

笔记本——设计报告

Java 课程项目

小组成员：

姓名：梁毓颖 学号：16340134

姓名：沈大伟 学号：16340190

中山大学软件学院

目录

项目介绍.....	2
提出背景.....	2
运用的知识点.....	2
系统设计.....	2
总体设计思路.....	2
类的设计.....	3
功能的实现.....	10
已实现的功能.....	10
待完善的功能.....	11
问题及解决方案.....	12
项目总结.....	12

项目介绍

该项目所完成的是一个类似于印象笔记的笔记本，用户注册登录后，可在程序中编写自己的笔记，对笔记进行分类管理，然后将笔记保留到本地文件或者网络中。

提出背景

我们在使用印象笔记的时候，感觉印象笔记这种可以随时记录笔记和管理笔记的功能，很适合我们大学生。很多时候，我们用手写来做笔记的时候，往往速度跟不上，而使用印象笔记这样电子版的笔记本，不仅能大大增加我们做笔记的速度，而且还很方便我们对笔记进行分类管理。另外，可插入图片这个功能也使我们做的笔记更加生动和完善。但是，印象笔记的普通用户一个月最多只能享受 60M 的笔记，要想有更好的体验，就必须得开通会员。而且，印象笔记只支持将笔记保存在网络上，而不能保存在本地。所以，为了让我们记笔记时拥有更好的体验，我们决定自己弄一个笔记本。

运用的知识点

基本知识点：封装、继承、（多态）、接口、异常处理、Socket 网络编程

Java Swing 图形界面

数据库：MySQL

系统设计

总体设计思路

首先，将整个项目划分为两个部分，客户端和服务端，其中，客户端采用图形界面。在客户端，采用三层架构的架构模式，将程序分为数据访问层、业务逻辑层、界面层，在实际开发的时候，将这三个层次的具体实现分开，然后通过接口将三个层次连接起来。服务端则不提供界面。


实现的时候，先将服务端实现，然后再实现客户端。实现客户端，则先将数据访问层和业务逻辑层的类实现出来。对于界面层，则分开各个组件单独实现，最后再将全部组件组装起来。

为了方便代码的管理，将不同模块和不同作用的类分开在不同的包。例如，服务端的类放在 `server` 包，客户端的底层放在 `client` 包，异常类放在 `client.error` 包，数据处理类放在 `client.data` 包，客户端和服务端的交互的类放在 `client.io` 包，界面层的类分别放在


client.ui.frame 包, client.ui.menu 包, client.ui.panel 包。

类的设计

客户端的数据处理类:

 DataControl
<ul style="list-style-type: none">• flag : boolean = false• noteGroups : Vector<NoteGroup>
<ul style="list-style-type: none">• getInstance () : DataControl• openSC ()• closeSC ()• getFlag () : boolean• getNativeNoteGroup () : Vector• getMySQLNoteGroup () : Vector• saveNativeNoteGroups ()• readNativeNoteGroups ()• createNativeNoteGroup (groupName : String,) : boolean• deleteNativeNoteGroup (groupName : String,) : boolean• createNativeNote (groupName : String, noteName : String, path : String,) : Note• deleteNativeNote (groupName : String, noteName : String,) : boolean• aboutMySQLNoteGroup (groupName : String, f : boolean,) : boolean• aboutMySQLNote (groupName : String, noteName : String, f : boolean,) : Note• readFileInMySQLGroup (chooseNg : NoteGroup,) : NoteGroup• download (groupName : String, noteName : String, path : String,) : File• saveFileInMySQLGroup (chooseNg : NoteGroup)• upload (groupName : String, noteName : String, file : File)• changeMySQLNoteName (groupName : String, oldNoteName : String, newNoteName : String,) : Note• changeMySQLNGName (oldNGName : String, newNGName : String,) : NoteGroup• login (email : String, password : String,) : boolean• register (email : String, password : String,) : boolean• getUser () : User• saveUser () : boolean

笔记本组的笔记显示类:

