

Email 邮件客户端

0 目录

- 1 个人信息
- 2 项目概览
- 3 功能介绍
- 4 设计模式简介
- 5 程序包组织与类文件描述
- 6 尚未解决的问题
- 7 后期工作

1 个人信息

姓名：紫梦 lan

电子邮件：dongdk.sysu@foxmail.com

2 项目概览

实验题目：Email 邮件客户端

开发环境：win8.1

开发工具：eclipse

开发依赖库：javamail, sqlite-jdbc

JDK 版本：1.8.0_25

开发语言：Java + Python

代码行数：

总共：11086

仅代码：7344

import 语句：653

换行符：1293

单行注释：1280

多行注释：1169

实验进展：

1 14.12.05 - 14.12.07:

部署 project 的开发环境及所需依赖库，熟悉 javamail 的 API.

2 14.12.08 - 14.12.10:

利用 Javamail 完成邮件客户端的界面设计，即用户登录.

3 14.12.10 - 14.12.13:

完成邮件客户端的主界面和发邮件界面， 并确定利用卡片布局来设计客户端的界面切换，如收发邮件界面的切换.

4 14.12.12 - 14.12.18:

完成邮件客户端的发邮件功能及界面，包括正常发邮件，定时发邮件，发附件

5 14.12.15~14.12.22:

完成邮件客户端的收邮件功能及界面，包括预览邮件，查看该邮件，下载附件，转发和回复该邮件.

6 14.12.20 - 14.12.24:

完成邮件内容的简单解析，并放弃卡片布局，实现收发邮件界面和主界面的分离.

7 14.12.23 - 14.12.25:

实现邮件客户端的记事本功能，即界面，简单实现记事本的查看和保存.

8 14.12.15 - 14.12.25:

不断优化代码和界面，实现了日志记录功能

9 14.12.26 - 14.12.28:

持续改进代码

打包代码(jar + exe)

写项目的报告

实验起止时间:

14.12.05 - 14.12.28

3 功能介绍

该邮件客户端实现的功能有:

1 登录

登录界面只是简单地模拟正常的邮箱登录，但是不会记录登录的状态，也就是一次登录就默认登录了，后续的操作都需要重新‘登录’，只是在用户透明的状态下登录，

进行重新认证，如发邮件时，重新连接到用户的邮箱服务器。

2 发邮件

登录成功后，会进入主界面，通过点击主界面左上方的发邮件按钮，来进入发邮件界面，允许用户可以添加一些小文件并进行附件发送，多收件人的邮件发送，定时邮件发送。

3 写记事本

进入主界面，可以点击主界面左下方的日历的某一天，进行记事本的记录，或者点击左上方的写记事本按钮，进行当天的记事本记录。

由于没有时间写数据库的邮件和记事本存储功能，所以就将记事本作为邮件的形式，简单地调用发邮件函数，讲记事本发送到用户的邮箱上。

4 收邮件

进入主界面，可以点击主界面左上方的收邮件按钮，进行预览邮件界面。

每个预览邮件界面只提供 5 条邮件(只是因为 POP3 的邮件解析效率太低，尤其在解析存在附件的邮件)，然后可以双击每条邮件的‘收件人’，‘发件人’，‘主题’和‘接收时间’条目或者右侧的‘查看该邮件’按钮进行详细地查看该邮件。

进入邮件查看界面，可以下载单个小文件附件(双击对应条目即可)或者下载全部附件(这里只可以下载小文件附件)，或者回复发件人，回复所有人，或者转发该邮件。

另外，在这里记事本也会当作邮件的方式查看。

5 收记事本

进入主界面，可以点击主界面左侧的记事本按钮，进入预览记事本界面。

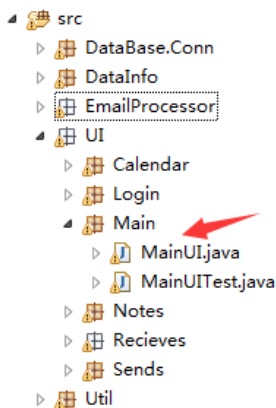
记事本预览界面只会显示记事本邮件，不会显示其他邮件，然后类似邮件预览界面一样，每个预览记事本界面只提供 5 条记事本，然后可以双击每个记事本的‘来自我的记事本’，‘主题’和‘接收时间’条目或者右侧的‘查看该记事本’按钮进行详细地查看该记事本。

然而记事本查看界面只允许用户查看该记事本，而不允许用户对其进行操作(因为 POP3 不会记录用户对邮件操作的状态，也就是不会同步用户对邮件操作的状态同步到邮箱服务器上。

4 设计模式简介

该邮件客户端的代码里面大量使用了单实例模式 (singleton) 和组合模式 (Composite Pattern).

