SmartReader

——基于Java的PDF阅读器（集成读书笔记功能）

学院： 计算机学院

班级： 110641班

姓名： 余浪

学号： 11061105

目录

[使用提示 3](#_Toc343779094)

[一、 项目背景 1](#_Toc343779095)

[二、 需求分析 1](#_Toc343779096)

[1. 面向的用户群体 1](#_Toc343779097)

[2. 软件功能要求 1](#_Toc343779098)

[三、 软件设计 2](#_Toc343779099)

[1. 设计思想 2](#_Toc343779100)

[2. 软件结构图 2](#_Toc343779101)

[3. 软件处理流程 3](#_Toc343779102)

[4. 软件实现 4](#_Toc343779103)

[**（1）** **源文件结构** 4](#_Toc343779104)

[**（2）** **类、部分字段及重要方法摘要** 4](#_Toc343779105)

[四、 运行结果 11](#_Toc343779106)

[五、 讨论与总结 14](#_Toc343779107)

使用提示

请务必确定PDFRender包已经加入到程序中才可运行！！

读书笔记必须先保存才能导出。

打开读书笔记功能时可能会卡壳，请耐心等待。

1. 项目背景

PDF是现在非常流行的文件格式，有广泛的应用，很多文件都是以pdf格式发布的。在众多pdf阅读软件中，adobe reader算是用户群最广的一款软件。然而，adobe reader虽然功能强大，然而在某些方面仍然差强人意。就作者的使用体验来说，adobe reader的书签功能十分差强人意，另外虽然最近的版本开始支持添加批注，然而就阅读体验来说，批注嵌入在pdf文件中，无法独立导出，是附属点缀，并不能满足做读书笔记的需求。因此，作者开发了一款基于java的pdf阅读器，实现了阅读器的基础功能，同时集成了读书笔记的功能，可以添加保存导出读书笔记。

1. 需求分析
2. 面向的用户群体

本软件主要针对需要阅读pdf文件的用户，尤其是阅读pdf电子书，在阅读同时喜欢写批注做读书笔记的用户群。

1. 软件功能要求

本软件主要功能是实现pdf文件的阅读、打印以及添加读书笔记功能。具体功能分析如下：

1. 首先要实现pdf文件的阅读，第一步就要选择打开文件。若直接让用户在繁多的文件中找pdf文件，若格式不正确再返回一个错误信息，这显然不够人性化。因此本软件首先要在选择文件时过滤掉其他格式的文件，只可见文件夹和pdf文件，给用户更好的体验。
2. 在实现打开文件功能之后，需要提供基本pdf阅读功能。为改善用户体验，要使内容界面能逐渐自动向下移动，与用户阅读速度同步，而不需用户一遍阅读一遍操作滚动条，使阅读器更切合用户需求。
3. 除了顺序阅读，还需要提供其他基础阅读功能，如手动换上下页，跳转，放大缩小页面，全屏阅读等。
4. 在实现基本pdf阅读器应该有的功能之外，要在阅读器中添加写读书笔记的功能模块，模块要支持编辑修改导出等功能。
5. 在打印文件功能模块，要实现页面设置，打印等功能。

功能框架见如下表格

功能表

|  |  |
| --- | --- |
| **功能** | **子项目** |
| PDF阅读基础功能 | 打开文件时筛选文件格式 |
| 阅读过程自动下移内容面板 |
| 页面跳转，上下页切换 |
| 调节内容显示大小 |
| 全屏阅读 |
|  |
| 打印文件功能 | 页面设置 |
| 打印输出 |
|  |
| 读书笔记功能 | 编辑读书笔记 |
| 导出读书笔记 |
| 保存读书笔记 |
| 删除读书笔记 |

1. 软件设计
2. 设计思想

本软件的设计大体分为三个部分。第一部分是对软件主界面的开发，包括图形界面的设计、各按钮的监听器设置以及内容面板的动态滚动；第二部分是菜单条下拉菜单功能的开发。第三部分是读书笔记模块的功能开发。

软件通过调用Java的PDFReander包来实现对PDF文件的解析，同时利用按钮及面板上添加的事件监听器，通过调用PDFReander提供的方法来响应用户相应的操作。而读书笔记模块则是一个相对独立的功能模块，主体读书笔记编辑区域其实即为一个TextField，通过调用相应的FileOutputStream及FileInputStream类来完成读书笔记的编辑，操作及导出等。读书笔记通过日期来作为关键字存储。

