## 小米便签开源代码的泛读报告

### 列出小米便签中的类及各类的主要作用

**表1 小米便签的代码构成及主要作用**

|  |  |  |  |
| --- | --- | --- | --- |
| **包** | **子包** | **类** | **主要作用** |
| data |  | contact | 联系人数据库，获取、查找以及判定联系人的信息。 |
| Notes | 便签数据库，Notes类中定义了很多常量，大多数是int型和String型，还定义了用于查询便签和文件夹的指针，记录便签相关属性和数据。 |
| NotesDatabaseHelper | 便签数据库，Notes类中定义了很多常量，大多数是int型和String型，还定义了用于查询便签和文件夹的指针，记录便签相关属性和数据。 |
| NotesProvider | 提供数据存储和获取的接口，可以在不同的程序之间实现数据的共享，主要是对URL的匹配、增、删、改、查。 |
| Gtask | data | MetaData | 本类继承于Task,主要用于记录数据的变化 |
| Node | 任务清单的抽象类，同步操作的基础数据类型，定义了相关指示同步操作的常量 |
| SqlData | 设置数据当中的数据，用于支持小米便签最底层的数据库相关操作 |
| SqlNode | 设置数据的内容，真正意义上的数据，用于支持小米便签最底层的数据库相关操作 |
| Task | 任务清单 |
| TaskList | 继承于Node的任务清单的内容 |
| exception | ActionFailureException | 处理功能失败的异常 |
| NetworkFailureException | 处理网络工作失败的异常 |
| remote | GTaskASyncTask | 设置任务的分享同步的工作 |
| GTaskClient | 设置任务对象 |
| GTaskManager | 处理任务的类 |
| GTaskSyncService | 设置任务分享的服务类 |
| model |  | Note | 对小米便签的单个便签项进行建模 |
| WorkingNote | 设置正在工作的便签项的基本操作功能和服务 |
| tool |  | BackupUtils | 用于备份数据，导出文件和便签。 |
| DataUtils | 编写便签的数据 |
| GTaskStringUtils | 定义了很多静态字符串，为了提供JsonObject中相应字符串的key，将这些静态的定义单独放到了一个类里面。 |
| ResourceParser | 获取相应的资源在程序中可以使用，例如便签的背景颜色等 |
| ui |  | AlarmAlertActivity | 闹钟 |
| AlarmInitReceiver | 闹钟界面的初始化 |
| AlarmReceiver | 用户使用闹钟这个功能最接近的一个包，AlarmAlertActivity在这个类中被启用 |
| DateTimePicker | 获取当前时间，在便签的左上 |
| DateTimePickerDialog | 设置时间的对话框 |
| DropdownMenu | 菜单 |
| FoldersListAdapter | 适配器，主要作用是便签数据库和用户的交互 |
| NoteEditActivity | 单个便签的界面 |
| NoteEditText | 编辑单个便签 |
| NoteItemData | 对文件的数据进行操作的类 |
| NotesListActivity | 进入便签之后的首个界面的交互，可完成的功能为新增便签，删除便签，进入已有便签，以及打开菜单 |
| NotesListAdapter | 目录的适配器，使note在目录上出现 |
| NotesListItem | 和NotesListActivity配套使用，使便签以目录的形式出现 |
| NotesPreferenceActivity | 颜色和字体大小的数据存储 |
| widget |  | NoteWidgetProvider | 在桌面创建小米便签的组件,点击桌面组件可以进入便签界面，可以新建或修改便签。 |
| NoteWidgetProvider\_2x | 在桌面创建小米便签的2x2的桌面小组件。 |
| NoteWidgetProvider\_4x | 在桌面创建小米便签的4x4的桌面小组件。 |