Singleton 是一种创建性模型, 它用来确保只产生一个实例, 并提供一个访问它的全局访问点. 在代码里面 MainUI 类文件(如下图所示), 作为整个邮件客户端的主界面, 这意味着它将是程序运行期间的唯一一个主界面实例. MainUI 控制着收发邮件和记事本界面(尽管代码里面收发邮件和记事本采用了多线程编程).

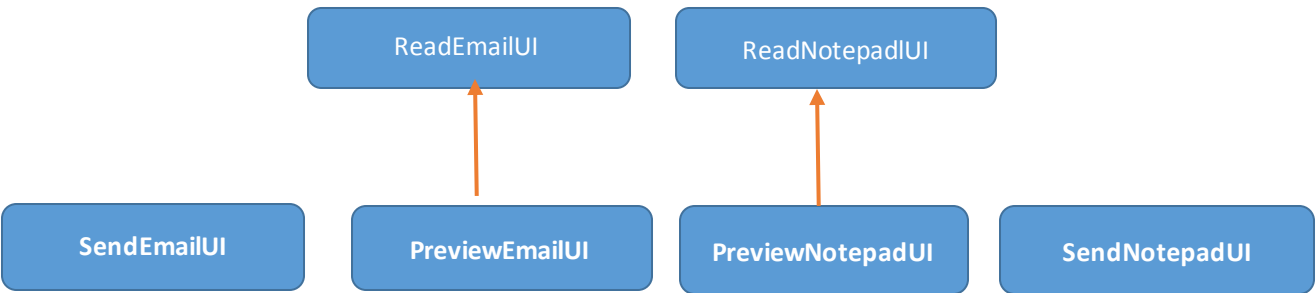


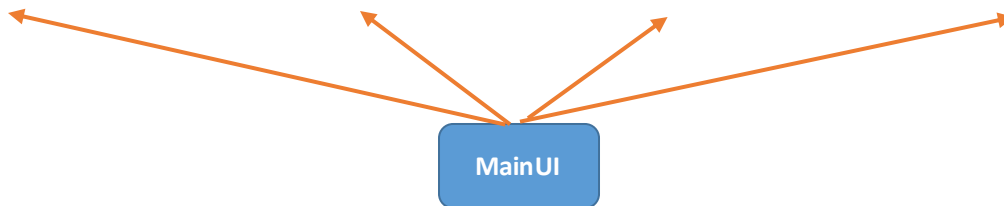
另外 EmailProcessor 里面的(如下图所示)

ReceivedEmailProcessor, ReceivedPop3Processor,
ReceivedIMAPProcessor, SendedSMTPProcessor

都是单实例的具体体现. 显然在收发邮件或记事本的过程中, 它们都是业务逻辑都是一样的, 即向邮件服务器发出连接认证, 然后 push 或者 pull 邮件. 所以它们可以作为单实例存在, 共享与这个程序运行期间.

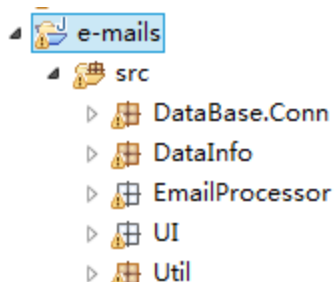
组合模式, 有时候又叫做部分-整体模式, 它使我们树型结构的问题中, 模糊了简单元素和复杂元素的概念, 客户程序可以像处理简单元素一样来处理复杂元素, 从而使得客户程序与复杂元素的内部结构解耦. 如代码里面的 MainUI 类, 它控制着收发邮件和记事本的界面, 使得它们作为 MainUI 里面的视图界面数据成员. 即 MainUI 类将对象组合成树形结构以表示“部分整体”的邮件客户端的界面层次结构. 另外组合模式使得用户对邮件客户端的单个界面对象和组合界面对象的使用具有一致性, 如下图所示:





