

# “小米便签” 精读笔记

项目组长：计科 2101 张文睿

组员 1：计科 2102 吴锦洋

组员 2：大数据 2102 吴佳瑞

组员 3：大数据 2102 凌国栩

目录

小组分工 ..... 6

    张文睿 ..... 6

    吴锦洋 ..... 6

    吴佳瑞 ..... 6

    凌国栩 ..... 6

功能一：新建便签 ..... 7

    功能介绍 ..... 7

    业务请求与响应过程 ..... 7

    类的作用 ..... 8

        界面层相关的类 ..... 8

        业务层相关的类 ..... 11

        模型层相关的类 ..... 12

        数据层相关的类 ..... 14

功能二：删除便签 ..... 15

    功能介绍 ..... 15

    业务请求与响应过程 ..... 15

    类的作用 ..... 16

        界面层相关的类 ..... 16

        业务层相关的类 ..... 19

        模型层相关的类 ..... 20

        数据层相关的类 ..... 22

功能三：移动便签 ..... 23

    功能介绍 ..... 23

    业务请求与响应过程 ..... 24

    类的作用 ..... 25

        界面层相关的类 ..... 25

        业务层相关的类 ..... 26

        模型层相关的类 ..... 28

        数据层相关的类 ..... 30

功能四：新建文件夹 ..... 32

功能介绍 .....	32
业务请求与响应过程 .....	32
类的作用 .....	33
界面层相关的类 .....	33
业务层相关的类 .....	38
模型层相关的类 .....	40
数据层相关的类 .....	44
功能五：查看文件夹 .....	45
功能介绍 .....	45
业务请求与响应过程 .....	45
类的作用 .....	46
界面层相关的类 .....	46
业务层相关的类 .....	48
模型层相关的类 .....	53
数据层相关的类 .....	55
功能六：删除文件夹 .....	56
功能介绍 .....	56
业务请求与响应过程 .....	57
类的作用 .....	58
界面层相关的类 .....	58
业务层相关的类 .....	60
模型层相关的类 .....	65
数据层相关的类 .....	66
功能七：修改文件夹名称 .....	68
功能介绍 .....	68
业务请求与响应过程 .....	69
类的作用 .....	70
界面层相关的类 .....	70
业务层相关的类 .....	75
模型层相关的类 .....	77
数据层相关的类 .....	81
功能八：导出便签 .....	82

功能介绍 .....	82
业务请求与响应过程 .....	83
类的作用 .....	85
界面层相关的类 .....	85
业务层相关的类 .....	90
模型层相关的类 .....	92
数据层相关的类 .....	97
功能九：关键字查询 .....	99
功能介绍 .....	99
业务请求与响应过程 .....	99
类的作用 .....	100
界面层相关的类 .....	100
业务层/模型相关的类：Activity .....	101
数据层相关的类 .....	101
功能十：修改字体大小 .....	103
功能介绍 .....	103
业务请求与响应过程 .....	103
类的作用 .....	104
界面层/业务层相关的类 .....	104
模型层相关的类 .....	106
数据层相关的类 .....	107
功能十一：修改背景颜色 .....	108
功能介绍 .....	108
业务请求与响应过程 .....	108
类的作用 .....	109
界面层/业务层相关的类 .....	109
模型层相关的类 .....	110
数据层相关的类 .....	111
功能十二：清单模式 .....	112
功能介绍 .....	112
业务请求与响应过程 .....	113
类的作用 .....	114

界面层相关的类.....	114
模型层相关的类.....	120
数据层相关的类.....	125
功能十三：发送到桌面 .....	127
功能介绍.....	127
业务请求与响应过程 .....	127
类的作用 .....	129
界面层相关的类.....	129
模型层相关的类.....	129
数据层相关的类.....	133
功能十四：自动保存便签.....	134
功能介绍.....	134
业务请求与响应过程 .....	135
类的作用 .....	136
界面层相关的类.....	136
模型层相关的类.....	137
数据层相关的类.....	140

# 小组分工

## 张文睿

精读分析以下功能：

新建文件夹

查看文件夹

删除文件夹

修改文件夹名称

## 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

## 吴佳瑞

新建便签

删除便签

移动便签

## 凌国栩

修改字体大小

修改背景颜色

关键词搜索

# 功能一：新建便签

## 功能介绍

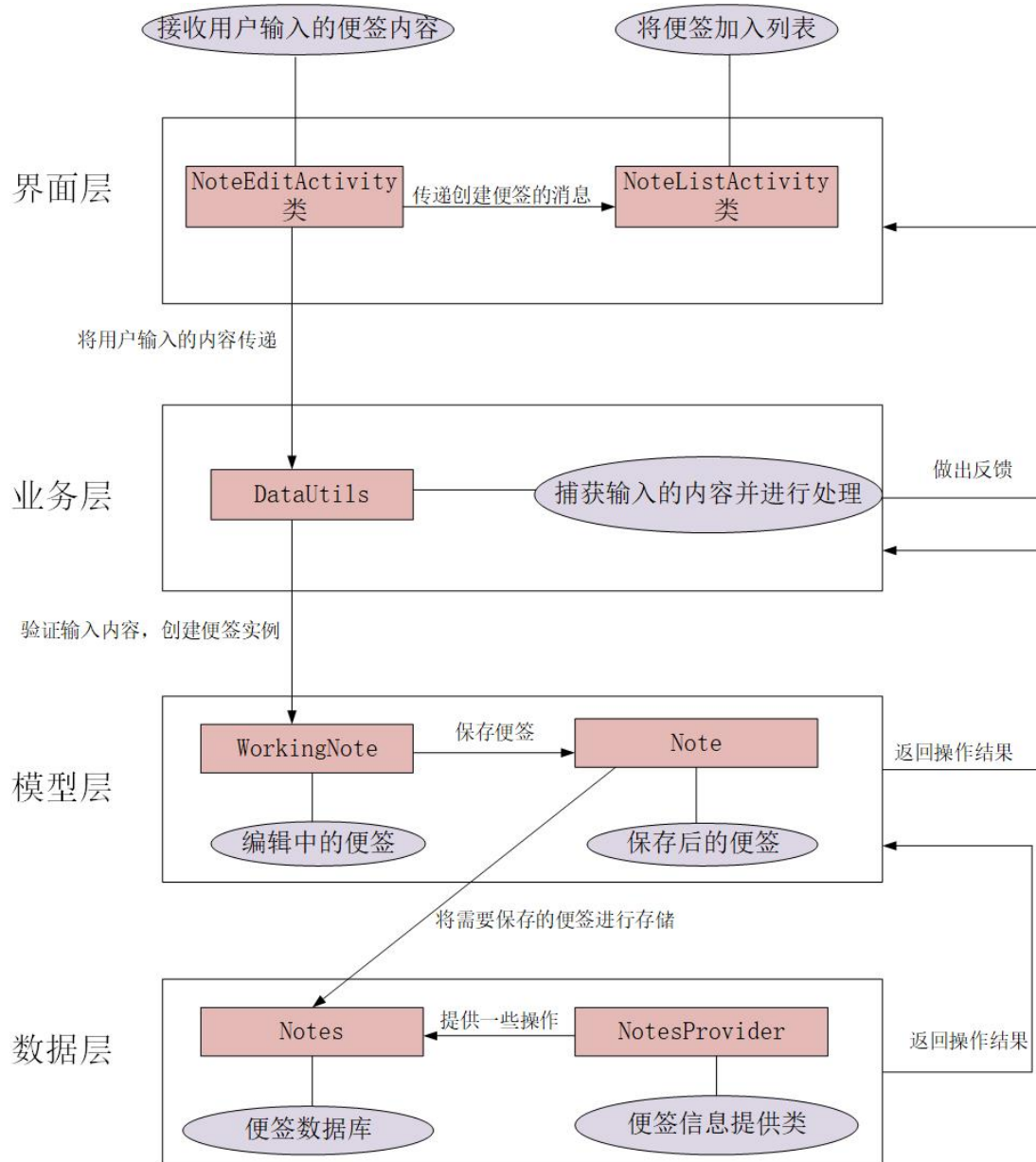
**功能概述：**新建一条便签并存储到本地数据库

**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

## 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。



## 类的作用

### 界面层相关的类

#### 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

(2) 类的精读:

类的成员

成员	说明
HeaderViewHolder	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	<p>super.onCreate (savedInstanceState): 调用父类的 onCreate 方法以执行基本的活动创建操作。</p> <p>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。</p>	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	<p>mWorkingNote: 便签保存在数据库中的形式</p> <p>outState: 用于存放已经被删除的便签名称</p>	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.getNotelId()) 保存到 outState 中，以便在稍后的状态恢复操作中使用。
onClick	<p>View v: 需要进行鼠标点击操作的视图</p> <p>Id: 视图名称</p>	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。

## 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。

### (2) 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	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

	对 象 ， 将 NoteEditActivity 类 作 为 目 标 Activity;		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。

## 业务层相关的类

### 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

#### （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。

# 模型层相关的类

## 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

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	用于判断是否值得保存，这可以和 saveNote 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

	存在，并且是否有本地修改		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。
--	--------------	--	---

## 数据层相关的类

### 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

**(2) 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

### 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

## 功能二：删除便签

### 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

### 业务请求与响应过程

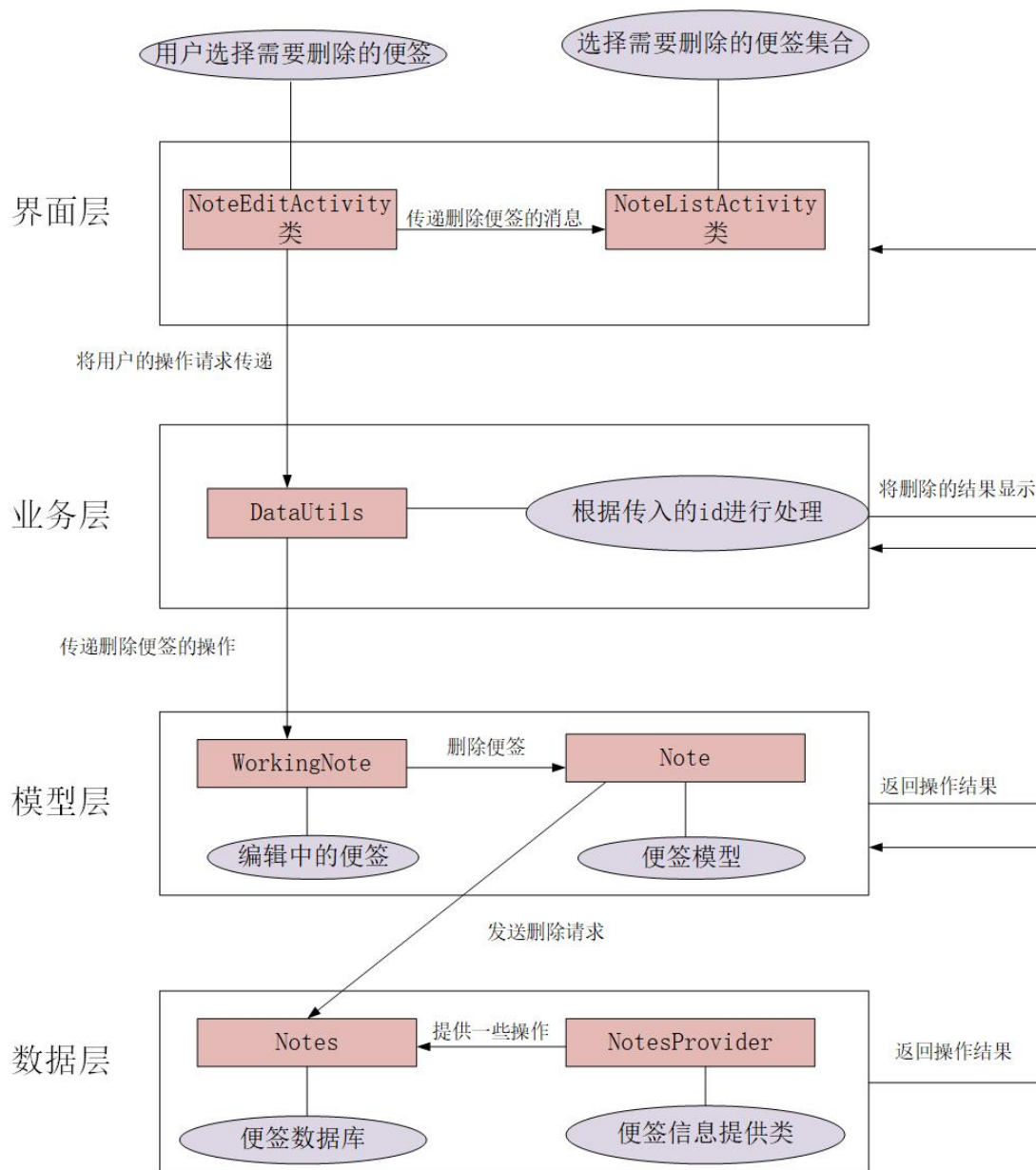
在删除便签的过程中，四个层级各自的作用大致如下：

**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。



## 类的作用

### 界面层相关的类

#### 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

(2) 类的精读:

