# JAVA实习作业一

## 计算机系111220086 欧北辰

2013年11月19日

**项目内容：**设计并实现一个简单的IFTTT窗口程序

**功能要求：**

* GUI窗口：设计良好的图形用户界面，界面中要求至少有菜单栏、输入区、文本域、主显区四个区域（其他图形元素可以自行按需添加）。
* 新建任务：一个任务包括一个THIS和一个THAT。用户可以通过图形化交互界面，根据可选的THIS和THAT来新建任务。
* 修改任务：用户可以修改已有的任务。
* 查看、删除任务：用户可以查看所有已有的任务，也可以选中并删除已有的任务。
* 任务控制：用户可以选择开始、暂停和结束当前的任务。
* 任务运行信息：在主界面显示当前正在运行的任务，并实时的输出当前任务的运行信息。
* 可选的THIS和THAT事件：可供用户选择的THIS事件必须包括定时和收到Gmail邮件，That事件必须包括发布新浪微博和发送Gmail邮件。
* 必须完成以下任务：
  + 在某个指定时间点，给某个指定邮箱发送一封邮件
  + 若某个指定邮箱收到邮件，则给另一个指定邮箱发送一封邮件
  + 在某个指定时间点，使用指定的新浪ID发布一条给定内容的新浪微博
  + 若某个指定邮箱收到邮件，使用指定的新浪ID发布一条给定内容的新浪微博

**项目要求：**

* 程序源代码中，对于主要的类、变量和方法要求有注释；
* 文档条理清晰，表述明确，主要用于清晰地说明作业的思想和实现方法；
* 提交文档、所有程序文件；（程序须可运行）；
* 界面尽量美观整洁
* 作业必须个人单独完成，若参考他人程序，请注明参考来源和参考内容，否则视为抄袭；

**项目实现：**

实验环境：

Microsoft Windows 7 SP1 X64

NetBeans IDE 7.2.1

JAVA SDK 1.6.0\_33

Weibo4j-oauth2.0-beta2.1.1.

Java Mail 1.5.1

Apache-commons-email 1.3.2

Apache-commons-net 3.3

dom4j-2.0.0

为什么选择NetBeans IDE：一是因为用惯了，之前用它开发过好几个GUI小程序，自动管理代码的功能很强大。GUI设计也比较直观。（并不是偷懒，Eclipse、MyEclipse、INTELLJIDEA装上插件也能直观地设计GUi。只是NetBeans更符合我的习惯，当然也有节约时间的想法在内。）

项目结构：

Project/src 项目源码

Project/src/oubeichen 我的代码包

Project/src/weibo4j Weibo SDK的包

Project/src/About.png “关于”窗口的背景图

Project/src/\*.properties Weibo SDK的默认配置文件，我没有修改移动防止出错

Project/lib 项目导入的库

Project/defaultmail.properties 发送邮件使用的默认邮箱的配置，使用程序前必须配置

Project/weiboauth.properties 微博授权的配置，可自动配置

Project/tasks.xml 程序中读取/保存功能使用的tasks.xml

Project/其他文件 IDE自动产生的配置、字节码、jar包等文件，与源代码无太多关联

Oubeichen包里的文件：

AboutDialog.java(extends jDialog) “关于”窗口，没有多少代码

DateChooser.java(extends jPanel) 网上借用的，给TextField添加弹出一个日历来选择日期功能的代码，如下图：



Getpage.java(普通class) 以前在网上借用并加以个人修改，来源已不可知。用于获取目标url页面的源代码。

MainFrame.java(extends jFrame) 主窗口，包含任务线程(RunningTask extends Task)的实现。

NewTaskDialog(extends jDialog) 新建/修改任务通用的对话框

Task.java(extends Thread) 存储任务信息（但不能运行）;

UpdateStatus.java(普通class) weibo SDK examples中的一个类，直接加以修改使用。

为什么要扩展Thread而不是实现Runnable：因为即便实现了Runnable，也需要通过Thread(<Runnable class>)来实现，这样只能直接管理Thread。而要获取Thread里面的信息数据是比较麻烦的，这样管理任务没法知道想要的是哪一个Thread。而对ArrayList采用指定index的方法不仅代码繁琐，而且还不安全 （多个ArrayList如果没有完全同步操作。则会导致这几个ArrayList中对应同一个任务的index不一样，最后完全崩溃）。个人认为，只有确认在内存中指向的位置一致，或者确认UID（每个Task有唯一的UID）一致，才能保证操作的安全。而根据此项目的当前需求，线程之间并不需要共享太多资源，所以直接扩展Thread的缺陷并没有产生问题。