5 程序包组织与类文件描述

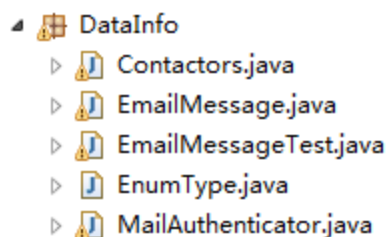
整体的程序包组织与类文件的视图如下：



整体的程序包组织由五部分组成：DataBase, DataInfo, EmailProcessor, UI 和 Util. (其中 DataBase 部分还没有实现). 下面讲详细地介绍 DataInfo, EmailProcessor, UI 和 Util 部分的具体实现和其实现的功能.

1 DataInfo

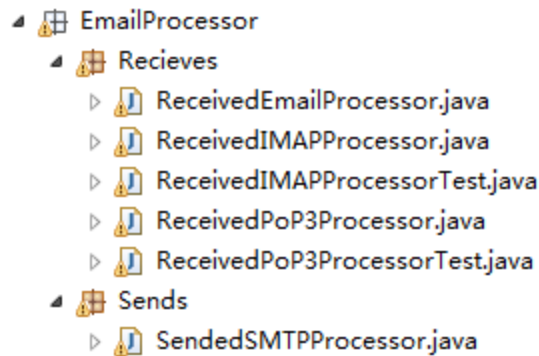
该包由几个数据结构文件组成：



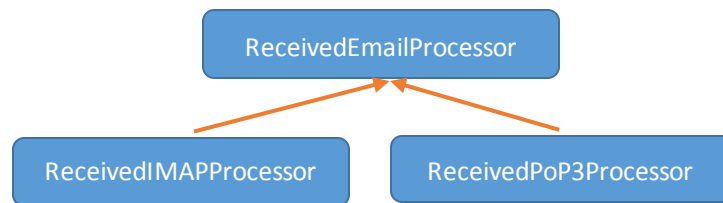
- 1 **EmailMessage** 类作为整个邮件客户端的核心数据结构，作为发邮件，收邮件，发记事本和收记事本的桥梁.
- 2 **Contactors** 则作为联系人的数据结构(暂时没有实现这个功能).
- 3 **EnumType** 则作为一些常量的类文件，用于记录一些配置常量.
- 4 **MailAuthenticator** 则作为连接邮箱服务器时，作为认证的数据结构.

2 EmailProcessor

该包则是收发邮件处理的业务逻辑包，以下所示：



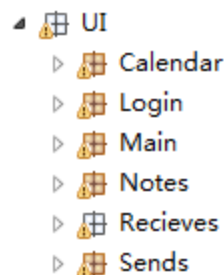
Receives 里面实现了 IMAP 和 POP3 的收邮件功能 (IMAP 目前有个 bug, 暂时用不了).



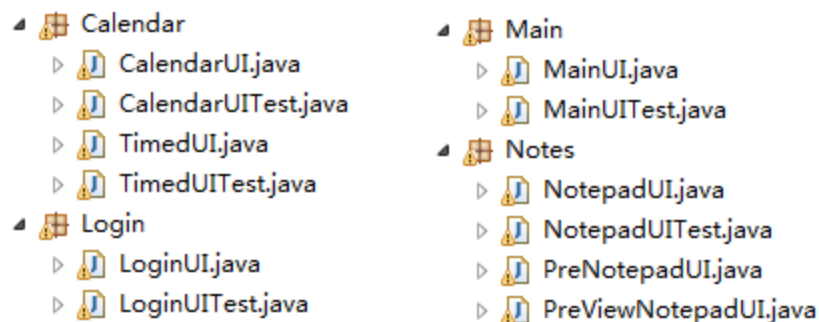
其中 **ReceivedEmailProcessor** 作为 **ReceivedPoP3Processor**, **ReceivedIMAPProcessor** 的共享文件, 也就是 **ReceivedEmailProcessor** 作为最基本的邮件解析类文件, 而 **ReceivedPoP3Processor** 作为 POP3 的邮件获取类文件, 而 **ReceivedIMAPProcessor** 作为 IMAP 的邮件获取类文件. 它们最后返回 **EmailMessage** 数组, 并传递给 **UI.Receives.PreView.PreReveivedUI** 类文件.

