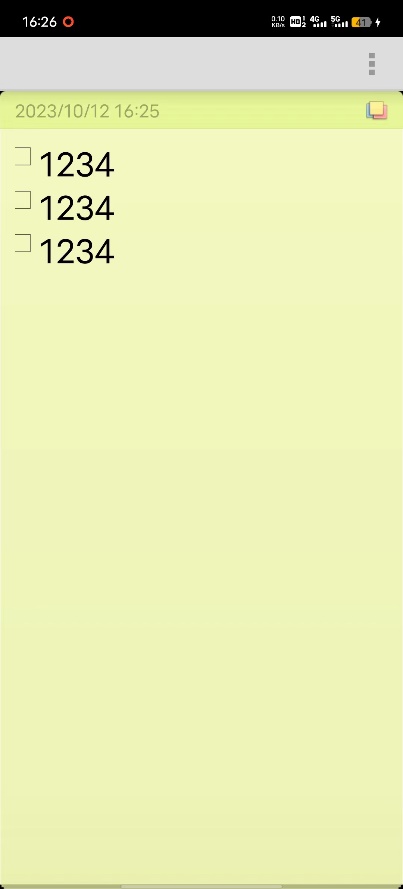
1. 功能十三：清单模式
   1. **功能介绍**

**功能概述：**进入清单模式，每一列变成待完成事项

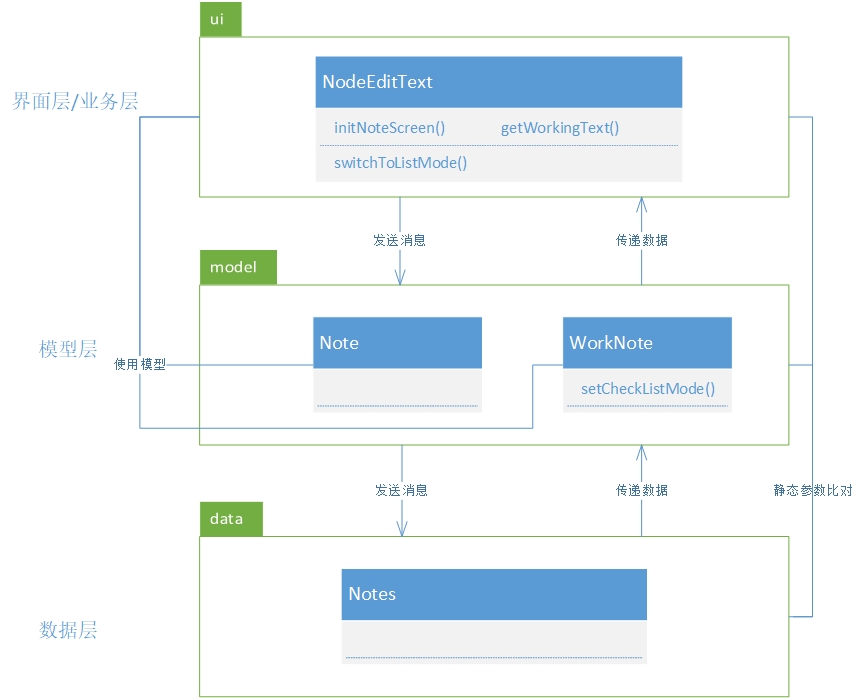
**操作步骤：**便签编辑页面右上角进入菜单，选择“进入清单模式”，选择后，在便签的每一行（段内部的自动换行除外）行首出现一个勾选框，用于当前便签下标记某事项的完成情况。如果该事项已完成，则用户在勾选框中轻触，此时勾选框中将出现一个对勾，框后的陈述文字被添加了中央删除线。（外侧文件夹并不能显示事项的完成状态，可优化）同理，在不需要清单模式的时候，便签编辑页面的右上角选择“退出清单模式”即可。

* 1. **业务请求与响应过程**

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

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

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

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

* 1. **类的作用**
     1. **界面层相关的类**

**1、类名：NoteEditText**

1. **类的作用：**

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

1. **类的精读：**

NoteEditText类的成员

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

NoteEditText类中的方法

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

* + 1. **模型层相关的类**

**1、 类名：**Note

1. **类的作用：**

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

1. **类的精读：**

Note类的成员

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

Note类的方法

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

* 1. **类名：**WorkingNote

1. **类的作用：**

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

1. **类的精读：**

WorkingNote类的成员

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

WorkingNote类的方法

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

* + 1. **数据层相关的类**

**1、****类名：Notes**

1. **类的作用：**

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

1. **类的精读：**

Note类的成员

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

Notes类没有方法