为默认颜色，设置了便签的小部件ID。  设置了，便签的小部件类型最后，该方法返回创建的 WorkingNote 实例。 |
| markDeleted | Mark：一个布尔值 | 用于设置便签的删除状态 | markDeleted 方法用于标记便签的删除状态。通过将布尔值 mark 赋值给成员变量 mIsDeleted，可以设置便签的删除状态。 |
| isWorthSaving | mIsDeleted：标记了删除操作  existInDatabase TextUtils.isEmpty  ：判断了当前便签是否存在于数据库中并且内容是否为空  existInDatabase()  mNote.isLocalModified()  判断当前便签是否在数据库中存在，并且是否有本地修改 | 用于判断是否值得保存，这可以和naveNote联动 | 检查是否标记了删除操作，如果已经删除，则可能认为不值得保存。  结合判断了当前便签是否存在于数据库中并且内容是否为空。如果便签既不在数据库中也没有内容，则可能认为不值得保存。  判断当前便签是否在数据库中存在，并且是否有本地修改。如果便签在数据库中存在但没有本地修改，则可能认为不值得保存。 |
| loadNote | mContext：目录 | 加载便签数据 | 通过提供的便签ID查询便签的数据。  如果查询结果不为 null，则将查询结果中的各个字段值分别赋给相应的成员变量。 |
| loadNoteData | mContext：目录 | 查询与特定ID 相关的数据。 | 进行数据库查询操作查询与特D相关的数据。通过循环逐个遍历查询结果中的行。  根据不同的数据类型，进行相应的操作 |

* + 1. **数据层相关的类**

**3.3.4.1、类名：Notes**

1. **类的作用：**

便签数据库，用于记录便签相关属性和数据

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| DataConstants | 用于指示一些常量 |
| NoteColumns、DataColumns | 用于定义一些final类型的值 |
| TextNote、CallNote | 对DataColumns进行接口的实现 |

此类由于是data类，用于保存和记录数据，没有相关的方法

* + - 1. **类名：NotesProvider**

1. **类的作用：**便签信息提供类
2. **类的精读：**

继承了 ContentProvider （Android 中用于提供数据访问接口的基类，可以从其他应用程序中访问和共享数据。）类

用于处理小米便签数据的内容提供器。它可能提供了一些方法来实现对便签数据的增删改查操作，以及与其他应用程序进行数据共享和访问的功能。

**类中的方法**

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| insert | URI：唯一标识符  ContentValues：目录 | 用于插入数据到数据库中 | 据提供的 uri，它将区分是插入便签数据还是插入其他数据。 |
| getType | URI：唯一标识符 | 判断给定的 Uri 是属于什么类型的数据 | 根据给定的 Uri 获取其类型，并返回一个字符串表示 |
| update | URI：唯一标识符  mMatcher：匹配器 | 用于更新数据库中数据 | 首先，通过传入的参数 uri  使用 mMatcher  .match(uri) 来执行一个匹配操作，判断要进行的操作类型。  根据匹配的结果，执行相应的操作：  如果更新的行数大于 0，表明有数据更新成功，则执行以下操作：  如果 updateData 为 true，表示数据项（Data）已更新，则通过调用 notifyChange() 方法通知数据项发生变化。  最后，返回更新的行数。 |

1. **功能四：新建文件夹**
   1. **功能介绍**

**功能概述：**新建一个文件夹，读取用户输入的文件夹名称，并将其存储到本地数据库。

**操作步骤：**在主界面的右下角有一个菜单栏，用户点击按钮，选择“新建文件夹”选项，并输入拟创建文件夹的名字，在点击确认后即可完成新文件夹的创建。

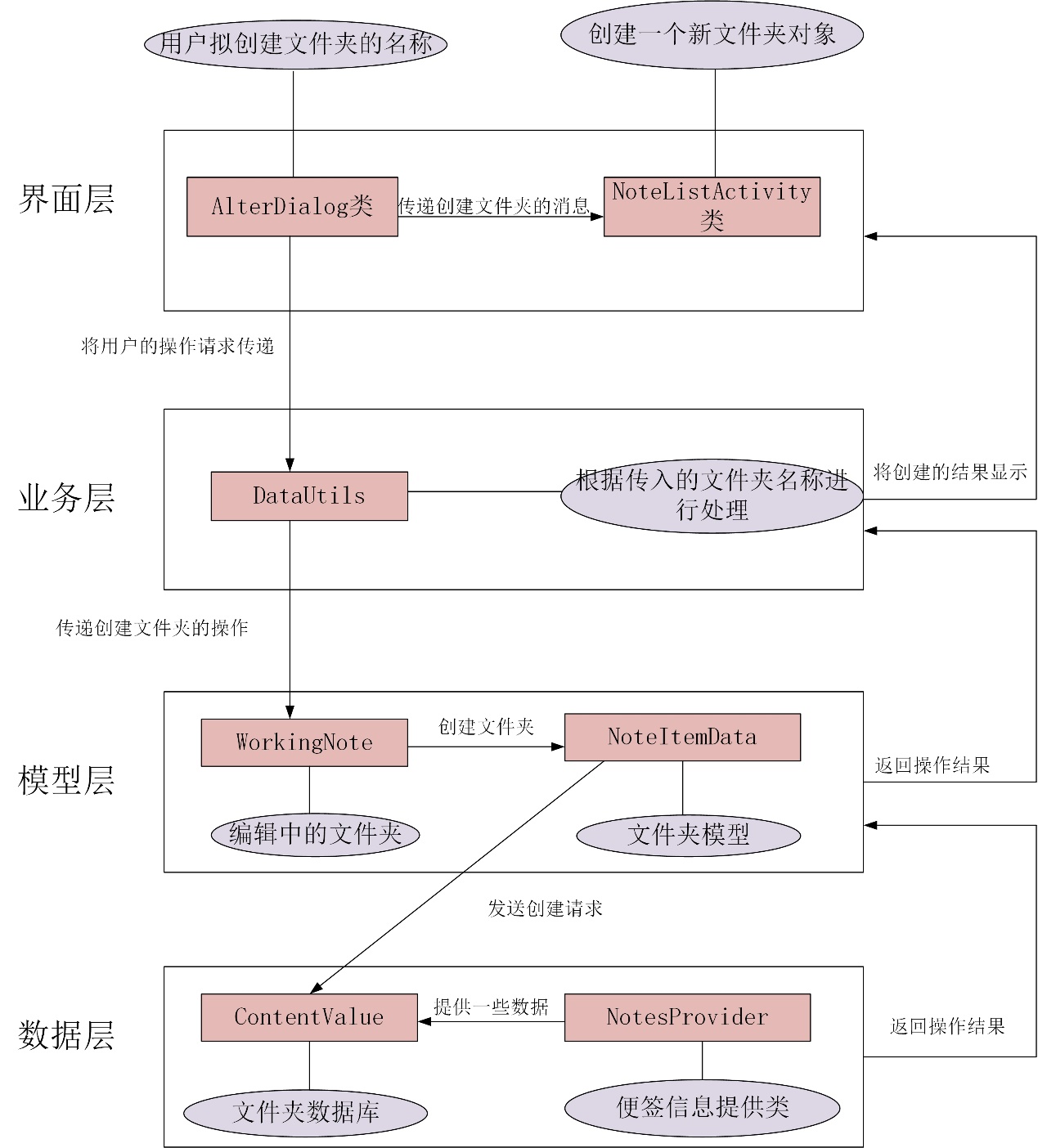
* 1. **业务请求与响应过程**

新建文件夹功能（下简称“功能”）的界面层主要由包ui中NotesListActivity类通过调用showCreateOrModifyFolderDialog()方法实现。showCreateOrModifyFolderDialog()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

功能的业务层由DataUtils类和NotesListActivity类实现。首先通过TextUtils的isEmpty()方法来判断文件夹名是否为空，其次调用checkVisibleFolderName()方法判断文件夹名是否存在，在不存在的情况下然后创建 ContentValues 对象并存储信息，最后使用 ContentResolver 的 insert()方法将文件夹中的信息插入到数据库中。

功能的模型层主要由model包中的WorkingNote类和NoteItemData类来实现，主要包括getFolderId()等方法。

功能的数据层主要由ContentValue类的对象来实现。创建一个ContentValue的对象，并调用put()方法将文件夹的名称和类型存入对象。最后使用ContentResolver的insert()方法将包含文件夹信息的ContentValues对象插入到数据库中。



* 1. **类的作用**
     1. **界面层相关的类**
        1. **类名：NoteListActivity**

1. 类的作用：

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode 是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount 表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用 mBackgroundQueryHandler 发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用 DataUtils.batchMoveToFolder 方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的Intent，并指定当前文件夹ID |
| batchDelete() | / | 批量删除便签或将它们移动到回收站 | 异步执行的任务，根据同步模式，直接删除便签 |
| deleteFolder(long folderId) | folderId 表示要删除的文件夹的ID | 删除文件夹及其相关操作 | 根据同步模式，直接删除文件夹 |
| openNode(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| openFolder(NoteItemData data) | 同上 | 打开文件夹 | 根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表 |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用 createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| **showCreateOrModifyFolderDialog(boolean create)** | **create 是一个布尔值，表示是创建文件夹还是修改文件夹** | **显示创建或修改文件夹的对话框** | **弹出一个弹窗，将输入参数传入函数** |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId 表示小部件的ID，appWidgetType 表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| exportNoteToText() | / | 将便签导出为文本文件 | 使用 BackupUtils 类执行导出操作，并根据结果显示相应的对话框提示 |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图; view：表示用户长按的特定列表项视图; position：表示被长按的列表项在适配器中的位置; id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为 mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

* + 1. **业务层相关的类**

**4.3.2.1类名：**DataUtils

1. **类的作用：**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

(**2) 类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |
|  |  |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

**4.3.3. 模型层相关的类**

**4.3.3.1、类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 标记当前便签状态 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |
| NoteSettingChangedListener | 定义便签设置改变监听器的接口 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getTitleBgResId() | 无 | 返回当前的标题颜色的ID值 | 通过背景颜色ID值来调用NoteBgResource内的方法来匹配返回标题的ID值 |
| WorkingNote(Context context, long folderId) | context:Android应用程序的上下文。  folderId:笔记所属文件夹的ID。 | 创建一个新的笔记对象 | 实例化对象 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getBgColorResId(): | 无 | 获取笔记背景颜色资源ID | Return直接返回 |
| getCheckListMode() | 无 | 获取笔记模式 | Return直接返回 |
| getModifiedDate() | 无 | 获取笔记修改日期 | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getContent() | 无 | 获取笔记内容 | Return直接返回 |
| getNoteId() | 无 | 获取笔记ID | Return直接返回 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| getTitleBgResId() | 无 | 获取笔记标题背景资源ID | Return直接返回 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context: Android 应用程序的上下文。  folderId: 笔记所属文件夹的ID。  widgetId: 笔记对应的小部件的ID。  widgetType: 笔记对应的小部件类型。  defaultBgColorId: 笔记的默认背景颜色ID。 | 创建一个空白笔记 | 通过调用一系列set方法，设置笔记的初始情况 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |

* + - 1. **类名：NoteItemData**

1. **类的作用：表示便签项的数据。该类通过从数据库查询结果的 Cursor 中获取相应的列数据来初始化实例。**
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mId | 便签项的ID |
| mId | 提醒时间 |
| mBgColorId | 背景颜色ID |
| mCreatedDate | 创建时间 |
| mHasAttachment | 是否有附件 |
| mModifiedDate | 最近修改时间 |
| mNotesCount | 笔记数量 |
| mParentId | 父文件夹ID |
| mSnippet | 便签内容的摘要 |
| mType | 便签类型 |
| mWidgetId | Widget的ID |
| mWidgetType | Widget类型 |
| mName | 联系人名字 |
| mPhoneNumber | 联系人电话号码 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteItemData(Context context, Cursor cursor) | Context:文本  Cursor：游标 | 构造函数，从数据库查询结果的Cursor中初始化便签项数据。 | 调用相关方法，实现数据库的查询和数据的初始化。 |
| isOneFollowingFolder() | 无 | 判断是否是一个便签项后面跟随一个文件夹 | 调用相关方法，实现是否跟随一个文件夹的判断 |
| isMultiFollowingFolder() | 无 | 判断是否是一个便签项后面跟随多个文件夹 | 调用相关方法，实现是否跟随多个文件夹的判断 |
| isLast() | 无 | 判断是否是列表中的最后一个项 | 调用相关方法，实现对是否是列表中的最后一个项的判断 |
| getCallName() | 无 | 获取联系人名字 | Return直接返回 |
| isFirst() | 无 | 判断是否是列表中的第一个项 | 调用相关方法，实现是否是列表中的第一个项的判断 |
| isSingle() | 无 | 判断列表是否只有一个项 | 调用相关方法，实现列表是否只有一个项的判断 |
| getId() | 无 | 获取便签项的ID | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getCreatedDate() | 无 | 获取创建时间 | Return直接返回 |
| hasAttachment() | 无 | 判断是否有附件 | 调用相关方法，实现对是否有附件的判断 |
| getModifiedDate() | 无 | 获取最近修改时间 | Return直接返回 |
| getBgColorId() | 无 | 获取背景颜色ID | Return直接返回 |
| getWidgetType() | 无 | 获取父文件夹ID | Return直接返回 |

* + 1. **数据层相关的类**

**4.3.4.1、类名：**ContentValues

1. **类的作用：此类用于存储 ContentResolver 可以处理的一组值。**
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mValues | HashMap对象，用以存储键值对 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| **put(String key, Byte value)** | Key：键  Value值 | 向 ContentValues 中添加一个 byte 类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Short value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Short**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Integer value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Integer**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Long value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Long**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Float value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Float**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key,Double value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Double**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, String value)** | Key：键  Value值 | 向 ContentValues 中添加一个**String**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, byte[] value)** | Key：键  Value值 | 向 ContentValues 中添加一个 byte数组类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **putNull(String key)** | Key：键 | 向 ContentValues 中添加一个空值的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **get(String key)**: | Key：键 | 根据键获取对应的值 | 直接调用HashMap中的get方法 |
| **containsKey(String key)** | Key：键 | 检查是否包含指定键 | 直接调用HashMap中的contains方法 |
| **remove(String key)** | Key：键 | 根据键移除键值对 | 直接调用HashMap中的remove方法 |
| **clear()** |  | 清空 ContentValues 中的所有键值对。 | 直接调用HashMap中的clear方法 |

1. **功能五：查看文件夹**
   1. **功能介绍**

**功能概述：**打开指定的文件夹，并根据文件夹类型更新界面显示，包括标题栏的文本和按钮的可见性。

**操作步骤：**在主界面的右下角有一个菜单栏，用户点击按钮，选择文件夹列表，点击选中拟查看的文件夹，在点击后自动显示拟查看文件夹的相关信息。

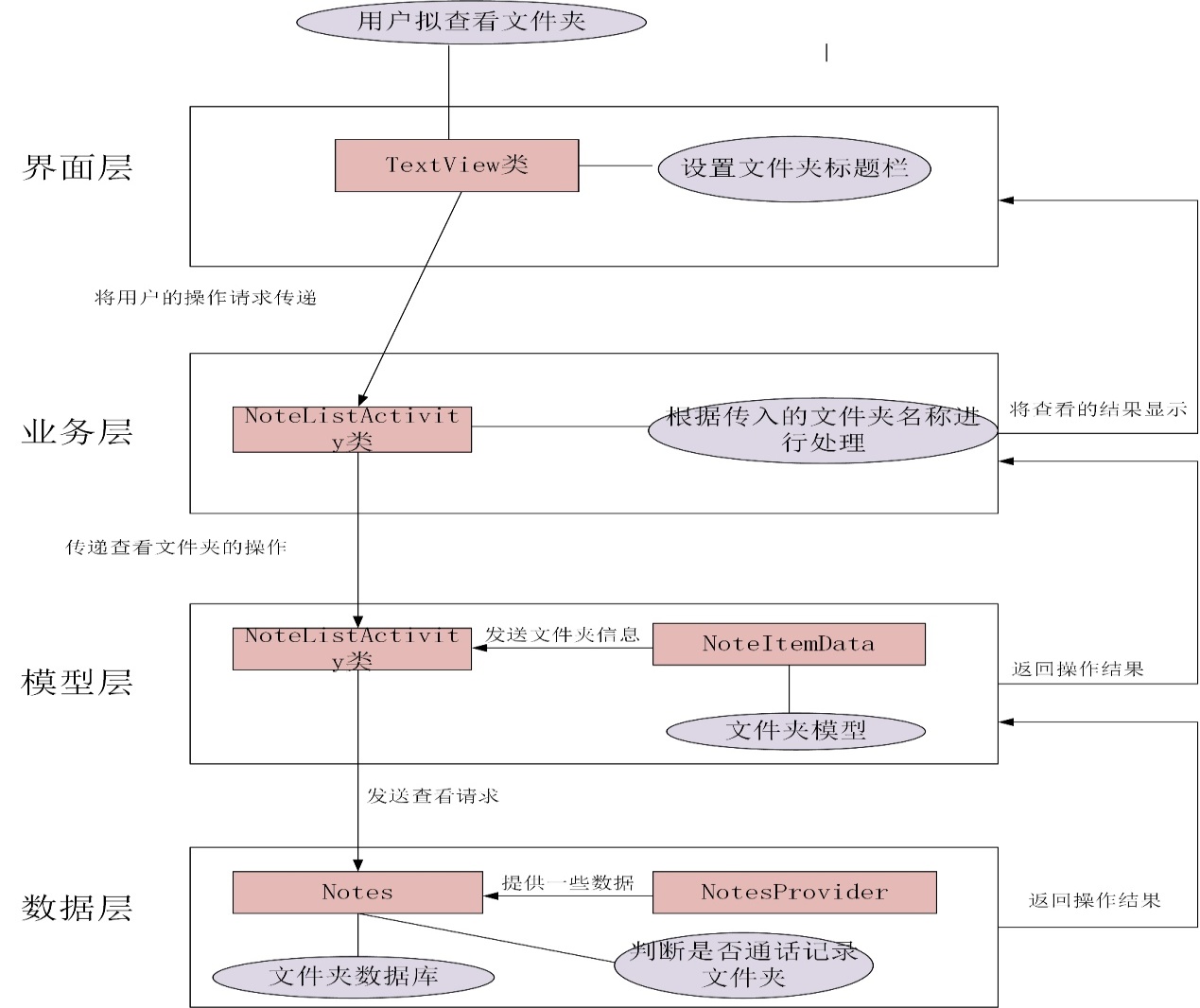
* 1. **业务请求与响应过程**

查看文件夹功能（下简称“功能”）的界面层主要由TextView类通过调用setText()方法和setVisibility()方法实现。setText()设置标题栏文本，而setVisibility()方法设置标题栏可见。

功能的业务层由openFolder()方法实现。首先通过 getId() 方法获取文件夹的 ID，并根据文件夹的 ID 进行不同的逻辑处理。如果文件夹是通话记录文件夹 (Notes.ID\_CALL\_RECORD\_FOLDER)，则界面层的标题栏文本设置为通话记录文件夹的名称，并将按钮隐藏。如果文件夹不是通话记录文件夹，则界面层的标题栏文本设置为文件夹的摘要，按钮保持可见。最后，设置标题栏可见。

功能的模型层主要由ui包中的NoteItemData类中的对象来实现，将该对象作为参数传递给openFolder方法，以表示文件夹信息。

功能的数据层主要由data包中Notes类中的方法来实现。调用ID\_CALL\_RECORD\_FOLDER用于检查是否是通话记录文件夹。



* 1. **类的作用**
     1. **界面层相关的类**

**5.3.1.1、类名：TextView**

1. **类的作用：**

该类是Android 中用于显示文本的视图控件，提供了丰富的方法用于控制文本的显示样式、内容和交互行为，使得开发者可以灵活地定制文本显示的方式和响应用户交互的方式。

1. **类的精读：**

TextView类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| BufferType | 指定文本的缓冲类型 |
| CharSequence | 表示文本内容 |

TextView类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setText(CharSequence text) | Text：字符串文本 | 设置TextView显示的文本内容 | 调用父类的setText()方法 |
| getText() |  | 获取TextView中显示的文本内容 | 直接选择Return对应文本内容 |
| setTextSize(float size) | size：文本字体大小 | 设置文本的字体大小 | 调用相关方法实现文本字体大小修改 |
| setTextColor(int color) | color：文本颜色 | 设置文本的颜色 | 调用相关方法实现文本字体颜色修改 |
| setTypeface(Typeface tf) | tf：字符串文本字体 | 设置文本的字体 | 调用相关方法实现文本字体修改 |
| setGravity(int gravity) | gravity：文本的对齐方式 | 设置文本的对齐方式 | 调用相关方法实现文本字体对齐方式修改 |
| setEllipsize(TextUtils.TruncateAt where) | TruncateAt where：省略方式 | 设置文本过长时的省略方式 | 调用相关方法实现文本省略方式修改 |
| setSingleLine(boolean singleLine) | singleLine：单行显示 | 设置是否单行显示文本 | 调用相关方法实现文本显示方式修改 |
| setLines(int lines) | lines：行数 | 设置文本显示的行数 | 调用相关方法实现文本显示行数修改 |
| setInputType(int type) | type：文本输入类型 | 设置文本输入类型 | 调用相关方法实现文本输入类型修改 |
| append(CharSequence text) | Text：字符串文本 | 在已有文本后追加新的文本 | 调用相关方法实现文本后追加文本功能 |
| setMovementMethod(MovementMethod movement) | movement：字符串文本移动方法 | 设置文本的移动方法，如滚动等。 | 调用相关方法实现文本移动方式修改 |
| setOnEditorActionListener(TextView.OnEditorActionListener l) | l：监听器 | 设置编辑操作的监听器。 | 调用相关方法实现编辑操作的监听器设置 |
| setOnClickListener(View.OnClickListener l) | l：监听器 | 设置点击事件监听器 | 调用相关方法实现点击操作的监听器设置 |

* + 1. **业务层相关的类**
       1. **类名：NoteListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode 是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount 表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用 mBackgroundQueryHandler 发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用 DataUtils.batchMoveToFolder 方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的 Intent，并指定当前文件夹ID |
| batchDelete() | / | 批量删除便签或将它们移动到回收站 | 异步执行的任务，根据同步模式，直接删除便签 |
| deleteFolder(long folderId) | folderId 表示要删除的文件夹的ID | 删除文件夹及其相关操作 | 根据同步模式，直接删除文件夹 |
| openNode(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| **openFolder(NoteItemData data)** | **data是一个 NoteItemData对象，表示要打开的便签数据** | **打开文件夹** | **根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表** |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用 createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| showCreateOrModifyFolderDialog(boolean create) | create 是一个布尔值，表示是创建文件夹还是修改文件夹 | 显示创建或修改文件夹的对话框 | 弹出一个弹窗，将输入参数传入函数 |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId 表示小部件的ID，appWidgetType 表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| exportNoteToText() | / | 将便签导出为文本文件 | 使用 BackupUtils 类执行导出操作，并根据结果显示相应的对话框提示 |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图; view：表示用户长按的特定列表项视图; position：表示被长按的列表项在适配器中的位置; id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为 mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

**5.3.3模型层相关的类**

**5.3.3.1、类名：**NoteItemData

1. **类的作用：**

表示便签项的数据。该类通过从数据库查询结果的Cursor中获取相应的列数据来初始化实例。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mId | 便签项的ID |
| mId | 提醒时间 |
| mBgColorId | 背景颜色ID |
| mCreatedDate | 创建时间 |
| mHasAttachment | 是否有附件 |
| mModifiedDate | 最近修改时间 |
| mNotesCount | 笔记数量 |
| mParentId | 父文件夹ID |
| mSnippet | 便签内容的摘要 |
| mType | 便签类型 |
| mWidgetId | Widget的ID |
| mWidgetType | Widget类型 |
| mName | 联系人名字 |
| mPhoneNumber | 联系人电话号码 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteItemData(Context context, Cursor cursor) | Context:文本  Cursor：游标 | 构造函数，从数据库查询结果的Cursor中初始化便签项数据。 | 调用相关方法，实现数据库的查询和数据的初始化。 |
| isOneFollowingFolder() | 无 | 判断是否是一个便签项后面跟随一个文件夹 | 调用相关方法，实现是否跟随一个文件夹的判断 |
| isMultiFollowingFolder() | 无 | 判断是否是一个便签项后面跟随多个文件夹 | 调用相关方法，实现是否跟随多个文件夹的判断 |
| isLast() | 无 | 判断是否是列表中的最后一个项 | 调用相关方法，实现对是否是列表中的最后一个项的判断 |
| getCallName() | 无 | 获取联系人名字 | Return直接返回 |
| isFirst() | 无 | 判断是否是列表中的第一个项 | 调用相关方法，实现是否是列表中的第一个项的判断 |
| isSingle() | 无 | 判断列表是否只有一个项 | 调用相关方法，实现列表是否只有一个项的判断 |
| getId() | 无 | 获取便签项的ID | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getCreatedDate() | 无 | 获取创建时间 | Return直接返回 |
| hasAttachment() | 无 | 判断是否有附件 | 调用相关方法，实现对是否有附件的判断 |
| getModifiedDate() | 无 | 获取最近修改时间 | Return直接返回 |
| getBgColorId() | 无 | 获取背景颜色ID | Return直接返回 |
| getWidgetType() | 无 | 获取父文件夹ID | Return直接返回 |