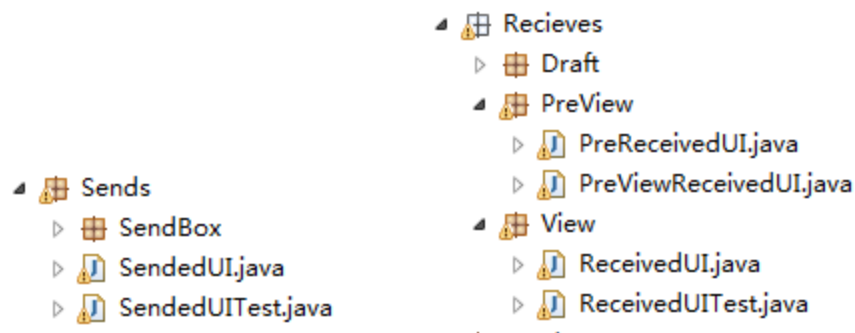
3 UI

该包主要实现邮件客户端的视图, 由下图所示:



下面是具体的包图:





下面将详细说该 UI 包的每个子包的具体实现功能：

1 Calendar 子包

Calendar 子包有两个类文件：CalendarUI 类实现的是主界面左下角的日历功能，为用户提供其进行记事本记录的日期；TimedUI 则为用户定时发邮件提供定时的时间。

2 Login 子包

Login 子包则提供了用户登录该邮件客户端的登录界面及登录功能的实现。具体由 LoginUI 类来提供实现。登录时，只是简单地检测用户邮件地址的合法性和用户邮件地址及其密码是否匹配，若匹配成功则转到该邮件客户端的主界面。

3 MainUI 子包

MainUI 子包中的 MainUI 类则是用户登录后的主界面，而 **MainUITest 类则是该邮件客户端的入口**。其中 MainUI 类则提供了收发邮件和记事本等的界面。

4 Notes 子包

Notes 子包实现了记事本的收发界面及其收发功能。其中 NotePadUI 类则是发记事本的界面实现，PreViewNotepadUI 类则是每个记事本预览界面中的一个记事本预览条目，而 PreNotepadUI 类则是记事本的预览界面（代码实现中，记事本预览界面只提供 5 条记事本预览条目）。

5 Sends 子包

Sends 子包提供了发邮件界面即发邮件功能的实现。发邮件界面提供了用户：选择添加小附件，删除添加的小附件，定时和正常发送的交互操作。

6 Receives 子包

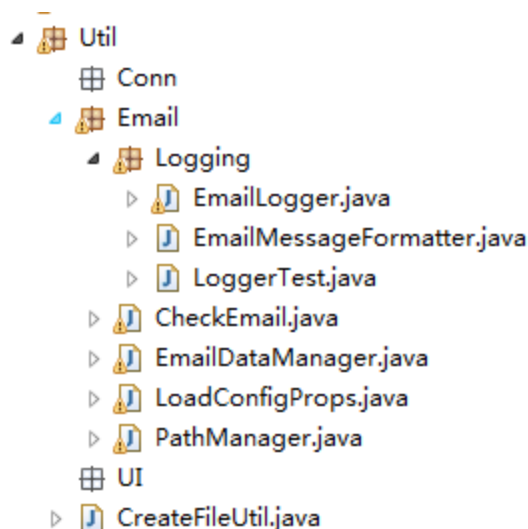
Receives 子包则类似 Notes 子包一样，提供了邮件预览和某个邮件查看的界面实现。其中邮件预览界面则由它的 PreView 子包提供实现，有个邮件查看的界面则由它的 View 子包提供实现。

类似地，每个邮件预览界面只提供 5 条简单的邮件预览条目，邮件预览条目则由 PreView 子包里面的 PreViewReceivedUI 类来实现，而邮件预览界面则由 PreView 子包里面的 PreReceivedUI 类来实现。

某个邮件查看的界面则类似于发邮件界面，由 ReceivedUI 类提供实现。其提供了邮件的附件(单个或者全部)下载功能，或者对此邮件进行回复，回复所有，转发等功能。