类的成员

成员	说明
HeaderViewHolder	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	<p>super.onCreate (savedInstanceState): 调用父类的 onCreate 方法以执行基本的活动创建操作。</p> <p>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。</p>	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	<p>mWorkingNote: 便签保存在数据库中的形式</p> <p>outState: 用于存放已经被删除的便签名称</p>	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.getNoteld()) 保存到 outState 中，以便在稍后的状态恢复操作中使用。
onClick	<p>View v: 需要进行鼠标点击操作的视图</p> <p>Id: 视图名称</p>	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	在函数中，会根据条件判断来决

		事件的分发	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

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。
- (2) 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。

## 业务层相关的类

### 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

#### (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

			<p>的数量。</p> <p>如果游标不为空（则移动游标到第一行</p> <p>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</p> <p>返回获取到的文件夹数量。</p>
batchDeleteNotes	<p>HashSet&lt;Long&gt;: id 列表</p> <p>ContentResolver: 一个内容解析器</p> <p>ContentUris.</p> <p>withAppendedId(): 用于构建便签的特定 URI。</p> <p>operationList 操作列表</p>	根据传入的便签 ID 集合进行批量删除操作	<p>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。</p> <p>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。</p> <p>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。</p> <p>如果删除操作成功，则返回 true，否则返回 false。</p>

## 模型层相关的类

### 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
getNewNoteld	ContentValues: 用于存储便签	获取新便签的	从数据库中获取一个新

	的各个属性和值 <b>Context:</b> 上下文参数 <b>folderId:</b> 文件夹名称	ID	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。
<b>Note</b>	<b>ContentValues:</b> 用于存储便签的各个属性和值	<b>Note</b> 类的构造函数。它用于初始化一个 <b>Note</b> 对象。	这是一个无参数的构造函数，用于创建 <b>Note</b> 对象。
<b>getNewNoteId</b>	<b>Context:</b> 上下文参数 <b>folderId:</b> 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 <b>noteId</b>
<b>setNoteValue</b>	<b>Key:</b> 需要被设置的属性 <b>Value:</b> 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 <b>key</b> 和 <b>value</b> ，将指定属性的新值添加到 <b>mNoteDiffValues</b> 中

## 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
<b>WorkingNote</b>	WorkingNote 的一些参数需要	私有构造方法	私有构造方法
<b>createEmptyNote</b>	<b>Context:</b> 上下文参数 <b>folderId:</b> 文件夹名称 <b>widgetId:</b> 插件 id	创建一个空的便签，并进行一些初始设置	创建了一个 <b>WorkingNote</b> 类的实例，这行代码设置了便签的背景颜色为默

	widgetType: 插件类型 defaultBgColorId: 用于颜色设置		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 <b>WorkingNote</b> 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 <b>mark</b> 赋值给 成员变量 <b>mIsDeleted</b> , 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 <b>naveNote</b> 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。

## 数据层相关的类

### 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

#### (2) 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 <b>final</b> 类型的值
TextNote、CallNote	对 <b>DataColumns</b> 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

## 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。

## 功能三：移动便签

### 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

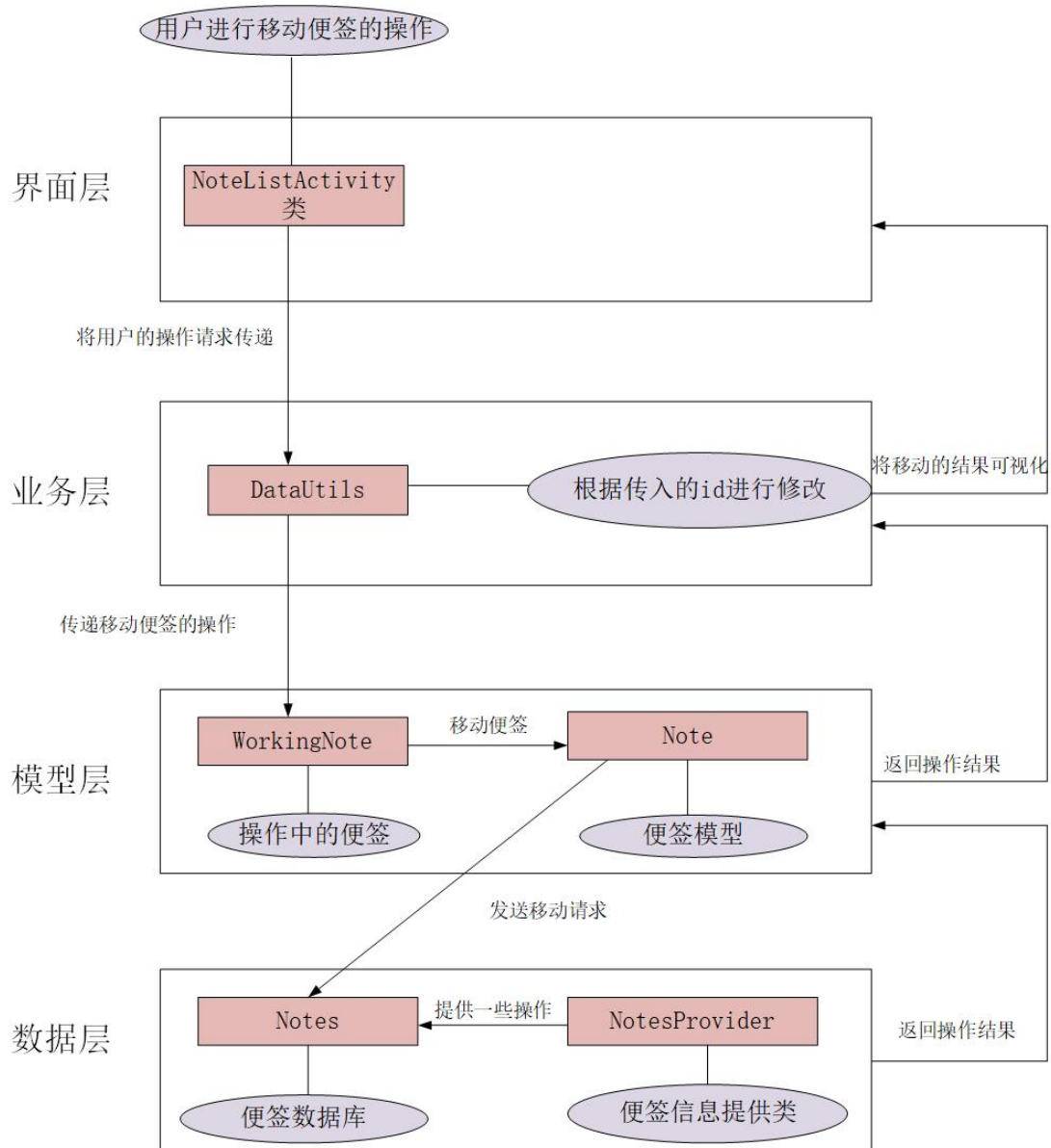
标签，点击左下方移动按钮，选择目标文件夹，就可以将快速创建的便签归类以进行有效管理

## 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。





## 类的作用

## 界面层相关的类

### 类名：NotesListActivity

#### (3) 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。

(4) 类的精读：

表 1 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

表 2 类中的方法

方法	参数解释	方法作用	实现过程
startAsyncNotesListQuery	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	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

业务层相关的类

类名：DataUtils

(3) 类的作用：

(4) 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

(5) 类的精读：

表 1 类的成员

成员	说明
TAG	public static final String TAG = "DataUtils";

表 2 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

			<p>中。</p> <p>使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。</p> <p>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。</p> <p>如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。</p>
--	--	--	---

## 模型层相关的类

### 类名：Note

(3) 类的作用：单个便签的模型

(4) 类的精读：

表 1 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";

表 2 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

### 类名: **WorkingNote**

(3) 类的作用: 正在编辑的便签的模型

(4) 类的精读:

表 1 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

表 2 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	<p>mIsDeleted: 标记了删除操作</p> <p>existInDatabase</p> <p>TextUtils.isEmpty</p> <p>: 判断了当前便签是否存在于数据库中并且内容是否为空</p> <p>existInDatabase()</p> <p>mNote.isLocalModified()</p> <p>判断当前便签是否在数据库中</p> <p>存在, 并且是否有本地修改</p>	用于判断是否值得保存, 这可以和 naveNote 联动	<p>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。</p> <p>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。</p> <p>判断当前便签是否在数据库中</p> <p>存在, 并且是否有本地修改。如果便签在数据库中</p> <p>存在但没有本地修改, 则可能认为不值得保存。</p>
loadNote	mContext: 目录	加载便签数据	<p>通过提供的便签 ID 查询便签的数据。</p> <p>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。</p>
loadNoteData	mContext: 目录	查询与特定 ID 相关的数据。	<p>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。</p> <p>根据不同的数据类型, 进行相应的操作</p>

## 数据层相关的类

### 类名: Notes

#### (3) 类的作用:

便签数据库, 用于记录便签相关属性和数据

#### (4) 类的精读:

表 1 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 DataColumnns 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

### 类名：NotesProvider

(3) 类的作用：便签信息提供类

(4) 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。

# 功能四：新建文件夹

## 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。

## 业务请求与响应过程

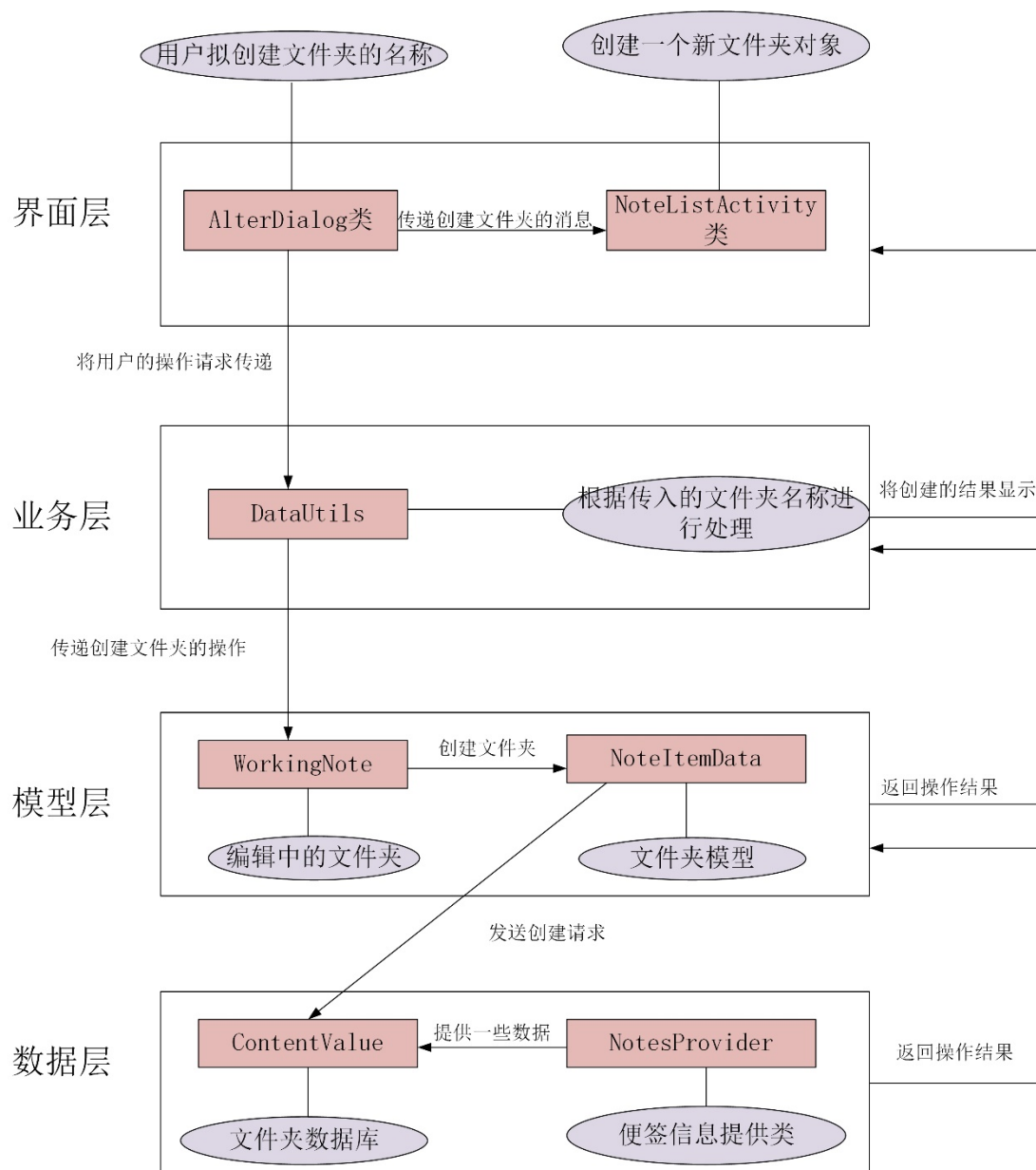
新建文件夹功能（下简称“功能”）的界面层主要由包 `ui` 中 `NotesListActivity` 类通过调用 `showCreateOrModifyFolderDialog()` 方法实现。`showCreateOrModifyFolderDialog()` 中使用 `AlertDialog.Builder` 构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

