**西安电子科技大学**

**Java程序设计 课程实验报告**

**实验名称 命令行个人信息管理程序（PIM GUI）**

计算机科学与技术 学院 2103051 班

成 绩

姓名 张平 学号 21030540006

姓名 张宏喆 学号 21030540005

实验日期 2022 年 05 月 30 日

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验内容  三、实验过程  四、实验结果分析  五、实验小结（实验过程感受和建议） |

# 实验目的

1. 了解Java GUI编程技术，使用Java面向对象设计思想进行设计。
2. 了解AWT类层次结构及基本组件。
3. 熟悉容器和布局管理器的基本概念。

4. 掌握使用JFrame、JPanel、JScrollPane、JButton等组件设计GUI的方法。

5. 掌握FlowLayout、BorderLayout、GridLayout布局管理器的使用方法。

6. 熟悉Java的事件处理模型。

7. 了解Swing常用组件的特点和使用方法。

# 实验内容

下面列出之前的实验要求。最开始的要求如下：

# **This assignment involves the creation of simple Personal Information Management system that can deal with 4 kinds of items: todo items, notes, appointments and contacts. Each of these kinds of items is described in more detail below. The assignment requires that you create a class for each item type, and that each class extends an abstract base class provided for you. In addition to creating the four classes, you need to create a manager class that supports some simple text-based commands for creating and managing items.**

注释：PIM可以处理4种类别事项：待办事项，备忘，约会和联系人，PIMEntity是公共抽象父类，创建PIMManager进行测试，(有给定名称的要按给定的名称)。

上次实验（PIM with I/O）在前面实验的基础上，改写基于命令行形式的个人信息管理（PIMCmd）程序，要求如下：

**The PIM with I/O assignment is a Command Oriented Personal Information Manager (第5章实验内容2) with same general idea (todos, appointments, etc.) stored in one(or more) file(s) accessed by I/O.**

**Java I/O: you need to be able to save a list of items to a file (and to load - read from a file). You can use a simple text file and the PIMEntity methods toString() and fromString() to generate/parse strings, or you can get fancy and create Serializable objects and use Object streams.**

本次实验则在前面两次实验的基础上，要求实现个人信息管理器（GUI版本）。在支持所有四种PIMEntity类型的基础上，通过图形用户界面提供（至少）以下内容：

1. 一次显示一个月的日历，允许用户更改月份（后退和前进按钮都可以）。日历中必须显示所有具有相关日期的条目的文本，即显示从本地文件检索到的、与日期相关的PIMEntity（待办事项和约会）。日历需要满足预览作业的要求，但不需要确保每个条目（待办事项、约会）在日历中都是完全可见的，只有部分文本出现在正确的日期就可以了。
2. 用户必须能够通过GUI创建新的PIM条目。
3. 用户必须能够通过GUI编辑现有的PIM条目。
4. 用户必须能够按条目类型查看所有PIM条目，因此应该有某种方式来查看所有待办事项或约会等。各种条目的显示没有必要的格式。
5. 用户必须有某种方式来保存/加载条目，可以是本地文件（文本文件或持久化对象存储）。
6. 必须有一个菜单栏，允许用户从支持的功能（查看日历、查看待办事项、查看联系人等）中进行选择。菜单栏还应包括用于创建新PIM条目的菜单项（“新TODO”、“新约会”等）。
7. 用户应该能够指定一次帐户名（用作PIMTenties的所有者）。每次程序启动时都会提示用户名，还可以将其保存在某种配置文件中（或在命令行上指定）。重要的问题是，用户不需要在每次创建新Entity时指定所有者。
8. 应使用泛型。如果从本地文件获取实体，则可以只使用IOException（抛出异常时）。RemotePIMCollection只是一个接口，用于从存储中获取PIMCollection。这并不意味着要使用特定的技术。
9. 面向命令的PIM相关类定义将用于GUI项目。以下内容应更改：

应更改PIMEntity，使其包含对所有者（字符串）的支持和指示项目是私有还是公共的布尔值。一般来说，使用的简单方案是，如果知道所有者名称，就可以访问私有项（这是由PIMCollection中的方法强制执行的，而不是单个PIMEntity派生类）。应更改4种类型的Pimentity中每种类型的构造函数，以支持owner和shared标志。fromString方法不再是接口PIMEntity的一部分。PIMCollection类支持更多的方法，特别是有一些方法可以获取公共项（当没有指定所有者时），以及获取与特定所有者对应的公共和私有项。