4 Util

该包由几个工具类文件组成：



其中 Conn 包和 UI 包为空(暂时没有实现)。

1 Logging 子包

这里实现了程序代码的日记记录，通过格式化的方式输出日记内容。日记内容包括了七个级别：Severe, Warning, Info, Config, Fine, Finer, Finest。

2 CheckEmail 类

提供了 EmailMessage 相关的操作，如对用户的登录认证，邮件地址的检测，发收邮件日期的格式化，邮件主题或内容编码解码等操作。

3 EmailDataManager 类

对登录的用户进行管理，并且为后续的收发邮件和记事本等操作给出用户的信息，以便重新连接邮件服务器和认证。

4 LoadConfigProps 类

对该邮件客户端的配置信息的管理。

5 PathManager 类

管理该邮件客户端的资源默认路径的生成, 获取的操作, 如日记文件的生成等.

6 CreateFileUtil 类

完成代码中需要创建文件夹或者文件(如临时文件)的操作.

6 尚未解决的问题

发邮件:

1 发送大附件或者多个附件问题

测试中, 发现上传大附件需要的时间较长, 而且发送的时候, 对方的邮件服务器会拒绝认证, 导致发送大附件或者多个附件时出错, 也会导致该程序的崩溃.

2 匿名发送问题

虽然代码里简单地实现了匿名发送邮件, 但是由于对方的邮件服务器, 如 QQ 邮箱服务器, 会嗅探出异常, 从而需要用户认证, 这样将导致匿名发送邮件失败, 即与正常发送邮件无差别.

收邮件:

1 POP3 获取邮件的效率很低

尤其在获取附件的信息的时候, 在代码里面, 每次只获取较少的邮件, 如 5 封, 然后渲染邮件

2 QQ 邮箱账户认证失败

QQ 用户登录后, 有时候在查看邮件的时候出错. 可能因为 QQ 邮件服务器对频繁操作的账户, 进行拒绝认证.

3 解析并渲染邮件内容的问题

由于只是简单地用 swing 里面的 JEditorPane 渲染邮件内容, 只能简单地渲染 html 和 plain 格式的邮件内容, 并不能解析和渲染其他格式的内容, 从而在查看邮件的时候, 并没有显示出邮件内容, 如:

```
content type: multipart/mixed;
    boundary="-----_NextPart_54977BE3_09C2E028_7D24CFAE"
content type: multipart/mixed;
    boundary="-----_Part_1424491_1623779250.1419496917507"
```

4 卡片布局中的卡片重影问题

在卡片设计中, 每个卡片都是一个 JPanel, JPanel 里面有一个 JScrollPane 组件, 但由于 JScrollPane 组件的大小默认为 JPanel 的大小, 导致在卡片切换时, JPanel 非 JScrollPane 部分不能切换, 从而导致了卡片切换时, 造成重影效果.

当将 JScrollPane 作为 JPanel 的唯一直接子组件时，但是难以控制 JScrollPane 的高宽，从而放弃这种做法。

7 后期工作

1 数据库的存储功能

从类包图里可以看到，里面有 DataBase 部分，由于个人开始该 project 的时间较晚，就来不及实现 DataBase 部分的代码。DataBase 部分代码所完成的功能有：

- a 实现邮件的存储
- b 实现记事本的存储
- c 实现联系人的存储

2 实现联系人的功能

- a 存储联系人，将联系人分为三类：常联系人，星标联系人，最近联系人
- b 基于联系人的邮件搜索，即给出与该联系人有关的邮件视图

3 附件的处理

- a 解决大附件的上传
- b 实现大附件的下载
- c 实现附件的转发

4 邮件内容的解析和渲染

- a 支持发邮件时的图片嵌进，html 格式的邮件内容发送
- b 在查看邮件内容时，尽可能的渲染邮件内容，也就是能支持多种格式的邮件内容还原，使得在该邮件客户端的邮件查看效果近似在浏览器查看邮件的效果

5 解决 POP3 的 parsing 效率

由于 POP3 在解析邮件的时候效率很低，尤其当邮件存在附件的时候，所以尽可能地提高其解析邮件的效果，降低邮件查看的延迟。

6 利用 IMAP 来解析邮件

IMAP 解析邮件并操作邮件，使得在该邮件客户端对邮件的操作，能同步到其邮件服务器上。