### 类间关系图

根据小米便签的开源代码，绘制其体系结构图(见如图1）。

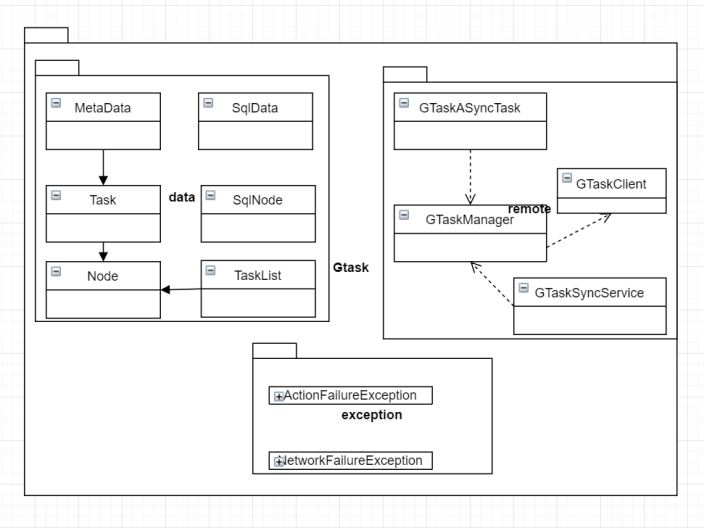


**图1. 小米便签体系结构图**

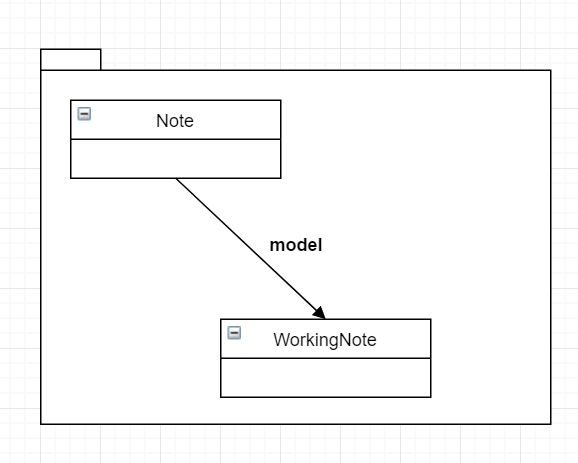
下图进一步描述了各个包中类间的关系。



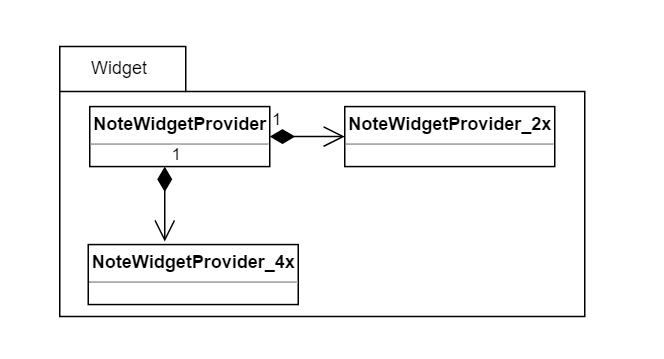
**图2. data包中类间的关系图**



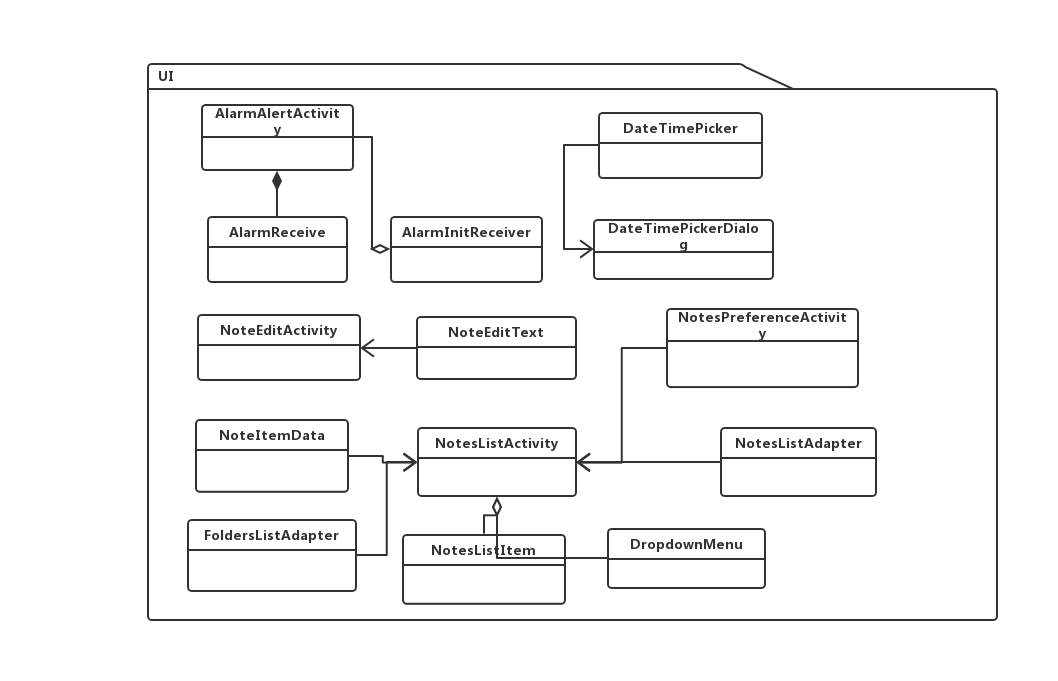
**图3. gtask包中类间的关系图**



**图4. model包中类间的关系图**



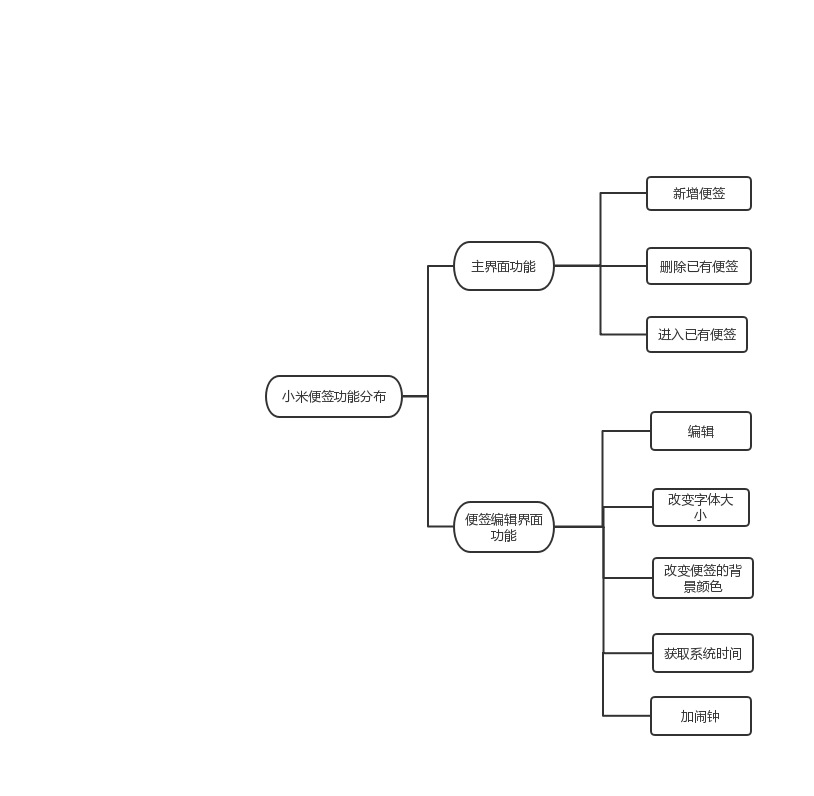
**图5. widget包中类间的关系图**



**图6. ui包中类间的关系图**

### 列出小米便签的基本功能

根据对小米便签代码的阅读和功能的理解，下图描述了小米便签的整体功能框架（如图7）。



**图7. 小米便签的功能分类**

1. **功能1：新建/删除/移动便签**

* 新建便签：通过小米便签软件的主界面下方的“写便签”快捷键和文件夹、便签视图下的选项“新建便签”可以在当前目录创建一个便签并打开进入文本编辑。其中主界面下的快速创建方式可以迅速创建一个待编辑的便签，用于满足临时迅速记录的需求。
* 删除便签：长按便签后，下方会出现拥有删除按键的菜单，可以对于选择的便签进行删除操作