1. 软件结构图

调用

响应操作

pdffile类实现相应操作

软件运行调用关系



软件包之间的组织关系

1. 软件处理流程



1. 软件实现
2. **源文件结构**

|  |  |
| --- | --- |
| **包（Package）** | **类（Class）** |
| main  主界面图形和相关功能的实现 | MainFrame、JPMouseAction、JPMouseMotionAction、  PageBuilder、PerformFullScreenMode |
| resource  程序用到的图片资料 |  |
| util  可供调用的工具类，封装了相应方法 | Buttonactivator、Buttoninit、CreatecdIcon、Fileoperator、GetPage、Pagecontroller、PrintThread |
| activity | SimulateActivity |
| note  用于处理读书笔记模块的类 | DateTime、LeftPane、MonthPane 、Motto、NoteBook、NotePane、YearMonth |

1. **类、部分字段及重要方法摘要**

* **main包**
* MainFrame 类

软件的主面板及相关功能实现

***重要的方法：***

* JMenuBar CreateMenuBar()

用于创建主面板上部的菜单条以及相应选项的下拉菜单，返回一个创建好的菜单条

* JToolBar CreateToolBar()

用于创建位于菜单条下方的工具栏的方法，返回一个创建好的工具条

在工具栏中添加相应的操作按钮之外，还添加显示当前页与总页数的JLabel以及填写要跳转的目标页数的JTextField

调用resource中的图片来创建按钮，同时关联按钮对应的action

* void note()

调用util包中NoteBook类构造方法，新建一个读书笔记面板

此方法在用户点击noteButton时将被调用

* void doFirst(), void doLast(), void doNext(), void doPrev()

用于pdf阅读的页数操作方法，分别实现回首页，转到末页，下一页，上一页

这组方法在用户点击firstButton,lastButton,nextButton,backButton时将分别被调用

* void doFullScreen()

用于实现全屏阅读的方法，进入全屏模式

此方法在用户点击fullScreenButton时将被调用

* void doOpen()

打开本地pdf文件的方法，大量调用PDFRender包中提供的类及方法实现

同时，方法还调用util中Buttonactivator封装的方法使工具栏中的按钮可用

* void gotoPage(int pagenum)

用于实现跳转到相应页数的方法，同时检查输入的pagenum是否合法，避免非法页数出现

* void doClose()

用于关闭当前打开的文件的方法

* void doPageSetup(), void doPrint()

用于实现页面设置，打印的方法

这些方法都在菜单栏中File的下拉菜单中触发

* void KeyPressed()

用于响应用户的键盘操作，实现软件的快捷键

***重要的对象：***

* jpmain

放置主内容的面板

* fullscreen

全屏阅读的面板

* thumbs

放置缩略图的面板

* pdffile

PDFRender包中的获取pdf文档的对象，软件对本地pdf文件的操作依赖于此

* pdfFilter

用于在选择打开文件时过滤掉非pdf或文件夹的文件

* curpage

用于控制当前内容的页码

* isDragged

用于判断用户鼠标是否点击在内容面板中，即面板是否被拖拽

* JPMouseAction类（final）**.**

响应鼠标动作的类

***重要的方法：***

* void mouseReleased(java.awt.event.MouseEvent e)，
* void mousePressed(java.awt.event.MouseEvent e)

这一组方法用于相应用户的鼠标动作，在鼠标点击和放开时分别设置鼠标形状

同时对于鼠标点击动作，同时不断记录鼠标位置，以实现面板随鼠标移动而移动

* JPMouseMotionAction类（final）

响应鼠标拖拽的类

***重要的方法：***

* void mouseDragged(java.awt.event.MouseEvent e)

用于响应鼠标拖拽，随着鼠标移动不断刷新内容面板位置

* **util包**
* Buttonactivator类

激活按钮的类，即用于处理按钮是否可用的类

***重要的方法：***

* static void activator(JButton jb)

封装用于激活按钮的方法，传递进JButton之后设为可用

当文件打开后再调用此方法

* Buttoninit类

初始化按钮的类，即调用resource中图片创建按钮的类

***重要的方法：***

* static void init(JButton jb,String st,String pt)