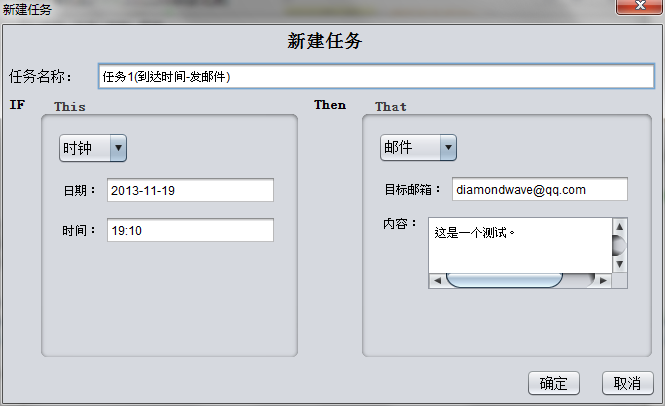
为什么要定义类中类：恕我技术拙略，当一个类需要回调调用这个类的类中的函数（如一个jDialog被jFrame产生，某时刻jDialog在运行一个函数，中途需要回调jFrame的某函数，来刷新jFrame中间控件的属性）时。除非将原来的类中间的函数设置成静态public函数以外，就只能弄类中类了。定义类中类可以方便地让里面的类调用外部类的函数。

处理逻辑：

* 新建任务/修改任务：

建立一个对话框来设置任务参数（根据根据THIS和THAT的选择显示不同控件。以及验证输入等细节就不多说了）。确认时将任务信息存储在NewTaskDialog中的一个Task类里。确认后在MainFrame里取得NewTaskDialog中那个Task类，加入任务列表Tasks(ArrayList)中。

修改任务和新建任务区别不大，只是建立窗口时，先在MainFrame里传入对应Task，在NewTaskDialog构造函数中自动将各控件内容填好，再显示出来而已。



* 查看、删除任务：

用一个ArrayList Tasks来存储建立的全部Task。然后用一个jList同步刷新显示Tasks的名称。选中一个任务时，修改一个TextArea的值来显示详细。删除时只需删除Tasks中的任务，刷新jList即可。



* 任务控制：用户可以选择开始、暂停和结束当前的任务。

用一个ArrayList RunningTasks来存储建立的RunningTask。

选择一个Task开始时，新建一个RunningTask，加入到RunningTasks中。Task实现了clonnable，所以此操作是深度复制。RunningTask的值一旦设定就不会随着原来的Task的值的更改而更改。这是为了保持正在运行的任务的正确性，不会因为原任务修改而导致错误。加入之后将此RunningTask执行start();

选择一个RunningTask暂停、恢复时，就将其执行suspend()或者resume()。此方法被认为不安全，但这里因为数据不会被修改，所以不用担心。如果要完全安全，只能重写整个Thread。

选择一个RunningTask停止时，就将其执行stop()，然后从RunningTasks移除。

* 任务运行信息：在主界面显示当前正在运行的任务，并实时的输出当前任务的运行信息。

同样用jList来同步刷新显示RunningTasks，选择jList中项目刷新两个TextArea。



* 可选的THIS和THAT事件：

处理细节以及测试已经耗费太多时间，就没有实现额外的事件，请见谅。

THIS均为While循环，直到满足条件才结束循环。

定时：每次判断当前时间与任务设定的关系（未到时间，到达时间，超过时间），未到时间则继续循环，到达时间则退出循环，超过时间则结束任务。

邮件：使用POP3获得邮箱信息，同上一次收取时信息进行比较。如果没有变化，则继续循环。如果有新增邮件，则退出循环。(我用了简单的apache-commons-net。)

THAT均为一次性执行。

发邮件：使用SMTP发送邮件。(我用了基于java mail的简单的apache-commons-email。)

Weibo：使用Weibo API。因为我们只是测试，没法通过新浪审核，所以只能使用测试账号。

关于Weibo授权：Weibo授权需要用户登陆一个网页，获取code返回给Weibo API后换成access\_token才能发送微博。Code手动复制当然很没必要。有人可能使用了htmlparser或者httpclient来模拟用户登录来获取code。当然我觉得这样有一定欺骗性，并没有用。

所以我是让Weibo API登陆后自动跳转到我自己域名中的一个页面http://oubeichen.com/oauth.php?code=XXXXXXXX，而这个页面会自动保存传入的code到一个txt。然后用Getpage.java里的函数读取oauth.php写好的文件来获取code。

实时显示任务信息只需要回调MainFrame里的更新函数即可。

当任务自动结束时，有些人放在原来的地方不管。而我认为这样不仅没有回收Thread的资源，而且有可能会产生两个相同UID的任务同时运行的问题。所以我在线程即将结束时再建立了一个线程，这个线程延迟一定时间后，删除原来那个RunningTask线程(RunningTask线程没法删除自己)。达到自动删除已经停止的任务的功能。

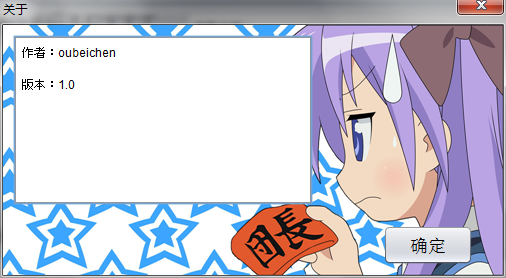
* 读取/保存：

使用dom4j的XML处理API来Tasks里所有Task。保存到/读取自tasks.xml.