* + 1. **数据层相关的类**

**5.3.4.1、****类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与便签关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表便签的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表便签是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的便签数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本便签的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本便签的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本便签的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本便签的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

1. **功能六：删除文件夹**
   1. **功能介绍**

**功能概述：**选中指定的文件夹，点击删除按钮，在确认后删除指定文件夹。

**操作步骤：**在主界面的右下角有一个菜单栏，用户点击按钮，选择文件夹列表，点击选中拟删除的文件夹，点击确认后删除对应文件夹。

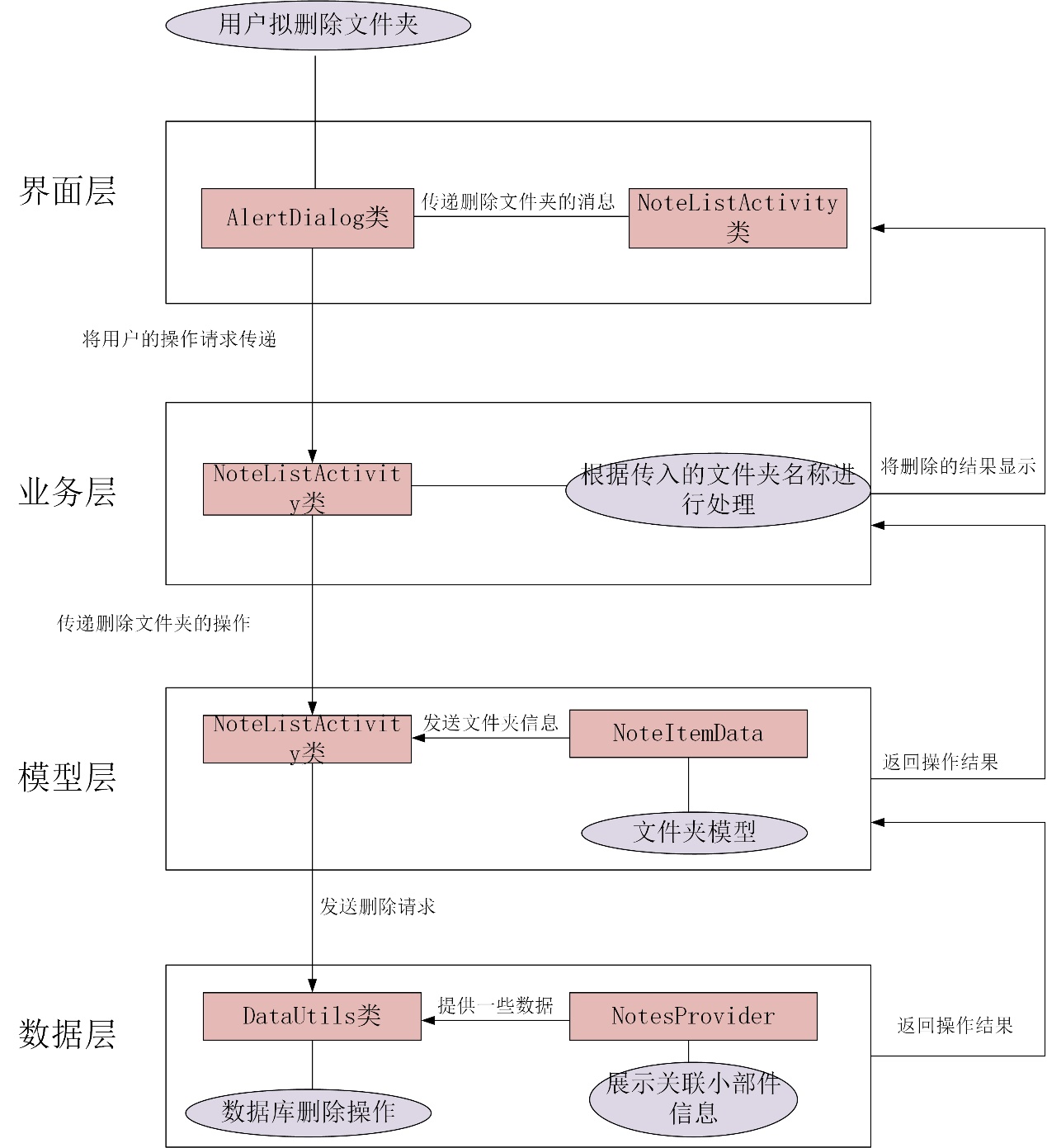
* 1. **业务请求与响应过程**

删除文件夹功能（下简称“功能”）的界面层主要由包ui中NotesListActivity类实现。showCreateOrModifyFolderDialog()中使用AlertDialog.Builder构建了一个对话框，显示一个确认取消按钮的对话框。

功能的业务层由deleteFolder()方法实现。首先判断文件夹 ID 是否为根文件夹，如果是则记录错误日志并返回，不进行删除。然后根据isSyncMode()判断同步模式，调用batchDeleteNotes()方法删除文件夹，或调用batchMoveToFolder()将文件夹及其内容移动到回收站，最后调用updateWidget()来更新与该文件夹相关联的小部件信息。

功能的模型层主要由ui包中的NoteItemData类中的对象来实现，将该对象作为参数传递给deleteFolder方法，以表示文件夹信息。

功能的数据层主要由tool包中DataUtil类中的方法,来实现获取文件夹关联的小部件信息和执行批量删除或移动操作。



* 1. **类的作用**
     1. **界面层相关的类**

**6.3.1.1、类名：AlterDialog**

1. **类的作用：**

该类是Android 中的对话框类，用于显示一个弹出式的对话框，可以显示一些信息、接收用户输入等。

1. **类的精读：**

AlterDialog类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| Button | 表示对话框中的按钮 |

TextView类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setMessage(CharSequence message) | message：字符串文本 | 设置对话框中显示的消息 | 调用相关方法实现消息的显示 |
| setTitle(CharSequence title) | title：字符串文本 | 设置对话框的标题 | 调用相关方法实现对话框标题的设置 |
| setIcon(int resId) | resId：对话框图标 | 设置对话框的图标 | 调用相关方法实现对话框标题的设置 |
| setCancelable(boolean cancelable) | cancelable：是否可取消 | 设置是否可以通过返回键取消对话框 | 调用相关方法实现是否可以通过返回键取消对话框的判断 |
| setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener) | text：文本  listener:事件监听器 | 设置对话框的 Neutral 按钮，可设置文本和点击事件 | 调用相关方法实现对话框对话框的 Neutral 按钮的设置 |
| setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener) | text：文本  listener:事件监听器 | 设置对话框的 Positive 按钮，可设置文本和点击事件 | 调用相关方法实现对话框Positive 按钮的设置 |
| setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener) | text：文本  listener:事件监听器 | 设置对话框的 Negative 按钮，可设置文本和点击事件 | 调用相关方法实现对话框Negative 按钮的设置 |
| setSingleChoiceItems(CharSequence[] items, int checkedItem, DialogInterface.OnClickListener listener) | []Items：列表  checkedItem：选中列表项  listener:事件监听器 | 设置单选列表项，可响应用户的选择 | 调用相关方法实现单选列表项的设置 |
| setItems(CharSequence[] items, DialogInterface.OnClickListener listener) | []Items：列表  listener:事件监听器 | 设置列表项，可响应用户的选择 | 调用相关方法实现列表项的设置 |
| setMultiChoiceItems(CharSequence[] items, boolean[] checkedItems, DialogInterface.OnMultiChoiceClickListener listener) | []Items：列表  checkedItems：选中列表项  listener:事件监听器 | 设置多选列表项，可响应用户的选择 | 调用相关方法实现多选列表项的设置 |
| setOnCancelListener(DialogInterface.OnCancelListener listener) | listener:事件监听器 | 设置对话框取消事件的监听器 | 调用相关方法实现对话框取消事件的监听器设置 |
| setOnDismissListener(DialogInterface.OnDismissListener listener) | listener:事件监听器 | 设置对话框关闭事件的监听器 | 调用相关方法实现对话框关闭事件的监听器设置 |
| setOnKeyListener(DialogInterface.OnKeyListener onKeyListener) | onKeyListener:键盘事件监听器 | 设置对话框键盘事件的监听器 | 调用相关方法实现对话框键盘事件的监听器设置 |
| setView(View view) | View：视图 | 设置对话框的自定义视图 | 调用相关方法实现对话框自定义视图的设置 |

* + 1. **业务层相关的类**
       1. **类名：NoteListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode 是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount 表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用 mBackgroundQueryHandler 发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用 DataUtils.batchMoveToFolder 方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的 Intent，并指定当前文件夹ID |
| **batchDelete()** | **/** | **批量删除便签或将它们移动到回收站** | **异步执行的任务，根据同步模式，直接删除便签** |
| **deleteFolder(long folderId)** | **folderId 表示要删除的文件夹的ID** | **删除文件夹及其相关操作** | **根据同步模式，直接删除文件夹** |
| openNode(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| openFolder(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开文件夹 | 根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表 |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用 createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| showCreateOrModifyFolderDialog(boolean create) | create 是一个布尔值，表示是创建文件夹还是修改文件夹 | 显示创建或修改文件夹的对话框 | 弹出一个弹窗，将输入参数传入函数 |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId 表示小部件的ID，appWidgetType 表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| exportNoteToText() | / | 将便签导出为文本文件 | 使用 BackupUtils 类执行导出操作，并根据结果显示相应的对话框提示 |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图; view：表示用户长按的特定列表项视图; position：表示被长按的列表项在适配器中的位置; id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为 mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

* + 1. **模型层相关的类**

**6.3.3.1、类名：**NoteItemData

1. **类的作用：**

表示便签项的数据。该类通过从数据库查询结果的Cursor中获取相应的列数据来初始化实例。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mId | 便签项的ID |
| mId | 提醒时间 |
| mBgColorId | 背景颜色ID |
| mCreatedDate | 创建时间 |
| mHasAttachment | 是否有附件 |
| mModifiedDate | 最近修改时间 |
| mNotesCount | 笔记数量 |
| mParentId | 父文件夹ID |
| mSnippet | 便签内容的摘要 |
| mType | 便签类型 |
| mWidgetId | Widget的ID |
| mWidgetType | Widget类型 |
| mName | 联系人名字 |
| mPhoneNumber | 联系人电话号码 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteItemData(Context context, Cursor cursor) | Context:文本  Cursor：游标 | 构造函数，从数据库查询结果的Cursor中初始化便签项数据。 | 调用相关方法，实现数据库的查询和数据的初始化。 |
| isOneFollowingFolder() | 无 | 判断是否是一个便签项后面跟随一个文件夹 | 调用相关方法，实现是否跟随一个文件夹的判断 |
| isMultiFollowingFolder() | 无 | 判断是否是一个便签项后面跟随多个文件夹 | 调用相关方法，实现是否跟随多个文件夹的判断 |
| isLast() | 无 | 判断是否是列表中的最后一个项 | 调用相关方法，实现对是否是列表中的最后一个项的判断 |
| getCallName() | 无 | 获取联系人名字 | Return直接返回 |
| isFirst() | 无 | 判断是否是列表中的第一个项 | 调用相关方法，实现是否是列表中的第一个项的判断 |
| isSingle() | 无 | 判断列表是否只有一个项 | 调用相关方法，实现列表是否只有一个项的判断 |
| getId() | 无 | 获取便签项的ID | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getCreatedDate() | 无 | 获取创建时间 | Return直接返回 |
| hasAttachment() | 无 | 判断是否有附件 | 调用相关方法，实现对是否有附件的判断 |
| getModifiedDate() | 无 | 获取最近修改时间 | Return直接返回 |
| getBgColorId() | 无 | 获取背景颜色ID | Return直接返回 |
| getWidgetType() | 无 | 获取父文件夹ID | Return直接返回 |

* + 1. **数据层相关的类**

**6.3.4.1 类名：Notes**

1. **类的作用：**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

1. **功能七：修改文件夹名称**
   1. **功能介绍**

**功能概述：**选中一个文件夹，读取用户输入的文件夹名称，将原有的文件夹名称进行修改，并将修改后的结果存储到本地数据库。

**操作步骤：**在主界面的右下角有一个菜单栏，用户点击按钮，选中特定文件夹，选择修改文件夹名称选项，输入拟修改的名称，点击确认按钮即可。

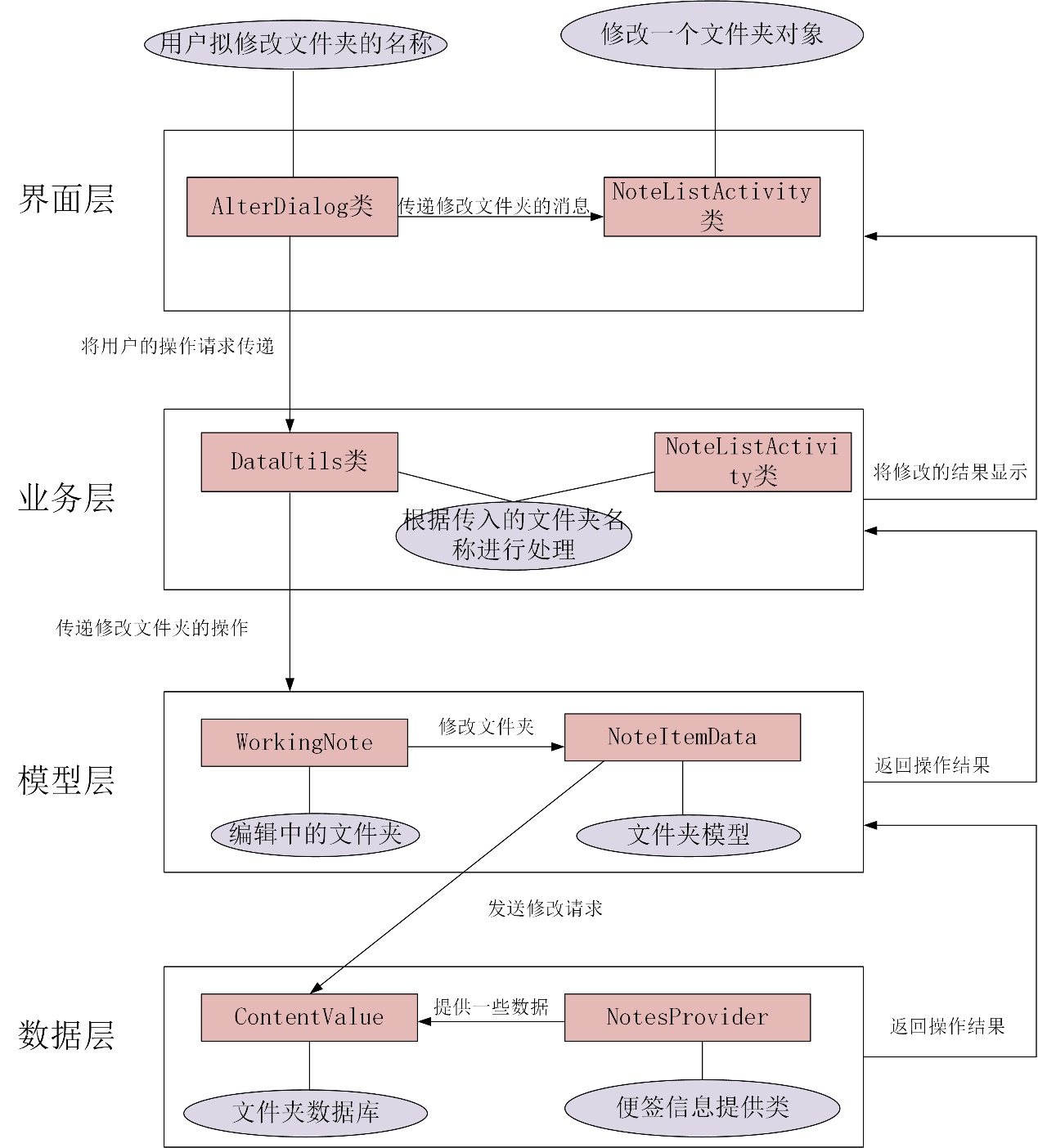
* 1. **业务请求与响应过程**

修改文件夹功能（下简称“功能”）的界面层主要由包ui中NotesListActivity类通过调用showCreateOrModifyFolderDialog()方法实现。showCreateOrModifyFolderDialog()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框，并调用showSoftInput()方法显示软键盘。

功能的业务层由DataUtils类和NotesListActivity类实现。首先通过TextUtils的isEmpty()方法来判断文件夹名是否为空，其次调用checkVisibleFolderName()方法判断文件夹名是否存在，在不存在的情况下然后创建 ContentValues 对象并存储信息，最后使用 ContentResolver 的 insert()方法将文件夹中的信息插入到数据库中。

功能的模型层主要由model包中的WorkingNote类和NoteItemData类来实现，主要包括getFolderId()等方法。

功能的数据层主要由ContentValue类的对象来实现。创建一个ContentValue的对象，并调用put()方法将文件夹的名称和类型存入对象。最后使用ContentResolver的insert()方法将包含文件夹信息的ContentValues对象插入到数据库中。



* 1. **类的作用**
     1. **界面层相关的类**
        1. **类名：NoteListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode 是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount 表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用 mBackgroundQueryHandler 发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用 DataUtils.batchMoveToFolder 方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的Intent，并指定当前文件夹ID |
| batchDelete() | / | 批量删除便签或将它们移动到回收站 | 异步执行的任务，根据同步模式，直接删除便签 |
| deleteFolder(long folderId) | folderId 表示要删除的文件夹的ID | 删除文件夹及其相关操作 | 根据同步模式，直接删除文件夹 |
| openNode(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| openFolder(NoteItemData data) | 同上 | 打开文件夹 | 根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表 |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用 createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| **showCreateOrModifyFolderDialog(boolean create)** | **create 是一个布尔值，表示是创建文件夹还是修改文件夹** | **显示创建或修改文件夹的对话框** | **弹出一个弹窗，将输入参数传入函数** |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId 表示小部件的ID，appWidgetType 表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| exportNoteToText() | / | 将便签导出为文本文件 | 使用 BackupUtils 类执行导出操作，并根据结果显示相应的对话框提示 |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图; view：表示用户长按的特定列表项视图; position：表示被长按的列表项在适配器中的位置; id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为 mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

* + 1. **业务层相关的类**

**7.3.2.1类名：**DataUtils

1. **类的作用：**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

(**2) 类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

* + 1. **模型层相关的类**

**7.3.3.1、类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 标记当前便签状态 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |
| NoteSettingChangedListener | 定义便签设置改变监听器的接口 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getTitleBgResId() | 无 | 返回当前的标题颜色的ID值 | 通过背景颜色ID值来调用NoteBgResource内的方法来匹配返回标题的ID值 |
| WorkingNote(Context context, long folderId) | context:Android应用程序的上下文。  folderId:笔记所属文件夹的ID。 | 创建一个新的笔记对象 | 实例化对象 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getBgColorResId(): | 无 | 获取笔记背景颜色资源ID | Return直接返回 |
| getCheckListMode() | 无 | 获取笔记模式 | Return直接返回 |
| getModifiedDate() | 无 | 获取笔记修改日期 | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getContent() | 无 | 获取笔记内容 | Return直接返回 |
| getNoteId() | 无 | 获取笔记ID | Return直接返回 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| getTitleBgResId() | 无 | 获取笔记标题背景资源ID | Return直接返回 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context: Android 应用程序的上下文。  folderId: 笔记所属文件夹的ID。  widgetId: 笔记对应的小部件的ID。  widgetType: 笔记对应的小部件类型。  defaultBgColorId: 笔记的默认背景颜色ID。 | 创建一个空白笔记 | 通过调用一系列set方法，设置笔记的初始情况 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |

* + - 1. **类名：NoteItemData**

1. **类的作用：表示便签项的数据。该类通过从数据库查询结果的 Cursor 中获取相应的列数据来初始化实例。**
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mId | 便签项的ID |
| mId | 提醒时间 |
| mBgColorId | 背景颜色ID |
| mCreatedDate | 创建时间 |
| mHasAttachment | 是否有附件 |
| mModifiedDate | 最近修改时间 |
| mNotesCount | 笔记数量 |
| mParentId | 父文件夹ID |
| mSnippet | 便签内容的摘要 |
| mType | 便签类型 |
| mWidgetId | Widget的ID |
| mWidgetType | Widget类型 |
| mName | 联系人名字 |
| mPhoneNumber | 联系人电话号码 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteItemData(Context context, Cursor cursor) | Context:文本  Cursor：游标 | 构造函数，从数据库查询结果的Cursor中初始化便签项数据。 | 调用相关方法，实现数据库的查询和数据的初始化。 |
| isOneFollowingFolder() | 无 | 判断是否是一个便签项后面跟随一个文件夹 | 调用相关方法，实现是否跟随一个文件夹的判断 |
| isMultiFollowingFolder() | 无 | 判断是否是一个便签项后面跟随多个文件夹 | 调用相关方法，实现是否跟随多个文件夹的判断 |
| isLast() | 无 | 判断是否是列表中的最后一个项 | 调用相关方法，实现对是否是列表中的最后一个项的判断 |
| getCallName() | 无 | 获取联系人名字 | Return直接返回 |
| isFirst() | 无 | 判断是否是列表中的第一个项 | 调用相关方法，实现是否是列表中的第一个项的判断 |
| isSingle() | 无 | 判断列表是否只有一个项 | 调用相关方法，实现列表是否只有一个项的判断 |
| getId() | 无 | 获取便签项的ID | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getCreatedDate() | 无 | 获取创建时间 | Return直接返回 |
| hasAttachment() | 无 | 判断是否有附件 | 调用相关方法，实现对是否有附件的判断 |
| getModifiedDate() | 无 | 获取最近修改时间 | Return直接返回 |
| getBgColorId() | 无 | 获取背景颜色ID | Return直接返回 |
| getWidgetType() | 无 | 获取父文件夹ID | Return直接返回 |

* + 1. **数据层相关的类**

**7.3.4.1、类名：**ContentValues

1. **类的作用：此类用于存储 ContentResolver 可以处理的一组值。**
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mValues | HashMap对象，用以存储键值对 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| **put(String key, Byte value)** | Key：键  Value值 | 向 ContentValues 中添加一个 byte 类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Short value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Short**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Integer value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Integer**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Long value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Long**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, Float value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Float**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key,Double value)** | Key：键  Value值 | 向 ContentValues 中添加一个**Double**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, String value)** | Key：键  Value值 | 向 ContentValues 中添加一个**String**类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **put(String key, byte[] value)** | Key：键  Value值 | 向 ContentValues 中添加一个 byte数组类型的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **putNull(String key)** | Key：键 | 向 ContentValues 中添加一个空值的键值对 | 直接调用HashMap中的put方法插入对应键值对 |
| **get(String key)**: | Key：键 | 根据键获取对应的值 | 直接调用HashMap中的get方法 |
| **containsKey(String key)** | Key：键 | 检查是否包含指定键 | 直接调用HashMap中的contains方法 |
| **remove(String key)** | Key：键 | 根据键移除键值对 | 直接调用HashMap中的remove方法 |
| **clear()** |  | 清空ContentValues 中的所有键值对。 | 直接调用HashMap中的clear方法 |