用于新建按钮封装的方法。将JButton初始设置为不可用，提示文本设置为st，调用名字为pt的图片作为按钮的样子。

* CreatecdIcon类

调用resource中图片创建按钮的类，封装了相应方法，使得指定图片不需要重复输入相同的路径名。

***重要的方法：***

* static ImageIcon add(String ImageName)

用于添加图片文件作为按钮，路径是一样的，只需寻找对应的图片名即可，为mainframe添加图片提供方便。

***重要的对象：***

* IconUrl

用于记录打开的目标文件的路径

* icon

方法创建的按钮对象，用于返回

* GetPage类

用于处理关于文件页码即当前页类

***重要的方法：***

* static String showPage (int curpage, PDFFile pdffile)

用于显示当前页码与总页码的方法，用于向工具栏中显示页码的JLabel对象返回需要的current page/ total pages格式的字符串，用于显示在工具栏中。

* Fileoperator类

用于操作文件的类，可继续扩展其功能，目前只实现文件页面设置的操作。

***重要的方法：***

* static void setup(PageFormat pformat)

用于打印时页面设置的方法，封装的方法在菜单中File下拉菜单中被调用

* Pagecontroller类

用于处理页码相关问题的类，主要用于限制不合法页码出现

***重要的方法：***

* static int checkPage(int pagenum, PDFFile pdffile)

封装的用于检测页码是否合法的方法，将不合法的页码修改为最近的值。（负数则修改为1，超过总页数的则修改为当前文件的末页）

* PrintThread类

打印线程，在点击Print时被调用。

* **Note包**

以下几个类均为实现读书笔记模块的类，读书笔记模块宏观上分左侧的日历面板（LeftPane），以及右侧的读书笔记编辑操作面板（NotePane）来实现。

* DateTime类

用于获取当前日期，以及处理日期相关问题的类

用于读书笔记模块左边面板的日历处理

***重要的方法：***

* static int getYear()， static int getMonth(), static int getDay()

这组封装的方法用于获得当前系统的年，月，日

在涉及日期处理的方法中会被频繁调用

* static int getMonDay(int year, int month)

获取当前月份有几天的方法。方法中考虑了润年及不同月的区别，返回相应月份对应的天数

天数将被用于读书笔记模块日历的生成。

***重要的对象：***

* today

记录当前的日期的Calendar变量。声明为static。

* LeftPane类

用于创建读书笔记模块左侧日历及时间面板的类，调用其他工具类来完成功能

***重要的方法：***

* LeftPane()

类的构造方法，用于新建左侧面板必要组件并将其添加进容器，同时设置布局与添加划分。

具体组件的定义在其他的类中实现。

* MonthPane类

用于处理读书笔记模块的日历处理类

***重要的方法：***

* MonthPane()

类的构造方法，用于创建日历面板，利用for循环来创建一系列显示日期的JTextField。

* void mcInit()

用于初始化日历面板的相关的JLabel以及JTextField，同时设置格式，字体，布局等

* void arrangeNum(int year, int month)

用于设置日历中日期显示以及设置特殊日期显示的方法

利用DateTime的方法来确定年月对应的日历

***重要的对象：***

* first，days

用于存储当前年月的月首在周几，以及一个月共有多少天

在生成日历的时候被用到

* title[]

用于存储在日历上访显示周几的JLabel

* Motto类

可扩展的类，用于在读书笔记模块中添加自己个性化图片

* NoteBook类

读书笔记模块的主框架，处理读书笔记模块的创建，以及相关功能的实现。

***重要的方法：***

* NoteBook()

完成面板创建及属性设置，以及相关组件的初始化及设置

* void bookInit()

初始化面板（调用leftpane和notepane来实现），同时获取当前年月日存进类变量中。

* void addEvent()

为左侧日历面板加事件监听，包括为年月微调器增加事件监听，和显示日期的JTextField用循环方法添加鼠标事件监听。

* void stateChanged(ChangeEvent e)

用于相应用户调整年月的方法，相应地调整日历以及刷新右侧读书笔记面板

* void noteBookRefresh()