 MyNoteShowPane
<ul style="list-style-type: none">• showPane : JPanel• textpanels : Vector<JTextPane>• editorKit : HTMLEditorKit• nameLabel : JLabel• sql_flag : boolean = false
<ul style="list-style-type: none">• changeNg (otherng : NoteGroup, othersql_flag : boolean)• saveNg ()• changeNgName ()• getNg () : NoteGroup• addJSeparator (height : int)• showNoteGroupName ()• getInstance () : MyNoteShowPane• changeShowPane ()

工作区类:

MyUtilPane
<ul style="list-style-type: none"> ◉ pmenu : JPopupMenu ◉ ngPmenu : JPopupMenu ◉ nPmenu : JPopupMenu ◉ noteItems : JMenuItem[*] { unique } ◉ image1 : Image ◉ image2 : Image ◉ image3 : Image ◉ image4 : Image ◉ image5 : Image ◉ image6 : Image ◉ image7 : Image ◉ image8 : Image ◉ image9 : Image
<ul style="list-style-type: none"> ● getInstance () : MyUtilPane ● setNoteItems () ● createNPmenu () ● createNgPmenu () ● createPmenu () ● addNL ()

发送邮件的类:

CheckMail
<ul style="list-style-type: none"> ◉ myEmailAccount : String = syslisp_chain@163.com { readonly } ◉ myEmailPassword : String = chain0981101082 { readonly } ◉ myEmailSMTPHost : String = smtp.163.com { readonly } ◉ receiveMailAccount : String ◉ subjectName : String ◉ content : String
<ul style="list-style-type: none"> ● createMimeMessage (session : Session,) : MimeMessage ● main (args : String[*])

用户信息显示和更改:

UserMessageDialog
<ul style="list-style-type: none"> ◉ serialVersionUID : long = 0 { readonly } ◉ font : Font ◉ in : Insets ◉ bg : ButtonGroup ◉ maleButton : JRadioButton ◉ femaleButton : JRadioButton
<ul style="list-style-type: none"> ● addSexChoose () ● addLabel (string : String, i : int, j : int, k : int, l : int)

注册界面：

RegisterDialog
<ul style="list-style-type: none">• curSec : int• frame : JFrame• password : JPasswordField• mail : JTextField• verifyCode : JTextField• lb_pwd : JLabel• lb_m : JLabel• lb_code : JLabel• error_message : JLabel• b1 : JButton• b2 : JButton• getEmail : JButton• userMail : String• userPassword : String• yanzhengCode : String = 123
<ul style="list-style-type: none">• init ()• sendEmail (accountMail : String,) : String• checkEmail (email : String,) : boolean

字体设置的类：

FontFrame

- `menuFont` : Font
 - `defaultPaneFont` : Font
 - `fontPane` : JPanel
 - `buttonPane` : JPanel
 - `presentPaneFont` : Font
 - `demoLabel` : JLabel
 - `presentFontFamaly` : String = 楷体
 - `presentFontStyle` : int = 0
 - `presentFontSize` : int = 14
 - `defaultFamalyItem` : int = 0
 - `defaultStyleItem` : int = 0
 - `defaultSizeItem` : int = 0
-
- `addButtons (constraints : GridBagConstraints)`
 - `addFontDemo (constraints : GridBagConstraints)`
 - `addFontFamily (constraints : GridBagConstraints)`
 - `addFontStyle (constraints : GridBagConstraints)`
 - `addFontSize (constraints : GridBagConstraints)`

客户端文件处理:

FileClient

- `path` : String = data/Client/new.txt
 - `socket` : Socket
 - `dis` : DataInputStream
 - `dos` : DataOutputStream
 - `fos` : FileOutputStream
 - `fis` : FileInputStream
-
- `close ()`
 - `send (id : String, noteName : String, file : File, isMessage : boolean)`
 - `download (email : String, groupName : String, noteName : String, mypath : String,) : File`
 - `upload (email : String, groupName : String, noteName : String, file : File, isExit : boolean)`
 - `sendFile (file : File)`
 - `receiveFile (mypath : String,) : File`
 - `receive (mypath : String,) : boolean`