1. **功能八：导出便签**
   1. **功能介绍**

**功能概述：**主界面的菜单栏中选择“导出文件”，在Android手机提供SD卡支持的情况下，能够将小米便签中的便签内容逐个转化为“.txt”格式的文本文档，并存储到SD卡中。

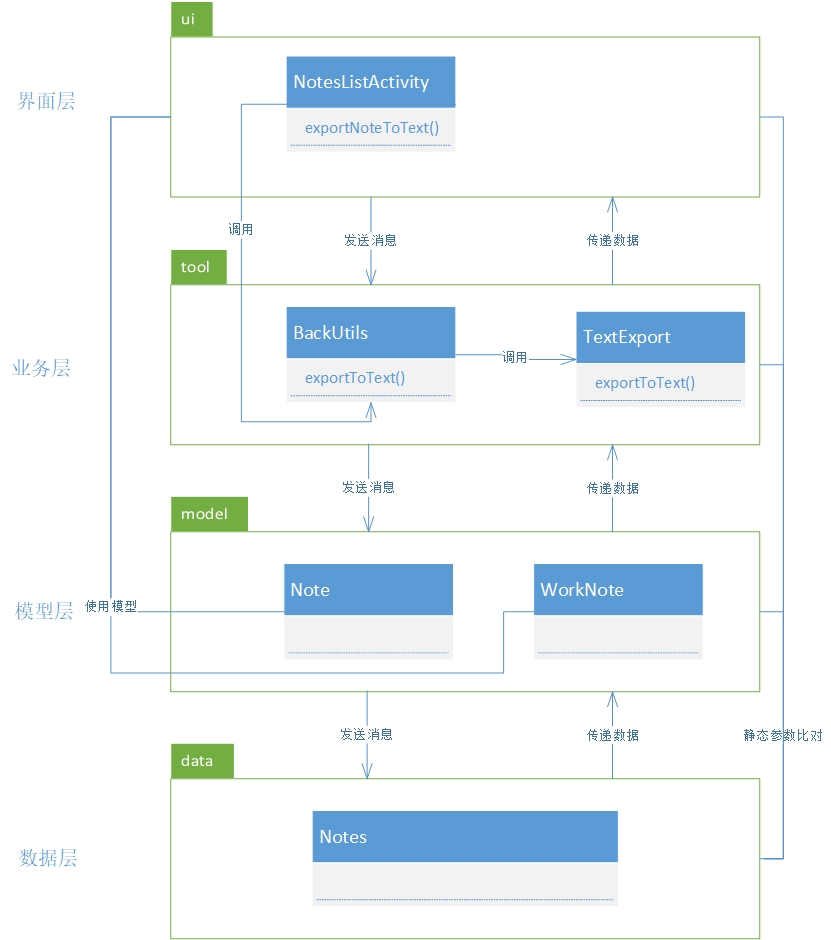
**操作步骤：**在主列表界面点击右下角菜单栏，点击“导出文本”按钮，软件自动检测Android手机是否有SD卡，有SD卡则导出文本到路径“mnt/sdcard/MIUI/notes”路径下；反之如果SD卡未挂载、SD卡正在被占用、系统出错则弹出错误信息提示弹窗。下图为成功导出界面和未成功导出界面（由于测试物理机、虚拟机均没有SD卡，成功用例图系网上查找）

* 1. **业务请求与响应过程**

导出便签功能（下简称“功能”）的界面层由包ui中NotesListActivity类调用exportNoteToText()方法实现。exportNoteToText()中新建了一个不可修改的BackupUtils备份工具类实例；新建一个异步任务AsyncTask执行具体的备份功能，在后台调用BackupUtils的exportToText()方法，在界面层接收下层返回的结果，并对执行结果进行弹窗提示。

功能的业务层由包tool中BackUtils类调用exportToText()方法实现。exportToText()中调用了内部类TextExport的exportToText()将便签导出生成的文本写入目标地址文件中，其中通过getExportToTextPrintStream()方法创建PrintStream，该Stream包含了generateFileMountedOnSDcard()方法用以生成待存储的文件。

功能的模型层由包model中的Note类和WorkingNote类分别实现，Note类定义了便签类的抽象参数，WorkingNote类定义了对便签操作的方法（如setBgColorId()、setCheckListMode()等）。

****功能的数据层由包data中的Notes类实现，Notes类具体定义了便签类的具体参数，将存储的便签数据向上传递。

* 1. **类的作用**
     1. **界面层相关的类**

**8.3.1.1、类名：NoteListActivity**

1. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用mBackgroundQueryHandler发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用DataUtils.batchMoveToFolder方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的 Intent，并指定当前文件夹ID |
| batchDelete() | / | 批量删除便签或将它们移动到回收站 | 异步执行的任务，根据同步模式，直接删除便签 |
| deleteFolder(long folderId) | folderId表示要删除的文件夹的ID | 删除文件夹及其相关操作 | 根据同步模式，直接删除文件夹 |
| openNode(NoteItemData data) | data是一个NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| openFolder(NoteItemData data) | 同上 | 打开文件夹 | 根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表 |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| showCreateOrModifyFolderDialog(boolean create) | create 是一个布尔值，表示是创建文件夹还是修改文件夹 | 显示创建或修改文件夹的对话框 | 弹出一个弹窗，将输入参数传入函数 |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId表示小部件的ID，appWidgetType表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| **exportNoteToText()** | **/** | **将便签导出为文本文件** | **使用BackupUtils类执行导出操作，并根据结果显示相应的对话框提示** |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图;view：表示用户长按的特定列表项视图;position：表示被长按的列表项在适配器中的位置;id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

* + 1. **业务层相关的类**
       1. **类名：**BackUtils

1. **类的作用：**

一个备份工具类，能够处理备份和导出便签数据到文本文件的功能，用于导出Android便签应用程序数据到文本文件的工具类

1. **类的精读：**

BackupUtils类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| TAG | 用于日志记录的标签，通常用于在应用中标识日志消息的来源 |
| sInstance | 用于实现单例模式的静态成员变量，用于存储BackupUtils的单一实例 |
| STATE\_SD\_CARD\_UNMOUONTED | 表示SD卡当前未挂载的状态 |
| STATE\_BACKUP\_FILE\_NOT\_EXIST | 表示备份文件不存在的状态 |
| STATE\_DATA\_DESTROIED | 表示数据损坏或格式不正确的状态 |
| STATE\_SYSTEM\_ERROR | 表示备份或恢复操作中发生运行时异常的状态 |
| STATE\_SUCCESS | 表示备份或恢复操作成功完成的状态 |
| mTextExport | TextExport类的实例，用于处理将便签数据导出为文本的操作 |
| TextExport.TEXT\_FORMAT | 存储用于格式化文本导出的字符串数组 |
| TextExport.mContext | 用于访问应用的资源和内容提供器 |
| TextExport.mFileName | 导出的文本文件名 |
| TextExport.mFileDirectory | 导出的文本文件目录 |

BackupUtils类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getInstance(Context context) | 传入一个Context参数，用于初始化该实例 | 获取BackupUtils的单例实例 | 传入一个 Context参数，获取单例实例 |
| TextExport(Context context) | 一个 Context 参数 | 初始化TextExport实例 | / |
| externalStorageAvailable() | / | 检查外部存储是否可用 | / |
| exportToText() | / | 导出便签数据为文本文件的方法 | 调用TextExport对象的函数 |
| getExportedTextFileName() | / | 获取导出的文本文件名 | 调用TextExport对象的函数 |
| getExportedTextFileDir() | / | 获取导出的文本文件目录 | 调用TextExport对象的函数 |
| TextExport.getFormat(int id) | id表示传入的id | 根据给定的格式ID返回对应的格式化字符串 | 返回为该id的格式化字符串 |
| TextExport.exportFolderToText(String folderId, PrintStreamps) | folderId是文件夹的id，PrintStream输出流 | 导出指定文件夹的便签到文本 | 创建文件夹游标，游标移动到传入id的文件夹 |
| TextExport.exportNoteToText(String noteId, PrintStreamps) | noteId便签的ID；PrintStream输出流 | 导出指定便签的内容到文本 | 创立数据游标，根据传入的id移动到该便签，开始导出 |
| TextExport.exportToText() | / | 导出便签数据为文本文件 | 获取文件夹id，调用exportFolderToText方法；获取便签id，调用exportNoteToText方法 |
| TextExport.getExportToTextPrintStream() | / | 获取一个PrintStream对象 | 获取输出地址，创建备份文件，初始化一个ps对象 |
| generateFileMountedOnSDcard(Context context, int filePathResId, int fileNameFormatResId) | 一个Context对象，一个文件路径资源ID；一个文件名格式资源 ID | 生成一个在SD卡上的文件用于存储导出的数据 | 将SD卡的根目录路径追加到 sb 中；将资源ID filePathResId对应的字符串追加到 sb 中 |

* + 1. **模型层相关的类**
       1. **类名：**Note

1. **类的作用：**

管便签（notes）的数据，包括创建、编辑、同步等操作。它维护便签的元数据和文本数据以及通话数据，并将这些数据同步到Android应用的数据库中

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNoteDiffValues | 一个ContentValues对象，用于存储便签的元数据的差异 |
| mNoteData | 一个NoteData对象，用于管理便签的文本数据和通话数据 |
| TAG | 一个用于日志记录的常量字符串 |

Note类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getNewNoteId(Context context, long folderId) | context：Android应用程序的上下文；folderId表示便签所属的文件夹的ID | 创建一个新的便签ID并插入数据库中，返回新便签的ID | 创建一个ContentValues对象values，用于存储便签的元数据，初始化values，从URI中提取新便签的ID并返回 |
| Note() | / | 构造函数，初始化mNoteDiffValues和mNoteData对象 | / |
| setNoteValue(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置便签元数据的键值对，用于标记元数据的本地修改 | 将键值对存储到mNoteDiffValues，并更新LOCAL\_MODIFIED和MODIFIED\_DATE的值 |
| setTextData(String key, String value) | String key：文本数据的键名；String value：文本数据的键值 | 设置文本数据的键值对 | 将键值对存储到mNoteData的文本数据中 |
| setTextDataId(long id) | id：文本数据的ID | 设置文本数据的ID | 将ID存储到mNoteData中 |
| getTextDataId() | / | 获取文本数据的ID | 返回mNoteData中的文本数据ID |
| setCallDataId(long id) | id：通话数据的ID | 设置通话数据的ID | 将ID存储到mNoteData中 |
| setCallData(String key, String value) | key：通话数据的键名；value：通话数据的键值 | 设置通话数据的键值对 | 将键值对存储到mNoteData的通话数据中 |
| isLocalModified() | / | 判断便签是否在本地有修改 | 检查mNoteDiffValues和mNoteData是否包含本地修改的数据，如果有则返回true，否则返回false |
| syncNote(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 同步便签到数据库中 | 如果没有本地修改，直接返回true表示同步成功；如果有本地修改，首先尝试更新便签的元数据 |
| NoteData.isLocalModified() | / | 判断文本数据或通话数据是否在本地有修改 | / |
| NoteData.setTextDataId(long id) | id表示传入的id | 设置文本数据的ID | / |
| NoteData.setCallDataId(long id) | id表示传入的id | 设置通话数据的ID | / |
| NoteData.setCallData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置通话数据的键值对 | / |
| NoteData.setTextData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置文本数据的键值对 | / |
| pushIntoContentResolver(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 将文本数据或通话数据同步到数据库中 | 根据是否存在本地修改，构建数据库操作，然后执行同步操作 |

* + - 1. **类名：**WorkingNote

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| WorkingNote(Context context, long folderId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID | 构造函数，用于创建一个新的工作便签 | / |
| WorkingNote(Context context, long noteId, long folderId) | context：Android应用程序的上下文；noteId：要加载的便签的ID；folderId：便签所属的文件夹ID | 构造函数，用于加载已有的工作便签 | / |
| loadNote() | / | 用于加载便签的元数据信息 | 创建游标，游标移动到制定，获取属性值并赋给对应便签属性 |
| loadNoteData() | / | 用于加载便签的内容数据 | 创建游标，游标移动到制定，获取文本内容并赋给 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID；widgetId：小部件的ID；widgetType：小部件类型；defaultBgColorId：默认的背景颜色ID | 创建一个新的空白便签 | 创建一个新的工作便签，并设置其属性，包括背景颜色、小部件ID等 |
| load(Context context, long id) | context：Android应用程序的上下文；id：要加载的便签的ID | 加载现有的工作便签 | 根据传入的条件查找便签，并加载 |
| saveNote() | / | 保存便签的方法，包括新建便签和已有便签的保存 | 判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库 |
| existInDatabase() | / | 判断便签是否已存在于数据库中 | 判断NoteId是否存在 |
| isWorthSaving() | / | 判断便签是否值得保存 | 考虑是否已删除、是否存在文本内容和是否有本地修改 |
| setOnSettingStatusChangedListener(NoteSettingChangedListener l) | 便签属性改变监听器 | 设置便签设置状态变化的监听器 | / |
| setAlertDate(long date, boolean set) | date：提醒日期的时间戳；set：是否设置提醒 | 设置提醒日期 | 如果提醒日期不同于当前日期，更新提醒日期并通知监听器 |
| markDeleted(boolean mark) | mark：是否标记为已删除 | 标记便签是否已删除 | 标记便签是否已删除，同时通知监听器 |
| setBgColorId(int id) | id：背景颜色的ID | 设置背景颜色的ID | 如果背景颜色ID不同于当前ID，更新背景颜色ID并通知监听器 |
| setCheckListMode(int mode) | mode：模式 | 设置便签的清单模式 | 如果模式不同于当前模式，更新模式并通知监听器 |
| setWidgetType(int type) | type：小部件类型 | 设置小部件的类型 | 确定widget类型是否变化，变化了就更新widget参数 |
| setWidgetId(int id) | id：小部件的ID | 设置小部件的ID | 确定widget的id是否变化，更新widget |
| setWorkingText(String text) | text：文本内容 | 设置便签的文本内容 | 如果输入内容和原文本不一致则进行内容更新 |
| convertToCallNote(String phoneNumber, long callDate) | phoneNumber：通话记录的电话号码；callDate：通话日期的时间戳 | 将便签转换为通话记录便签 | 将传入的电话号码设置为电话数据 |
| hasClockAlert() | / | 判断便签是否有提醒设置 | / |
| getContent() | / | 获取便签的文本内容 | / |
| getAlertDate() | / | 获取便签的提醒日期 | / |
| getModifiedDate() | / | 获取便签的最后修改日期 | / |
| getBgColorResId() | / | 获取背景颜色的资源ID | / |
| getBgColorId() | / | 获取背景颜色的ID | / |
| getTitleBgResId() | / | 获取标题栏背景颜色的资源ID | / |
| getCheckListMode() | / | 获取便签的清单模式 | / |
| getNoteId() | / | 获取便签的ID | / |
| getFolderId() | / | 获取便签所属的文件夹ID | / |
| getWidgetId() | / | 获取小部件的ID | / |
| getWidgetType() | / | 获取小部件的类型 | / |
| NoteSettingChangedListener | / | 内部接口，定义了便签设置状态变化的监听器方法 | / |

* + 1. **数据层相关的类**

**8.3.4.1、类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表笔记或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表笔记或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表笔记的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与笔记关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表笔记的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表笔记是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的笔记数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本笔记的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本笔记的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本笔记的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本笔记的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本笔记数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

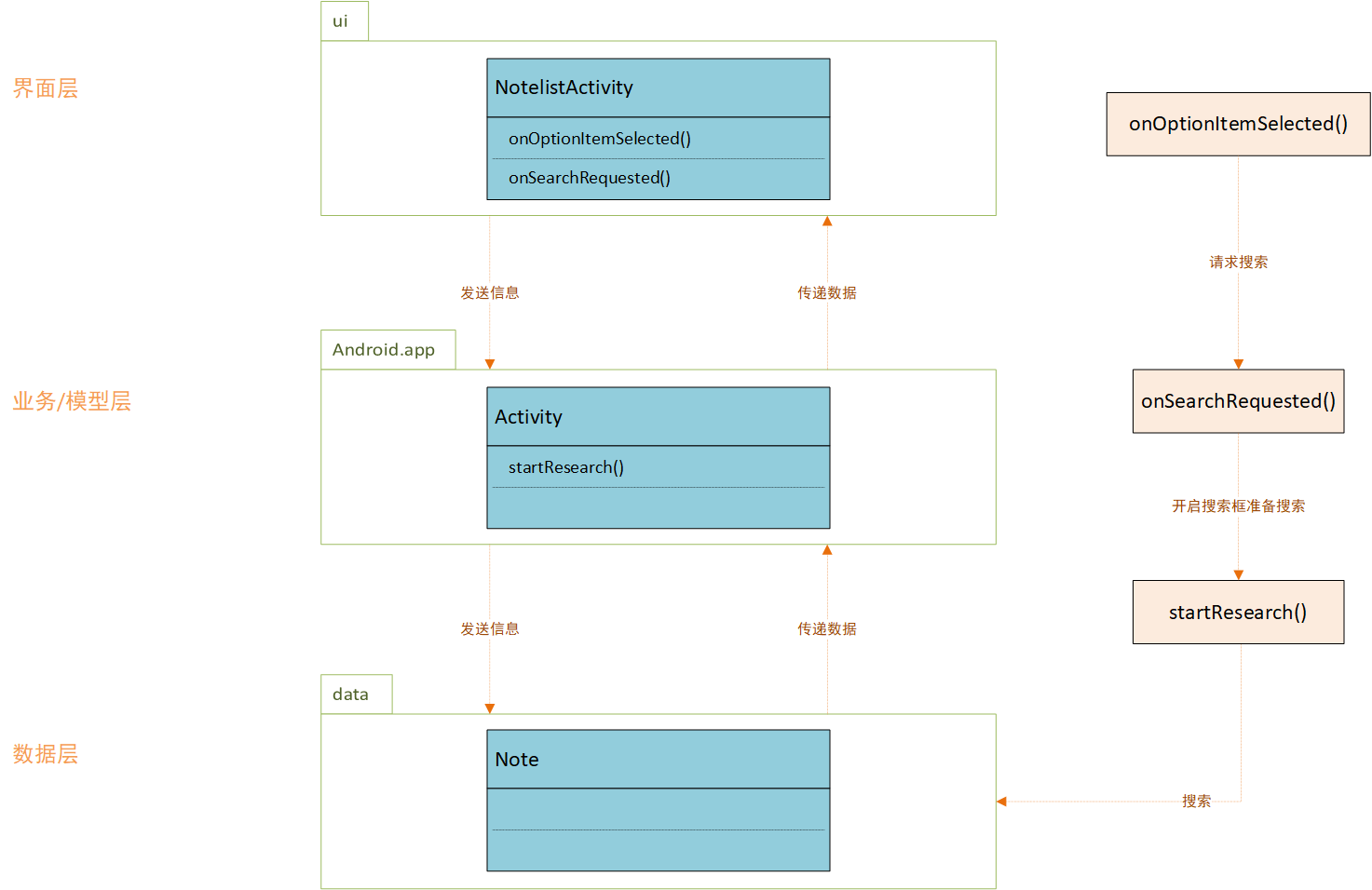
1. **功能九：关键字查询**
   1. **功能介绍**

**功能概述：**通过关键字查询所有便签匹配的内容，并显示在界面上

**操作步骤：**在右下角的菜单中，选择search功能，在顶部的搜索框中输入自己需要的搜索的关键字，在点击搜索。

* 1. **业务请求与响应过程**

首先通过NotelistActivity中的onOptionItemSelected()的方法选择搜索后调用方法onSearchRequested()，在调用Android包中Activity类中startResearch()方法调出搜索引擎框，并完成搜索功能，向data包中note数据进行搜索。



* 1. **类的作用**
     1. **界面层相关的类**

**9.3.1.1、类名：NotelistActivit**

1. **类的作用：**

该类是完成小米便签菜单的显示和交互功能，调用出菜单，并完成用户点击菜单上功能项的交互功能。如搜索，设置

1. **类的精读：**列出类中的主要成员（方法）及各成员（方法）的解释

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| mState | 表示当前的列表编辑状态 |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mContentResolver | 用于访问应用数据 |
| NORMAL\_SELECTION | 用于数据库查询表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 用于数据库查询表示根文件夹的筛选条件 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState，用于保存Activity的状态信息 | 用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| onStart() | 无 | Activity的生命周期方法 | 调用  startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| IntResources() | 无 | 初始化各种资源，应用在Acitivity的onCreate()的方法中 | 直接为监视器，视图等成员直接进行赋值 |
| updateMenu() | 无 | 更新菜单选项的状态和文本的内容 | 检查和跟新Selcect All菜单项，根据已选笔记情况设置它的状态和文本 |
| onDestroyActionMode(ActionMode mode) | mode:表示当前的菜单栏 | 在多选操作结束后恢复到正常应用界面状态 | 取消笔记列表模式，恢复笔记列表长点击模式，使新建笔记可见 |
| finishActionMode() | 无 | 结束多选模式 | 调用finish的方法 |
| onItemCheckedStateChanged (ActionMode mode, int position, long id, boolean checked) | mode:当前的上下文操作对象。  position：选中项的位置  id：选中项的id  checked：表示该项是否选中 | 更新菜单和反应当前的选中状态 | 调用mNotesListAdapter.set  ChekedItem()的方法 |
| onMenuItemClick(MenuItem item) | item:表示被点击的菜单项 | 处理不同点击菜单对应的功能事件。 | 根据不同的点击结果处理不同事件 |
| onTouch(View v,MotionEvent event) | v:当前视图  event:触击事件 | 实现特殊的交互效果 | 通过捕捉触击的位置，匹配是否符合ui范围 |
| onOptionsItemSelected(MenuItem item) | item:当前菜单项 | 实现对应点击的菜单功能 | 通过不同的点击执行不同的功能 |
| onSearchRequested() | 无 | 调用Android包自带的搜索框，和搜索功能完成搜索 | 调用startSearch()方法 |

* + 1. **业务层/模型相关的类：Activity**

**Activity**是**Android**包中的类，里面自带有搜索框和搜索引擎的功能。

### 9.3.3.数据层相关的类

**9.3.3.1、类名：Note**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表笔记或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表笔记或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表笔记的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与笔记关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表笔记的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表笔记是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的笔记数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本笔记的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本笔记的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本笔记的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本笔记的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本笔记数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Note类没有方法

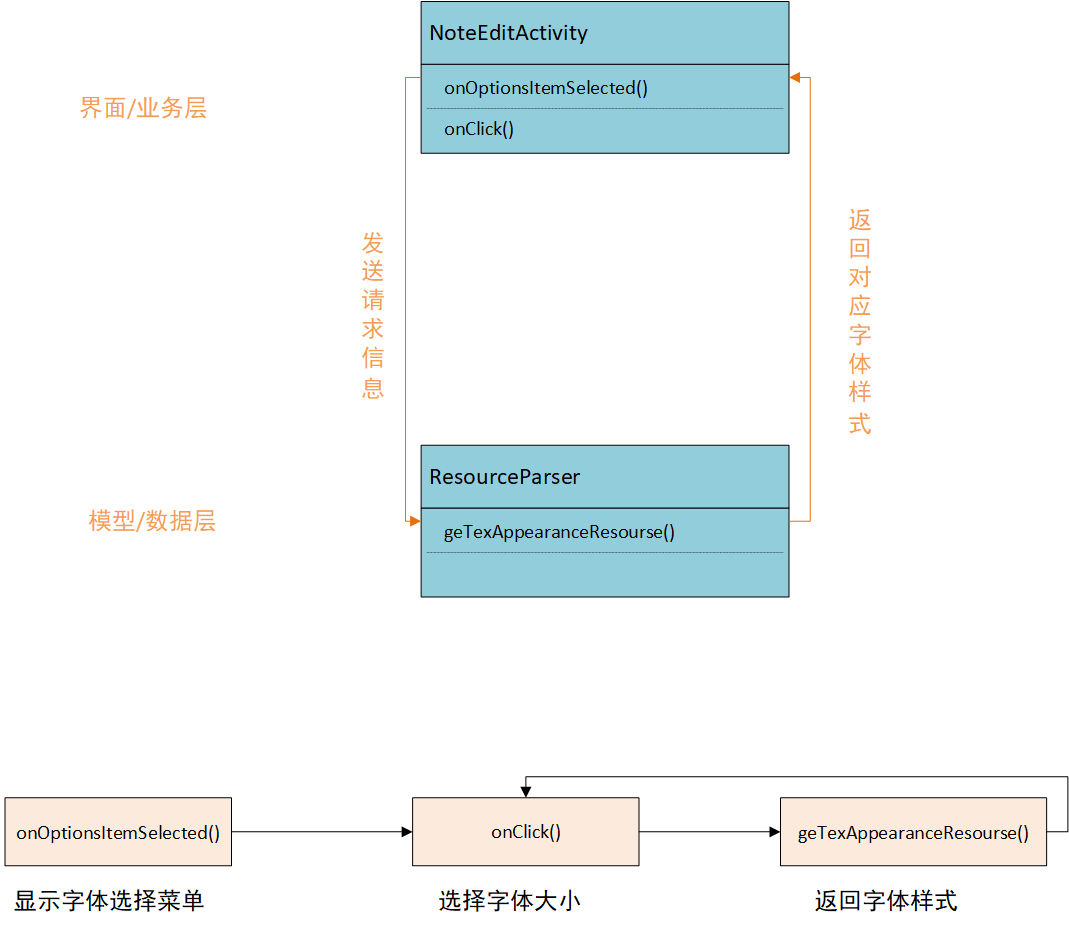
1. **功能十：修改字体大小**
   1. **功能介绍**

**功能概述：**修改便签内字体的显示大小

**操作步骤：**在主界面的右上角点击菜单按钮，在点击Font Size，从large，small，normal，super中选择自己需要的字体的大小。

* 1. **业务请求与响应过程**

修改字体大小的功能，由界面层UI包中NoteEditActivity类中的onOPtionsItemSelected()方法来显示菜单向，然后用户点击后，有click方法读取选中的字体大小，并存储响应的配置信息，然后调用模型层**ResourceParser**类中getTexAppearanceResource()的方法返回对应的字体样式值，由edittor成员调用对应显示的方法，把修改好的字体样式显示出来。所以界面层和业务层都是由NoteEditActivity类完成的，而模型和数据层则是有ResourceParser类来负责。