刷新右侧读书笔记面板，包括刷新上部显示的日期以及读书笔记编辑区域的内容。

***重要的对象：***

* lp,np

新建的左侧日历面板以及右侧的读书笔记编辑面板

* year, month, day

用于记录正在处理的日期

* NotePane类

用于处理右侧读书笔记面板的类，完成初始化及相关设置。

***重要的方法：***

* NotePane()

用于创建右侧的读书笔记面板，同时完成相关设置，如布局设置等；创建并初始化各个组件

* void noteInit()

用于初始化读书笔记编辑面板，新建编辑区，更新日期，设置字体等。

* void buttonInit()

用于初始化读书笔记面板的按钮，完成新建按钮，设置显示文本，向面板中添加按钮等功能。

* void addEvent()

为读书笔记编辑面板下方的三个按钮添加事件监听

* void actionPerformed(ActionEvent e)

为三个按钮和相应的方法建立关联，在用户点击时调用相应方法

* void setDateInfo(int y, int m, int d)

设置读书笔记面板上方日期显示的方法

* void refreshContent(int year, int month, int day)

更新指定年月日的读书笔记的内容的方法（读书笔记是以日期为关键字组织的）

* void save(int year, int month, int day)
* void delete(int year, int month, int day)
* void export()

实现读书笔记保存，删除，导出功能的方法，在用于点击读书笔记面板下方三个按钮时将分别被调用。

***重要的对象：***

* Hashtable table

用于存放读书笔记的哈希表，读书笔记的存储同对应日期关联

* JLabel dateInfo

用于显示当前日期的JLabel对象，位于读书笔记编辑区域的上方

* YearMonth类

用于控制左侧日历面板的类

***重要的方法：***

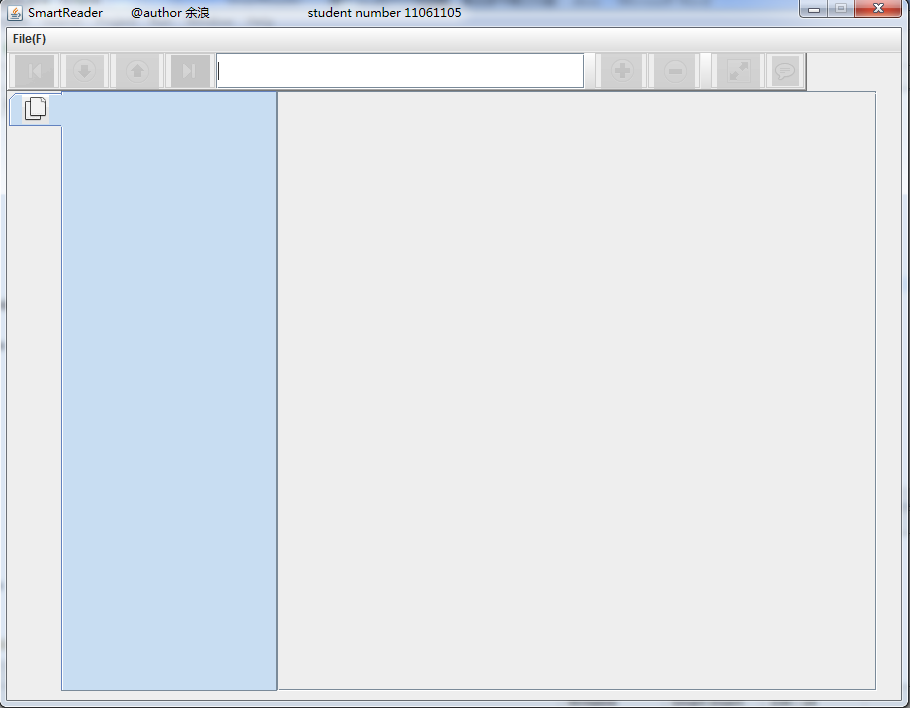
* void ymInit()