1. **功能2：改变背景颜色**

* 进入便签编辑界面后，点击右上角的方块图标可以选择该便签的背景颜色。

1. **功能3：加一个闹钟**

* 进入便签编辑界面后，点击右上角的闹钟图标，可添加闹钟

1. **功能4：添加桌面组件**

* 小米便签自带一个2x2的桌面小工具和一个4x4的桌面小工具，可以通过自己的手机上拖出桌面挂件就可以使用这个功能，点击这个桌面挂件就可以新建编辑便签

1. **功能5：呼叫电话**

* 在便签中的号码，长按后会出现呼叫电话选项

1. **功能7：导出到文本**

* 在软件下方的菜单栏点开后会有导出选项，点击后，选择中的便签会导出到sd卡中

1. **功能8：分享**

* 在文本编辑界面，双击文本，会在文本下方出现菜单，可对于选中文本进行分享操作。

1. **功能9：备份**

### 软件功能与类间的对应关系

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **功能名称** | **实现模块** | **实现方法** |
| 1 | 新建便签 | NoteEditActivity | createNewNote() |
| 2 | Gtask设置和同步 | GTaskManager | syncFolder() |
| 3 | 导出到文本 | GTaskManager | doContentSync() |
| 4 | 分享 | GTaskASyncTask | startSync() |
| 5 | 添加桌面组件 | NotesProvider | onDeleted()、onUpdate() |
| 6 | 备份 | BackupUtils | exportToText() |
| 7 | 删除便签 | NoteEditActivity | deleteFolder（） |
| 8 | 启动闹钟 | AlarmReceive | onReceive() |
| 9 | 获取系统时间 | NoteEditActivity | initNoteScreen() |
| 10 | 设置背景颜色 | NoteEditActivity | dispatchTouchEvent() |
| 11 | 拨打电话 | NoteEditActivity | initActivityState() |

### 收获

* 在进行小米便签运行环境配置（Eclipse、AndroidStudio）的过程中掌握了多种插件的使用，这些插件可以大大提高代码分析的效率；
* 在进行小米便签源代码的泛读的过程中，对于添加桌面组件的功能有了了解与掌握，之前在本身的使用中就觉得桌面组件，例如超级课程表查看课表，Days Matter查看倒计时，不用打开软件就可以在桌面上清楚的看到。AppWidgetProvider本身是一个广播接收者, 通过继承类、重写方法等也算知道在 Android 如何使用 Widget了。
* 通过这次的作业，我了解到了更多java的组件，进一步地了解了对于listview的使用以及适配器的使用。
* 学习到了如何获取系统时间，以及其他的调用系统本来具有的一些资源。
* 对安卓开发的掌握更加熟练，对一些视图及监听方法的实现有了更深入的了解，对一些控件的属性等也更加的熟悉。
* 因为用到了Android和Java的知识点，又进一步巩固了JAVA相关的知识。  
  对于功能无法实现、运行错误、Bug修改，极大限度的磨练了我们的耐心和接受度。由于版本较老，很多功能方法无法实现。自己又不得不重新学习了解。这次作业除了飞巩固丰富了知识以外，更加磨练了自己。

### 存在的问题

* 由于对Java语言和Andriod开发的架构不够熟悉，在阅读代码中存在很多障碍，导致对代码具体理解不够清晰，需要在精读过程中加深理解；
* 一开始导入代码的时候，由于不熟悉包的引用的问题，所以一开始所有的包无法使用，后来才意识到了这个问题。
* 源代码中有很多方法过老，已经弃用，所以在修改这一部分的代码的时候花了一些时间。