# 实验过程

1. **实验环境**

操作系统：Windows 11

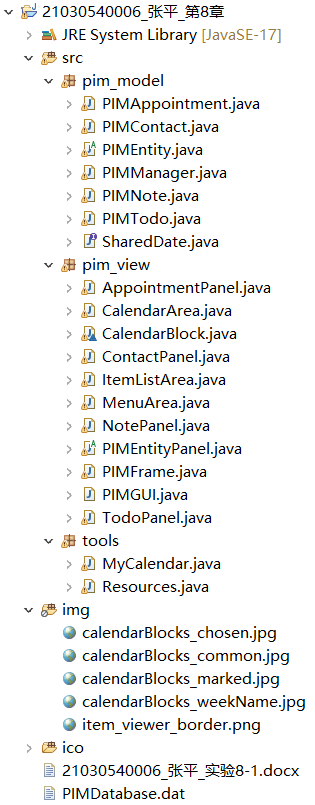
集成开发环境：Eclipse IDE for Enterprise Java and Web Developers (includes Incubating components) 2022-03 (4.23.0)

1. **题目分析**

在上次实验的基础上（创建可序列化的对象、并使用对象输入输出流来实现对象的持久化存储），我们这次来实现GUI版本的个人信息管理程序。

限于只有两个人，且个人能力有限，加上面临期末复习压力，我们无法完全实现所有的需求，具体来说7)、8)、9)这三项需求难以完成。特别地，对所有者Owner这一项需求我们几人存在理解上的分歧，不太清楚如何随着账户名的变化来使用对应的公共和私有项。

经过讨论，我们决定使用MVC模型-视图-控制器设计模式。模型（model）对应PIMEntity类（PIMTodo、PIMContact、PIMAppointment、PIMNote）和PIMManager、以及GUI组件中组件的状态、文本域中的文本等，视图（view）对应GUI界面（PIMGUI、PIMFrame、MenuFrame、CalendarArea类等），控制器对应事件监听器，用来处理用户输入事件、实现前后端交互。最后，完成的项目结构如下所示：



各个类之间的关系比较简单，抽象类PIMEntity是PIMTodo、PIMContact、PIMAppointment、PIMNote的父类，它实现Serializable接口，这样其子类也可以被序列化；PIMManager和PIMEntity是组合关系，它管理一组PIMEntity对象；PIMGUI创建主窗口（PIMFrame）的实例，PIMFrame持有一个PIMManager对象，同时与菜单窗格MenuFrame、日历区域类CalendarArea是组合关系。

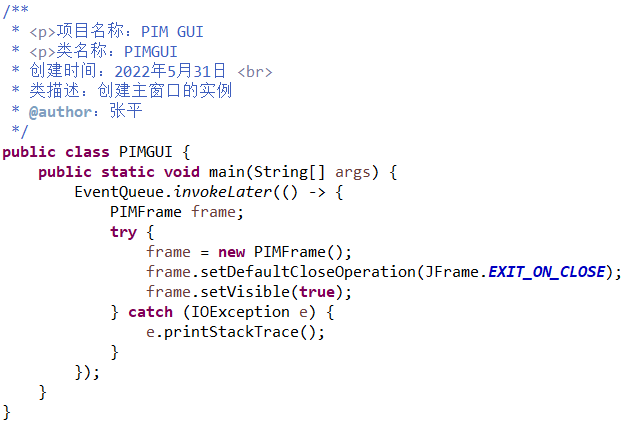
在程序代码中，我们首先进入PIMGUI的main方法，初始化PIMFrame用户主界面。PIMFrame会实例化一个PIMManager对象用来管理PIMEntity列表，同时用MenuFrame生成菜单栏（包括File、Edit、Navigate、Help四个菜单）。

File菜单包含Save、Save As、Load、Load As和Exit菜单项，Edit菜单包括四类PIMEntity对象的创建命令，Navigate用来实现日历的跳转和各类事项的展示，包括Jump To Date、Jump To Today、Show All等多个菜单项，Help菜单的About菜单项说明程序功能。

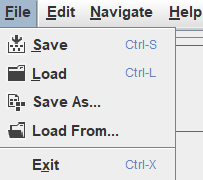
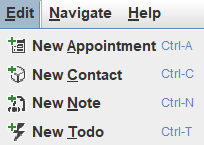
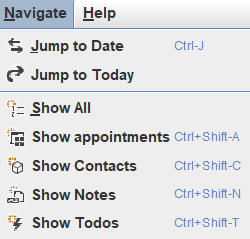
Edit菜单中没有删除和修改菜单项，是因为PIMFrame还会在界面左侧创建一个条目列表，简要显示PIMEntity信息，点击相应事件会弹出窗口、可以编辑信息。PIMFrame类还在界面右侧实例化一个CalendarArea，动态绘制某年某月的日历，我们可以在Navigate中调整对应年月。点击日历中的某天（有背景颜色即代表有事项），就会在左侧显示对应日期的PIMEntity事件信息（PIMAppointment和PIMTodo）；如果没有事项，就会弹出一个窗口，提示是否创建与该日期有关的事项。