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。





## 类的作用

## 界面层相关的类

### 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。

#### (2) 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 <code>ListView</code> 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 <code>ContentResolver</code> 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
<code>onCreate(Bundle savedInstanceState)</code>	<code>savedInstanceState</code> 是一个 <code>Bundle</code> 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 <code>initResources()</code> 方法初始化资源	首先调用了父类 <code>onCreate</code> 方法，然后通过 <code>setContentView</code> 方法设置了布局为 <code>R.layout.note_list</code>
<code>onActivityResult(int requestCode, int resultCode, Intent data)</code>	<code>requestCode</code> 是请求码，用于标识请求的来源	用于处理从其他 Activity 返回的结果	处理 Activity 返回结果的方法，当其他

	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	果，例如打开便签或创建新便签后的操作	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 and 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.set CheckedItem()
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 mBackgroundQuery Handler 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

## 业务层相关的类

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	<p><b>resolver:</b> 用于与应用程序的数据提供者进行通信。</p> <p><b>id:</b> 要移动的便签的唯一标识符。</p> <p><b>srcFolderId:</b> 便签当前所在的文件夹的唯一标识符。</p> <p><b>desFolderId:</b> 要移动到的目标文件夹的唯一标识符。</p>	将指定的便签移动到不同的文件夹中	<p>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。</p> <p>将更新后的 <b>values</b> 对象应用于指定的便签。指定要更新的便签的位置。</p>
batchMoveToFolder	<p><b>resolver:</b> 用于交互。</p> <p><b>ids:</b> 包含要移动的便签的 ID。</p> <p><b>folderId:</b> 要移动到的目标文件夹的 ID。</p>	将一组便签移动到指定文件夹	<p>对于 <b>ids</b> 中的每个便签 ID，创建一个 <b>Operation</b> 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。</p> <p>将创建的 <b>Operation</b> 对象添加到 <b>operationList</b> 中。</p> <p>使用 <b>resolver</b> 对象调用 <b>applyBatch()</b> 方法，将 <b>operationList</b> 中的操作应用到 <b>ContentProvider</b> 中。</p> <p>根据返回的结果进行判断：如果 <b>results</b> 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 <b>false</b>。</p> <p>如果没有抛出异常，则返回 <b>true</b>。否则，记录异常信息并返回 <b>false</b>。</p>

## 模型层相关的类

### 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。

(2) 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
NoteSettingChangeListener	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

## 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
mCreateDate	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

# 数据层相关的类

## 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

<b>putNull(String key)</b>	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
<b>get(String key):</b>	Key: 键	根据键获取对应的值	直接调用 HashMap 中的 get 方法
<b>containsKey(String key)</b>	Key: 键	检查是否包含指定键	直接调用 HashMap 中的 contains 方法
<b>remove(String key)</b>	Key: 键	根据键移除键值对	直接调用 HashMap 中的 remove 方法
<b>clear()</b>		清空 ContentValues 中的所有键值对。	直接调用 HashMap 中的 clear 方法

# 功能五：查看文件夹

## 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。

## 业务请求与响应过程

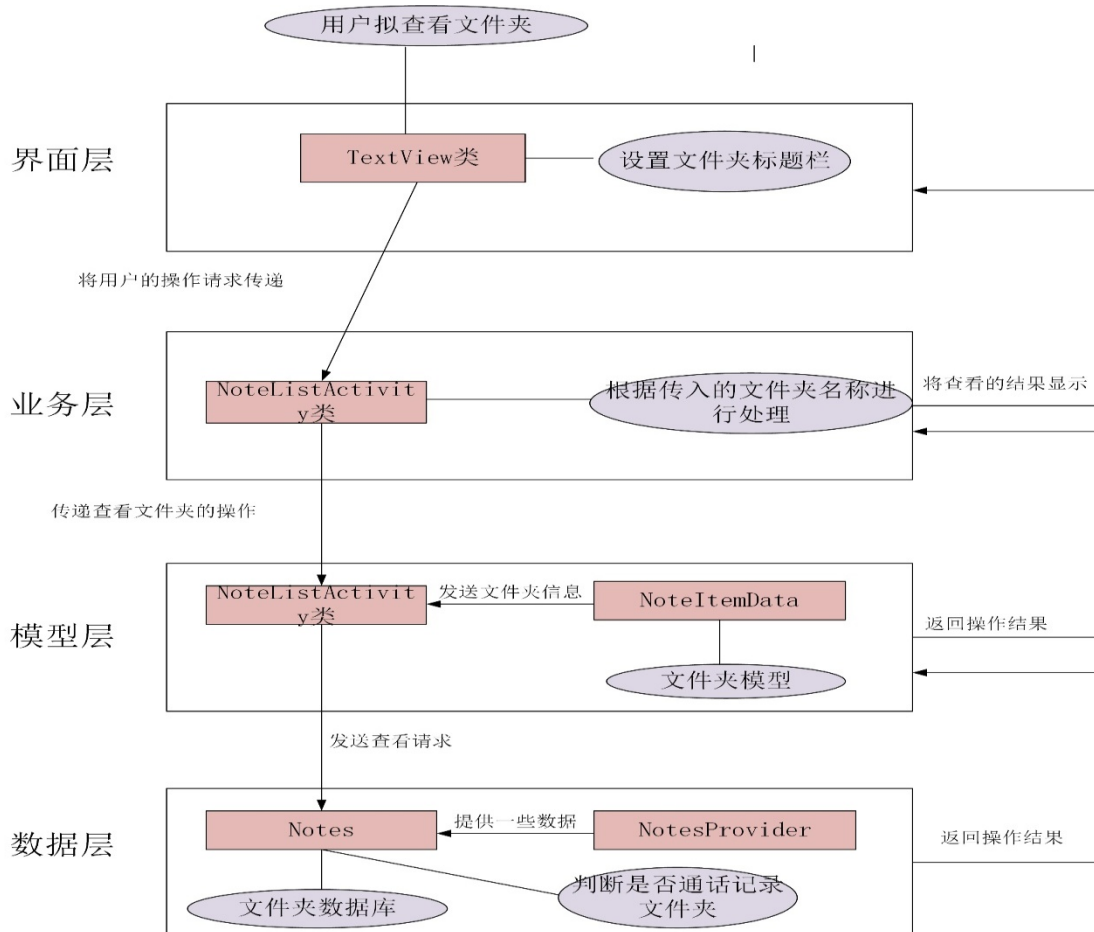
查看文件夹功能（下简称“功能”）的界面层主要由 TextView 类通过调用 setText()方法和 setVisibility()方法实现。setText()设置标题栏文本，而 setVisibility()方法设置标题栏可见。

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。



## 类的作用

### 界面层相关的类

#### 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。

##### (2) 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

## 业务层相关的类

### 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。

#### (2) 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	<p>首先获取 <b>SharedPreferences</b> 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 (<b>R.raw.introduction</b>) 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 <b>true</b>, 以避免重复插入</p>
onStart()	/	Activity 的生命周期方法, 在 Activity 即将可见时调用	调用 <b>startAsyncNotesListQuery()</b> 方法开始异步查询便签列表数据
initResources()	/	初始化 Activity 的资源 and 视图	<p>初始化了 <b>mContentResolver</b>、<b>mBackgroundQueryHandler</b>、<b>mCurrentFolderId</b> 等成员变量, 以及便签列表的视图 (<b>mNotesListView</b>) 和适配器 (<b>mNotesListAdapter</b>) 等</p>
updateMenu()	/	更新多选模式下的操作菜单	<b>selectedCount</b> 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)	<p><b>ActionMode mode:</b> 表示当前的多选模式操作栏; <b>Menu menu:</b> 表示多选操作栏的菜单</p>	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
<b>openFolder(NoteItemData data)</b>	<b>data 是一个 NoteItemData 对象, 表示要打开的便签数据</b>	<b>打开文件夹</b>	<b>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</b>
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 <b>BackupUtils</b> 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 ( <b>Notes.TYPE_FOLDER</b> ), 则将文件夹上下文菜单创建监听器设置为 <b>mFolderOnCreateContextMenuListener</b> 。这样,当用户点击该文件夹时,将弹出文件夹上下文菜单

## 模型层相关的类

### 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
mCreateDate	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
getCreateDate()	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	Return 直接返回

		间	
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

## 数据层相关的类

### 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。

#### （2）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_TEMPORARY_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_FOLDER	标识回收站文件夹常量
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

## 功能六：删除文件夹

### 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。



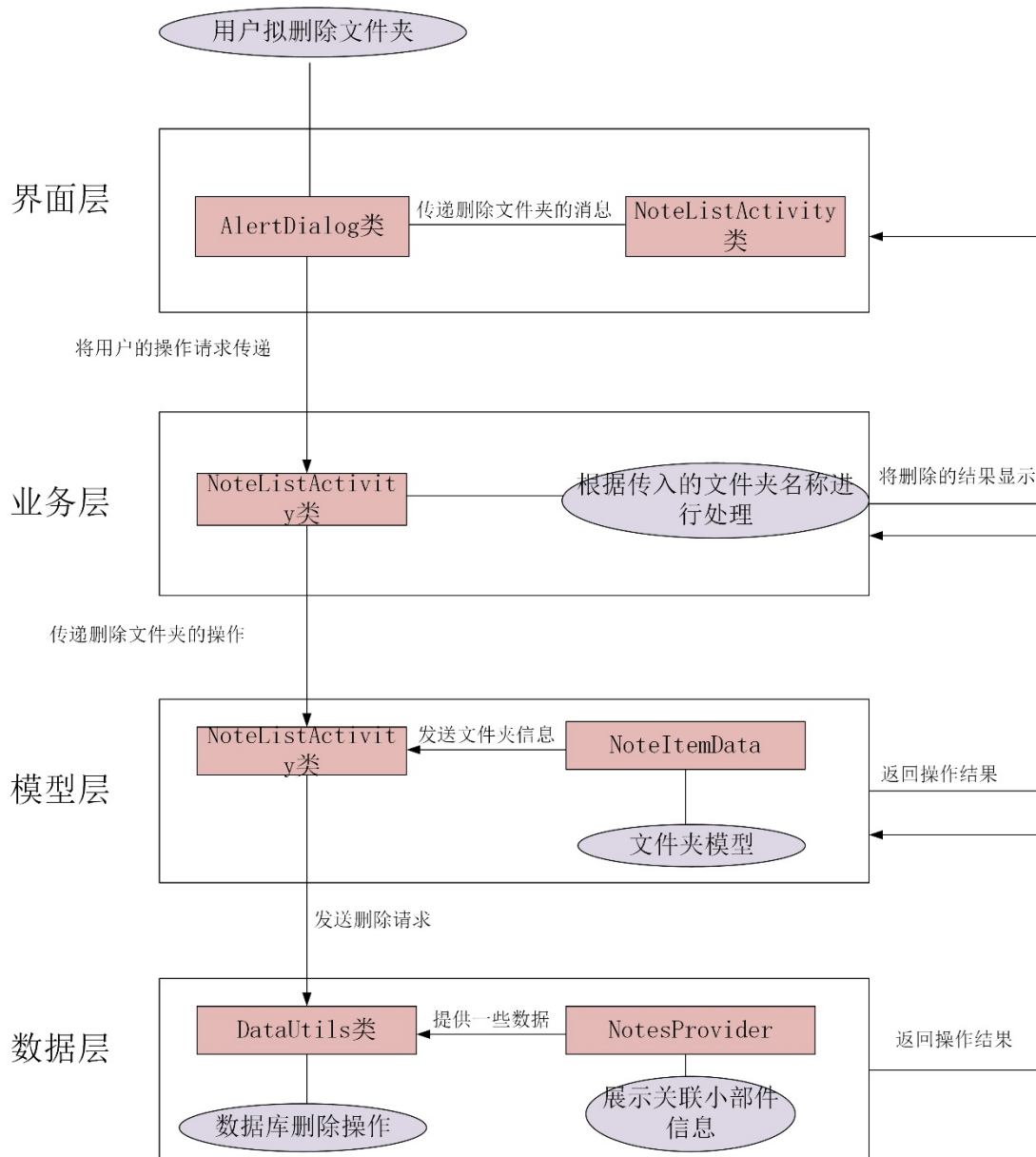
## 业务请求与响应过程

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。



## 类的作用

## 界面层相关的类

### 类名：AlertDialog

#### (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。

#### (2) 类的精读：

### AlertDialog 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

## 业务层相关的类

### 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。

#### （2） 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 and 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.set CheckedItem()
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 mBackgroundQuery Handler 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
<b>batchDelete()</b>	/	<b>批量删除便签或将它们移动到回收站</b>	<b>异步执行的任务, 根据同步模式, 直接删除便签</b>
<b>deleteFolder(long folderId)</b>	<b>folderId 表示要删除的文件夹的 ID</b>	<b>删除文件夹及其相关操作</b>	<b>根据同步模式, 直接删除文件夹</b>
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