* 1. **类的作用**
     1. **界面层/业务层相关的类**

**10.3.1.1、类名：NoteEditActivity**

1. **类的作用：**

该类是完成小米便签的编辑活动，在该界面可以进行便签的字体选择，闹钟提醒的设置，和分享该便签等功能。完成菜单业务的各种对应的个性化设置的功能。例如修改字体大小和背景颜色等等

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 用于保存一组视图的引用，有修改日期文本，背景颜色，提醒图标，和提醒日期文本的视图。 |
| mFontSizeSelector | 私有化一个界面，即字体大小选择器，用于对字体大小操作 |
| mFontsizeId | 用于操作字体的大小 |
| PREFERENCE\_FONT\_SIZE | 用做共享首选项中的键 |
| mFontsizeId | 用于操作字体的大小 |
| sFontSizeBtnsMap | 保存字体大小和mFontsizeid间的映射关系 |
| sFontSelectorSelectionMap | 保存字体大小选择器中对应按钮与字体大小间的映射关系。 |
| mNoteEditor | 编辑实现设置字体大小的样式 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| dispactchTouchEvent(MotionEventev) | ev:屏幕触碰的操作 | 使得颜色选择器和字体大小的选择器在屏幕上可见。同时用于在其他特殊操作下隐藏选择器 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| InRangeOfView(View view , MotionEventev) | view ：用来表示ui容器，容器的可触范围。  ev：屏幕触碰操作 | 用来确定触碰操作是否在给定的view范围内 | 先确定容器的范围location，在用ev的getX与getY操作判断操作点是否位于location内 |
| InitResource() | 无 | 初始化UI界面的各种元素和资源 | 通过给定id，R.id类的内容来赋值 |
| onBackPressed() | 无 | 确保用户按下返回按钮时，未保存的便签内容可以得到保存 | 先检查用户是否处于其他设置状态，否则则直接调用saveNote()的方法保存内容。同时调用super.onBackPressed方法来执行默认返回操作。 |
| clearSettingState() | 无 | 如果需要时，保证颜色选择器和字体大小选择器在不使用的时候是隐藏的 | 用判断结构，如果当前选择器为可见，则设置为不可见。 |
| onClick(View v) | v：菜单选择器内对应向的id值 | 用来实现用户点击与界面中视图的交互，根据不同的点击内容，执行不同的点击结果。例如修改字体的大小和修改背景颜色 | View.OnclickListener的接口来监听点击事件，在onClick方法中实现对应操作 |
| onOPtionsItemSelected(MenuItem item) | Item;菜单点击事件 | 根据不同的菜单点击事件，使菜单可见，或执行当前菜单内对应的功能 | 通过获得item的事件id，来匹配不同的菜单内的功能，然后再执行对应功能的内容。 |
| InitNoteScreen() | 无 | 加载设置的字体大小样式等样式个性化内容到屏幕上。 | 调用mNoteEditor中的setTextAppearance的方法来实现。 |

* + 1. **模型层相关的类**

**10.3.2.1、类名：ResourceParser**

1. **类的作用：**保存字体大小的匹配信息，和完成设置对应字体大小的样式
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TextAppearanceResources | 存储字体样式的信息，提供获取样式的返回值的方法 |
| TEXT\_SMALL | 设置对应字体大小样式对应的int值 |
| TEXT\_MEDIUM |
| TEXT\_LARGE |
| TEXT\_SUPER |
| BG\_DEFAULT\_FONT\_SIZE | 设置默认字体大小样式的初始值为MEDIUM |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getTexAppearanceResourse(int id) | Id:标识确定字体样式的大小。 | 用来确保id的值在正确的范围内，当大于正确范围时将字体大小重新设为默认值。如果符合的话返回当前字体id对应的字体大小设置样式。 | 判断id是否在正确的范围，在做出修改或者不修改为默认值的选择。 |
| getResourcesSize() | 无 | 返回字体样式范围的最大值 | 用return语句实现。 |

* + 1. **数据层相关的类**

**10.3.3.1、类名：NoteEditActivity**

1. **类的作用：**保存用户的配置信息，如当前便签的修改字体大小和背景颜色的简单的配置信息。
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mSharedPrefs | 用来保存配置信息。 |
| PREFERENCE\_FONT\_SIZE | 用做共享首选项中的键标识配置信息。 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onClick | v | 用来实现用户点击与界面中视图的交互，根据不同的点击内容，执行不同的点击结果。例如修改字体的大小和修改背景颜色。同时保存配置信息。 | View.OnclickListener的接口来监听点击事件，在onClick方法中实现对应操作 |
| onOPtionsItemSelected | Item;菜单点击事件 | 根据不同的菜单点击事件，使菜单可见，或执行当前菜单内对应的功能 | 通过获得item的事件id，来匹配不同的菜单内的功能，然后再执行对应功能的内容。 |

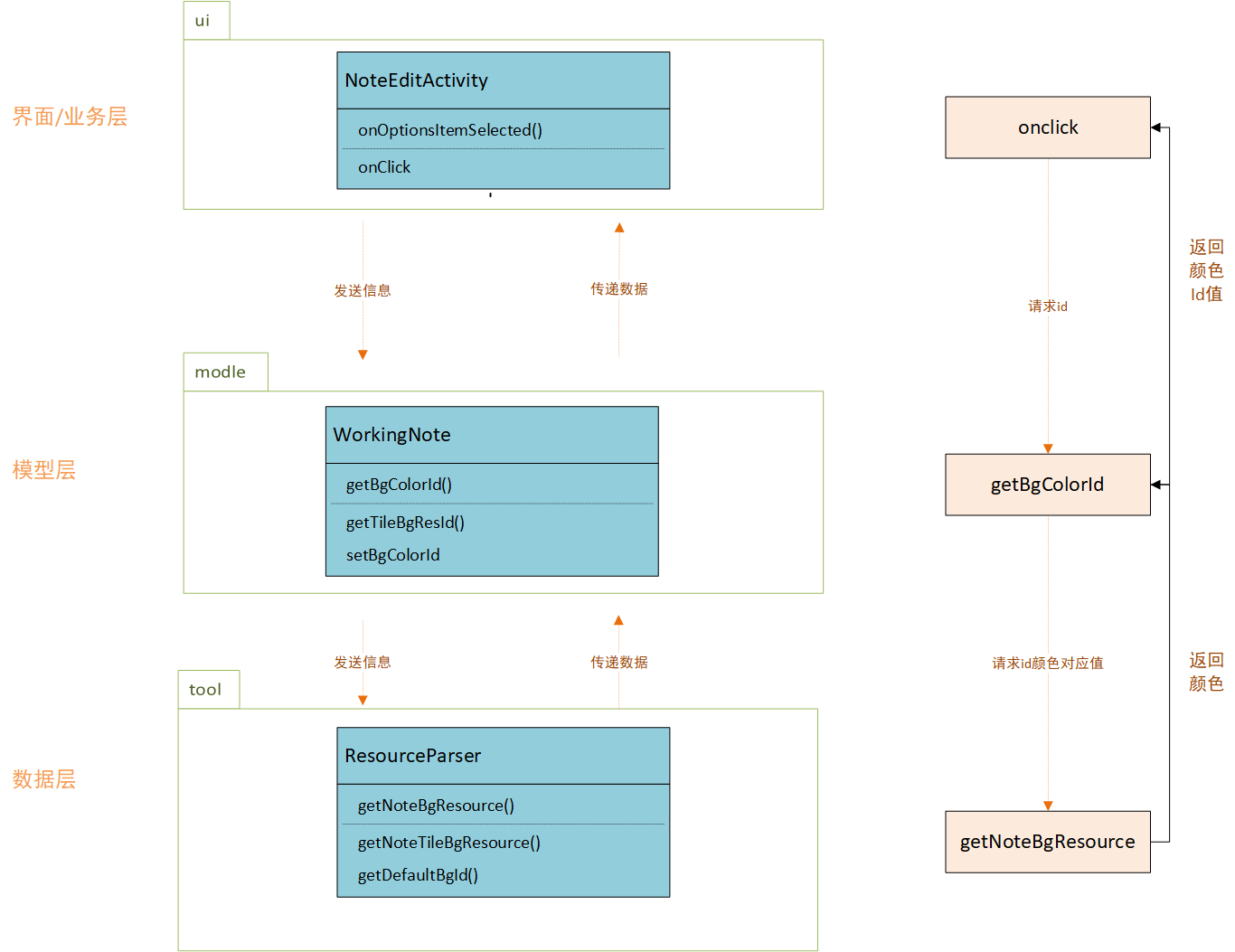
1. **功能十一：修改背景颜色**
   1. **功能介绍**

**功能概述：**修改便签的背景颜色设置

**操作步骤：**在主界面的右上角，有颜色的缩略图，点击然后在颜色菜单中，选择需要的颜色，完成修改。

* 1. **业务请求与响应过程**

功能是由UI包中类NoteEditActivity中的onclick()方法获得的背景颜色的请求id值从modle包中的WorkingNote类中getBgColorId()方法获得返回值，在向tool包中的ResourceParser类请求颜色返回值，并在onClick()中方法完成设置。



* 1. **类的作用**
     1. **界面层/业务层相关的类**

**11.3.1.1、类名：NoteEditActivity**

1. **类的作用：**

该类是显示背景颜色的选择器，和完成背景颜色选择修改的业务。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 用于保存一组视图的引用，有修改日期文本，背景颜色，提醒图标，和提醒日期文本的视图。 |
| sBgSelectorBtnsMap | 保存背景颜色选择器中选项和对应颜色的映射关系 |
| sBgSelectorSelectionMap | 保存颜色id值与对应颜的映射关系 |
| mNoteBgColorSelectot | 背景颜色的选择器 |
| mWorkingNote | 当前操作的便签项 |
| mNoteEditorPanel | 操作背景颜色的修改 |
| mHeadViewPanel | 操作标题颜色的修改 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| dispactchTouchEvent() | ev:屏幕触碰的操作 | 使得颜色选择器和字体大小的选择器在屏幕上可见。同时用于在其他特殊操作下隐藏选择器 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| InRangeOfView(View view, MotionEventev) | view ：用来表示ui容器，容器的可触范围。  ev：屏幕触碰操作 | 用来确定触碰操作是否在给定的view范围内 | 先确定容器的范围location，在用ev的getX与getY操作判断操作点是否位于location内 |
| InitResource() | 无 | 初始化UI界面的各种元素和资源 | 通过给定id，R.id类的内容来赋值 |
| onBackPressed() | 无 | 确保用户按下返回按钮时，未保存的便签内容可以得到保存 | 先检查用户是否处于其他设置状态，否则则直接调用saveNote()的方法保存内容。同时调用super.onBackPressed方法来执行默认返回操作。 |
| clearSettingState() | 无 | 如果需要时，保证颜色选择器和字体大小选择器在不使用的时候是隐藏的 | 用判断结构，如果当前选择器为可见，则设置为不可见。 |
| onClick(View v) | v:菜单选择器内对应向的id值 | 用来实现用户点击与界面中视图的交互，根据不同的点击内容，执行不同的点击结果。例如修改字体的大小和修改背景颜色 | View.OnclickListener的接口来监听点击事件，在onClick方法中实现对应操作 |
| onOptionsItemSelected(MenuI item) | Item;菜单点击事件 | 根据不同的菜单点击事件，使菜单可见，或执行当前菜单内对应的功能 | 通过获得item的事件id，来匹配不同的菜单内的功能，然后再执行对应功能的内容。 |
| InitNoteScreen() | 无 | 加载设置的字体大小样式等样式个性化内容到屏幕上 | 调用mNoteEditor中的setTextAppearance的方法来实现。 |
| onBackgroundColorChanged() | 无 | 修改当前的背景颜色 | 通过id值来修改对应的背景颜色 |

* + 1. **模型层相关的类**

**11.3.2.1、类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 标记当前便签状态 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |
| NoteSettingChangedListener | 定义便签设置改变监听器的接口 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getTitleBgResId() | 无 | 返回当前的标题颜色的ID值 | 通过背景颜色ID值来调用NoteBgResource内的方法来匹配返回标题的ID值 |

* + 1. **数据层相关的类**

**11.3.3.1、类名：ResourceParser**

1. **类的作用：**

存储背景颜色的默认值，和修改背景颜色所对应的数据。同时提供便签不同位置的背景颜色ID值

1. **类的精读：**

类的成员

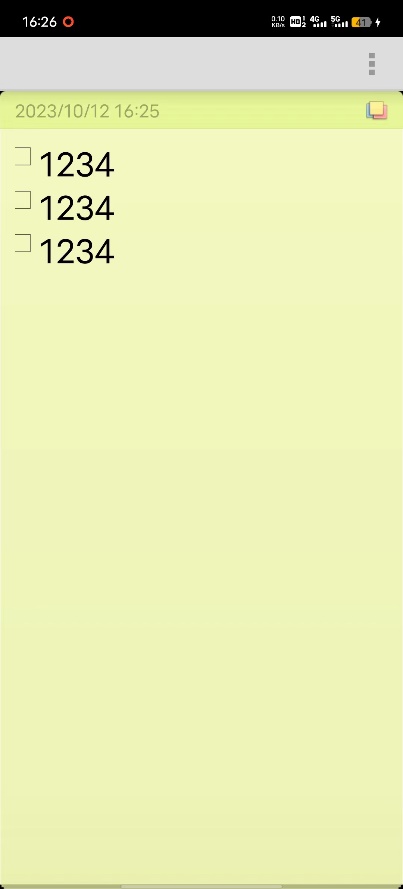
|  |  |
| --- | --- |
| 成员 | 说明 |
| NoteBgResources | 存储便签背景颜色和标题颜色的对应值 |
| BG\_DEFAULT\_COLOR | 存储默认的颜色背景值 |
| NoteItemBgResources | 用于管理小米便签应用中笔记列表项的不同背景资源 |
| Yellow | 设置颜色对应的int值 |
| Blue |
| White |
| Green |
| Red |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getNoteBgResource(int id) | Id:用来表示当前颜色的值 | 返回对应id的背景颜色 | 数组内匹配id值返回对应的颜色 |
| getNoteTileBgResource(int id) | Id:用来表示当前颜色的值 | 返回对应id的标题颜色 | 数组内匹配id值返回对应的颜色 |
| getDefaultBgId(Context contex) | context:程序的上下文 | 用于获取默认颜色的id值 | 获取用户的默认共享首选项来确定用户是否启用了自定义背景颜色设置，如果启用则返回一个随机的颜色值。否则则返回默认背景颜色值 |

1. **功能十二：清单模式**
   1. **功能介绍**

**功能概述：**进入清单模式，每一列变成待完成事项

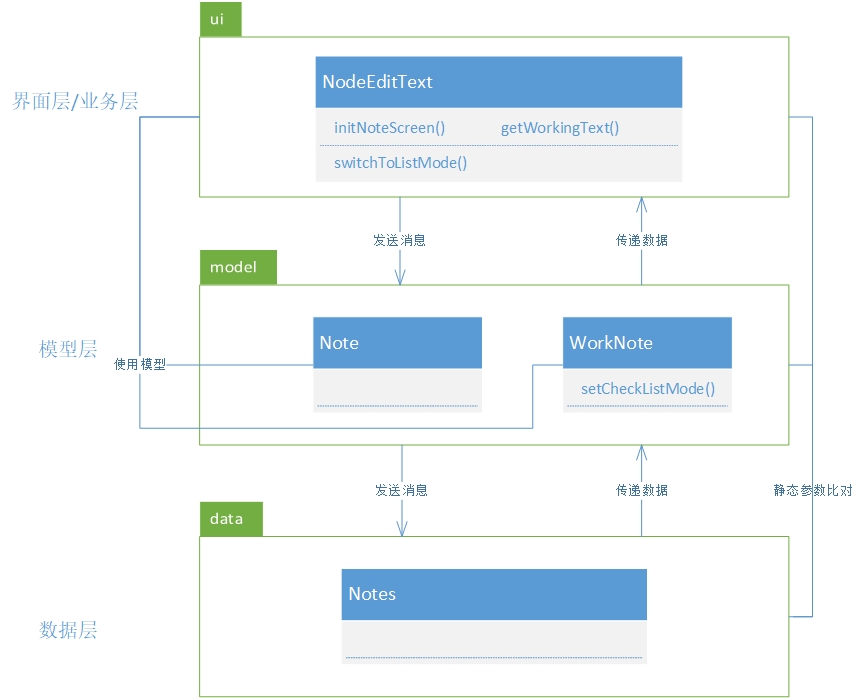
**操作步骤：**便签编辑页面右上角进入菜单，选择“进入清单模式”，选择后，在便签的每一行（段内部的自动换行除外）行首出现一个勾选框，用于当前便签下标记某事项的完成情况。如果该事项已完成，则用户在勾选框中轻触，此时勾选框中将出现一个对勾，框后的陈述文字被添加了中央删除线。（外侧文件夹并不能显示事项的完成状态，可优化）同理，在不需要清单模式的时候，便签编辑页面的右上角选择“退出清单模式”即可。

* 1. **业务请求与响应过程**

清单模式功能（下简称“功能”）的界面层由包ui中NoteEditText类中多个方法共同实现。首先在初始化界面的时候调用initNoteScreen()方法进行初始化在这里需要进行一次清单模式的判断，如果是则将传入内容修改成清单的格式，调用switchToListMode(String text)方法；判断清单项是否完成，调用getWorkingText()展示所有清单项的完成状态，所有状态的改变都会向下传递至模型层进行相关存储的修改。

功能由于属于前端界面的显示变化，也就是说界面层和业务层重合，界面层直接收集模型层向上传递的数据然后进行显示。

功能的模型层由包model中WorkingNote类中setCheckListMode(int mode)实现，对模式进行判断。

****功能的数据层由包data中Notes类实现，数据层存储的一些常量进行比对，进行模式识别。

* 1. **类的作用**
     1. **界面层相关的类**

**12.3.1.1、类名：NoteEditText**

1. **类的作用：**

该类是便签编辑操作的集成，对便签界面进行初始化，提供设置提醒时间弹窗，对便签界面的菜单按钮时间进行定义，对便签的删改操作进行定义，实现了清单模式变化和清单项完成文本变化等操作

1. **类的精读：**

NoteEditText类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 头部视图类 |
| sBgSelectorBtnsMap | 一个映射，将背景选择按钮的资源ID映射到背景颜色的整数值 |
| sBgSelectorSelectionMap | 一个映射，将背景颜色的整数值映射到选中的背景选择按钮的资源ID |
| sFontSizeBtnsMap | 一个映射，将字体大小按钮的资源ID映射到字体大小的整数值 |
| sFontSelectorSelectionMap | 将字体大小的整数值映射到选中的字体选择按钮的资源ID |
| TAG | 一个字符串常量，标签 |
| mNoteHeaderHolder | 用于保存便签的头部视图元素 |
| mHeadViewPanel | 便签编辑器中的头部视图容器 |
| mNoteBgColorSelector | 用于选择便签的背景颜色 |
| mFontSizeSelector | 用于选择文本的字体大小 |
| mNoteEditor | 一个EditText控件，用于编辑便签的文本内容 |
| mNoteEditorPanel | 便签编辑器的容器 |
| **mWorkingNote** | **包含有关便签的状态信息** |
| mSharedPrefs | 一个SharedPreferences对象，用于存储和管理应用程序的偏好设置 |
| mFontSizeId | 存储所选的字体大小的整数值 |
| PREFERENCE\_FONT\_SIZE | 字体大小偏好设置的键名 |
| SHORTCUT\_ICON\_TITLE\_MAX\_LEN | 快捷图标的标题最大长度 |
| **TAG\_CHECKED** | **表示待办事项列表中的某任务已完成** |
| **TAG\_UNCHECKED** | **表示待办事项列表中的某任务未完成** |
| mEditTextList | 包含编辑文本的列表 |
| mUserQuery | 保存用户的查询文本 |
| mPattern | 用于在文本中进行匹配和搜索 |

NoteEditText类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState用于在重建活动时存储先前的状态信息 | Android Activity的生命周期方法之一，用于创建活动时初始化界面和资源 | 首先调用父类的onCreate方法，然后加载与活动相关的布局文件（R.layout.note\_edit）。接着，它检查是否存在先前保存的状态信息（savedInstanceState），并根据情况初始化活动状态或结束活动 |
| onRestoreInstanceState(Bundle savedInstanceState) | savedInstanceState包含之前保存的状态信息 | 用于在活动被系统销毁后重新创建时，恢复之前的状态信息 | 如果存在Intent.EXTRA\_UID的键在savedInstanceState中，它将创建一个新的Intent，将之前保存的UID添加到其中，并尝试使用该Intent初始化活动状态 |
| initActivityState(Intent intent) | intent包含用于初始化活动的信息 | 用于初始化活动的状态，包括加载或创建便签 | 根据传入的Intent的不同操作（ACTION） |
| onResume() | / | 用于在活动重新进入前台时进行初始化和更新界面 | 调用initNoteScreen()方法来初始化便签编辑界面 |
| initNoteScreen() | / | 初始化便签编辑界面的各个部分，包括文本样式、文本内容、背景颜色等 | 根据便签的内容和属性，设置文本样式、内容，以及相关的背景颜色。如果便签具有提醒信息，它还会在界面上显示提醒的日期和图标 |
| showAlertHeader() | / | 在界面上显示或隐藏便签的提醒信息 | 根据便签是否具有提醒信息，它决定显示或隐藏提醒日期和图标，并根据提醒的时间情况显示相应的文本（已过期或相对时间） |
| onNewIntent(Intent intent) | intent包含新的Intent信息 | 用于在活动已经存在的情况下处理新的Intent | 首先调用父类的onNewIntent方法，然后使用传入的Intent重新初始化活动状态 |
| onSaveInstanceState(Bundle outState) | outState存储活动状态信息 | 保存活动的状态信息，以便在需要时进行恢复 | 如果便签还没有存在于数据库中（没有分配ID），则首先尝试保存便签。然后，将便签的ID放入outState中，以便在重建活动时使用 |
| dispatchTouchEvent(MotionEvent ev) | MotionEvent ev包含有关触摸事件的信息 | 处理触摸事件分发，用于检测用户的点击位置以隐藏背景颜色选择器和字体大小选择器 | 检查背景颜色选择器和字体大小选择器是否可见，如果不在它们的范围内点击，则隐藏它们。最后，调用父类的dispatchTouchEvent方法继续事件分发 |
| inRangeOfView(View view, MotionEvent ev) | view要检查的视图；ev触摸事件信息 | 检查触摸事件是否在指定视图内 | 获取视图的位置信息，然后检查触摸事件的坐标是否在视图的范围内。如果在范围内，返回true，否则返回false |
| initResources() |  | 初始化活动所需的资源和视图 | 初始化了活动的各个部分，包括头部视图、便签编辑器、背景颜色选择器、字体大小选择器等 |
| onPause() | / | 活动进入后台时保存便签数据 | 首先调用父类的onPause方法，然后尝试保存便签数据，并清除设置状态 |
| updateWidget() | / | 更新小部件的显示 | 根据活动中的便签的小部件类型，创建一个对应的Intent，并发送广播以更新小部件的显示 |
| onClick(View v) | v点击的视图对象 | 处理点击事件 | 根据点击的视图的ID，执行不同的操作。例如，当点击背景颜色选择按钮时，显示颜色选择器；当点击背景颜色选择器中的颜色时，更改便签的背景颜色等 |
| onBackPressed() | / | 处理用户按下返回按钮时的行为 | 首先调用clearSettingState方法，以确保任何设置状态都被清除。然后尝试保存便签，并调用父类的onBackPressed方法以返回 |
| clearSettingState() | / | 清除设置状态 | 检查颜色选择器和字体大小选择器是否可见，如果可见，则隐藏它们。如果成功隐藏，返回true，否则返回false |
| onBackgroundColorChanged() | / | 处理背景颜色改变时的操作 | 根据便签的新背景颜色，更新界面元素的颜色 |
| onPrepareOptionsMenu(Menu menu) | menu菜单对象 | 准备选项菜单 | 清除任何现有的菜单项，然后根据便签的状态和属性，动态生成适当的菜单项，包括切换列表模式、设置提醒等操作 |
| **onOptionsItemSelected(MenuItem item)** | **item表示用户选择的菜单项** | **处理用户在选项菜单中选择的操作** | **根据用户的不同选择进行查找** |
| setReminder() | / | 用于设置提醒时间 | 创建一个日期时间选择对话框，允许用户选择提醒时间。选择时间后，将提醒时间设置给当前便签 |
| sendTo(Context context, String info) | context上下文对象；info要分享的便签内容 | 将便签内容分享到支持Intent.ACTION\_SEND动作和text/plain类型的应用程序 | 创建一个Intent，设置EXTRA\_TEXT为便签内容，类型为text/plain，然后启动适用的应用程序以分享内容 |
| createNewNote() | / | 创建一个新的便签 | 首先保存当前正在编辑的便签，然后结束当前活动。接下来，创建一个新的NoteEditActivity意图，以新建便签的方式启动该活动 |
| deleteCurrentNote() | / | 删除当前便签 | 如果当前便签已存在于数据库中，将其ID添加到HashSet中，并根据是否启用同步模式执行不同的删除操作。如果不是同步模式，调用DataUtils. batchDeleteNotes来删除便签；如果是同步模式，将便签移动到回收站。最后，将当前便签标记为已删除 |
| isSyncMode() | / | 检查是否处于同步模式 | 检查应用的同步账户名称是否为空，如果不为空则表示处于同步模式，返回true，否则返回false |
| onClockAlertChanged(long date, boolean set) | date提醒时间；set是否设置提醒） | 处理提醒时间的更改 | 如果当前便签未存在于数据库中，首先保存便签。然后创建一个用于提醒的Intent，并根据set参数设置或取消提醒。最后，根据提醒是否设置，调用showAlert Header方法以更新提醒状态的显示 |
| updateWidget() | / | 更新小部件的显示 | 根据便签的小部件类型，创建一个对应的Intent，然后发送广播以更新小部件的显示 |
| onEditTextDelete(int index, String text) | index文本编辑框索引；String text要删除的文本 | 处理文本编辑框中的文本删除操作 | 从文本编辑框列表中删除指定索引的文本编辑框，同时调整其他文本编辑框的索引，然后将被删除的文本追加到前一个文本编辑框的末尾 |
| onEditTextEnter(int index, String text) | index文本编辑框索引；text要插入的文本 | 处理文本编辑框中的文本插入操作 | 创建一个新的文本编辑框，将其插入到指定索引处，同时调整其他文本编辑框的索引 |
| **switchToListMode(String text)** | **text要分割为列表的文本** | **切换为列表模式，将文本分成多个项目** | **将文本按行分割，创建文本编辑框列表，并将每一行文本作为列表项目添加到列表中。最后，隐藏原始的文本编辑器，显示文本编辑框列表** |
| getHighlightQueryResult(String fullText, String userQuery) | fullText完整文本；userQuery用户查询 | 创建一个具有高亮显示查询结果的Spannable文本 | 首先创建一个SpannableString，然后使用正则表达式将用户查询高亮显示在完整文本中，最后返回带有高亮的Spannable文本 |
| **getListItem(String item, int index)** | **item列表项文本；index列表项索引** | **创建一个包含列表项的视图** | **从布局文件R.layout.note\_ed it\_list\_item创建一个新视图，获取文本编辑框和复选框，设置文本样式、文本内容和复选框的监听器。根据列表项是否以特定标记（TAG\_CHECKED 或 TAG\_UNCHECKED）开头，设置复选框的选中状态和文本内容，并应用高亮查询结果。返回创建的视图** |
| onTextChange(int index, boolean hasText) | index文本编辑框索引；hasText文本编辑框是否包含文本 | 处理文本编辑框中的文本变化事件 | 根据文本编辑框是否包含文本，设置相应文本编辑框的复选框的可见性 |
| **onCheckListModeChanged(int oldMode, int newMode)** | **oldMode原模式；newMode新模式** | **处理待办事项列表模式的切换** | **如果新模式是待办事项列表模式，将编辑框内容切分为多个项目，然后显示列表项** |
| **getWorkingText()** | **/** | **获取清单模式下任务状态** | **判断是否处于清单模式，对于每一项任务是否执行完成进行状态区分并呈现不同状态** |
| saveNote() | / | 保存便签 | 获取工作文本，然后调用saveNote方法保存便签。如果保存成功，设置结果码为RESULT\_OK表示创建或编辑便签 |
| sendToDesktop() | / | 将便签发送到桌面 | 如果便签不存在于数据库中，首先保存便签。然后，创建一个Intent，用于创建快捷方式。将快捷方式信息设置为便签的内容和图标，然后发送广播以创建快捷方式。如果便签没有ID（新建便签且内容为空），显示错误消息 |
| makeShortcutIconTitle(String content) | content便签内容 | 生成快捷方式图标的标题 | 移除内容中的特定标记，然后将标题限制在最大长度内 |
| showToast(int resId) | resId消息资源ID | 显示短时长的Toast消息 | 调用重载函数showToast(int resId, int duration) |
| showToast(int resId, int duration) | resId消息资源ID；duration Toast显示时长 | 显示自定义时长的Toast消息 | / |