字符串查找:

FindStringDialog
<ul style="list-style-type: none"> serialVersionUID : long = 0 { readonly } font : Font in : Insets back : boolean = false isSelect : boolean = false error : JLabel
<ul style="list-style-type: none"> scope () direction () addLabel (string : String, i : int, j : int, k : int, l : int) main (args : String[*])

客户端:


SystemClient
<ul style="list-style-type: none"> socket : Socket is : ObjectInputStream os : ObjectOutputStream
<ul style="list-style-type: none"> register (email : String, password : String,) : boolean login (email : String, password : String,) : User saveUser (user : User,) : boolean aboutNoteGroup (email : String, noteGroupName : String, f : boolean,) : User aboutNote (email : String, noteGroupName : String, noteName : String, f : boolean,) : User changeNGName (email : String, oldNGName : String, newNGName : String,) : User changeNGNoteName (email : String, groupName : String, oldNoteName : String, newNoteName : String,) : User close ()

登录类和笔记基本类:


LoginFrame
<ul style="list-style-type: none"> frame : JFrame account : JTextField password : JPasswordField enter : JButton reload : JButton quit : JButton error_message : JLabel
<ul style="list-style-type: none"> loginIn (index : int) main (args : String[*])

Note
<ul style="list-style-type: none"> serialVersionUID : long { readonly } name : String path : String createDate : Date changeDate : Date
<ul style="list-style-type: none"> getCreateDate () : Date getChangeDate () : Date setChangeDate () getName () : String setName (name : String) getPath () : String setPath (path : String) toString () : String

笔记本类:

 NoteGroup
<ul style="list-style-type: none"> • <u>serialVersionUID</u> : long { readonly } • name : String • notes : Vector<Note>
<ul style="list-style-type: none"> • getName () : String • setName (name : String) • addNote (note : Note) • removeNote (fileName : String,) : boolean • getAllNote () : Vector • hasNote (noteName : String,) : boolean • getNote (noteName : String,) : Note • toString () : String

用户类:

 User
<ul style="list-style-type: none"> • <u>serialVersionUID</u> : long = 0 { readonly } • email : String • password : String • name : String • sex : String • friends : Vector<String> • noteGroups : Vector<NoteGroup>
<ul style="list-style-type: none"> • toString () : String • getPassword () : String • setPassword (password : String) • getName () : String • setName (name : String) • getEmail () : String • setEmail (email : String) • getSex () : String • setSex (s : String) • addNoteGroup (noteGroup : NoteGroup) • addNote (noteGroupName : String, note : Note) • removeNoteGroup (name : String) • removeNote (noteGroupName : String, fileName : String) • getAllNoteGroup () : Vector • hasNoteGroup (noteGroupName : String,) : boolean • hasNote (noteGroupName : String, noteName : String,) : boolean • getNote (noteGroupName : String, noteName : String,) : Note • getNoteGroup (noteGroupName : String,) : NoteGroup • addFriend (friend : String) • removeFriend (friend : String) • getAllFriend () : Vector

数据库管理类:

MySQLManager
<ul style="list-style-type: none"> driver : String = com.mysql.jdbc.Driver url : String = jdbc:mysql://localhost:3306/notepad user : String = root password : String = 960919 connection : Connection statement : Statement resultSet : ResultSet ps : PreparedStatement
<ul style="list-style-type: none"> getInstance() : MySQLManager getConnection() add(email : String, password : String,) : boolean remove(email : String, removeFile : boolean,) : boolean query(email : String,) : User update(email : String, type : String, obj : Object,) : boolean addFile(email : String, groupName : String, noteName : String, file : File,) : boolean readFile(email : String, groupName : String, notename : String,) : File removeFile(email : String, groupName : String, noteName : String,) : boolean changeNote(email : String, groupName : String, noteName : String, newName : String, f : boolean,) : boolean updateFile(email : String, groupName : String, noteName : String, file : File,) : boolean queryMessage(email : String,) : File addMessage(email : String, fileName : String, file : File,) : boolean