		面	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

## 模型层相关的类

### 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
getCreateDate()	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

## 数据层相关的类

### 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。

(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。如果结果不为空或第一个结果不为空，表示删除失败。 如果删除操作成功，则返回 true，否则返回 false。

moveNoteToFoler	<p><b>resolver:</b> 用于与应用程序的数据提供者进行通信。</p> <p><b>id:</b> 要移动的便签的唯一标识符。</p> <p><b>srcFolderId:</b> 便签当前所在的文件夹的唯一标识符。</p> <p><b>desFolderId:</b> 要移动到的目标文件夹的唯一标识符。</p>	将指定的便签移动到不同的文件夹中	<p>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。</p> <p>将更新后的 <b>values</b> 对象应用于指定的便签。指定要更新的便签的位置。</p>
batchMoveToFolder	<p><b>resolver:</b> 用于交互。</p> <p><b>ids:</b> 包含要移动的便签的 ID。</p> <p><b>folderId:</b> 要移动到的目标文件夹的 ID。</p>	将一组便签移动到指定文件夹	<p>对于 <b>ids</b> 中的每个便签 ID，创建一个 <b>Operation</b> 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。</p> <p>将创建的 <b>Operation</b> 对象添加到 <b>operationList</b> 中。</p> <p>使用 <b>resolver</b> 对象调用 <b>applyBatch()</b> 方法，将 <b>operationList</b> 中的操作应用到 <b>ContentProvider</b> 中。</p> <p>根据返回的结果进行判断：如果 <b>results</b> 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 <b>false</b>。</p> <p>如果没有抛出异常，则返回 <b>true</b>。否则，记录异常信息并返回 <b>false</b>。</p>

## 功能七：修改文件夹名称

### 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。

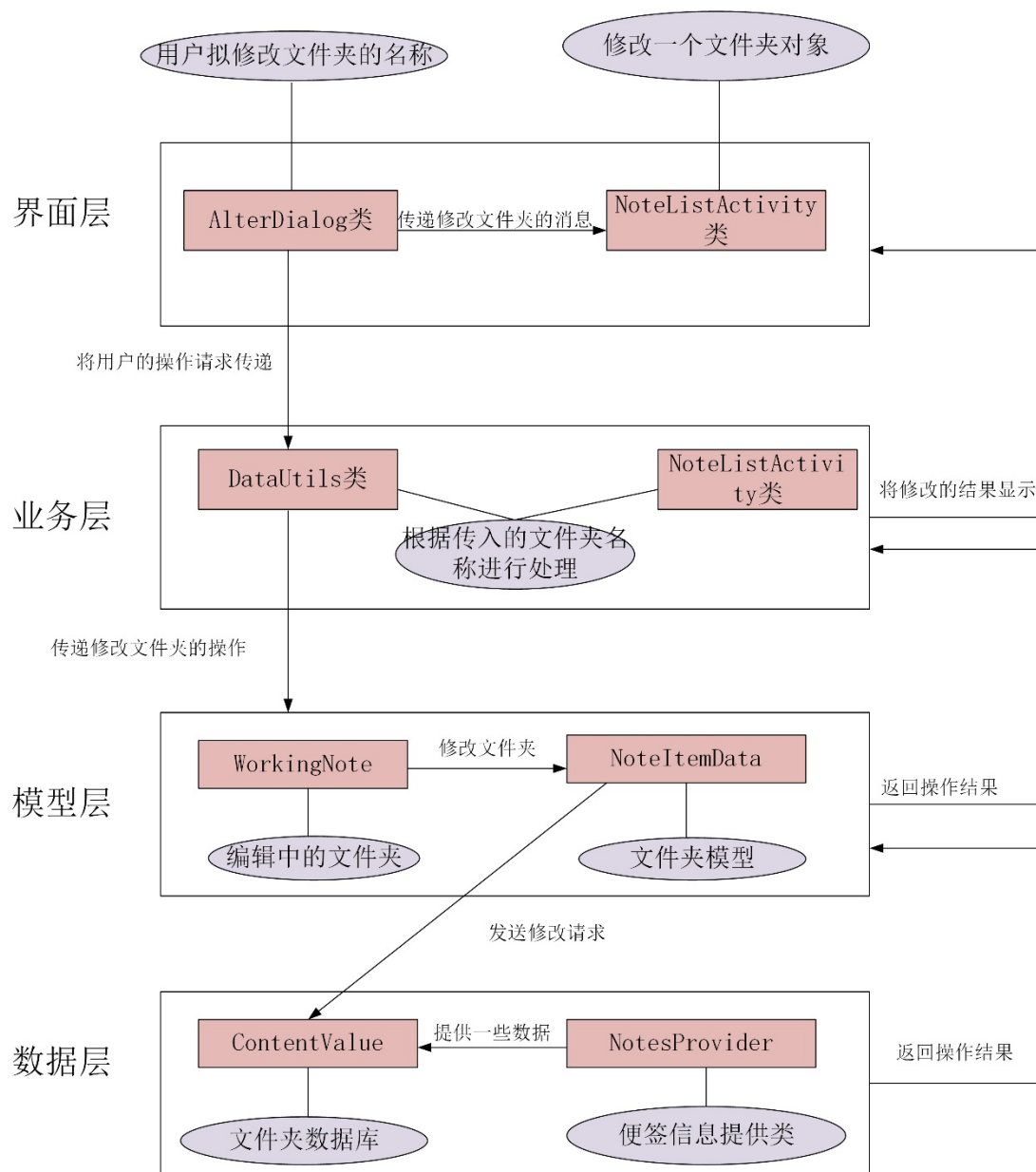
## 业务请求与响应过程

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。



## 类的作用

## 界面层相关的类

### 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。

#### (2) 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	requestCode 是请求	用于处理从其他	处理 Activity 返回

resultCode, Intent data)	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	Activity 返回的结果，例如打开便签或创建新便签后的操作	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 and 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.set CheckedItem()
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 mBackgroundQuery Handler 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

## 业务层相关的类

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	将指定的便签	创建一个对象, 用于存

	<p>提供者进行通信。</p> <p><b>id:</b> 要移动的便签的唯一标识符。</p> <p><b>srcFolderId:</b> 便签当前所在的文件夹的唯一标识符。</p> <p><b>desFolderId:</b> 要移动到的目标文件夹的唯一标识符。</p>	移动到不同的文件夹中	<p>储要更新的便签的新值。之后更新便签的属性值。</p> <p>将更新后的 <b>values</b> 对象应用于指定的便签。指定要更新的便签的位置。</p>
batchMoveToFolder	<p><b>resolver:</b> 用于交互。</p> <p><b>ids:</b> 包含要移动的便签的 ID。</p> <p><b>folderId:</b> 要移动到的目标文件夹的 ID。</p>	将一组便签移动到指定文件夹	<p>对于 <b>ids</b> 中的每个便签 ID，创建一个 <b>Operation</b> 对象，并设置更新的 <b>URI</b> 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。</p> <p>将创建的 <b>Operation</b> 对象添加到 <b>operationList</b> 中。</p> <p>使用 <b>resolver</b> 对象调用 <b>applyBatch()</b> 方法，将 <b>operationList</b> 中的操作应用到 <b>ContentProvider</b> 中。</p> <p>根据返回的结果进行判断：如果 <b>results</b> 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 <b>false</b>。</p> <p>如果没有抛出异常，则返回 <b>true</b>。否则，记录异常信息并返回 <b>false</b>。</p>

## 模型层相关的类

### 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。

## (2) 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
NoteSettingChangeListener	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

## 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
mCreateDate	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



# 数据层相关的类

## 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

<code>get(String key):</code>	Key: 键	根据键获取对应的值	直接调用 HashMap 中的 <code>get</code> 方法
<code>containsKey(String key)</code>	Key: 键	检查是否包含指定键	直接调用 HashMap 中的 <code>contains</code> 方法
<code>remove(String key)</code>	Key: 键	根据键移除键值对	直接调用 HashMap 中的 <code>remove</code> 方法
<code>clear()</code>		清空 ContentValues 中的所有键值对。	直接调用 HashMap 中的 <code>clear</code> 方法

# 功能八：导出便签

## 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上角菜单栏，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）



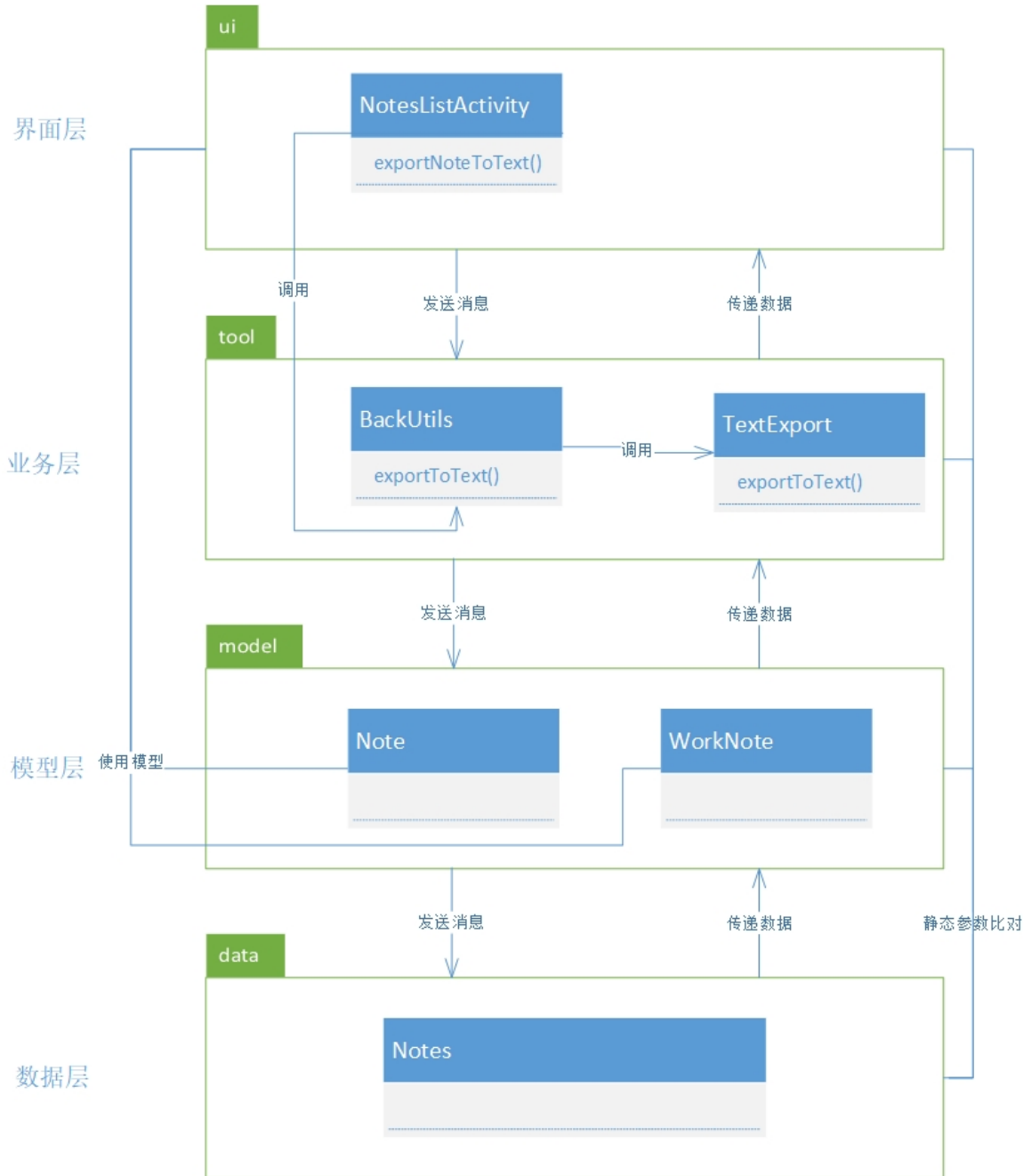
## 业务请求与响应过程

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。



# 类的作用

## 界面层相关的类

### 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。

(2) 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 <b>ListView</b> 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 <b>ContentResolver</b> 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