1. **代码实现**

由于代码较长，这里只展示关键部分。PIMGUI类很简单，在其main方法（整个GUI程序的起点）中用事件分派线程创建PIMFrame类对象，初始化并显示用户界面：

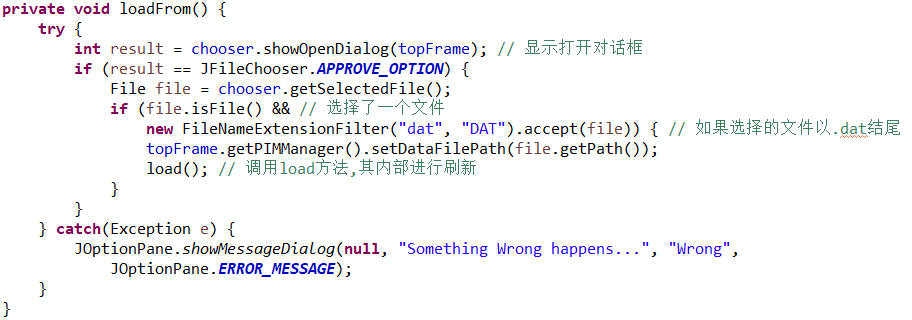


菜单栏是本程序的关键部分，它实现了程序中的大部分功能调用。菜单栏实现后的效果如下。为了美观，我们添加了图标；为了使用方便，我们还设置了助记符和相应的快捷键：

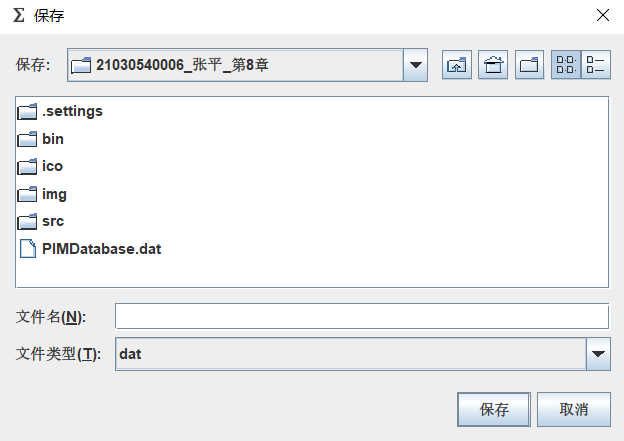
下面简要介绍菜单栏的实现代码。菜单栏中File菜单的关键功能是Save As和Load From。为了实现这两种功能，我们使用了Swing提供的JFileChooser类，用来创建文件保存/打开对话框。对于另存为功能，我们可以选择一个目录来自动新建一个数据文件，或者自定义数据文件名；对于加载功能，我们必须选择一个扩展名为.dat的文件。



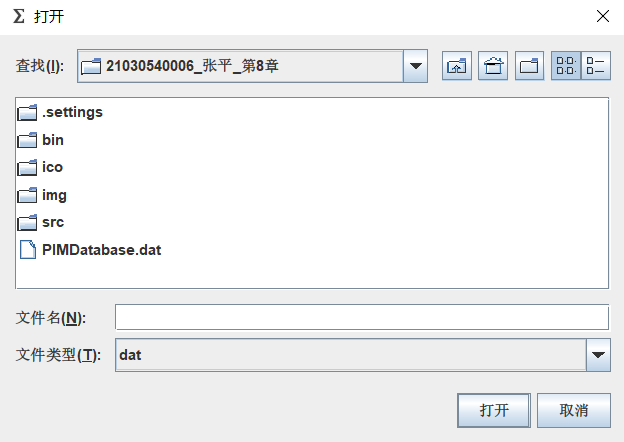


这两种功能实现后的窗口如下所示。点击Save As，首先弹出提示框，点击确认后会弹出文件保存对话框，保存之后还会报告结果：





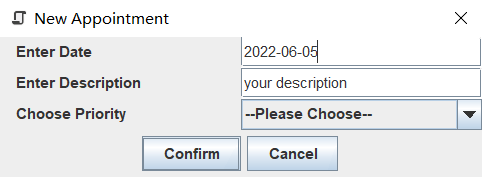
点击Load From则会直接弹出文件打开对话框，选择文件、完成数据加载后同样会报告结果：