服务端文件处理线程类:

FileSocketThread
<ul style="list-style-type: none"> path : String = data/Server/ socket : Socket id : String dis : DataInputStream dos : DataOutputStream fos : FileOutputStream fis : FileInputStream message : String
<ul style="list-style-type: none"> getId() : String sendFileToFriend(id : String, file : File, isMessage : boolean,) : boolean sendFile(file : File, isMessage : boolean) receiveFile(fileName : String,) : File run()

服务端线程类:

SystemSocketThread
<ul style="list-style-type: none"> socket : Socket ois : ObjectInputStream oos : ObjectOutputStream message : String email : String password : String
<ul style="list-style-type: none"> dealWith() run()

主面板编辑区类:

```
MyNotePa
* menuPan : JPanel
* toolPan : JPanel
* sp : JScrollPan
* textPan : JTextPan
* sc : StyleCenter
* doc : DefaultStyledDocum
* unde : UndoManag
* cutActio : Actio
* copyActio : Actio
* pasteActio : Actio
* alignmentBc : JComboBox<Strin
* fontFamiliesBc : JComboBox<Stein
* fontSizesBc : JComboBox<Strin
* leftIndentButt : JButte
* deleteIndentButt : JButte
* boldButte : JButte
* italicButte : JButte
* underlineButte : JButte
* insertBreakButt : JButte
* subscriptButt : JButte
* superscriptButt : JButte
* imageButte : JButte
* file : File
* leftInden : int - 5
* fileName : JTextField
* noteGroupNam : JLabel
* noteNew : JLabel
* hideFile : boolean - fal
* isSql : boolean - fal
* getInitName() : MyNotePa
* setNoteAndGrou(note : Note, noteGroupName : String, isSql : boolean, file
* changeNoteGrou(oldName : String, newName : Str
* addLabel(text : String, x : int, button : JBu
* addTextPan()
* addToolPan()
* addMenuPan()
* upFile()
* readFile(file : File
* removeFile(fileName : Strin
* isEdit() : boolean
* getUndoActio() : UndoActio
* getRedoActio() : RedoActio
* getCutActio() : Actio
* getCopyActio() : Actio
* getPasteActio() : Actio
* doLeftIndentButt()
* doDeleteIndentButt()
* doBoldButt()
* doItalicButt()
* doUnderlineButt()
* doInsertBreakButt()
* doSubscriptButt()
* doSuperscriptButt()
* find(str : String, newStr : String, back : boolean, isSelect : boolean
* openWithStyle()
* saveWithStyle()
* startNewDocum()
* openDocum()
* saveDocum() : boolean
* saveDocument() : boolean
* exit()
* clear()
* selectAll()
* addData()
* stringToUppr()
* stringToLowe()
* changeStrin()
* setAlignmentBc(i : int
* setFontFamiliesB(i : int
* setFontSizesB(i : int
* setFontFamilyFont(f) : <...
```

功能的实现

已实现的功能

- 1、用户登录：客户端将账号和密码传至服务端和数据库进行匹配。
- 2、用户注册：客户通过邮箱注册，服务端向客户邮箱发送验证码，当客户输入的验证码正确时，再和数据库比对，若邮箱还没注册，则注册成功。