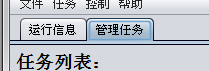


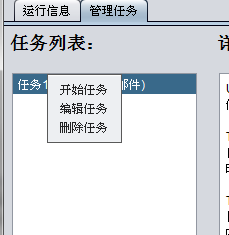
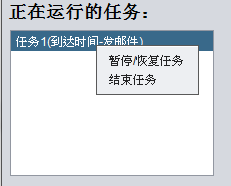
* 关于：

啥也没有，只是凑个数



* 其他细节：

jTabbedPanel 

右键菜单 

双击任务列表中的任务：编辑任务

双击正在运行任务列表中的任务：暂停、恢复任务

**参考：**

使用java mail api 和 apache mail api 发送email 代码示例

<http://houfeng0923.iteye.com/blog/1014475>

新浪微博api：

<http://open.weibo.com/wiki/%E5%BE%AE%E5%8D%9AAPI>

JAVA对象拷贝：

<http://www.blogjava.net/JAVA-HE/archive/2008/04/28/196727.html>

Apache commons：

<http://commons.apache.org/>

JAVA格式化输出XML：

<http://blog.csdn.net/welcome000yy/article/details/7760767>

JAVA Swing 日期选择控件(有大量参考代码，非核心功能，应该没关系吧)：

<http://zgdeng.iteye.com/blog/1405650>

Apache commons email samples

<http://commons.apache.org/proper/commons-net/examples/mail/>

以及许许多多不知名的，打开看了那么一下就关闭了的页面。

以及百度知道、知乎、天涯问答、Stackoverflow 等国内外问答网站、CSDN等论坛。

**已知不足：**

1.一个稍具规模的程序自然是各种容易发现的和不容易发现的BUG，不然大型软件也不会那么多补丁。

2.THIS、THAT事件过少。

3.重要信息如邮箱密码、Oauth数据、Weibo API配置均以明文存储。（我数学不好想不到一个较安全的并且可逆的加密算法，弱智算法当然是给程序的后续扩展自添麻烦）。

4.判断是否收到邮件不够严谨（因为POP3和IMAP的限制，我也没有好的办法），如果同时有其他邮箱提供商或者客户端使用这些协议访问该邮箱，可能出现无法得知收到了新邮件。

5.关于定时，有人说可用Timer（swing和util中各一个），但我为了简化代码，还是用的简单的while(不符合条件){sleep()}。

6.前面已经提到了暂停/恢复任务使用的Thread.suspend()和Thread.resume()并不安全，Task直接extends Thread也有缺陷，这个需要修改许多地方，所以暂时没能解决。

7.没有用到数据库而使用的XML，这样扩展到多用户Web界面不方便。

8.邮箱登陆我是通过用(pop.<domain>/smtp.<domain>)来当做pop3/smtp域名的，而且使用了SSL(支援Gmail)。所以不是这种规则的pop3/smtp服务器（如admin@oubeichen.com 的pop3服务器域名为pop.qq.com），以及不支持SSL的邮箱提供商都会有问题。

9.You tell me.

**感想：**

JAVA不愧是时常占据编程语言排行榜榜首的大热门语言。扩展性很高，IDE的智能化也到了一定境界。编写程序入门门槛低，实现功能也简单。

说来惭愧。对于GUI设计，我如果不是捣鼓Visual Studio来弄一个明明很简单但非得装.NET framework 4.5才能正常运行的程序，就只会用JAVA了。

想想自己弄的创新项目是JAVA，自己没事捣鼓的安卓源码也是JAVA、JNI混合，以前给字幕组设计的一个简单时间轴查错软件也是（<http://github.com/oubeichen/TSGSUB>），再之前的数据结构给模拟pagerank软件写的更简单的GUI也是JAVA。

但JAVA也不是没缺点，如果在虚拟机中运行，效率很低，而且再怎么样也无法使用多核（也就是说JAVA的多线程其实是时间片轮之类的算法，而不是真正同时运行）。当然我们这些普通用户使用的JVM也没法JIT。还有32位JVM也有这里那里的限制。怪不得安卓经常被人诟病，最近4.4开始狠下心来整改，使用新的ART。

所以理论上又要追求效率又追求可移植性应该首选C和C++，无奈离开了Visual Studio的GUI编程看得人头疼，而且有同学（NOIp提高组一等奖水平）高中花了一个暑假才写了一个五子棋的先例，所以一般都是望而却步。

事实上长期使用软件，收到脚本语言、批处理、Linux终端的影响又觉得GUI不算什么，命令行程序才是王道，那些民用需要高效率的软件基本上都可以用命令行解决吧。（平时喜欢玩的视频转码相关的软件基本上都可以命令行，如x264，mkvmerge，gpac，neroaac，又比如说服务器搭建，apache，mysql都可以命令行解决，所以可以写自动脚本节省了大量时间精力，貌似IIS不行）

不过这一切的一切对于精通多门编程语言的学神和学霸们来说都不是问题。现在也是，当我们最后一个晚上才拼死拼活弄出一个Demo来，还差点因为误操作导致代码被删除（伟大的数据恢复工具！）的时候，他们不是在完善更多的功能、为实验三做准备，就是投身于设计算法的精妙数学事业中。正所谓：

（此处略去打油诗一首）