<code>initResources()</code>	/	初始化 Activity 的资源 and 视图	初始化了 <code>mContentResolver</code> 、 <code>mBackgroundQueryHandler</code> 、 <code>mCurrentFolderId</code> 等成员变量，以及便签列表的视图（ <code>mNotesListView</code> ）和适配器（ <code>mNotesListAdapter</code> ）等
<code>updateMenu()</code>	/	更新多选模式下的操作菜单	<code>selectedCount</code> 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
<code>onPrepareActionMode(ActionMode mode, Menu menu)</code>	<b>ActionMode mode:</b> 表示当前的多选模式操作栏； <b>Menu menu:</b> 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
<code>onActionItemClicked(ActionMode mode, MenuItem item)</code>	<b>ActionMode mode:</b> 表示当前的多选模式操作栏； <b>MenuItem item:</b> 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
<code>onDestroyActionMode(ActionMode mode)</code>	<b>ActionMode mode:</b> 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
<code>finishActionMode()</code>	/	结束操作模式	调用 <code>mActionMode.finish()</code>
<code>onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked)</code>	<b>ActionMode mode:</b> 表示当前的多选模式操作栏； <b>int position:</b> 表示被选中或取消选中的便签项的位置； <b>long id:</b> 表示被选中或取消选中的便签项的 ID； <b>boolean</b>	用于更新已选择项的状态和更新菜单	调用 <code>NitesListAdapter.setCheckedItem()</code>

	<b>checked:</b> 表示便签项是否被选中		
<code>onMenuItemClick(MenuItem item)</code>	<b>MenuItem item:</b> 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
<code>startAsyncNotesListQuery()</code>	/	异步查询并加载便签列表数据	使用 <code>mBackgroundQueryHandler</code> 发起异步查询,根据当前文件夹 ID 来加载相应的便签数据
<code>showFolderListMenu(Cursor cursor)</code>	<b>cursor</b> 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 <code>DataUtils.batchMoveToFolder</code> 方法将选定的便签移动到所选文件夹
<code>createNewNote()</code>	/	创建新的便签	创建一个新的便签编辑界面的 <b>Intent</b> , 并指定当前文件夹 ID
<code>batchDelete()</code>	/	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
<code>deleteFolder(long folderId)</code>	<b>folderId</b> 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
<code>openNode(NoteItemData data)</code>	<b>data</b> 是一个 <b>NoteItemData</b> 对象,表示要打开的便签数据	打开便签	创建一个 <b>Intent</b> 以查看指定便签的内容
<code>openFolder(NoteItemData data)</code>	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
<code>onClick(View v)</code>	<b>v</b> 是触发点击事件的视图	处理点击事件	调用 <code>createNewNote()</code> 方法创建新便签
<code>showSoftInput()</code>	/	显示软键盘	调用相关方法
<code>hideSoftInput(View view)</code>	<b>view</b> 是一个视图对象,用于获取窗口令牌	隐藏软键盘	调用相关方法



	以隐藏软键盘		
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	构建查询，获取可以

		标的文件夹列表	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
<code>onItemLongClick(AdapterView&lt;?&gt; parent, View view, int position, long id)</code>	<code>parent</code> : 表示触发长按事件的父级适配器视图; <code>view</code> : 表示用户长按的特定列表项视图; <code>position</code> : 表示被长按的列表项在适配器中的位置; <code>id</code> : 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 ( <code>Notes.TYPE_FOLDER</code> ),则将文件夹上下文菜单创建监听器设置为 <code>mFolderOnCreateContextMenuListener</code> 。这样,当用户点击该文件夹时,将弹出文件夹上下文菜单

## 业务层相关的类

### 类名: **BackupUtils**

**(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

**(2) 类的精读:**

BackupUtils 类的成员

成员	说明
<code>TAG</code>	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
<code>sInstance</code>	用于实现单例模式的静态成员变量,用于存储 <code>BackupUtils</code> 的单一实例
<code>STATE_SD_CARD_UNMOUONTED</code>	表示 SD 卡当前未挂载的状态
<code>STATE_BACKUP_FILE_NOT_EXIST</code>	表示备份文件不存在的状态
<code>STATE_DATA_DESTROYIED</code>	表示数据损坏或格式不正确的状态
<code>STATE_SYSTEM_ERROR</code>	表示备份或恢复操作中发生运行时异常的状态
<code>STATE_SUCCESS</code>	表示备份或恢复操作成功完成的状态
<code>mTextExport</code>	<code>TextExport</code> 类的实例,用于处理将

	便签数据导出为文本的操作
<code>TextExport.TEXT_FORMAT</code>	存储用于格式化文本导出的字符串数组
<code>TextExport.mContext</code>	用于访问应用的资源和内容提供者
<code>TextExport.mFileName</code>	导出的文本文件名
<code>TextExport.mFileDirectory</code>	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
<code>getInstance(Context context)</code>	传入一个 <b>Context</b> 参数，用于初始化该实例	获取 <b>BackupUtils</b> 的单例实例	传入一个 <b>Context</b> 参数，获取单例实例
<code>TextExport(Context context)</code>	一个 <b>Context</b> 参数	初始化 <b>TextExport</b> 实例	/
<code>externalStorageAvailable()</code>	/	检查外部存储是否可用	/
<code>exportToText()</code>	/	导出便签数据为文本文件的方法	调用 <b>TextExport</b> 对象的函数
<code>getExportedTextFileName()</code>	/	获取导出的文本文件名	调用 <b>TextExport</b> 对象的函数
<code>getExportedTextFileDir()</code>	/	获取导出的文本文件目录	调用 <b>TextExport</b> 对象的函数
<code>TextExport.getFormat(int id)</code>	<b>id</b> 表示传入的 <b>id</b>	根据给定的格式 <b>ID</b> 返回对应的格式化字符串	返回为该 <b>id</b> 的格式化字符串
<code>TextExport.exportFolderToText(String folderId, PrintStream ps)</code>	<b>folderId</b> 是文件夹的 <b>id</b> , <b>PrintStream</b> 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 <b>id</b> 的文件夹
<code>TextExport.exportNoteToText(String noteId, PrintStream ps)</code>	<b>noteId</b> 便签的 <b>ID</b> ; <b>PrintStream</b> 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 <b>id</b> 移动到该便签，开始导出
<code>TextExport.exportToText()</code>	/	导出便签数据为文本文件	获取文件夹 <b>id</b> ，调用 <b>exportFolderToText</b> 方法；获取便签 <b>id</b> ，调用 <b>exportNoteToText</b> 方法
<code>TextExport.getExportToTextPrintStream()</code>	/	获取一个 <b>PrintStream</b>	获取输出地址，创建备份文件，初始化一个 <b>ps</b>

		对象	对象
<code>generateFileMountedOnSDcard(Context context, int filePathResId, int fileNameFormatResId)</code>	一个 <b>Context</b> 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 <b>sb</b> 中；将资源 ID <b>filePathResId</b> 对应的字符串追加到 <b>sb</b> 中

## 模型层相关的类

### 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

#### （2） 类的精读：

Note 类的成员

成员	说明
<code>mNoteDiffValues</code>	一个 <b>ContentValues</b> 对象，用于存储便签的元数据的差异
<code>mNoteData</code>	一个 <b>NoteData</b> 对象，用于管理便签的文本数据和通话数据
<code>TAG</code>	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
<code>getNewNoteId(Context context, long folderId)</code>	<b>context</b> : Android 应用程序的上下文； <b>folderId</b> 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 <b>ContentValues</b> 对象 <b>values</b> ，用于存储便签的元数据，初始化 <b>values</b> ，从 URI 中提取新便签的 ID 并返回
<code>Note()</code>	/	构造函数，初始化 <b>mNoteDiffValues</b> 和 <b>mNoteData</b> 对象	/

<code>setNoteValue(String key, String value)</code>	<b>String key:</b> 元数据的键名; <b>String value:</b> 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 <b>mNoteDiffValues</b> , 并更新 <b>LOCAL_MODIFIED</b> 和 <b>MODIFIED_DATE</b> 的值
<code>setTextData(String key, String value)</code>	<b>String key:</b> 文本数据的键名; <b>String value:</b> 文本数据的键值	设置文本数据的键值对	将键值对存储到 <b>mNoteData</b> 的文本数据中
<code>setTextDataId(long id)</code>	<b>id:</b> 文本数据的 ID	设置文本数据的 ID	将 ID 存储到 <b>mNoteData</b> 中
<code>getTextDataId()</code>	/	获取文本数据的 ID	返回 <b>mNoteData</b> 中的文本数据 ID
<code>setCallDataId(long id)</code>	<b>id:</b> 通话数据的 ID	设置通话数据的 ID	将 ID 存储到 <b>mNoteData</b> 中
<code>setCallData(String key, String value)</code>	<b>key:</b> 通话数据的键名; <b>value:</b> 通话数据的键值	设置通话数据的键值对	将键值对存储到 <b>mNoteData</b> 的通话数据中
<code>isLocalModified()</code>	/	判断便签是否在本地上有修改	检查 <b>mNoteDiffValues</b> 和 <b>mNoteData</b> 是否包含本地修改的数据, 如果有则返回 <b>true</b> , 否则返回 <b>false</b>
<code>syncNote(Context context, long noteId)</code>	<b>context:</b> Android 应用程序的上下文; <b>noteId:</b> 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 <b>true</b> 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
<code>NoteData.isLocalModified()</code>	/	判断文本数据或通话数据是否在本地上有修改	/
<code>NoteData.setTextDataId(long id)</code>	<b>id</b> 表示传入的 id	设置文本数据的 ID	/
<code>NoteData.setCallDataId(long id)</code>	<b>id</b> 表示传入的 id	设置通话数据的 ID	/
<code>NoteData.setCallData(String key, String value)</code>	<b>String key:</b> 元数据的键名; <b>String value:</b> 元数据的键值	设置通话数据的键值对	/

<code>NoteData.setTextData(String key, String value)</code>	<b>String key:</b> 元数据的键名; <b>String value:</b> 元数据的键值	设置文本数据的键值对	/
<code>pushIntoContentResolver(Context context, long noteId)</code>	<b>context:</b> Android 应用程序的上下文; <b>noteId:</b> 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

## 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。

### (2) 类的精读:

WorkingNote 类的成员

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

folderId)	用程序的上下文; folderId: 便签所属的文件夹 ID	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
setOnSettingStatusChangeListener(NoteSettingChangeListener 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	/
NoteSettingChangeListener	/	内部接口，定义了便签设置状态变化的监听器方法	/

## 数据层相关的类

### 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。

#### （2）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_TEMPORARY_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_FOLDER	标识回收站文件夹常量
INTENT_EXTRA_ALERT_DATE	在 <b>Intent</b> 中作为键，传递提醒日期信息
INTENT_EXTRA_BACKGROUND_ID	在 <b>Intent</b> 中作为键，用来传递背景颜色的标识符
INTENT_EXTRA_WIDGET_ID	在 <b>Intent</b> 中作为键，用来传递小部件的 ID
INTENT_EXTRA_WIDGET_TYPE	在 <b>Intent</b> 中作为键，用来传递小部件的类型信息
INTENT_EXTRA_FOLDER_ID	在 <b>Intent</b> 中作为键，用来传递文件夹的 ID
INTENT_EXTRA_CALL_DATE	在 <b>Intent</b> 中作为键，传递通话记录的日期信息
TYPE_WIDGET_INVALID	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	用于指示是否在本机修改
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

## 功能九：关键字查询

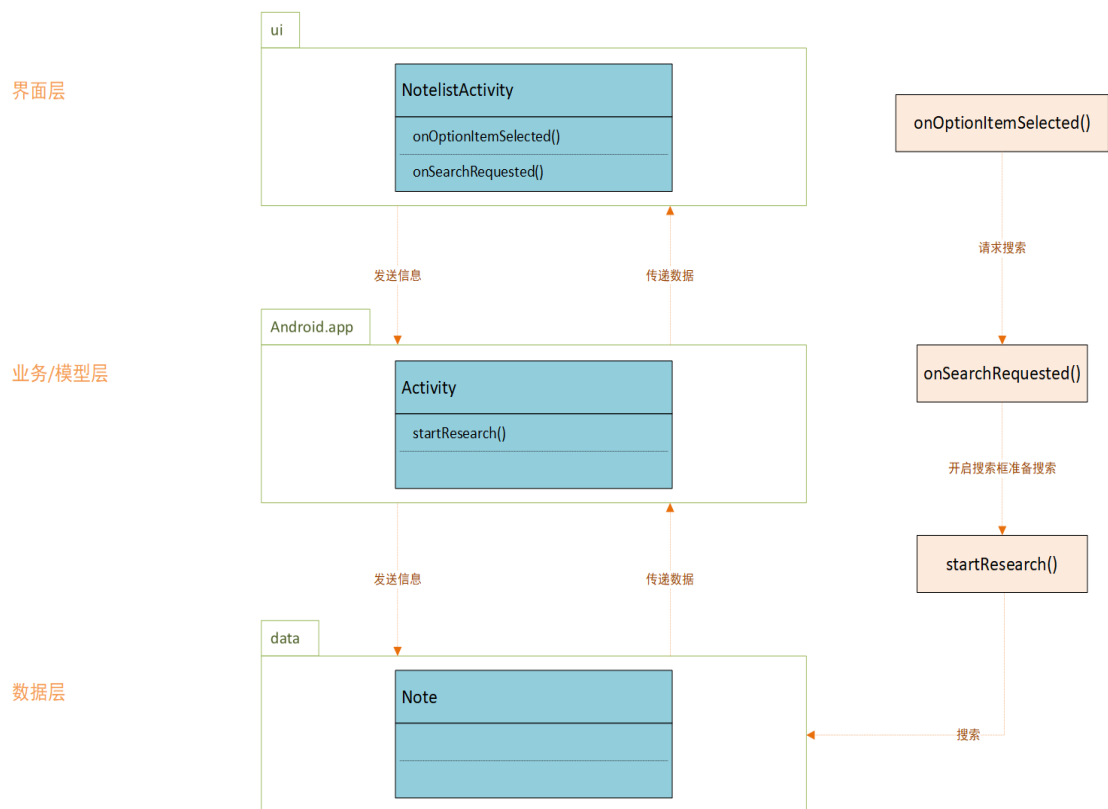
### 功能介绍

**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。

### 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。



# 类的作用

## 界面层相关的类

### 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

(2) 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	检查和更新 Select 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.setCheckedItem() 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