### 12.3.2. 模型层相关的类

**12.3.2.1、类名：**Note

1. **类的作用：**

管便签（notes）的数据，包括创建、编辑、同步等操作。它维护便签的元数据和文本数据以及通话数据，并将这些数据同步到Android应用的数据库中

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNoteDiffValues | 一个ContentValues对象，用于存储便签的元数据的差异 |
| mNoteData | 一个NoteData对象，用于管理便签的文本数据和通话数据 |
| TAG | 一个用于日志记录的常量字符串 |

Note类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getNewNoteId(Context context, long folderId) | context：Android应用程序的上下文；folderId表示便签所属的文件夹的ID | 创建一个新的便签ID并插入数据库中，返回新便签的ID | 创建一个ContentValues对象values，用于存储便签的元数据，初始化values，从URI中提取新便签的ID并返回 |
| Note() | / | 构造函数，初始化mNoteDiffValues和mNoteData对象 | / |
| setNoteValue(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置便签元数据的键值对，用于标记元数据的本地修改 | 将键值对存储到mNoteDiffValues，并更新LOCAL\_MODIFIED和MODIFIED\_DATE的值 |
| setTextData(String key, String value) | String key：文本数据的键名；String value：文本数据的键值 | 设置文本数据的键值对 | 将键值对存储到mNoteData的文本数据中 |
| setTextDataId(long id) | id：文本数据的ID | 设置文本数据的ID | 将ID存储到mNoteData中 |
| getTextDataId() | / | 获取文本数据的ID | 返回mNoteData中的文本数据ID |
| setCallDataId(long id) | id：通话数据的ID | 设置通话数据的ID | 将ID存储到mNoteData中 |
| setCallData(String key, String value) | key：通话数据的键名；value：通话数据的键值 | 设置通话数据的键值对 | 将键值对存储到mNoteData的通话数据中 |
| isLocalModified() | / | 判断便签是否在本地有修改 | 检查mNoteDiffValues和mNoteData是否包含本地修改的数据，如果有则返回true，否则返回false |
| syncNote(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 同步便签到数据库中 | 如果没有本地修改，直接返回true表示同步成功；如果有本地修改，首先尝试更新便签的元数据 |
| NoteData.isLocalModified() | / | 判断文本数据或通话数据是否在本地有修改 | / |
| NoteData.setTextDataId(long id) | id表示传入的id | 设置文本数据的ID | / |
| NoteData.setCallDataId(long id) | id表示传入的id | 设置通话数据的ID | / |
| NoteData.setCallData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置通话数据的键值对 | / |
| NoteData.setTextData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置文本数据的键值对 | / |
| pushIntoContentResolver(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 将文本数据或通话数据同步到数据库中 | 根据是否存在本地修改，构建数据库操作，然后执行同步操作 |

* + - 1. **类名：**WorkingNote

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| WorkingNote(Context context, long folderId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID | 构造函数，用于创建一个新的工作便签 | / |
| WorkingNote(Context context, long noteId, long folderId) | context：Android应用程序的上下文；noteId：要加载的便签的ID；folderId：便签所属的文件夹ID | 构造函数，用于加载已有的工作便签 | / |
| loadNote() | / | 用于加载便签的元数据信息 | 创建游标，游标移动到制定，获取属性值并赋给对应便签属性 |
| loadNoteData() | / | 用于加载便签的内容数据 | 创建游标，游标移动到制定，获取文本内容并赋给 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID；widgetId：小部件的ID；widgetType：小部件类型；defaultBgColorId：默认的背景颜色ID | 创建一个新的空白便签 | 创建一个新的工作便签，并设置其属性，包括背景颜色、小部件ID等 |
| load(Context context, long id) | context：Android应用程序的上下文；id：要加载的便签的ID | 加载现有的工作便签 | 根据传入的条件查找便签，并加载 |
| saveNote() | / | 保存便签的方法，包括新建便签和已有便签的保存 | 判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库 |
| existInDatabase() | / | 判断便签是否已存在于数据库中 | 判断NoteId是否存在 |
| isWorthSaving() | / | 判断便签是否值得保存 | 考虑是否已删除、是否存在文本内容和是否有本地修改 |
| setOnSettingStatusChangedListener(NoteSettingChangedListener l) | 便签属性改变监听器 | 设置便签设置状态变化的监听器 | / |
| setAlertDate(long date, boolean set) | date：提醒日期的时间戳；set：是否设置提醒 | 设置提醒日期 | 如果提醒日期不同于当前日期，更新提醒日期并通知监听器 |
| markDeleted(boolean mark) | mark：是否标记为已删除 | 标记便签是否已删除 | 标记便签是否已删除，同时通知监听器 |
| setBgColorId(int id) | id：背景颜色的ID | 设置背景颜色的ID | 如果背景颜色ID不同于当前ID，更新背景颜色ID并通知监听器 |
| setCheckListMode(int mode) | mode：模式 | 设置便签的清单模式 | 如果模式不同于当前模式，更新模式并通知监听器 |
| setWidgetType(int type) | type：小部件类型 | 设置小部件的类型 | 确定widget类型是否变化，变化了就更新widget参数 |
| setWidgetId(int id) | id：小部件的ID | 设置小部件的ID | 确定widget的id是否变化，更新widget |
| setWorkingText(String text) | text：文本内容 | 设置便签的文本内容 | 如果输入内容和原文本不一致则进行内容更新 |
| convertToCallNote(String phoneNumber, long callDate) | phoneNumber：通话记录的电话号码；callDate：通话日期的时间戳 | 将便签转换为通话记录便签 | 将传入的电话号码设置为电话数据 |
| hasClockAlert() | / | 判断便签是否有提醒设置 | / |
| getContent() | / | 获取便签的文本内容 | / |
| getAlertDate() | / | 获取便签的提醒日期 | / |
| getModifiedDate() | / | 获取便签的最后修改日期 | / |
| getBgColorResId() | / | 获取背景颜色的资源ID | / |
| getBgColorId() | / | 获取背景颜色的ID | / |
| getTitleBgResId() | / | 获取标题栏背景颜色的资源ID | / |
| getCheckListMode() | / | 获取便签的清单模式 | / |
| getNoteId() | / | 获取便签的ID | / |
| getFolderId() | / | 获取便签所属的文件夹ID | / |
| getWidgetId() | / | 获取小部件的ID | / |
| getWidgetType() | / | 获取小部件的类型 | / |
| NoteSettingChangedListener | / | 内部接口，定义了便签设置状态变化的监听器方法 | / |

* + 1. **数据层相关的类**

**12.3.3.1、类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

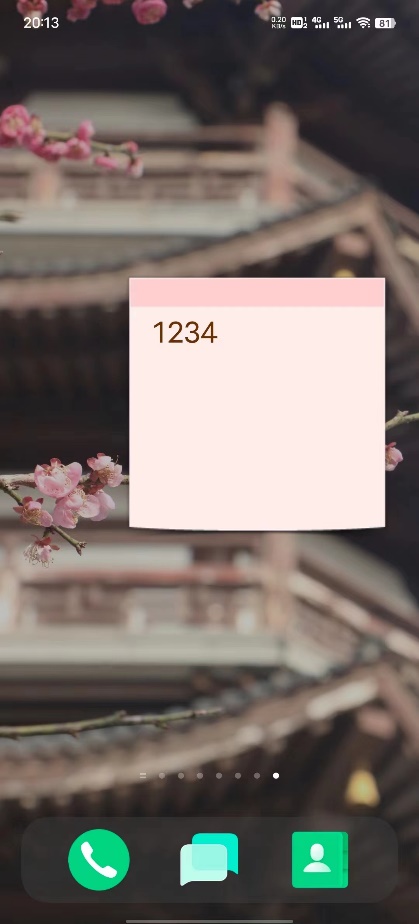
Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与便签关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表便签的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表便签是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的便签数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本便签的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本便签的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本便签的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本便签的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

1. **功能十三：发送到桌面**
   1. **功能介绍**

**功能概述：**将编辑好的便签发送到桌面上

**操作步骤：**在Android操作系统的桌面创建小米便签的小部件后，在编写便签完成后，编辑页面菜单栏选择“发送到桌面”，便可在便签小部件上显示当前便签的内容

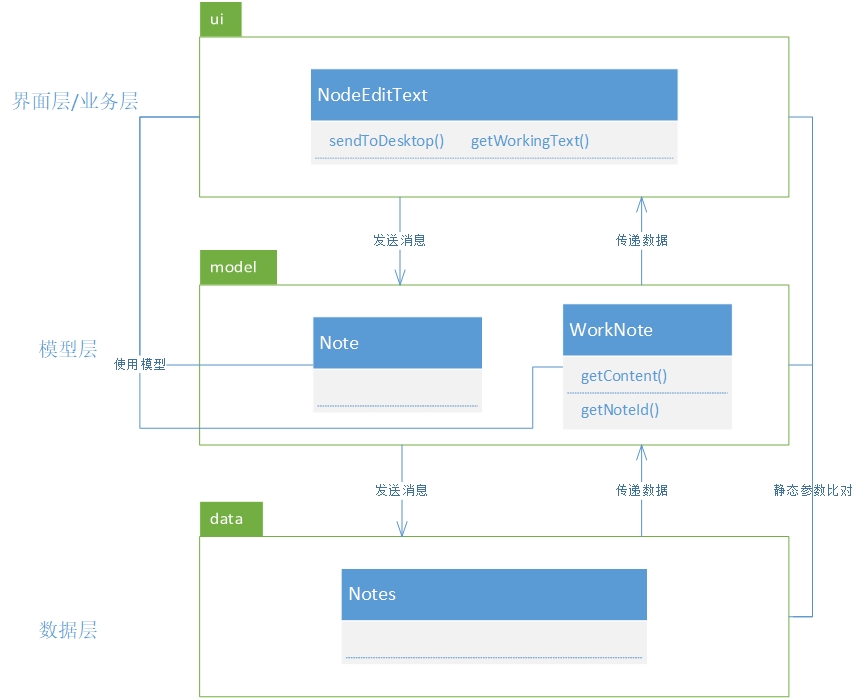
* 1. **业务请求与响应过程**

发送到桌面功能（下简称“功能”）的界面层由包ui中NodeEditActivity类的sendToDesktop()实现。方法检查当前编辑的笔记是否已存在于数据库中以确保它在发送到桌面之前已经保存，接着检查当前编辑的笔记是否具有有效的笔记ID（Note ID），然后创建一个新的意图（Intent）对象，用于在桌面上创建一个快捷方式（Shortcut），设置快捷方式的意图（Intent），将它指向 NoteEditActivity 类，并将操作设置为 Intent.ACTION\_VIEW以指示打开笔记编辑活动，将笔记的唯一标识（Note ID）添加到快捷方式的意图中以便在打开笔记时可以检索到正确的笔记。设置快捷方式的名称、设置快捷方式的图标资源。

功能直接在界面层创建一个快捷方式，并通过Intent操作设置直接发送到桌面，将下层模型层获取的文本内容注入快捷方式中，和业务层重合

功能的模型层由包model的类WorkingNotes中getContent()、getNoteId()方法实现，获取当前便签的文本内容并注入小部件中。

功能的数据层由包data的类Notes实现，调用Notes中的静态参数进行数据的获取。

****

* 1. **类的作用**
     1. **界面层相关的类**

**13.3.1.1、类名：NoteEditText**

1. **类的作用：**

该类是便签编辑操作的集成，对便签界面进行初始化，提供设置提醒时间弹窗，对便签界面的菜单按钮时间进行定义，对便签的删改操作进行定义，实现了清单模式变化和清单项完成文本变化等操作

1. **类的精读：(仅列举相关)**

NoteEditText类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| **mWorkingNote** | **包含有关便签的状态信息** |

NoteEditText类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| **saveNote()** | **/** | **保存便签** | **获取工作文本，然后调用saveNote方法保存便签。如果保存成功，设置结果码为RESULT\_OK表示创建或编辑便签** |
| **sendToDesktop()** | **/** | **将便签发送到桌面** | **如果便签不存在于数据库中，首先保存便签。然后，创建一个Intent，用于创建快捷方式。将快捷方式信息设置为便签的内容和图标，然后发送广播以创建快捷方式。如果便签没有ID（新建便签且内容为空），显示错误消息** |

* + 1. **模型层相关的类**

**13.3.2.1类名：Note**

1. **类的作用：**

管便签（notes）的数据，包括创建、编辑、同步等操作。它维护便签的元数据和文本数据以及通话数据，并将这些数据同步到Android应用的数据库中

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNoteDiffValues | 一个ContentValues对象，用于存储便签的元数据的差异 |
| mNoteData | 一个NoteData对象，用于管理便签的文本数据和通话数据 |
| TAG | 一个用于日志记录的常量字符串 |

Note类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getNewNoteId(Context context, long folderId) | context：Android应用程序的上下文；folderId表示便签所属的文件夹的ID | 创建一个新的便签ID并插入数据库中，返回新便签的ID | 创建一个ContentValues对象values，用于存储便签的元数据，初始化values，从URI中提取新便签的ID并返回 |
| Note() | / | 构造函数，初始化mNoteDiffValues和mNoteData对象 | / |
| setNoteValue(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置便签元数据的键值对，用于标记元数据的本地修改 | 将键值对存储到mNoteDiffValues，并更新LOCAL\_MODIFIED和MODIFIED\_DATE的值 |
| setTextData(String key, String value) | String key：文本数据的键名；String value：文本数据的键值 | 设置文本数据的键值对 | 将键值对存储到mNoteData的文本数据中 |
| setTextDataId(long id) | id：文本数据的ID | 设置文本数据的ID | 将ID存储到mNoteData中 |
| getTextDataId() | / | 获取文本数据的ID | 返回mNoteData中的文本数据ID |
| setCallDataId(long id) | id：通话数据的ID | 设置通话数据的ID | 将ID存储到mNoteData中 |
| setCallData(String key, String value) | key：通话数据的键名；value：通话数据的键值 | 设置通话数据的键值对 | 将键值对存储到mNoteData的通话数据中 |
| isLocalModified() | / | 判断便签是否在本地有修改 | 检查mNoteDiffValues和mNoteData是否包含本地修改的数据，如果有则返回true，否则返回false |
| syncNote(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 同步便签到数据库中 | 如果没有本地修改，直接返回true表示同步成功；如果有本地修改，首先尝试更新便签的元数据 |
| NoteData.isLocalModified() | / | 判断文本数据或通话数据是否在本地有修改 | / |
| NoteData.setTextDataId(long id) | id表示传入的id | 设置文本数据的ID | / |
| NoteData.setCallDataId(long id) | id表示传入的id | 设置通话数据的ID | / |
| NoteData.setCallData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置通话数据的键值对 | / |
| NoteData.setTextData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置文本数据的键值对 | / |
| pushIntoContentResolver(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 将文本数据或通话数据同步到数据库中 | 根据是否存在本地修改，构建数据库操作，然后执行同步操作 |

**13.3.2.2. 类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| **saveNote()** | **/** | **保存便签的方法，包括新建便签和已有便签的保存** | **判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库** |
| **getContent()** | **/** | **获取便签的文本内容** | **/** |

* + 1. **数据层相关的类**

**13.3.3.1、类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与便签关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表便签的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表便签是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的便签数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本便签的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本便签的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本便签的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本便签的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

1. **功能十四：自动保存便签**
   1. **功能介绍**

**功能概述：**在输入有效内容后，便签能够自动保存

**操作步骤：**在主列表界面新建一个便签，在便签输入有效内容，然后退出后，能够看到之前填写的便签已经被保存**。**

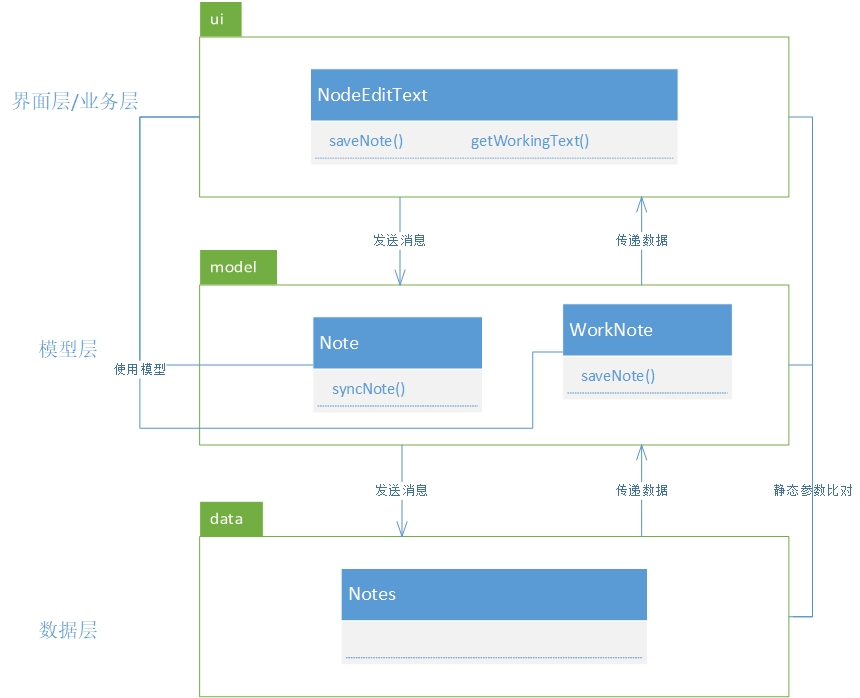
* 1. **业务请求与响应过程**

自动保存便签功能（下简称“功能”）的界面层由包ui中类NoteEditActivity的saveNote()方法实现，首先获取当前便签文本状态，然后调用类WorkingNote的saveNote()方法保存笔记。

功能直接调用模型层的类WorkingNote，因此可以知道功能的界面层和业务层重合。

功能的模型层由包model中WorkingNote类的saveNote()方法实现，首先判断是否值得存储，调用类Note中的syncNote(Context context, long noteId)方法同步笔记。

功能的数据层由包data中Notes类实现

****

* 1. **类的作用**
     1. **界面层相关的类**

**14.3.1.1、类名：NoteEditText**

1. **类的作用：**

该类是便签编辑操作的集成，对便签界面进行初始化，提供设置提醒时间弹窗，对便签界面的菜单按钮时间进行定义，对便签的删改操作进行定义，实现了清单模式变化和清单项完成文本变化等操作

1. **类的精读：**

NoteEditText类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| **mWorkingNote** | **包含有关便签的状态信息** |

NoteEditText类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| **getWorkingText()** | **/** | **获取清单模式下任务状态** | **判断是否处于清单模式，对于每一项任务是否执行完成进行状态区分并呈现不同状态** |
| **saveNote()** | **/** | **保存便签** | **获取工作文本，然后调用saveNote方法保存便签。如果保存成功，设置结果码为RESULT\_OK表示创建或编辑便签** |

* + 1. **模型层相关的类**

**14.3.2.1、类名：**Note

1. **类的作用：**

管理便签（notes）的数据，包括创建、编辑、同步等操作。它维护便签的元数据和文本数据以及通话数据，并将这些数据同步到Android应用的数据库中

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNoteDiffValues | 一个ContentValues对象，用于存储便签的元数据的差异 |
| mNoteData | 一个NoteData对象，用于管理便签的文本数据和通话数据 |
| TAG | 一个用于日志记录的常量字符串 |

Note类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getNewNoteId(Context context, long folderId) | context：Android应用程序的上下文；folderId表示便签所属的文件夹的ID | 创建一个新的便签ID并插入数据库中，返回新便签的ID | 创建一个ContentValues对象values，用于存储便签的元数据，初始化values，从URI中提取新便签的ID并返回 |
| Note() | / | 构造函数，初始化mNoteDiffValues和mNoteData对象 | / |
| setNoteValue(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置便签元数据的键值对，用于标记元数据的本地修改 | 将键值对存储到mNoteDiffValues，并更新LOCAL\_MODIFIED和MODIFIED\_DATE的值 |
| setTextData(String key, String value) | String key：文本数据的键名；String value：文本数据的键值 | 设置文本数据的键值对 | 将键值对存储到mNoteData的文本数据中 |
| setTextDataId(long id) | id：文本数据的ID | 设置文本数据的ID | 将ID存储到mNoteData中 |
| getTextDataId() | / | 获取文本数据的ID | 返回mNoteData中的文本数据ID |
| setCallDataId(long id) | id：通话数据的ID | 设置通话数据的ID | 将ID存储到mNoteData中 |
| setCallData(String key, String value) | key：通话数据的键名；value：通话数据的键值 | 设置通话数据的键值对 | 将键值对存储到mNoteData的通话数据中 |
| isLocalModified() | / | 判断便签是否在本地有修改 | 检查mNoteDiffValues和mNoteData是否包含本地修改的数据，如果有则返回true，否则返回false |
| syncNote(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 同步便签到数据库中 | 如果没有本地修改，直接返回true表示同步成功；如果有本地修改，首先尝试更新便签的元数据 |
| NoteData.isLocalModified() | / | 判断文本数据或通话数据是否在本地有修改 | / |
| NoteData.setTextDataId(long id) | id表示传入的id | 设置文本数据的ID | / |
| NoteData.setCallDataId(long id) | id表示传入的id | 设置通话数据的ID | / |
| NoteData.setCallData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置通话数据的键值对 | / |
| NoteData.setTextData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置文本数据的键值对 | / |
| pushIntoContentResolver(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 将文本数据或通话数据同步到数据库中 | 根据是否存在本地修改，构建数据库操作，然后执行同步操作 |