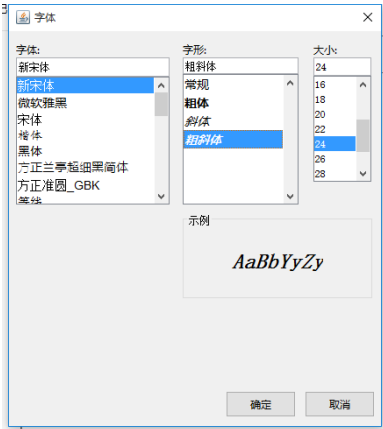


笔记本

- 3、菜单的功能：

文件 编辑 格式 其他 帮助

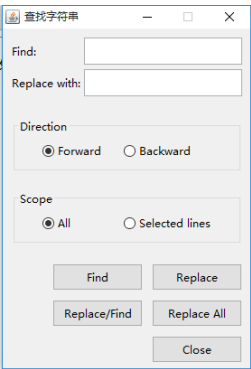
- 4、文件选项中的功能：新建笔记、打开笔记、保存、另存为、带格式保存、带格式读取、退出、文件上传、文件下载。
- 5、编辑选项中的功能：撤销、粘贴、复制、查找、替换、大小写转换、时间插入。
- 6、格式选项中的功能：字体设置、段落设置、样式设置。



- 7、其他选项中，有外嵌的编辑器、设计器和阅读器，但这些阅读器尚未和笔记本相关联。
- 8、帮助选择中，有快捷键提示和用户信息显示和更改等。



- 9、在笔记本组或笔记本中，可通过鼠标右击，弹出菜单，菜单上有新建笔记本、新建笔记本组、重命名、删除笔记等功能。
- 10、自己实现的编辑器，编辑器中除了基本的字体大小、格式、颜色等的设置外，还可设置文档的缩进，可在笔记中插入图片等。此外，实现了字符查找功能(快捷键为 Ctrl+F)，可精确查找笔记中的一段字符，并且可对字符内容进行替换，界面如下：



待完善的功能

- 1、菜单中的页面设置功能尚未完成。

- 2、用户间的聊天室功能和笔记共享功能，实现了逻辑处理层，UI 层尚未完成。
- 3、将外嵌的编辑器、设计器和阅读器和笔记本的内容连接起来。

问题及解决方案

- 1、无法将保存进数据库的文本修改。解决方案：删掉原本的文本，将修改后的文本重新保存进数据库。
- 2、文件的格式、字体设置以及图片等无法保存。解决方案：将整个面板 `JTextPane` 序列化保存进 `TXT` 文档，打开时再将已序列化的内容读取出来。
- 3、不知道怎样用一个数据处理的中心来同时操做处理三个面板以及三个面板之间如何相互联系。解决方案：用单例模式。
- 4、为了添加项目符号和编号等格式，把 `JTextPane` 从 `TXT` 格式改为 `TXT/HTML` 格式，发现 `HTML` 格式与字体设置有很多不兼容，而且项目符号和编号也难以处理，字符串的查找和替换也出现了问题。解决方案：改回 `TXT` 模式，放弃添加项目符号，编号等格式。
- 5、注册时向用户的邮箱发送验证码时，有时候会被邮箱官方网站判定是垃圾邮件而无法发送。该问题尚未解决，从网上查找的方案是复制一份发送的邮件给自己，但是，尝试过后，发现这个问题还是存在。

项目总结

一开始决定弄这个笔记本的时候，以为难度不会很大，但是，真正实现起来才发现，难度要比自己想象中高出许多。首先是设计的时候没有完全设计好，实现过程中突然想到一些新的功能想要加入到程序中，然后，又不得不把整个程序重新编写。其次，因为之前对 `java` 的图形界面并不是很掌握，很多时候用到一个功能时得花很长时间去百度或者查 `API` 文档来熟悉某个方法的使用。而且，加上时间方面没处理好，遇到考试周的时候还有挺多想实现的东西没有实现出来，造成了时间冲突，最后，一些之前已经写好了的组件和功能，因为不够时间，没办法组装到程序当中。

虽然看起来这个笔记本应该很简单，但真正实现起来才发现，里面的很复杂的功能虽然没有，但是功能比较多，而且繁杂，有事后往往为了实现这个功能，可能得把整个模块都给改了，造成了很多时间上的浪费。

总的来说，这次项目虽然花费了我们很多时间，也有好几次因为找不出 `Bug` 而抓狂，但通过这个项目，我们不仅掌握好了课堂上学到的关于 `java` 的基础知识，也自学了 `java` 对数据库的处理和 `java` 的图形界面，给了我们很大的收获。