## 业务层/模型相关的类：Activity

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

## 数据层相关的类

### 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。

## （2） 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_TEMPORARY_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_FOLDER	标识回收站文件夹常量
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	用于指示是否在本机修改

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

## 功能十：修改字体大小

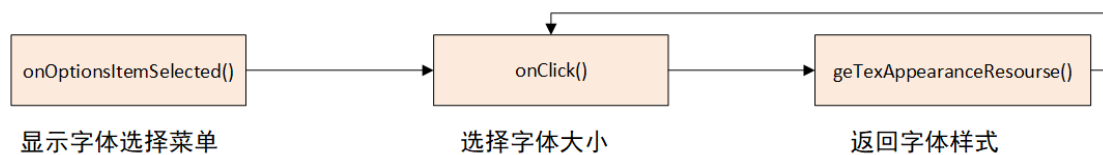
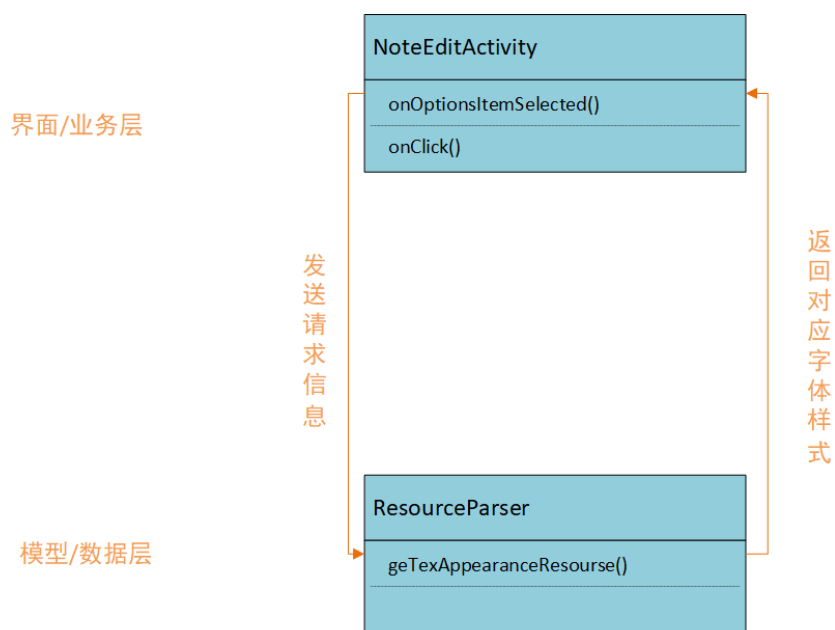
### 功能介绍

**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。

### 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()方法来显示菜单项，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 editor 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。



## 类的作用

### 界面层/业务层相关的类

#### 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

##### (2) 类的精读：

##### 类的成员

成员	说明
----	----



HeaderViewHolder	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
dispatchTouchEvent(MotionEvent ev)	ev: 屏幕触摸的操作	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 , MotionEvent ev)	view : 用来表示 ui 容器，容器的可触范围。 ev: 屏幕触摸操作	用来确定触摸操作是否在给定的 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	用判断结构，如果当前选择器为可见，则设置为不可见。

		藏的	
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。

## 模型层相关的类

### 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
getTexAppearanceResource(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。

## 数据层相关的类

### 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
mSharedPreferences	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。

# 功能十一：修改背景颜色

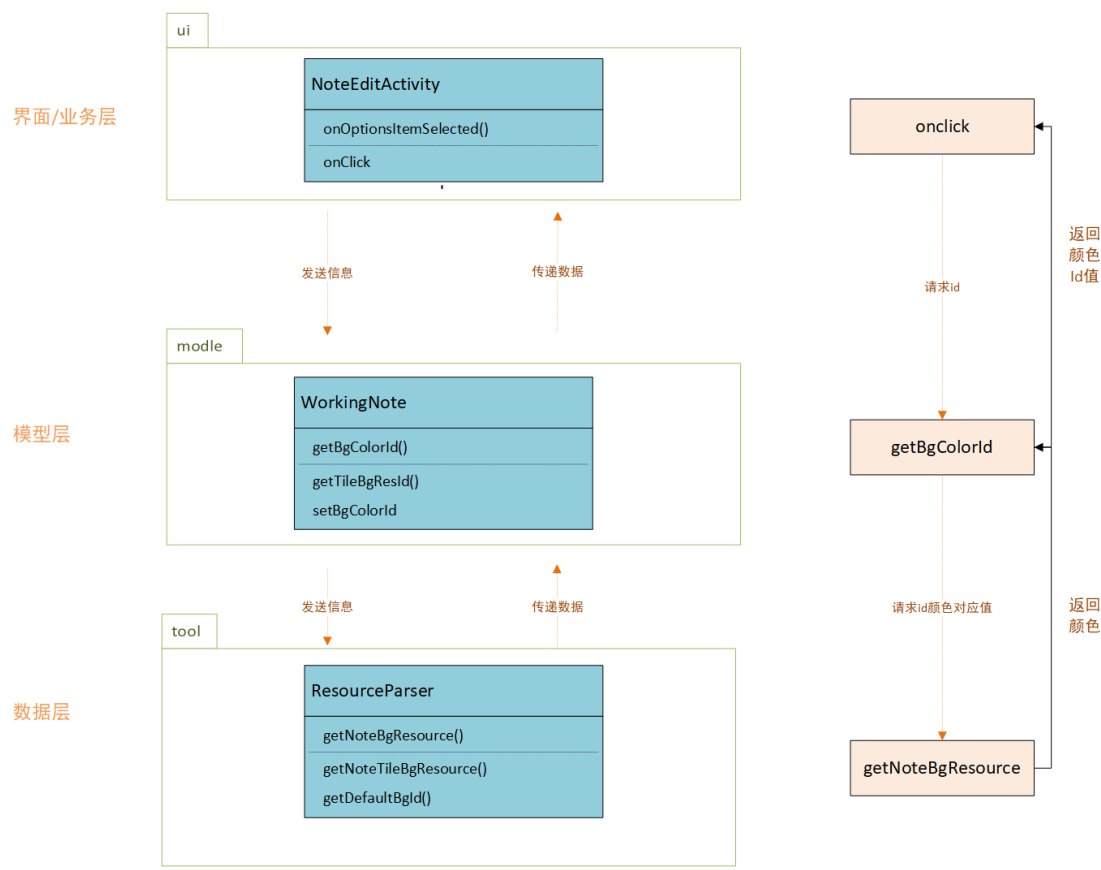
## 功能介绍

**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。

## 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。



# 类的作用

## 界面层/业务层相关的类

### 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。
- (2) 类的精读：

类的成员

成员	说明
ViewHolder	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 值与对应颜色的映射关系
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
mHeaderViewPanel	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
dispatchTouchEvent()	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, 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	先检查用户是否处于其

		回按钮时，未保存的便签内容可以得到保存	他设置状态，否则则直接调用 <code>saveNote()</code> 的方法保存内容。同时调用 <code>super.onBackPressed</code> 方法来执行默认返回操作。
<code>clearSettingState()</code>	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。
<code>onClick(View v)</code>	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	<code>View.OnClickListener</code> 的接口来监听点击事件，在 <code>onClick</code> 方法中实现对应操作
<code>onOptionsItemSelected (MenuItem item)</code>	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 <code>item</code> 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。
<code>InitNoteScreen()</code>	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 <code>mNoteEditor</code> 中的 <code>setTextAppearance</code> 的方法来实现。
<code>onBackgroundColorChanged()</code>	无	修改当前的背景颜色	通过 id 值来修改对应的背景颜色

## 模型层相关的类

### 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。

## (2) 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
NoteSettingChangeListener	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