* + - 1. **类名：**WorkingNote

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| **saveNote()** | **/** | **保存便签的方法，包括新建便签和已有便签的保存** | **判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库** |
| **existInDatabase()** | **/** | **判断便签是否已存在于数据库中** | **判断NoteId是否存在** |
| **isWorthSaving()** | **/** | **判断便签是否值得保存** | **考虑是否已删除、是否存在文本内容和是否有本地修改** |

* + 1. **数据层相关的类**

**14.3.3.1、Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与便签关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表便签的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表便签是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的便签数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本便签的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本便签的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本便签的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本便签的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

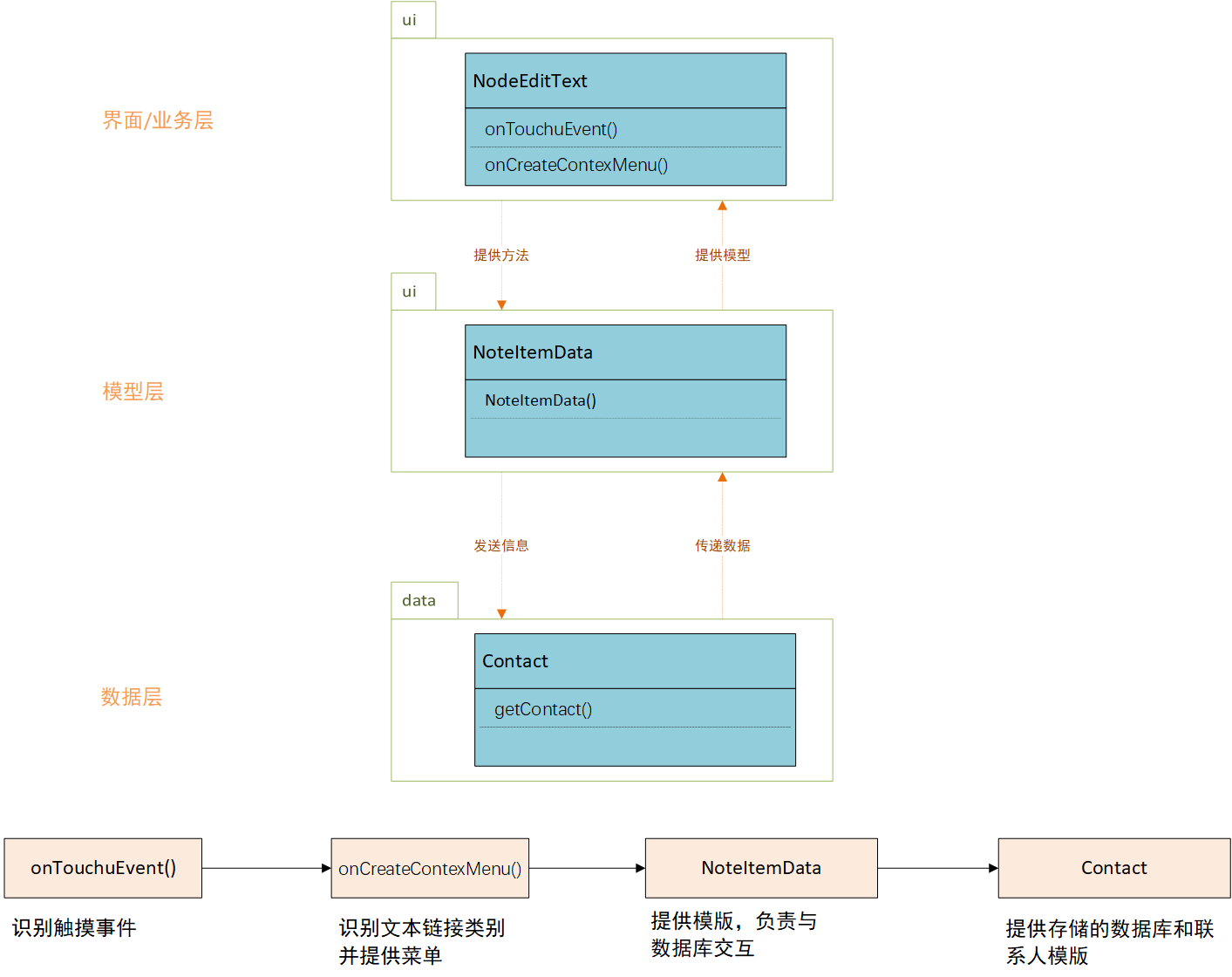
1. **功能十五：识别电话号码**
   1. **功能介绍**

**功能概述：**当在便签内输入电话号码的时候，点击可以出现对应的呼叫电话或者增添到手机通讯录的功能，并将其设置为高亮的蓝色，识别为电话号码。

**操作步骤：**新建一个便签，然后在便签中输入一个电话号码，在点击电话号码可以出现对应的呼叫号码操作等操作的菜单。然后号码会自动标识为高亮的蓝色以区分普通的便签内容。

* 1. **业务请求与响应过程**

首先是类NodeEditText中onCreateContextMenu()方法识别到文本中是否包含链接如电话号码和email地址和url网址，然后onTouchEvent()触摸操作识别用户的点击然后再显示对应连接的菜单，如果已有定义的链接资源方案操作则使用定义的方案，否则则使用R.string.note\_link\_other来使用其他链接方案，根据链接的不同使用不同的默认菜单，在从onMenuItemClick()方法中完成菜单功能与用户点击的交互操作，最终完成电话号码的识别功能。联系人的信息则由Contact类利用Android自带的data包数据库来完成存储。



* 1. **类的作用**
     1. **界面层/业务层相关的类**

**15.3.1.1、类名：**NodeEditText

1. **类的作用：**

完成电话号码的识别，和提供处理不同链接方案的方法，同时显示其对应的界面菜单。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | 用于记录日志和标记信息 |
| mIndex | 标识文本编辑界面的位置 |
| mSelectionStarBeforeDelete | 记录删除键按下前的位置 |
| SCHEME\_TEL | 标识链接的值与方案 |
| SCHEME\_HTTP |
| SCHEME\_EMAIL |
| sSchemaACctionResMap | Hash表存储标识值并用做匹配 |
| mOnTextViewChangeListener | 监听文本框中的变化事件 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteEditText(Context context) | context:用于访问其他资源和服务 | 类的构造函数，用于创造新的实例时，做出对应的赋值操作 | 调用父类EditText的的构造函数并传递context参数 |
| setIndex(int index) | index:用于传递新的索引值 | 用于在对象创建后或在运行时更改对象的状态或属性。 | 将代码传递给mIdex |
| setOnTextViewChangeListener(OnTextViewChangeListener listener) | listener:监听器 | 方法用于注册监听器，以便在文本内容发生变化时执行相应的操作。这样，外部代码可以通过设置监听器来定义 NoteEditText 对象的行为和响应。 | 直接通过给mOnTextViewChangeListener赋值为listener。 |
| onTouchuEvent(MotionEvent event) | event:监听的当前触摸事件 | 监听屏幕上的触摸事件 | 当触摸时，调用MotionEvent.ACTION\_DOWN,然后获取事件的坐标，并根据文本编辑框调整坐标的位置。 |
| onKeyDown(int keyCode , KeyEvent event) | keyCode:表示用户按下的键码  event:表示键盘事件信息 | 这个方法用于处理回车键和删除键的特殊情况。在按下回车键时，它检查是否设置了文本编辑监听器，如果设置了，就让监听器处理回车事件；在按下删除键时，它记录当前光标位置以备后续使用。其他按键事件将继续传递以由其他代码处理。在用户按下键盘时触发。 | 用switch语句检查按下的键是否为回车键或删除键在执行对应的操作。 |
| onKeyUp(int keyCode , KeyEvent event) | keyCode:表示用户按下的键码  event:表示键盘事件信息 | 这个方法用于处理回车键和删除键的特殊情况。在按下回车键时，它检查是否设置了文本编辑监听器，如果设置了，就让监听器处理回车事件；在按下删除键时，它记录当前光标位置以备后续使用。其他按键事件将继续传递以由其他代码处理。在用户按下键盘松开后触发。 | 用switch语句检查按下的键是否为回车键或删除键在执行对应的操作。 |
| onFoucusChanged(booleanfoucused, int direction, RectpreviouslyFoucusedRect) | foucused:布尔值，表示文本框控件当前是否具有焦点。  direction:用于指示焦点状态的改变方向。  previouslyFocusedRect: 这是一个矩形对象，表示在失去焦点之前，文本框控件的焦点矩形区域。 | 用于监听文本框控件的焦点状态变化，并根据焦点状态和文本内容是否为空来触发相应的文本变化事件。这可以用于执行与焦点状态和文本内容相关的操作。 | 首先检查是否设置了mOnTextViewChangeListener，即是否有文本视图变化的监听器,如果控件失去焦点或者获得焦点分别执行对应的操作。 |
| onCreateContexMenu(ContextMenu menu) | menu:作为创建的上下文菜单的对象 | 检查所选文本是否包含链接，根据链接的方案为用户提供相应的上下文菜单项，以便执行链接的操作，例如拨打电话、打开网页等。 | 先检查文本编辑框中文本是否属于Spanned类型在通过匹配sSchemaActionResMap来决定如何处理这个链接，然后在通过R.string.note\_link\_other方法使用默认资源id，menu.add调用对应的菜单。 |

* + 1. **模型层相关的类**

**15.3.2.1、类名：NoteItemData**

1. **类的作用:**
2. **类的精读:**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mId | 便签项的ID |
| mId | 提醒时间 |
| mBgColorId | 背景颜色ID |
| mCreatedDate | 创建时间 |
| mHasAttachment | 是否有附件 |
| mModifiedDate | 最近修改时间 |
| mNotesCount | 笔记数量 |
| mParentId | 父文件夹ID |
| mSnippet | 便签内容的摘要 |
| mType | 便签类型 |
| mWidgetId | Widget的ID |
| mWidgetType | Widget类型 |
| mName | 联系人名字 |
| mPhoneNumber | 联系人电话号码 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| NoteItemData(Context context, Cursor cursor) | context:用于访问其他资源和服务  cursor:访问数据库的游标 | 类的构造函数，用于创造新的实例时，从数据库中的查找结果，做出对应的赋值操作 | 用cursor在数据库中进行查询匹配，在对类中的值初始化。 |
| checkPosition(Cursor cursor) | cursor:游标 | 用于确定笔记列表中的当前游标位置的特殊情况，以便在用户界面中进行适当的显示和处理。 | 先检查游标是否在笔记列表的第一项和最后一项，然后在检查游标中的项数是否为1，在检查笔记类型在做出对应的赋值操作 |
| isOneFollowingFolder() | 无 | 判断是否是一个便签项后面跟随一个文件夹 | 返回当前布尔值mIsOneNoteFollowingFolder作为判断 |
| isMultiFollowingFolder() | 无 | 判断是否是一个便签项后面跟随多个文件夹 | 返回当前布尔值mIsMultiNotesFollowingFolder作为判断 |
| isLast() | 无 | 判断是否是列表中的最后一个项 | 返回当前布尔值mIsLastItem作为判断 |
| getCallName() | 无 | 获取拨打联系人的姓名 | Return返回当前姓名 |
| isFirst() | 无 | 判断是否是列表中的第一个项 | 返回当前布尔值mIsFirstItem作为判断 |
| getId() | 无 | 获取便签的id | Return直接返回 |
| gerBgcolorId | 无 | 获取背景颜色的id |
| getAlertDate() | 无 | 获取提醒的日期 |
| getCreatedDate() | 无 | 获取创建的时间 |
| hasAttachment() | 无 | 判断是否有附件 |
| getModifiedDate() | 无 | 获取最近修改时间 |
| isSingle() | 无 | 判断列表是否只有一个项 | 返回当前布尔值mIsOnlyOneItem作为判断 |
| getType() | 无 | 获取数据类型 | Return直接返回对应值 |
| getNotesCount() | 无 | 获取便签数量 |
| getWidgetType() | 无 | 获取小部件的类型 |
| getWidgetId() | 无 | 获取小部件的id |
| getSnippet() | 无 | 获取文本片段 |
| hasAlert() | 无 | 判断是否有提醒 | 通过比较是否存在提醒日期来判断 |
| isCallRecord() | 无 | 判断当前笔记是否通话记录 | 通过比较 mParentId 是否等于 Notes.ID\_CALL\_RECORD\_FOLDER 并且 mPhoneNumber 不为空来确定笔记是否为通话记录。 |
| gerNoteType(Cursor cursor) | cursor：游标 | 获取便签的工作类型 | 通过游标返回当前便签的类型 |

* + 1. **数据层相关的类**

**15.3.3.1、类名：Contact**

1. **类的作用：**

创建联系人的数据库，并设置联系人id的构建，提供获取联系人的方法。采用Android自带的data包的数据库来完成存储，用phone来构建对应的字段。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| sContactCache | 用于缓存电话号码和联系人名字的对应关系。如果已经查询过某个电话号码，它会将结果存储在这个缓存中，以便后续快速查询 |
| TAG | 标识文本编辑界面的位置 |
| CALLER\_ID\_SELECTION | 这是一个 SQL 查询字符串，用于根据电话号码查询联系人的 SQL 语句。它使用电话号码与数据库中的数据匹配，同时完成id构建 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getContact(Context contex, String phoneNumber) | context:用于访问应用程序的资源和服务  phoneNumber:要查询人的电话号码 | 用于获取联系人的名字 | 先检查缓存sContactCache中是否有对应的号码，如果没有在创立条件selection利用cusor在data包的数据库中完成查询。如果有则返回对应的联系人的名字 |

1. **功能十六：同步**
   1. **功能介绍**

**功能概述：**将本地的便签事项上传到云服务器，或将 Google Task 上的

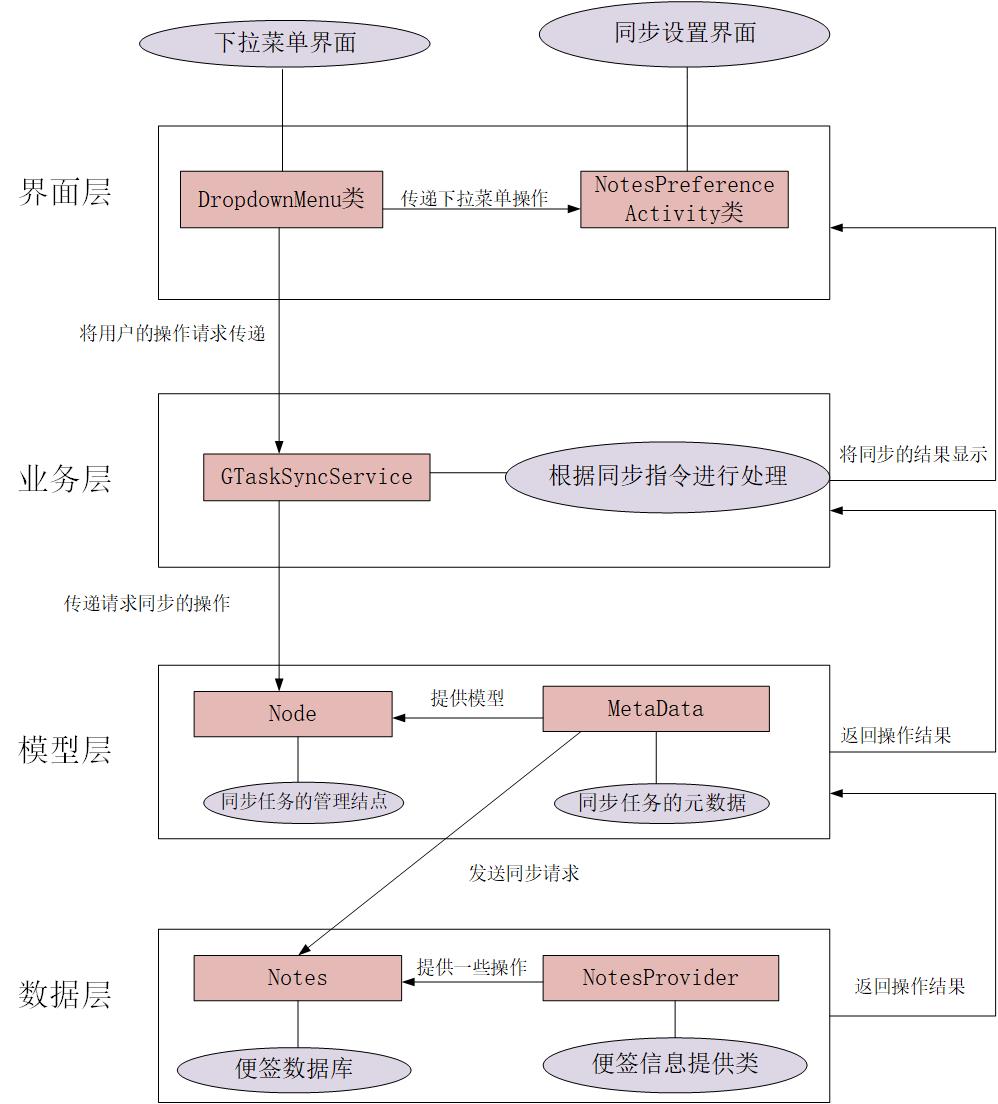
便签下载到本地

**操作步骤：**主界面的菜单栏中选择“同步”，登录 Google 账号，能够与 Google Task 备忘录同步便签记录，将本地的便签事项上传到云服务器，或将 Google Task 上的便签下载到本地，实现数据备份与迁移。

* 1. **业务请求与响应过程**

在同步的过程中，四个层级各自的作用大致如下：

1. **界面层**：用户在小米便签的界面上进行同步操作，例如点击同步按钮或下拉刷新
2. **业务层**：小米便签客户端会根据用户的同步操作，调用相应的业务逻辑处理程序，例如上传便签内容或下载同步的便签内容。
3. **模型层**：小米便签客户端会根据业务逻辑处理程序的要求，调用相应的模型层程序，例如读取或写入本地数据库中的便签内容。
4. **数据层**：小米便签客户端会根据模型层程序的要求，访问本地数据库或向小米云服务器发送网络请求，例如上传或下载便签内容。



* 1. **类的作用**
     1. **界面层相关的类**

**16.3.1.1、类名：DropdownMenu**

1. **类的作用：**

下拉菜单界面，在Android 应用程序中创建一个下拉菜单，让用户可以选择不同的选项。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mButton | 一个 Button 对象，用于触发下拉菜单的显示 |
| mPopupMenu; | 一个PopupMenu 对象，用于显示下拉菜单的弹出窗口。 |
| mMenu; | 一个 Menu 对象，用于存储下拉菜单的选项。 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| DropdownMenu | 上下文对象Context、  按钮对象Button  菜单资源ID menuId | 将一个按钮对象和一个下拉菜单对象关联起来，并在按钮被点击时显示下拉菜单 | 创建一个按钮对象mButton，并设置其背景为下拉图标。  创建一个PopupMenu对象mPopupMenu，并将其与按钮对象mButton关联起来。  创建一个Menu对象mMenu，并从菜单资源ID menuId中加载菜单项，将菜单项添加到mMenu中。  为按钮对象mButton设置一个点击事件监听器，当按钮被点击时，显示下拉菜单mPopupMenu。 |

* + - 1. **类名：NotesPreferenceActivity**

1. **类的作用：**

该类是小米便签的同步设置界面

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| getLastSyncTime | 用来获取上一次同步时间的，它接收一个Context 对象作为参数，然后从 SharedPreferences 中读取上一次同步时间的值，并返回该值。 |
| mNotesListAdapter | 用于管理便签列表展示的适配器（Adapter）实例。 |
| mNotesListView; | 用于显示便签列表的ListView 对象。 |
| BackgroundQueryHandler | 用于在后台执行异步查询操作 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate | Bundle icicle：  保存 Activity 的状态信息 | 对便签进行按照日期的排序 | 在后台异步查询便签列表数据，并根据一些条件进行筛选，以获取与当前文件夹相关的便签数据，并按照类型和修改日期降序进行排序。 |
| loadAccountPreference | defaultAccount：  获取名称 | 加载用户账户偏好设置，例如用户的同步账户、同步频率、同步时间等。 | 这个方法会调用其他类的方法，来获取和设置用户的账户偏好设置。在加载用户账户偏好设置时，这个方法可能会将用户的账户偏好设置显示在界面上 |
| showSelectAccountAlertDialog | mOriAccounts： accounts;  mHasAddedAccount = false; | 显示一个对话框，让用户选择要同步的Google账户 | 创建一个AlertDialog.Builder对象，用于构建对话框。  创建一个自定义标题视图用于显示对话框的标题和说明。  获取用户的Google账户列表，并将默认账户设置为选中状态。  当用户选择了要同步的账户后，将选中的账户保存到本地，并刷新UI界面。  显示对话框。 |

* + 1. **业务层相关的类**
       1. **类名：**GTaskSyncService

1. **类的作用：**

GTask 同步任务，用于提供同步服务（开始、取消同步），发送广播

**（2）类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| ACTION\_STRING\_NAME | ACTION\_STRING\_NAME ："sync\_action\_type"; |
| GTASK\_SERVICE\_BROADCAST\_NAME | net.micode.notes.gtask.remote.gtask\_sync\_service" |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| startSync | / | 启动同步操作，创建并执行NoteSyncTask类的异步任务。 | 创建一个新的类的异步任务，然后向其他组件发送广播，通知它们同步操作已经开始。接着，开始执行异步任务。在异步任务执行完成后，向其他组件发送广播，通知它们同步操作已经完成。最后，停止SyncManager服务的运行。 |
| onStartCommand | Intent intent表示接收到的同步指令  int flags表示启动服务时的标志位  int startId表示服务的启动ID。 | 接收并处理从其他组件发送过来的同步指令，例如启动同步操作、取消同步操作等。 | 接收到同步指令时，方法会根据指令的类型，调用相应的方法，来启动或取消同步操作。在处理同步指令时，返回START\_STICKY标志位，表示服务在被系统杀死后会自动重启。如果接收到的同步指令无效，则返回默认值。 |

**16.3.2.2、类名：**GTaskStringUtils

1. **类的作用：**

同步中使用的字符串工具类，为jsonobject 提供 string 对象

**（2）类的精读：**

这个类中是一些常量的定义，用于Google Task的JSON数据解析。这些常量定义了JSON数据中的各种字段名称，例如GTASK\_JSON\_ACTION\_ID表示JSON数据中的action\_id字段，GTASK\_JSON\_ACTION\_TYPE\_CREATE表示JSON数据中的action\_type字段的值为create等。这些常量的作用是方便程序员在代码中使用这些字段名称，避免硬编码，提高代码的可读性和可维护性。在Google Task的JSON数据解析过程中，程序员可以使用这些常量来获取JSON数据中的相应字段的值，从而实现数据的解析和处理。

* + 1. **模型层相关的类**

**16.3.3.1、类名：MetaData**

1. **类的作用：**关于同步任务的元数据
2. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | 成员名 |
| mRelatedGid | Null |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setMeta | gid是一个字符串类型的参数，表示Google Task中的任务ID  metaInfo是一个JSONObject类型的参数，表示Google Task中任务的元数据信息 | 设置便签的元数据信息 | 首先将gid添加到metaInfo中，然后将metaInfo转换成字符串，通过调用setNotes()方法将其设置为便签的内容。同时，设置便签的元数据名称。 |
| setContentByRemoteJSON | 远程JSON对象，包含便签的标题、内容、创建时间、修改时间等信息 | 从远程JSON对象中解析便签内容，并将解析结果存储到NoteModel对象中。 | 首先将远程JSON对象中的通用属性解析并存储然后，获取便签的元数据信息，接着，它将元数据信息解析成JSON对象，并从中获取GTask ID如果解析过程中出现异常，它会在日志中记录错误信息。这样，NoteModel对象就包含了从远程JSON对象中解析出的便签内容和元数据信息，可以用于后续的同步操作。 |