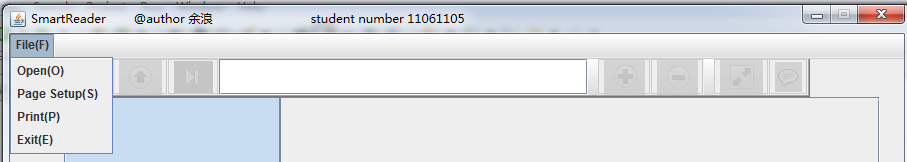
设置当前年月的方法，同时加入年月微调器，并对其进行基本设置。

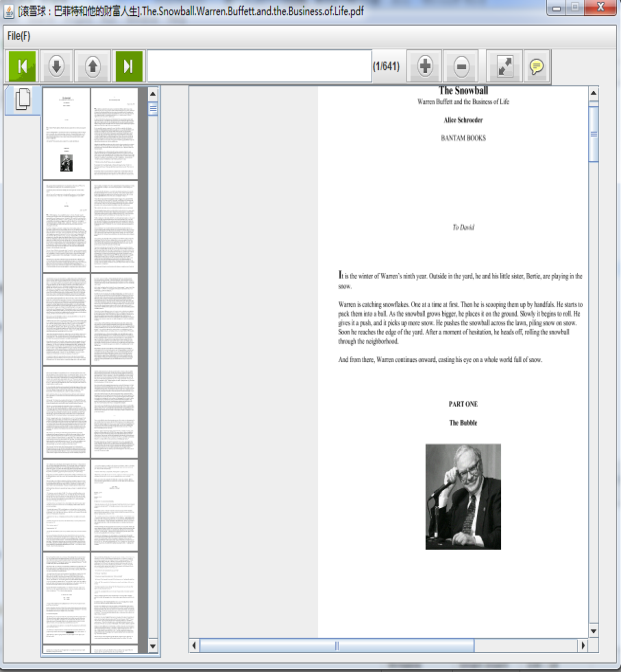
***重要的对象：***

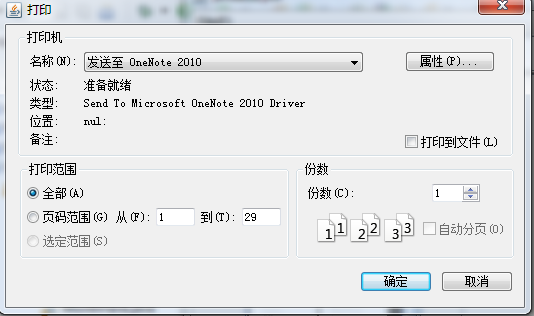
* JSpinner showYear，showMonth

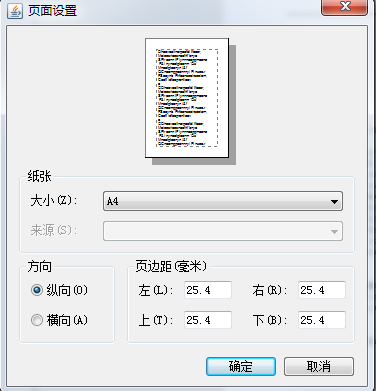
用于日历面板下方的年月微调器

1. 运行结果











1. 讨论与总结

本软件用于pdf文件的阅读，读书笔记的编辑与导出，功能实用，界面较为人性化。但是限于时间仓促，自身水平有限，一些功能存在漏洞，另外软件的性能不高。就目前情况而言，已知的不足主要有：

1. 程序运行不够流畅，尤其是在打开读书笔记模块时容易出现卡壳。
2. 在不关闭整个软件的情况下可以实现再次打开文件时记住最近访问的目录，然而没有配置文件导致一旦关闭整个软件，再次运行时打开文件的默认目录不再是上一次打开的目录。
3. 软件的界面大小不支持调节，为固定大小。

作者在读书笔记模块中倾注了大量心血，希望能在Adobe Reader现有功能的基础上提供更实用的功能。程序具有很好的扩张性，可继续添加其他的功能，例如标注，高亮等功能并在工具栏中添加相关按钮。

在此次大作业的完成过程中，我不但对Java的许多特性，比如swing图形界面开发，线程的使用程等有了较为深入的了解，更是逐渐理解了Java程序设计的思想，包括私有化数据，方法以及封装，继承等。特别是在对PDF文件的处理方面，对Java提供的PDFRender包有了更深刻的认识，熟悉了其中提供的类和方法的使用。首次驾驭这种规模的程序，让我对类的组织与类之间的相互关系有了更深刻的理解。总之，经过这次的实践，我对Java有了更深刻的认识和理解，编程能力有了很大的提升。