## 数据层相关的类

### 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

(2) 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

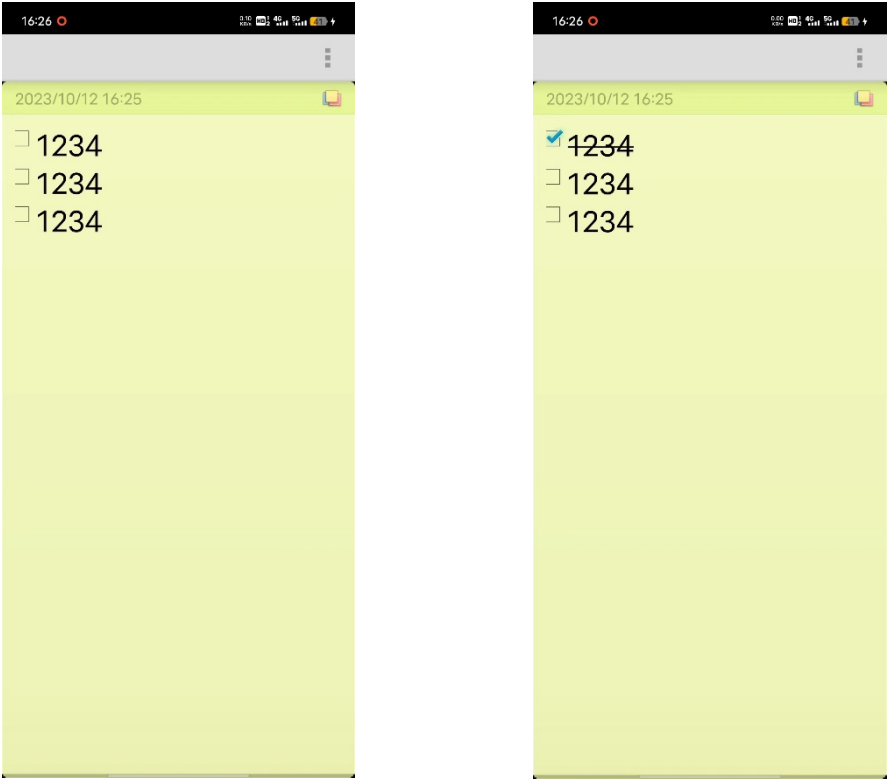
## 功能十二：清单模式

### 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。



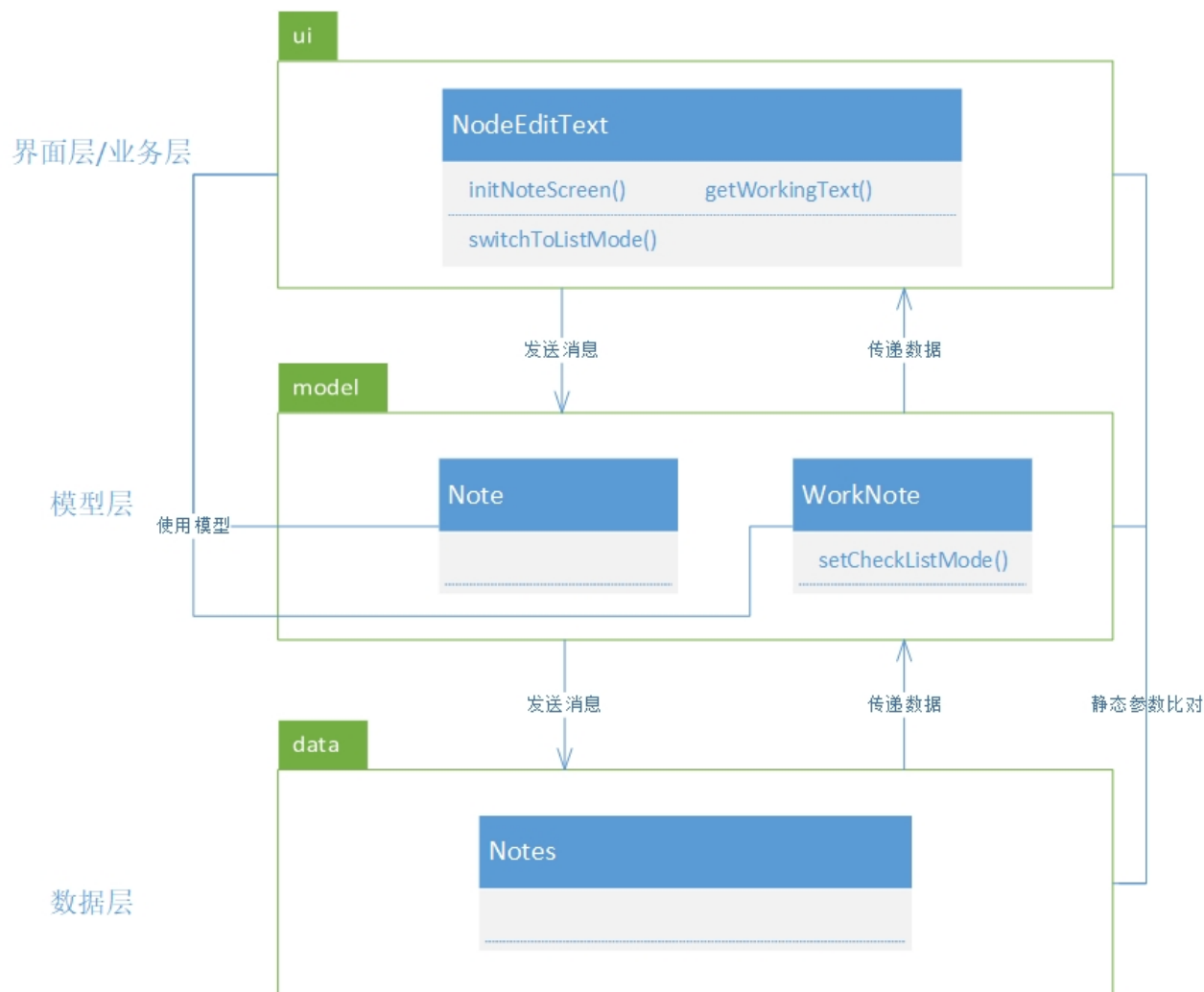
## 业务请求与响应过程

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。



## 类的作用

## 界面层相关的类

### 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

#### (2) 类的精读：

NoteEditText 类的成员

成员	说明
ViewHolder	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
mHeaderViewPanel	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
mSharedPreferences	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	用于在活动被系统销毁时恢复状态	如果存在

<code>savedInstanceState()</code>	包含之前保存的状态信息	毁后重新创建时，恢复之前的状态信息	<code>Intent.EXTRA_UID</code> 的键在 <code>savedInstanceState</code> 中，它将创建一个新的 <code>Intent</code> ，将之前保存的 <code>UID</code> 添加到其中，并尝试使用该 <code>Intent</code> 初始化活动状态
<code>initActivityState(Intent intent)</code>	<code>intent</code> 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 <code>Intent</code> 的不同操作（ <code>ACTION</code> ）
<code>onResume()</code>	/	用于在活动重新进入前台时进行初始化和更新界面	调用 <code>initNoteScreen()</code> 方法来初始化便签编辑界面
<code>initNoteScreen()</code>	/	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
<code>showAlertHeader()</code>	/	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）
<code>onNewIntent(Intent intent)</code>	<code>intent</code> 包含新的 <code>Intent</code> 信息	用于在活动已经存在的情况下处理新的 <code>Intent</code>	首先调用父类的 <code>onNewIntent</code> 方法，然后使用传入的 <code>Intent</code> 重新初始化活动状态
<code>onSaveInstanceState(Bundle outState)</code>	<code>outState</code> 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 <code>ID</code> ），则首先尝试保存便签。然后，将便签的 <code>ID</code> 放入 <code>outState</code> 中，以便在重建活动时使用
<code>dispatchTouchEvent(MotionEvent ev)</code>	<code>MotionEvent ev</code> 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 <code>dispatchTouchEvent</code> 方法继续事件分发
<code>inRangeOfView(View view, MotionEvent ev)</code>	<code>view</code> 要检查的视图； <code>ev</code> 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 <code>true</code> ，否则返回 <code>false</code>
<code>initResources()</code>		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 <b>onPause</b> 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 <b>Intent</b> ，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 <b>ID</b> ，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 <b>clearSettingState</b> 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 <b>onBackPressed</b> 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 <b>true</b> ，否则返回 <b>false</b>
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 <b>Intent.ACTION_SEND</b> 动作和 <b>text/plain</b> 类型的应用程序	创建一个 <b>Intent</b> ，设置 <b>EXTRA_TEXT</b> 为便签内容，类型为 <b>text/plain</b> ，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 <b>NoteEditActivity</b> 意图，以新建便签的方式启动该活动
<code>deleteCurrentNote()</code>	/	删除当前便签	如果当前便签已存在于数据库中，将其 <b>ID</b> 添加到 <b>HashSet</b> 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 <b>DataUtils.batchDeleteNotes</b> 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
<code>isSyncMode()</code>	/	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 <b>true</b> ，否则返回 <b>false</b>
<code>onClockAlertChanged(long date, boolean set)</code>	<code>date</code> 提醒时间； <code>set</code> 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 <b>Intent</b> ，并根据 <code>set</code> 参数设置或取消提醒。最后，根据提醒是否设置，调用 <b>showAlert Header</b> 方法以更新提醒状态的显示
<code>updateWidget()</code>	/	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 <b>Intent</b> ，然后发送广播以更新小部件的显示
<code>onEditTextDelete(int index, String text)</code>	<code>index</code> 文本编辑框索引； <code>String text</code> 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
<code>onEditTextEnter(int index, String text)</code>	<code>index</code> 文本编辑框索引； <code>text</code> 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
<code>switchToListMode(String text)</code>	<code>text</code> 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, 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	/

## 模型层相关的类

### 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

#### （2） 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 <code>mNoteData</code> 对象	
<code>setNoteValue(String key, String value)</code>	<b>String key:</b> 元数据的键名; <b>String value:</b> 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 <code>mNoteDiffValues</code> , 并更新 <code>LOCAL_MODIFIED</code> 和 <code>MODIFIED_DATE</code> 的值
<code>setTextData(String key, String value)</code>	<b>String key:</b> 文本数据的键名; <b>String value:</b> 文本数据的键值	设置文本数据的键值对	将键值对存储到 <code>mNoteData</code> 的文本数据中
<code>setTextDataId(long id)</code>	<b>id:</b> 文本数据的 ID	设置文本数据的 ID	将 ID 存储到 <code>mNoteData</code> 中
<code>getTextDataId()</code>	/	获取文本数据的 ID	返回 <code>mNoteData</code> 中的文本数据 ID
<code>setCallDataId(long id)</code>	<b>id:</b> 通话数据的 ID	设置通话数据的 ID	将 ID 存储到 <code>mNoteData</code> 中
<code>setCallData(String key, String value)</code>	<b>key:</b> 通话数据的键名; <b>value:</b> 通话数据的键值	设置通话数据的键值对	将键值对存储到 <code>mNoteData</code> 的通话数据中
<code>isLocalModified()</code>	/	判断便签是否在本地上有修改	检查 <code>mNoteDiffValues</code> 和 <code>mNoteData</code> 是否包含本地修改的数据, 如果有则返回 <code>true</code> , 否则返回 <code>false</code>
<code>syncNote(Context context, long noteId)</code>	<b>context:</b> Android 应用程序的上下文; <b>noteId:</b> 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 <code>true</code> 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
<code>NoteData.isLocalModified()</code>	/	判断文本数据或通话数据是否在本地上有修改	/
<code>NoteData.setTextDataId(long id)</code>	<b>id</b> 表示传入的 id	设置文本数据的 ID	/
<code>NoteData.setCallDataId(long id)</code>	<b>id</b> 表示传入的 id	设置通话数据的 ID	/
<code>NoteData.setCallData(String key, String value)</code>	<b>String key:</b> 元数据的键名; <b>String value:</b> 元数	设置通话数据的键值对	/

	据的键值		
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

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。

(2) 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
----	------	------	------

<code>WorkingNote(Context context, long folderId)</code>	<b>context:</b> Android 应用程序的上下文; <b>folderId:</b> 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	/
<code>WorkingNote(Context context, long noteId, long folderId)</code>	<b>context:</b> Android 应用程序的上下文; <b>noteId:</b> 要加载的便签的 ID; <b>folderId:</b> 便签所属的文件夹 ID	构造函数，用于加载已有的工作便签	/
<code>loadNote()</code>	/	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
<code>loadNoteData()</code>	/	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
<code>createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId)</code>	<b>context:</b> Android 应用程序的上下文; <b>folderId:</b> 便签所属的文件夹 ID; <b>widgetId:</b> 小部件的 ID; <b>widgetType:</b> 小部件类型; <b>defaultBgColorId:</b> 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
<code>load(Context context, long id)</code>	<b>context:</b> Android 应用程序的上下文; <b>id:</b> 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
<code>saveNote()</code>	/	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，创建新便签并获其 ID; 同步便签的元数据和内容数据到数据库
<code>existInDatabase()</code>	/	判断便签是否已存在于数据库中	判断 <b>NoteId</b> 是否存在
<code>isWorthSaving()</code>	/	判断便签是否值得	考虑是否已

		保存	删除、是否存在文本内容和是否有本地修改
setOnSettingStatusChangeListener(NoteSettingChangeListener 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	/
NoteSettingChangeListener	/	内部接口，定义了便签设置状态变化的监听器方法	/

## 数据层相关的类

### 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。

#### （2） 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_TEMPORARY_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_FOLDER	标识回收站文件夹常量
INTENT_EXTRA_ALERT_DATE	在 <b>Intent</b> 中作为键，传递提醒日期信息
INTENT_EXTRA_BACKGROUND_ID	在 <b>Intent</b> 中作为键，用来传递背景颜色的标识符
INTENT_EXTRA_WIDGET_ID	在 <b>Intent</b> 中作为键，用来传递小部件的 <b>ID</b>
INTENT_EXTRA_WIDGET_TYPE	在 <b>Intent</b> 中作为键，用来传递小部件的类型信息
INTENT_EXTRA_FOLDER_ID	在 <b>Intent</b> 中作为键，用来传递文件夹的 <b>ID</b>
INTENT_EXTRA_CALL_DATE	在 <b>Intent</b> 中作为键，传递通话记录的日期信息
TYPE_WIDGET_INVALID	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 <b>MIME</b> 类型
DataConstants.CALL_NOTE	表示通话记录的 <b>MIME</b> 类型
CONTENT_NOTE_URI	表示用于查询便签的 <b>URI</b>
CONTENT_DATA_URI	表示用于查询数据的 <b>URI</b>
NoteColumns.ID	代表唯一行的 <b>ID</b>
NoteColumns.PARENT_ID	代表便签或文件夹的父项的 <b>ID</b> ，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 <b>ID</b>
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 <b>ID</b>
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 <b>ID</b>
NoteColumns.LOCAL_MODIFIED	用于指示是否在本机修改
NoteColumns.ORIGIN_PARENT_ID	代表在移动到临时文件夹之前的原始父项 <b>ID</b>
NoteColumns.GTASK_ID	代表 <b>Google</b> 任务（ <b>Google Task</b> ）的 <b>ID</b>
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 <b>MIME</b> 类型
TextNote.CONTENT_ITEM_TYPE	用于标识单个文本便签的 <b>MIME</b> 类型
TextNote.CONTENT_URI	表示访问文本便签数据的 <b>URI</b>
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 <b>MIME</b> 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