**16.3.3.2、类名：Node**

1. **类的作用：**同步任务的管理结点，用于设置、保存同步动作的信息
2. **类的精读：**

Node 类是一个抽象类，不能直接实例化，需要通过继承来实现具体的节点类型。Node 类中定义了一些抽象方法，需要在子类中实现。例如，getCreateAction() 方法和 getUpdateAction() 方法需要根据具体的节点类型来生成相应的创建和更新操作；setContentByRemoteJSON() 方法和 setContentByLocalJSON() 方法需要根据具体的节点类型来解析远程或本地 JSON 数据，并设置节点内容；getLocalJSONFromContent() 方法需要根据具体的节点类型来生成节点内容的本地 JSON 数据；getSyncAction() 方法需要根据具体的节点类型和同步状态来生成相应的同步操作。Node 类中还定义了一些成员变量，例如 mGid、mName、mLastModified 和 mDeleted，用于存储节点的基本信息。子类可以通过调用 Node 类的方法来设置或获取这些成员变量的值。

方法作用

|  |  |
| --- | --- |
| **方法** | **作用** |
| getCreateAction(int actionId) | 获取创建节点的操作 |
| getUpdateAction(int actionId)： | 获取更新节点的操作 |
| setContentByRemoteJSON  (JSONObjectjs) | 根据远程 JSON 数据设置节点内容 |
| setContentByLocalJSON(JSONObjectjs) | 根据本地 JSON 数据设置节点内容。 |
| getLocalJSONFromContent() | 获取节点内容的本地 JSON 数据。 |
| getSyncAction(Cursor c) | 获取节点的同步操作。 |
| setGid(String gid) | 设置节点的全局唯一标识符 |
| setName(String name)： | 设置节点的名称。 |
| setLastModified(long lastModified) | 设置节点的最后修改时间。 |
| setDeleted(boolean deleted)： | 设置节点是否被删除。 |
| getGid()： | 获取节点的全局唯一标识符 |
| getName()： | 获取节点的名称。 |
| getLastModified()： | 获取节点的最后修改时间。 |
| getDeleted()： | 获取节点是否被删除。 |

* + 1. **数据层相关的类**

**16.3.4.1、类名：Notes**

1. **类的作用：**

便签数据库，用于记录便签相关属性和数据

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |

此类由于是data类，用于保存和记录数据，没有相关的方法

* + - 1. **类名：NotesProvider**

1. **类的作用：**便签信息提供类
2. **类的精读：**

继承了ContentProvider（Android 中用于提供数据访问接口的基类，可以从其他应用程序中访问和共享数据。）类

用于处理小米便签数据的内容提供器。它可能提供了一些方法来实现对便签数据的增删改查操作，以及与其他应用程序进行数据共享和访问的功能。

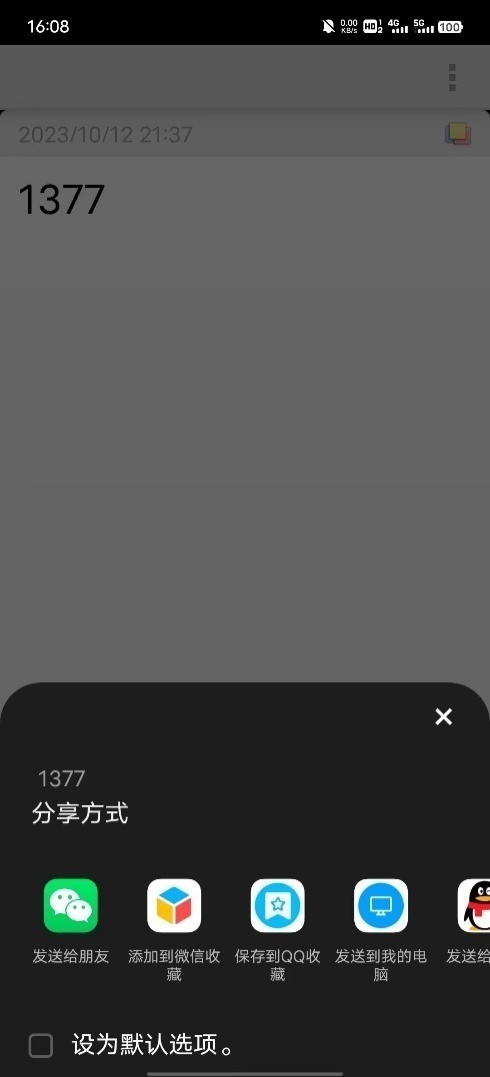
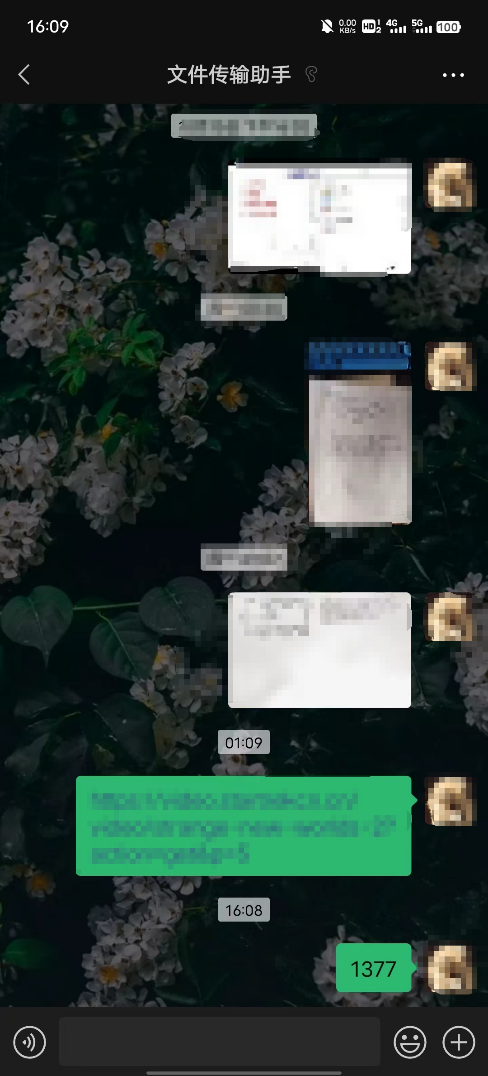
**类中的方法**

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| insert | URI：唯一标识符  ContentValues：目录 | 用于插入数据到数据库中 | 据提供的uri，它将区分是插入便签数据还是插入其他数据。 |
| getType | URI：唯一标识符 | 判断给定的 Uri 是属于什么类型的数据 | 根据给定的 Uri 获取其类型，并返回一个字符串表示 |
| delete | URI：唯一标识符  deleteData:布尔值  selection：删除选择内容 | 一个删除便签或数据的方法 | 通过匹配 Uri 来确定要删除的是便签还是数据。根据不同的 Uri 类型，采取不同的删除操作。  在删除便签时，根据传入的选择条件和参数执行删除操作。将删除影响的行数保存到 count 变量中。 |

1. **功能十七：分享便签**

## 17.1．功能介绍

**功能概述：**点击分享，将笔记内容分享出去

**操作步骤：**在便签的文本编辑界面，点击右上角菜单栏，点击“分享”按钮，自动获取当前终端所有可以分享的APP，选中其中某一APP后，将便签以文本内容的格式分享出去。

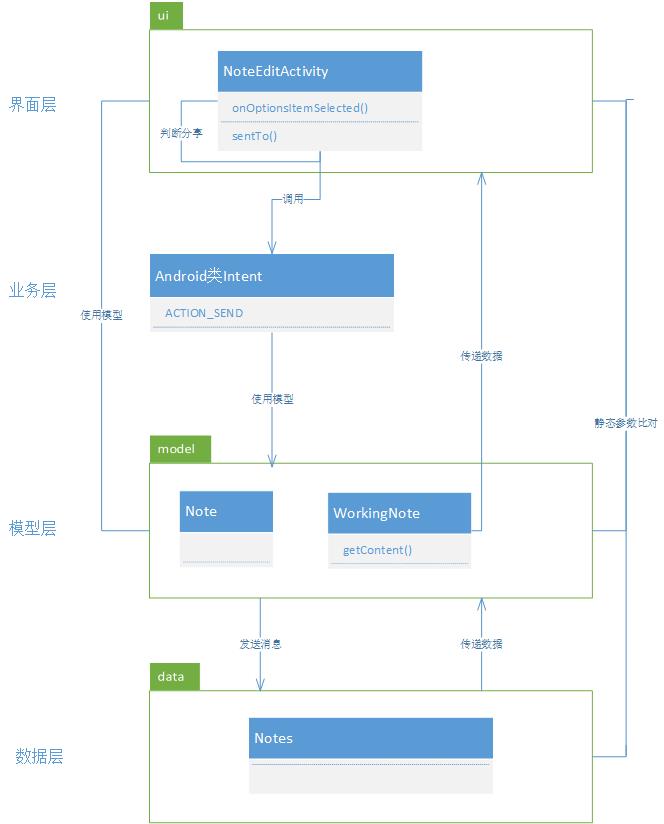
## 17.2．业务请求与响应过程

分享功能（下简称“功能“）的界面层由包ui中类NoteEditActivity的sendTo()方法实现。在按下“分享”按钮后，执行按钮项目判断，然后调用getWorkingText()方法获取当前便签中的内容，然后调用sendTo()方法实现分享功能。sendTo()方法使用Android Sharesheet将文本内容从一个Activity发送到另一个Activity。

功能的业务层为Android内置的类Intent，实现封装好的事件ACTION\_SEND，将获取到的便签内容以文本格式分享出去。

功能的模型层由包model的类WorkingNotes中getContent()方法实现，获取当前便签的文本内容，并向上转发。

功能的数据层由包data的类Notes实现，调用Notes中的静态参数进行数据的获取。



## 17.3. 类的作用

### 17.3.1. 界面层相关的类

**17.3.1.1**、类名：**NoteEditText**

1. **类的作用：**

该类是便签编辑操作的集成，对便签界面进行初始化，提供设置提醒时间弹窗，对便签界面的菜单按钮时间进行定义，对便签的删改操作进行定义，实现了清单模式变化和清单项完成文本变化等操作

1. **类的精读：**

NoteEditText类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 头部视图类 |
| sBgSelectorBtnsMap | 一个映射，将背景选择按钮的资源ID映射到背景颜色的整数值 |
| sBgSelectorSelectionMap | 一个映射，将背景颜色的整数值映射到选中的背景选择按钮的资源ID |
| sFontSizeBtnsMap | 一个映射，将字体大小按钮的资源ID映射到字体大小的整数值 |
| sFontSelectorSelectionMap | 将字体大小的整数值映射到选中的字体选择按钮的资源ID |
| TAG | 一个字符串常量，标签 |
| mNoteHeaderHolder | 用于保存便签的头部视图元素 |
| mHeadViewPanel | 便签编辑器中的头部视图容器 |
| mNoteBgColorSelector | 用于选择便签的背景颜色 |
| mFontSizeSelector | 用于选择文本的字体大小 |
| mNoteEditor | 一个EditText控件，用于编辑便签的文本内容 |
| mNoteEditorPanel | 便签编辑器的容器 |
| **mWorkingNote** | **包含有关便签的状态信息** |
| mSharedPrefs | 一个SharedPreferences对象，用于存储和管理应用程序的偏好设置 |
| mFontSizeId | 存储所选的字体大小的整数值 |
| PREFERENCE\_FONT\_SIZE | 字体大小偏好设置的键名 |
| SHORTCUT\_ICON\_TITLE\_MAX\_LEN | 快捷图标的标题最大长度 |
| TAG\_CHECKED | 表示待办事项列表中的某任务已完成 |
| TAG\_UNCHECKED | 表示待办事项列表中的某任务未完成 |
| mEditTextList | 包含编辑文本的列表 |
| mUserQuery | 保存用户的查询文本 |
| mPattern | 用于在文本中进行匹配和搜索 |

NoteEditText类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState用于在重建活动时存储先前的状态信息 | Android Activity的生命周期方法之一，用于创建活动时初始化界面和资源 | 首先调用父类的onCreate方法，然后加载与活动相关的布局文件（R.layout.note\_edit）。接着，它检查是否存在先前保存的状态信息（savedInstanceState），并根据情况初始化活动状态或结束活动 |
| onRestoreInstanceState(Bundle savedInstanceState) | savedInstanceState包含之前保存的状态信息 | 用于在活动被系统销毁后重新创建时，恢复之前的状态信息 | 如果存在Intent.EXTRA\_UID的键在savedInstanceState中，它将创建一个新的Intent，将之前保存的UID添加到其中，并尝试使用该Intent初始化活动状态 |
| initActivityState(Intent intent) | intent包含用于初始化活动的信息 | 用于初始化活动的状态，包括加载或创建便签 | 根据传入的Intent的不同操作（ACTION） |
| onResume() | / | 用于在活动重新进入前台时进行初始化和更新界面 | 调用initNoteScreen()方法来初始化便签编辑界面 |
| initNoteScreen() | / | 初始化便签编辑界面的各个部分，包括文本样式、文本内容、背景颜色等 | 根据便签的内容和属性，设置文本样式、内容，以及相关的背景颜色。如果便签具有提醒信息，它还会在界面上显示提醒的日期和图标 |
| showAlertHeader() | / | 在界面上显示或隐藏便签的提醒信息 | 根据便签是否具有提醒信息，它决定显示或隐藏提醒日期和图标，并根据提醒的时间情况显示相应的文本（已过期或相对时间） |
| onNewIntent(Intent intent) | intent包含新的Intent信息 | 用于在活动已经存在的情况下处理新的Intent | 首先调用父类的onNewIntent方法，然后使用传入的Intent重新初始化活动状态 |
| onSaveInstanceState(Bundle outState) | outState存储活动状态信息 | 保存活动的状态信息，以便在需要时进行恢复 | 如果便签还没有存在于数据库中（没有分配ID），则首先尝试保存便签。然后，将便签的ID放入outState中，以便在重建活动时使用 |
| dispatchTouchEvent(MotionEventev) | MotionEventev包含有关触摸事件的信息 | 处理触摸事件分发，用于检测用户的点击位置以隐藏背景颜色选择器和字体大小选择器 | 检查背景颜色选择器和字体大小选择器是否可见，如果不在它们的范围内点击，则隐藏它们。最后，调用父类的dispatchTouchEvent方法继续事件分发 |
| inRangeOfView(View view, MotionEventev) | view要检查的视图；ev触摸事件信息 | 检查触摸事件是否在指定视图内 | 获取视图的位置信息，然后检查触摸事件的坐标是否在视图的范围内。如果在范围内，返回true，否则返回false |
| initResources() |  | 初始化活动所需的资源和视图 | 初始化了活动的各个部分，包括头部视图、便签编辑器、背景颜色选择器、字体大小选择器等 |
| onPause() | / | 活动进入后台时保存便签数据 | 首先调用父类的onPause方法，然后尝试保存便签数据，并清除设置状态 |
| updateWidget() | / | 更新小部件的显示 | 根据活动中的便签的小部件类型，创建一个对应的Intent，并发送广播以更新小部件的显示 |
| onClick(View v) | v点击的视图对象 | 处理点击事件 | 根据点击的视图的ID，执行不同的操作。例如，当点击背景颜色选择按钮时，显示颜色选择器；当点击背景颜色选择器中的颜色时，更改便签的背景颜色等 |
| onBackPressed() | / | 处理用户按下返回按钮时的行为 | 首先调用clearSettingState方法，以确保任何设置状态都被清除。然后尝试保存便签，并调用父类的onBackPressed方法以返回 |
| clearSettingState() | / | 清除设置状态 | 检查颜色选择器和字体大小选择器是否可见，如果可见，则隐藏它们。如果成功隐藏，返回true，否则返回false |
| onBackgroundColorChanged() | / | 处理背景颜色改变时的操作 | 根据便签的新背景颜色，更新界面元素的颜色 |
| onPrepareOptionsMenu(Menu menu) | menu菜单对象 | 准备选项菜单 | 清除任何现有的菜单项，然后根据便签的状态和属性，动态生成适当的菜单项，包括切换列表模式、设置提醒等操作 |
| **onOptionsItemSelected(MenuItem item)** | **item表示用户选择的菜单项** | **处理用户在选项菜单中选择的操作** | **根据用户的不同选择进行查找** |
| setReminder() | / | 用于设置提醒时间 | 创建一个日期时间选择对话框，允许用户选择提醒时间。选择时间后，将提醒时间设置给当前便签 |
| **sendTo(Context context, String info)** | **context上下文对象；info要分享的便签内容** | **将便签内容分享到支持Intent.ACTION\_SEND动作和text/plain类型的应用程序** | **创建一个Intent，设置EXTRA\_TEXT为便签内容，类型为text/plain，然后启动适用的应用程序以分享内容** |
| createNewNote() | / | 创建一个新的便签 | 首先保存当前正在编辑的便签，然后结束当前活动。接下来，创建一个新的NoteEditActivity意图，以新建便签的方式启动该活动 |
| deleteCurrentNote() | / | 删除当前便签 | 如果当前便签已存在于数据库中，将其ID添加到HashSet中，并根据是否启用同步模式执行不同的删除操作。如果不是同步模式，调用DataUtils. batch DeleteNotes来删除便签；如果是同步模式，将便签移动到回收站。最后，将当前便签标记为已删除 |
| isSyncMode() | / | 检查是否处于同步模式 | 检查应用的同步账户名称是否为空，如果不为空则表示处于同步模式，返回true，否则返回false |
| onClockAlertChanged(long date, boolean set) | date提醒时间；set是否设置提醒） | 处理提醒时间的更改 | 如果当前便签未存在于数据库中，首先保存便签。然后创建一个用于提醒的Intent，并根据set参数设置或取消提醒。最后，根据提醒是否设置，调用showAlert Header方法以更新提醒状态的显示 |
| updateWidget() | / | 更新小部件的显示 | 根据便签的小部件类型，创建一个对应的Intent，然后发送广播以更新小部件的显示 |
| onEditTextDelete(int index, String text) | index文本编辑框索引；String text要删除的文本 | 处理文本编辑框中的文本删除操作 | 从文本编辑框列表中删除指定索引的文本编辑框，同时调整其他文本编辑框的索引，然后将被删除的文本追加到前一个文本编辑框的末尾 |
| onEditTextEnter(int index, String text) | index文本编辑框索引；text要插入的文本 | 处理文本编辑框中的文本插入操作 | 创建一个新的文本编辑框，将其插入到指定索引处，同时调整其他文本编辑框的索引 |
| switchToListMode(String text) | text要分割为列表的文本 | 切换为列表模式，将文本分成多个项目 | 将文本按行分割，创建文本编辑框列表，并将每一行文本作为列表项目添加到列表中。最后，隐藏原始的文本编辑器，显示文本编辑框列表 |
| getHighlightQueryResult(String fullText, String userQuery) | fullText完整文本；userQuery用户查询 | 创建一个具有高亮显示查询结果的Spannable文本 | 首先创建一个SpannableString，然后使用正则表达式将用户查询高亮显示在完整文本中，最后返回带有高亮的Spannable文本 |
| getListItem(String item, int index) | item列表项文本；index列表项索引 | 创建一个包含列表项的视图 | 从布局文件R.layout.note\_edit\_list\_item创建一个新视图，获取文本编辑框和复选框，设置文本样式、文本内容和复选框的监听器。根据列表项是否以特定标记（TAG\_CHECKED 或 TAG\_UNCHECKED）开头，设置复选框的选中状态和文本内容，并应用高亮查询结果。返回创建的视图 |
| onTextChange(int index, booleanhasText) | index文本编辑框索引；hasText文本编辑框是否包含文本 | 处理文本编辑框中的文本变化事件 | 根据文本编辑框是否包含文本，设置相应文本编辑框的复选框的可见性 |
| onCheckListModeChanged(int oldMode, int newMode) | oldMode原模式；newMode新模式 | 处理待办事项列表模式的切换 | 如果新模式是待办事项列表模式，将编辑框内容切分为多个项目，然后显示列表项 |
| **getWorkingText()** | **/** | **获取清单模式下任务状态及内容** | **判断是否处于清单模式，对于每一项任务是否执行完成进行状态区分并呈现不同状态** |
| saveNote() | / | 保存便签 | 获取工作文本，然后调用saveNote方法保存便签。如果保存成功，设置结果码为RESULT\_OK表示创建或编辑便签 |
| sendToDesktop() | / | 将便签发送到桌面 | 如果便签不存在于数据库中，首先保存便签。然后，创建一个Intent，用于创建快捷方式。将快捷方式信息设置为便签的内容和图标，然后发送广播以创建快捷方式。如果便签没有ID（新建便签且内容为空），显示错误消息 |
| makeShortcutIconTitle(String content) | content便签内容 | 生成快捷方式图标的标题 | 移除内容中的特定标记，然后将标题限制在最大长度内 |
| showToast(int resId) | resId消息资源ID | 显示短时长的Toast消息 | 调用重载函数showToast(int resId, int duration) |
| showToast(int resId, int duration) | resId消息资源ID；duration Toast显示时长 | 显示自定义时长的Toast消息 | / |

### 17.3.2. 业务层相关的类

**17.3.2.1**、类名：Intent

1. **类的作用：**是一种运行时绑定（run-time binding）机制，可以应用于两个应用间的通讯交互，也能够应用于在同一个应用下不同组件的交互，由于是android的内置类，在此不展开解释。

### 17.3.3. 模型层相关的类

**17.3.3.1、类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| **mContent** | **便签的文本内容** |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| WorkingNote(Context context, long folderId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID | 构造函数，用于创建一个新的工作便签 | / |
| WorkingNote(Context context, long noteId, long folderId) | context：Android应用程序的上下文；noteId：要加载的便签的ID；folderId：便签所属的文件夹ID | 构造函数，用于加载已有的工作便签 | / |
| loadNote() | / | 用于加载便签的元数据信息 | 创建游标，游标移动到制定，获取属性值并赋给对应便签属性 |
| loadNoteData() | / | 用于加载便签的内容数据 | 创建游标，游标移动到制定，获取文本内容并赋给 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID；widgetId：小部件的ID；widgetType：小部件类型；defaultBgColorId：默认的背景颜色ID | 创建一个新的空白便签 | 创建一个新的工作便签，并设置其属性，包括背景颜色、小部件ID等 |
| load(Context context, long id) | context：Android应用程序的上下文；id：要加载的便签的ID | 加载现有的工作便签 | 根据传入的条件查找便签，并加载 |
| saveNote() | / | 保存便签的方法，包括新建便签和已有便签的保存 | 判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库 |
| existInDatabase() | / | 判断便签是否已存在于数据库中 | 判断NoteId是否存在 |
| isWorthSaving() | / | 判断便签是否值得保存 | 考虑是否已删除、是否存在文本内容和是否有本地修改 |
| setOnSettingStatusChangedListener(NoteSettingChangedListener l) | 便签属性改变监听器 | 设置便签设置状态变化的监听器 | / |
| setAlertDate(long date, boolean set) | date：提醒日期的时间戳；set：是否设置提醒 | 设置提醒日期 | 如果提醒日期不同于当前日期，更新提醒日期并通知监听器 |
| markDeleted(boolean mark) | mark：是否标记为已删除 | 标记便签是否已删除 | 标记便签是否已删除，同时通知监听器 |
| setBgColorId(int id) | id：背景颜色的ID | 设置背景颜色的ID | 如果背景颜色ID不同于当前ID，更新背景颜色ID并通知监听器 |
| setCheckListMode(int mode) | mode：模式 | 设置便签的清单模式 | 如果模式不同于当前模式，更新模式并通知监听器 |
| setWidgetType(int type) | type：小部件类型 | 设置小部件的类型 | 确定widget类型是否变化，变化了就更新widget参数 |
| setWidgetId(int id) | id：小部件的ID | 设置小部件的ID | 确定widget的id是否变化，更新widget |
| setWorkingText(String text) | text：文本内容 | 设置便签的文本内容 | 如果输入内容和原文本不一致则进行内容更新 |
| convertToCallNote(String phoneNumber, long callDate) | phoneNumber：通话记录的电话号码；callDate：通话日期的时间戳 | 将便签转换为通话记录便签 | 将传入的电话号码设置为电话数据 |
| hasClockAlert() | / | 判断便签是否有提醒设置 | / |
| **getContent()** | **/** | **获取便签的文本内容** | **/** |
| getAlertDate() | / | 获取便签的提醒日期 | / |
| getModifiedDate() | / | 获取便签的最后修改日期 | / |
| getBgColorResId() | / | 获取背景颜色的资源ID | / |
| getBgColorId() | / | 获取背景颜色的ID | / |
| getTitleBgResId() | / | 获取标题栏背景颜色的资源ID | / |
| getCheckListMode() | / | 获取便签的清单模式 | / |
| getNoteId() | / | 获取便签的ID | / |
| getFolderId() | / | 获取便签所属的文件夹ID | / |
| getWidgetId() | / | 获取小部件的ID | / |
| getWidgetType() | / | 获取小部件的类型 | / |
| NoteSettingChangedListener | / | 内部接口，定义了便签设置状态变化的监听器方法 | / |

### 17.3.4.数据层相关的类

**17.3.4.1、类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants.NOTE | 表示文本便签的MIME类型 |
| DataConstants.CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表便签或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表便签或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表便签的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与便签关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表便签的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表便签是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的便签数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本便签的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本便签的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本便签的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本便签的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本便签数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法