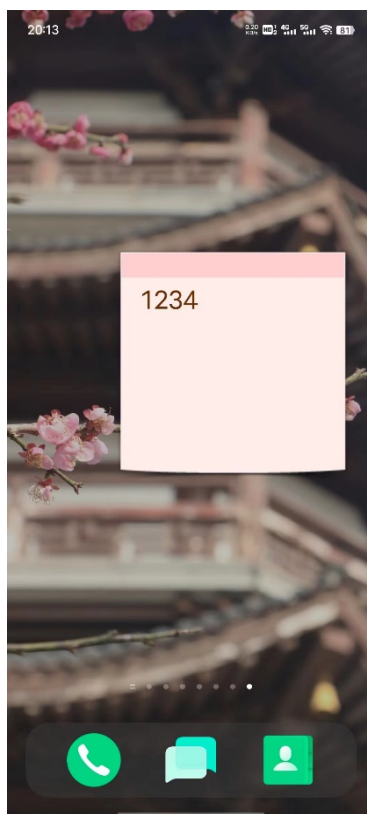
Notes 类没有方法

## 功能十三：发送到桌面

### 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



### 业务请求与响应过程

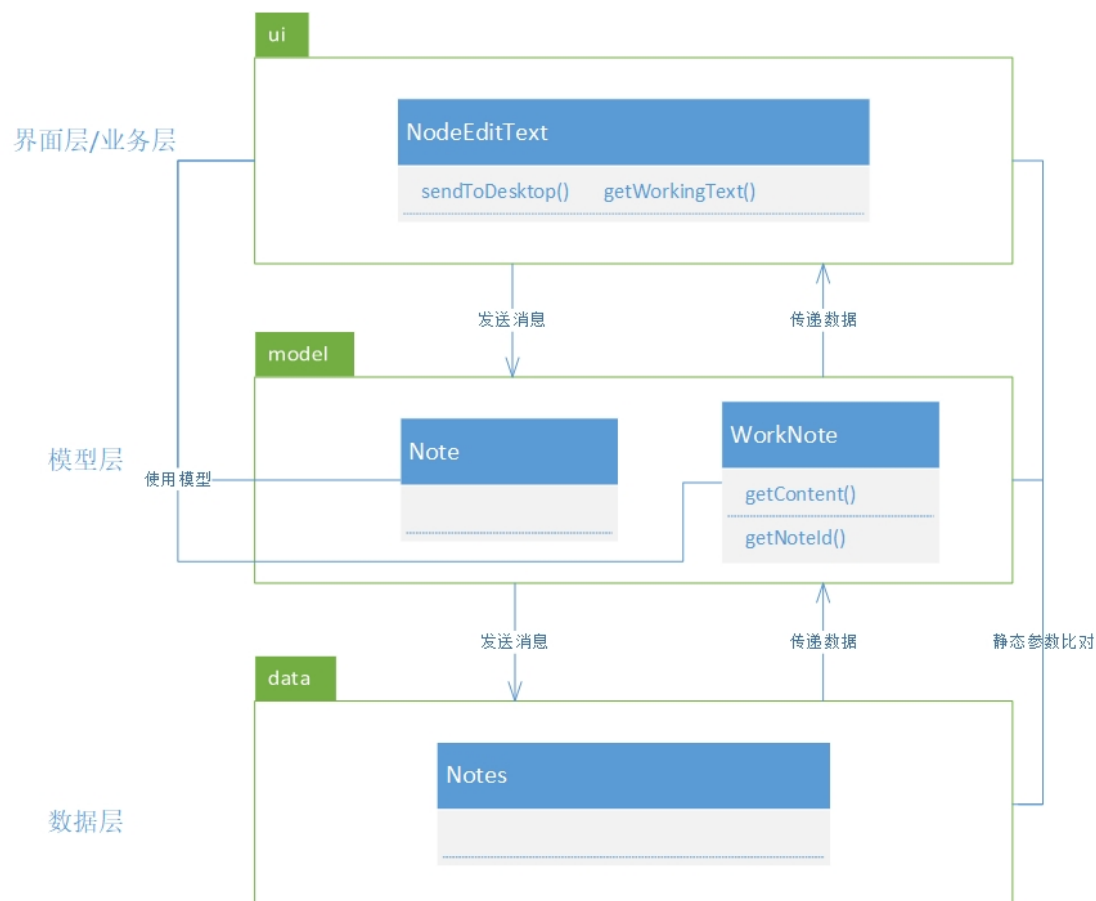
发送到桌面功能（下简称“功能”）的界面层由包 ui 中 NodeEditActivity 类

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。





# 类的作用

## 界面层相关的类

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

(2) 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

## 模型层相关的类

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

（2） 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

## 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。

### （2） 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	/

## 数据层相关的类

### 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。

#### (2) 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_TEMPORARY_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_FOLDER	标识回收站文件夹常量
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_INVALID	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	用于指示是否在本机修改
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

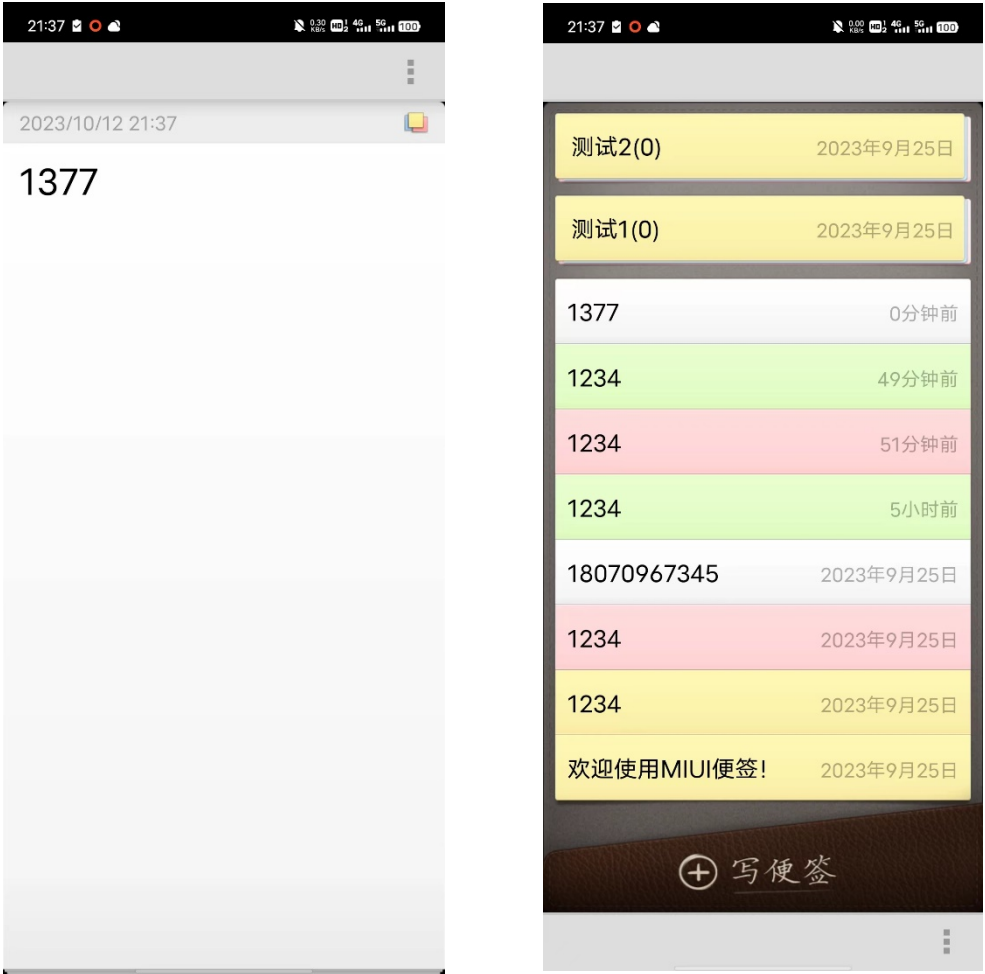
Notes 类没有方法

## 功能十四：自动保存便签

### 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。



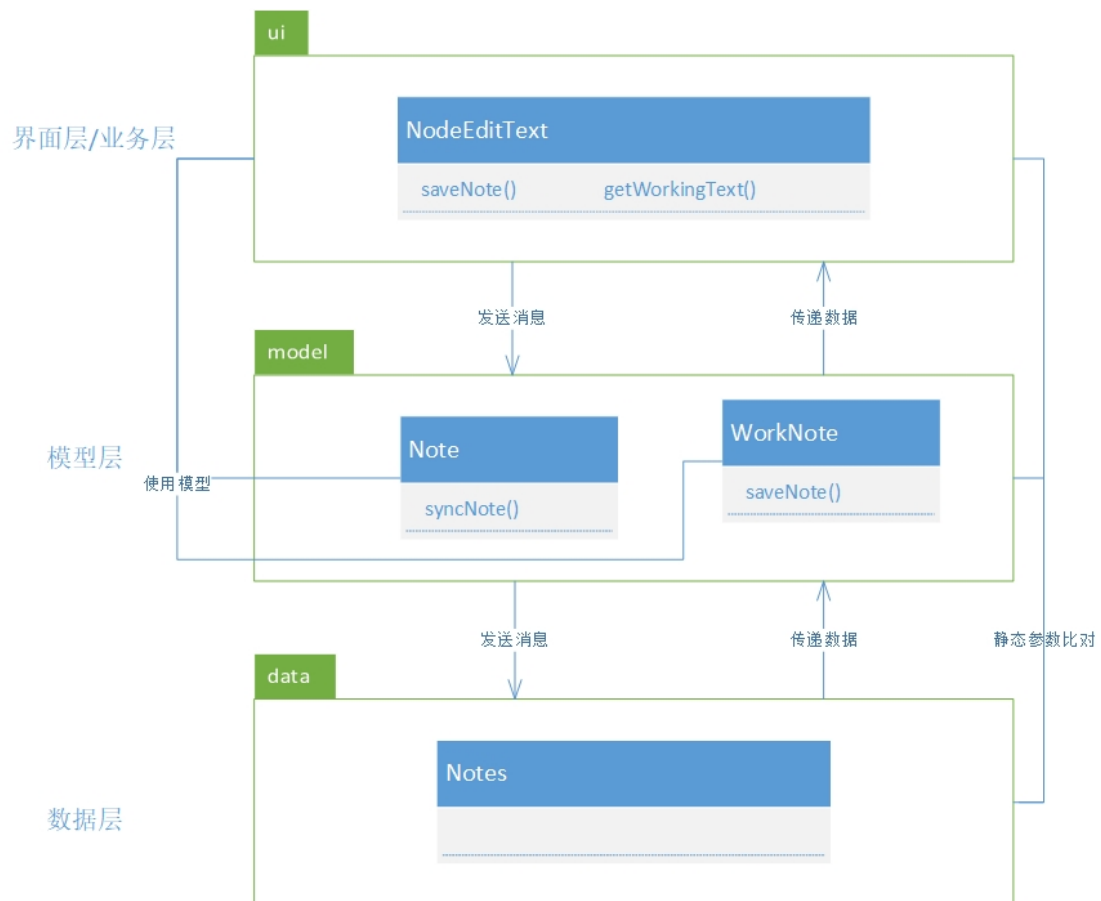
## 业务请求与响应过程

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



## 类的作用

## 界面层相关的类

### 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

#### (2) 类的精读：

NoteEditText 类的成员

成员	说明
<b>mWorkingNote</b>	<b>包含有关便签的状态信息</b>

NoteEditText 类中的方法

方法	参数解释	方法作用	实现过程
----	------	------	------



<code>getWorkingText()</code>	/	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
<code>saveNote()</code>	/	保存便签	获取工作文本，然后调用 <code>saveNote</code> 方法保存便签。如果保存成功，设置结果码为 <code>RESULT_OK</code> 表示创建或编辑便签

## 模型层相关的类

### 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

（2） 类的精读：

Note 类的成员

成员	说明
<code>mNoteDiffValues</code>	一个 <code>ContentValues</code> 对象，用于存储便签的元数据的差异
<code>mNoteData</code>	一个 <code>NoteData</code> 对象，用于管理便签的文本数据和通话数据
<code>TAG</code>	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
<code>getNewNoteId(Context context, long folderId)</code>	<code>context</code> : Android 应用程序的上下文； <code>folderId</code> 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 <code>ContentValues</code> 对象 <code>values</code> ，用于存储便签的元数据，初始化 <code>values</code> ，从 URI 中提取新便签的 ID 并返回
<code>Note()</code>	/	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	设置通话数据的键值对	/

	value: 元数据的键值		
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

## 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。

### (2) 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
<code>saveNote()</code>	/	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
<code>existInDatabase()</code>	/	判断便签是否已存在于数据库中	判断 <code>NoteId</code> 是否存在
<code>isWorthSaving()</code>	/	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

## 数据层相关的类

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。

### （2） 类的精读：

Note 类的成员

成员	说明
<code>AUTHORITY</code>	定义了用于访问数据的内容提供者的权限
<code>TAG</code>	用于标识日志输出的标签
<code>TYPE_NOTE</code>	普通便签的类型常量
<code>TYPE_FOLDER</code>	文件夹的类型常量
<code>TYPE_SYSTEM</code>	系统的类型常量

ID_ROOT_FOLDER	标识根文件夹的常量
ID_TEMPORARY_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_FOLDER	标识回收站文件夹常量
INTENT_EXTRA_ALERT_DATE	在 <b>Intent</b> 中作为键，传递提醒日期信息
INTENT_EXTRA_BACKGROUND_ID	在 <b>Intent</b> 中作为键，用来传递背景颜色的标识符
INTENT_EXTRA_WIDGET_ID	在 <b>Intent</b> 中作为键，用来传递小部件的 ID
INTENT_EXTRA_WIDGET_TYPE	在 <b>Intent</b> 中作为键，用来传递小部件的类型信息
INTENT_EXTRA_FOLDER_ID	在 <b>Intent</b> 中作为键，用来传递文件夹的 ID
INTENT_EXTRA_CALL_DATE	在 <b>Intent</b> 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	用于指示是否在本机修改
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