1. **功能十八：添加提醒**
   1. **功能介绍**

**功能概述：**对于特定便签可以选择增添提醒，使得对应时间的时候自动显示的闹钟图标，提醒用户当前时间的事项。

**操作步骤：**便签编辑界面选择“添加提醒”选项，弹出时间选择框，确定提醒时间后便签显示闹钟图标，标志提醒时间。

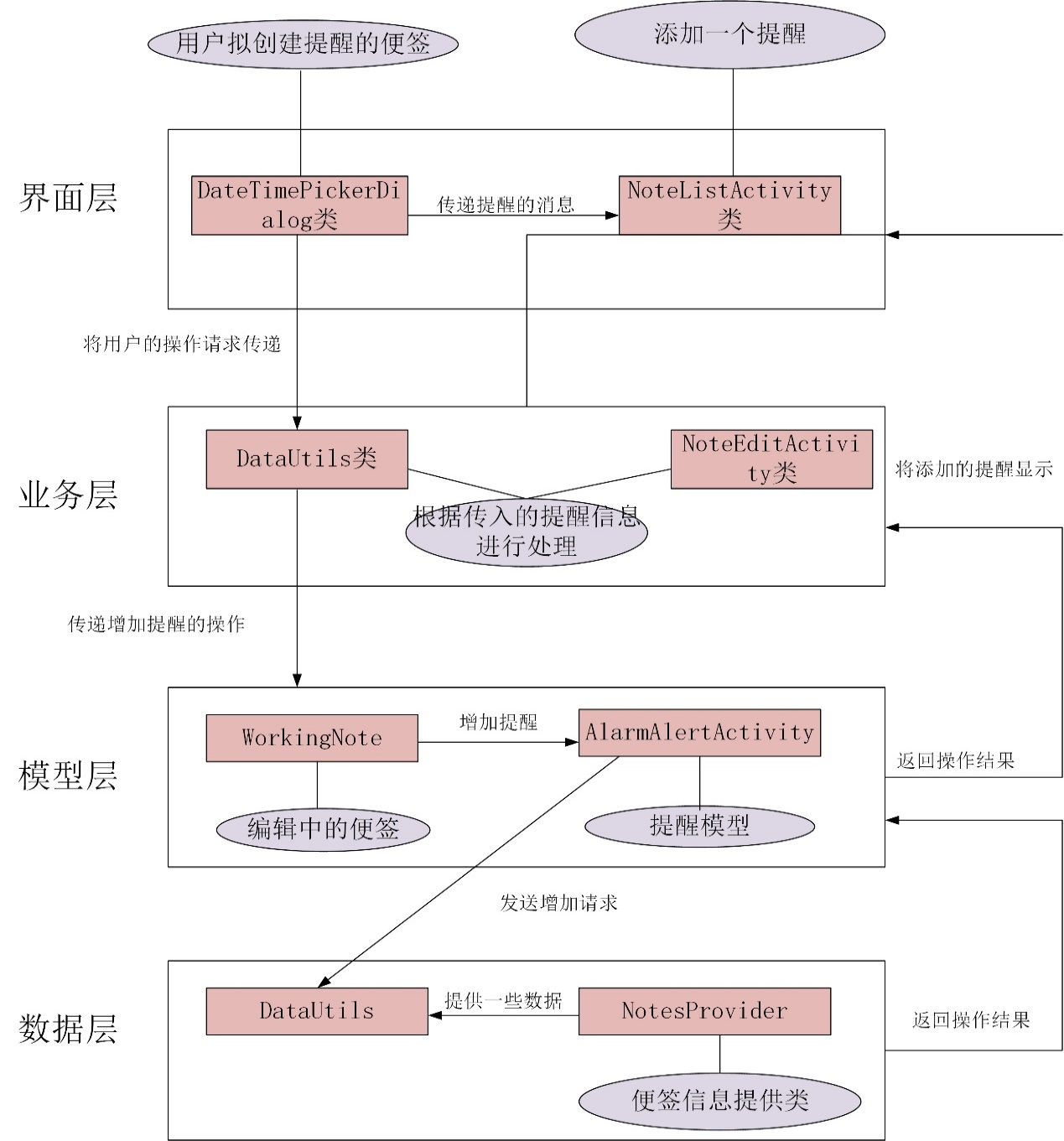
* 1. **业务请求与响应过程**

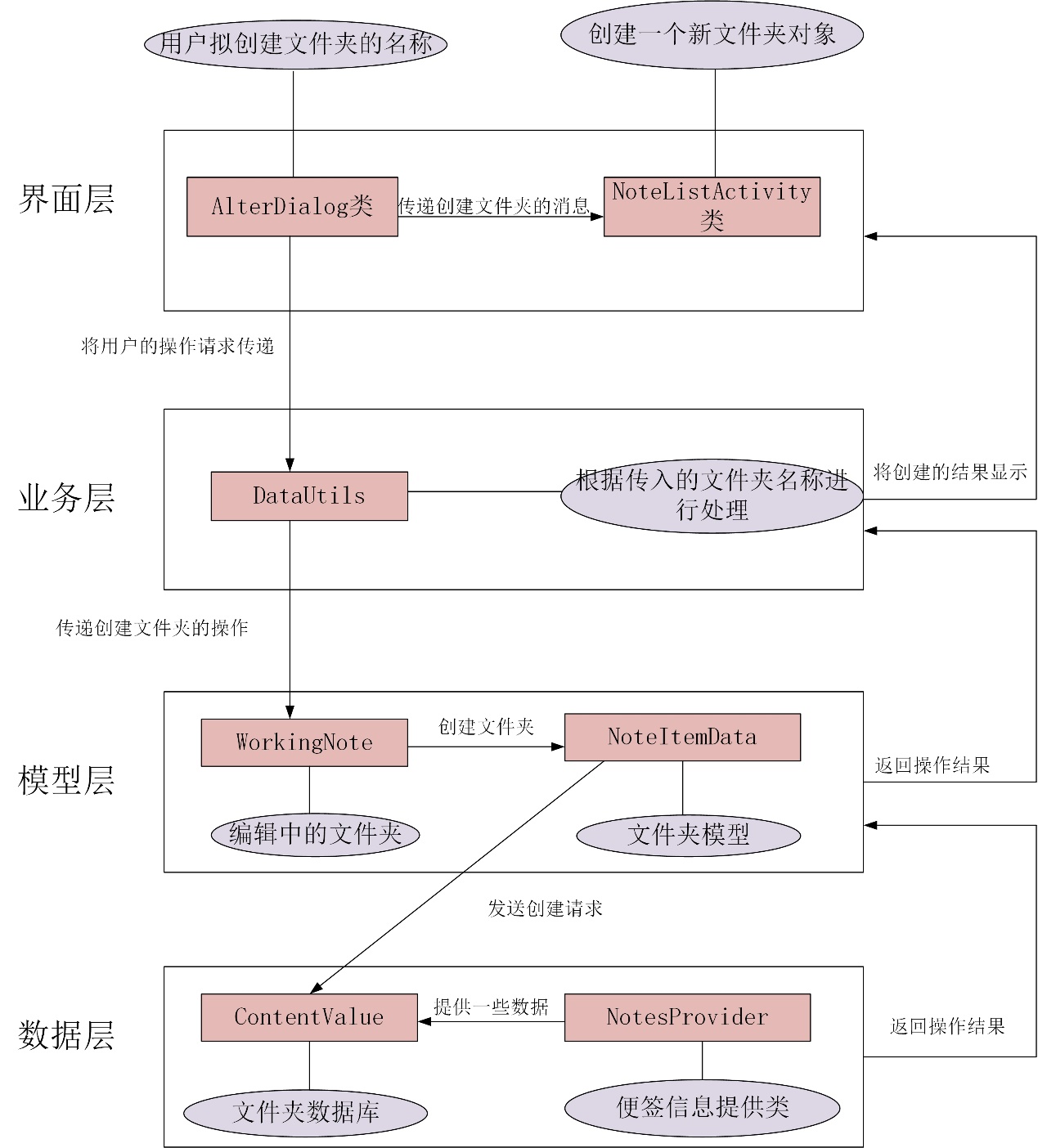
添加提醒功能（下简称“功能”）的界面层主要由包ui中DateTimePickerDialog类通过调用show()方法实现。show()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

功能的业务层由NoteEditActivity类中的setReminder()方法实现。首先，创建一个 DateTimePickerDialog 对象，传入当前系统时间作为初始时间。然后设置一个 OnDateTimeSetListener，用于监听用户设置日期和时间的操作。当用户设置完日期和时间后，调用 mWorkingNote.setAlertDate() 方法，将设置的日期和时间以及一个布尔值传递给该方法。最后显示 DateTimePickerDialog 对话框，供用户选择日期和时间

功能的模型层主要由ui包中的AlarmAlertActivity类来实现，主要包括onClick()和onDismiss()方法等。

功能的数据层主要由DataUtils类的对象来实现。包括了警报的信息，注释的内容、标题等等。





* 1. **类的作用**
     1. **界面层相关的类**
        1. **类名：****DateTimePickerDialogy**

1. **类的作用：**该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。
2. **类的精读：**

**DateTimePickerDialog**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mDate | 存储日期和时间 |
| mIs24HourView | 表示是否使用24小时制显示时间 |
| mOnDateTimeSetListener | 监听用户设置日期和时间的操作 |
| mDateTimePicker | 显示日期和时间选择器 |

**DateTimePickerDialog**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| set24HourView(boolean is24HourView) | is24HourView:判断是否使用24小时制显示时间 | 设置是否使用24小时制显示时间 | 将接收到的布尔值参数 is24HourView 赋值给成员变量 mIs24HourView，以便在更新对话框标题时使用。 |
| setOnDateTimeSetListener(OnDateTimeSetListener callBack) | callBack：回调监听器对象 | 设置日期时间设置完成后的回调监听器 | 将传入的回调监听器对象赋值给成员变量 mOnDateTimeSetListener，以便在用户设置完日期时间后触发相应的回调方法。 |
| updateTitle(long date) | Date:日期 | 根据传入的日期更新对话框的标题 | 定义一个标志变量 flag，用于指定日期和时间格式的显示方式。然后，将 mIs24HourView 是否为 true 的判断结果作为一个条件标志位添加到 flag 中。最后，调用 DateUtils 类的 formatDateTime() 方法，传入上下文、日期、标志等参数，生成格式化的日期时间字符串，并将其赋值给对话框的标题 |
| onClick(DialogInterface arg0, int arg1) | arg0：DialogInterface对象  arg1：参数 | 处理点击对话框按钮的事件。调用回调监听器的方法将设置的日期时间传递给回调方法。 | 调用回调监听器的 OnDateTimeSet() 方法，将当前选定的日期时间（通过 mDate.getTimeInMillis() 获取）作为参数传递给回调方法。 |
| onDateTimeChanged(DateTimePicker view, int year, int month, int dayOfMonth, int hourOfDay, int minute) | View：DateTimePicker对象  Year：年  Month：月  dayOfMonth：日期  hourOfDay：小时  minute：分钟 | 更新 mDate 的值，并调用 updateTitle() 方法更新对话框标题 | 更新了 mDate 对象的各个字段，以反映用户所选的日期和时间。然后，调用updateTitle() 方法，将更新后的日期设置为对话框的标题。 |

* + 1. **业务层相关的类**

**18.3.2.1、类名：NoteEditActivity**

1. **类的作用：**

保存用户的配置信息，如当前便签的修改字体大小和背景颜色的简单的配置信息。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| HeadViewHolder | 用于保存一组视图的引用，有修改日期文本，背景颜色，提醒图标，和提醒日期文本的视图。 |
| sBgSelectorBtnsMap | 保存背景颜色选择器中选项和对应颜色的映射关系 |
| sBgSelectorSelectionMap | 保存颜色id值与对应颜的映射关系 |
| mNoteBgColorSelectot | 背景颜色的选择器 |
| mWorkingNote | 当前操作的便签项 |
| mNoteEditorPanel | 操作背景颜色的修改 |
| mHeadViewPanel | 操作标题颜色的修改 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| dispactchTouchEvent() | ev:屏幕触碰的操作 | 颜色选择器和字体大小的选择器在屏幕上可见。同时用于在其他特殊操作下隐藏选择器 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| InRangeOfView(View view, MotionEvent ev) | view ：用来表示ui容器，容器的可触范围。  ev：屏幕触碰操作 | 用来确定触碰操作是否在给定的view范围内 | 先确定容器的范围location，在用ev的getX与getY操作判断操作点是否位于location内 |
| InitResource() | 无 | 初始化UI界面的各种元素和资源 | 通过给定id，R.id类的内容来赋值 |
| onBackPressed() | 无 | 确保用户按下返回按钮时，未保存的便签内容可以得到保存 | 先检查用户是否处于其他设置状态，否则则直接调用saveNote()的方法保存内容。同时调用super.onBackPressed方法来执行默认返回操作。 |
| clearSettingState() | 无 | 如果需要时，保证颜色选择器和字体大小选择器在不使用的时候是隐藏的 | 用判断结构，如果当前选择器为可见，则设置为不可见。 |
| onClick(View v) | v:菜单选择器内对应向的id值 | 用来实现用户点击与界面中视图的交互，根据不同的点击内容，执行不同的点击结果。例如修改字体的大小和修改背景颜色 | View.OnclickListener的接口来监听点击事件，在onClick方法中实现对应操作 |
| onOptionsItemSelected(MenuI item) | Item;菜单点击事件 | 根据不同的菜单点击事件，使菜单可见，或执行当前菜单内对应的功能 | 通过获得item的事件id，来匹配不同的菜单内的功能，然后再执行对应功能的内容。 |
| InitNoteScreen() | 无 | 加载设置的字体大小样式等样式个性化内容到屏幕上 | 调用mNoteEditor中的setTextAppearance的方法来实现。 |
| onBackgroundColorChanged() | 无 | 修改当前的背景颜色 | 通过id值来修改对应的背景颜色 |
| setReminder() | 无 | 设置提醒 | 创建一个 DateTimePickerDialog 对象，传入当前系统时间作为初始时间。然后设置一个 OnDateTimeSetListener 监听器，在用户设置完日期和时间后调用 mWorkingNote.setAlertDate() 方法来设置提醒日期和时间。 |
| sendTo(Context context, String info) | context ：上下文对象  info ：要分享的便签内容 | 分享便签内容给支持 Intent.ACTION\_SEND 动作和 text/plain 类型的应用程序 | 创建一个发送文本的意图，并将便签内容设置为 EXTRA\_TEXT 的额外数据，然后启动适合的应用程序 |
| createNewNote() | 无 | 用于创建新的笔记 | 保存当前正在编辑的笔记；然后，启动一个新的 NoteEditActivity 来创建一个新的笔记。 |
| deleteCurrentNote() | 无 | 删除当前笔记 | 删除当前笔记。如果笔记存在于数据库中，它会将笔记标记为已删除，并根据同步模式批量删除或移动到垃圾箱文件夹 |
| onClockAlertChanged(long date, boolean set) | date ：提醒日期和时间  set ：是否设置提醒 | 设置或取消闹钟提醒 | 根据是否设置提醒，创建一个隐式意图，关联到 AlarmReceiver 类，并使用笔记的 URI 数据。然后创建一个 PendingIntent 对象，并使用 AlarmManager 设置闹钟提醒。如果没有输入任何内容（即没有笔记 ID），则显示错误消息提醒用户输入内容 |
| isSyncMode() | 无 | 检查是否处于同步模式 | 检查是否处于同步模式。它根据用户在设置中配置的同步账号名称来判断。 |

* + 1. **模型层相关的类**

**18.3.3.1、类名：AlarmAlertActivity**

1. **类的作用：**

该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。

1. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **数据层相关的类**

**18.3.4.1、类名：**DataUtils

1. **类的作用**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

1. **功能十九：关闭提醒**
   1. **功能介绍**

**功能概述：**对于特定便签可以选择关闭提醒，使得对应时间的时候自动显示的闹钟图标被关闭，不再提醒用户当前时间的事项。

**操作步骤：**提醒时间到后，操作系统弹出对话框显示便签内容并响铃，闹钟变为“已过期”。用户点击关闭提醒后，闹钟不再响起。

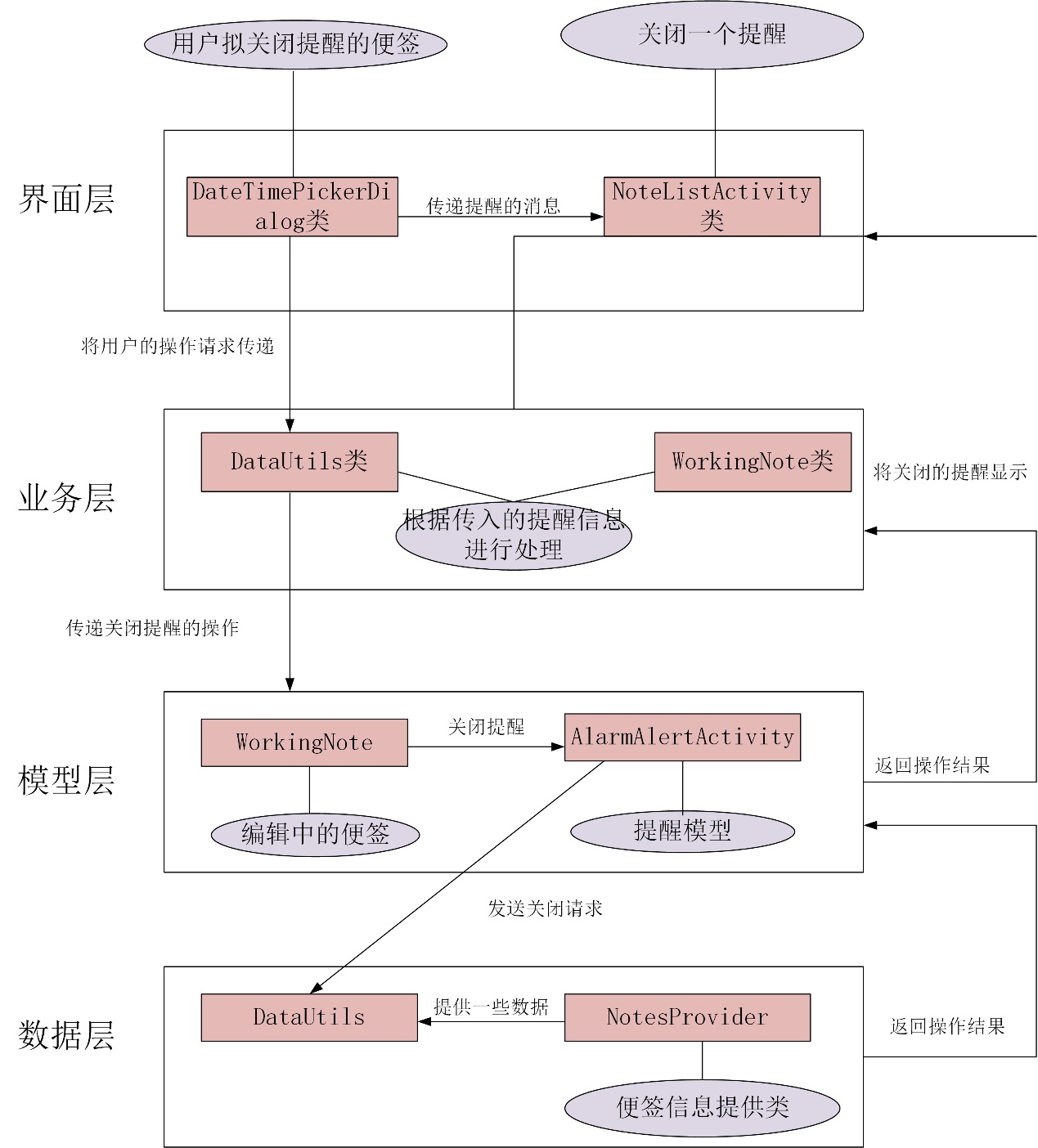
* 1. **业务请求与响应过程**

关闭提醒功能（下简称“功能”）的界面层主要由包ui中AlarmAlertActivity类通过调用showActionDialog()方法实现。showActionDialog()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

功能的业务层由WorkingNote类中的setAlertDate()方法实现。首先，通过比较传入的日期参数(date)与当前的警报日期(mAlertDate)，判断是否需要进行更新。接着，将更新后的 mAlertDate 的值以字符串形式存储到相应的注释字段。如果存在 mNoteSettingStatusListener 对象（即监听器对象），则调用 onClockAlertChanged()方法，将新的日期和设置状态作为参数传递给监听器。这样做是为了通知外部组件或模块，在警报日期发生改变时可以执行相关操作。

功能的模型层主要由ui包中的AlarmAlertActivity类来实现，主要包括onClick()和onDismiss()方法等。

功能的数据层主要由DataUtils类的对象来实现。包括了警报的信息，注释的内容、标题等等。



* 1. **类的作用**
     1. **界面层相关的类**
        1. **类名：AlarmAlertActivity**

1. **类的作用：**该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。
2. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **业务层相关的类**

**19.3.2.1、类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 标记当前便签状态 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |
| NoteSettingChangedListener | 定义便签设置改变监听器的接口 |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getTitleBgResId() | 无 | 返回当前的标题颜色的ID值 | 通过背景颜色ID值来调用NoteBgResource内的方法来匹配返回标题的ID值 |
| WorkingNote(Context context, long folderId) | context:Android应用程序的上下文。  folderId:笔记所属文件夹的ID。 | 创建一个新的笔记对象 | 实例化对象 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getBgColorResId(): | 无 | 获取笔记背景颜色资源ID | Return直接返回 |
| getCheckListMode() | 无 | 获取笔记模式 | Return直接返回 |
| getModifiedDate() | 无 | 获取笔记修改日期 | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getContent() | 无 | 获取笔记内容 | Return直接返回 |
| getNoteId() | 无 | 获取笔记ID | Return直接返回 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| getTitleBgResId() | 无 | 获取笔记标题背景资源ID | Return直接返回 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context: Android 应用程序的上下文。  folderId: 笔记所属文件夹的ID。  widgetId: 笔记对应的小部件的ID。  widgetType: 笔记对应的小部件类型。  defaultBgColorId: 笔记的默认背景颜色ID。 | 创建一个空白笔记 | 通过调用一系列set方法，设置笔记的初始情况 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| isWorthSaving() | 无 | 判断当前注释是否值得保存 | 如果注释已被标记为删除(mIsDeleted)，或者注释既不在数据库中也没有内容且为空(TextUtils.isEmpty(mContent))，或者注释存在于数据库中但未进行本地修改(!mNote.isLocalModified())，则返回false；否则，返回true。 |
| **setAlertDate(long date, boolean set)** | **date：警报日期。**  **set：指示是否要设置警报日期。** | **设置警报日期并更新相关字段的值。** | **如果传入的日期与当前警报日期不相等，则更新警报日期(mAlertDate)，并将警报日期存储到注释字段中。**  **如果存在警报设置状态更改监听器(mNoteSettingStatusListener)，则通知外部组件警报日期的更改。** |
| markDeleted(boolean mark) | mark：指示是否标记为删除 | 将注释标记为已删除或未删除 | 根据传入的标记值(mark)更新mIsDeleted成员变量，表示是否标记为删除。  如果存在小部件ID和小部件类型，并且警报设置状态更改监听器不为空，则通知外部组件小部件的更改 |

* + 1. **模型层相关的类**

**19.3.3.1、类名：AlarmAlertActivity**

1. **类的作用：**该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。
2. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **数据层相关的类**

**19.3.4.1、类名：**DataUtils

1. **类的作用**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

1. **类的精读：**

类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |

1. **收获和问题**

## 20.1. 收获

（1）通过这两次的精读，仔细分析了小米便签的源码，逐渐厘清软件工程的四层架构，分别搞清楚了不同层所承担的不同任务，对于Android开发有了一定的了解，对于功能的实现机理有了详细的了解，同时不同层次间的相互调用关系也能够比较清楚的分析出来，学会了绘制UML图，学会了在软件开发过程中各种文档的撰写。

（2）在这两次的精读中，从业务的处理请求和响应过程的思路着手，分析四个层次结构在不同功能中的作用，在源码的阅读中，逐渐清楚了各个层次自己的“位置”所在，对他们的详细作用也有了一定的研究和理解，在绘制调用关系图时，对小米便签整体的类间的协同也有了更加深刻的认识。学习源码是一个复杂的过程，从泛读到精读的过程中，也逐渐明晰了面对复杂且大型的工程时，该怎么做才能庖丁解牛一般快速切入，不至于毫无头绪焦头烂额。

## 20.2. 存在的问题

1. 某些方法的复杂度太高了，嵌套了多层if-else，且有些方法的调用很复杂，类间关系非常模糊。
2. GoogleTask功能没有用。
3. 年代久远造成一些方法已被弃用，版本不适配。
4. 有些功能雷同的类会在不同包中反复出现，如果能够将这些类进行修改，让他们能够兼容，会让复杂的结构更简洁。
5. 很多方法已经被更高级的方法取代。