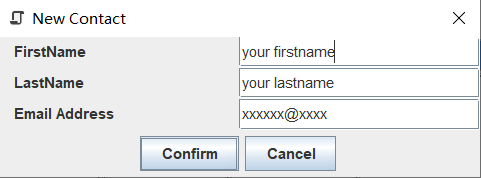


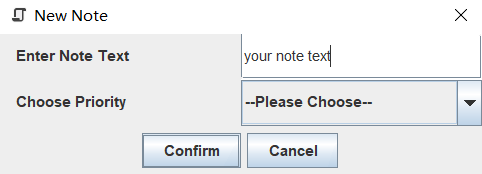
菜单栏中Edit菜单用来创建事项，四个菜单项对应的四个界面类都继承自抽象类PIMEntityPanel，它的showDialog方法可用来创建对话框。只要继承该抽象类，并在子类的构造函数中自定义要显示的Panel，就可以使用showDialog显示不同的对话框。我们的这一实现，一定程度上参考了JDialog类提供多个预定义对话框的方法。

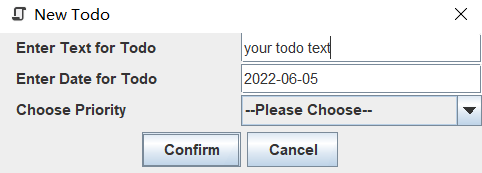


点击Edit菜单中的几个菜单项，结果如下所示。显示的几个对话框中设置了默认的文本：

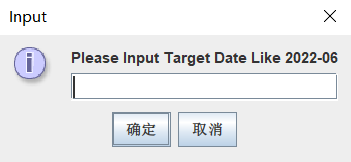






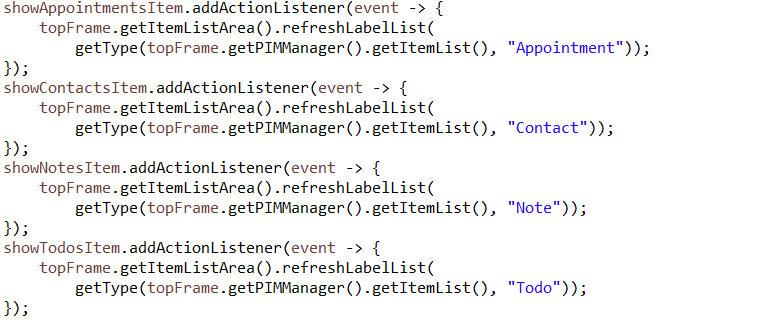


Navigate菜单中，只有Jump To Date有对话框，要求用户输入要跳转的年月。如果输出日期，它会弹出报错窗口：





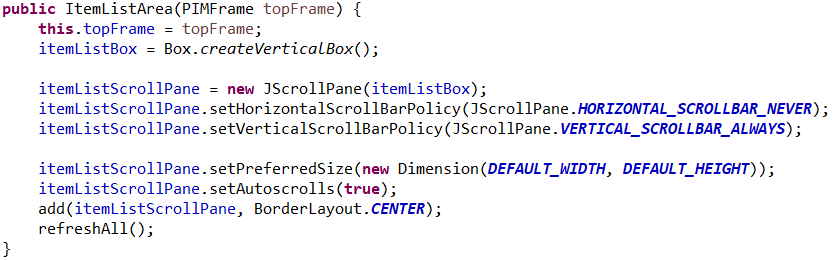
其他几个Show菜单项，对应的监听器代码只是简单调用了ItemListArea提供的、用来刷新左侧事件列表区域的方法：



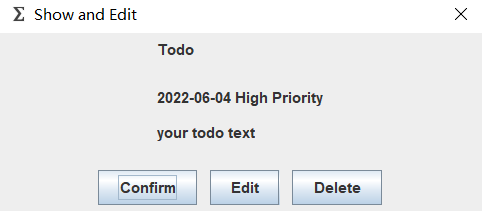
最后是Help菜单，它的About PIM GUI菜单项界面如下所示：

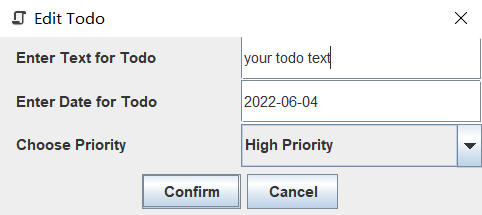


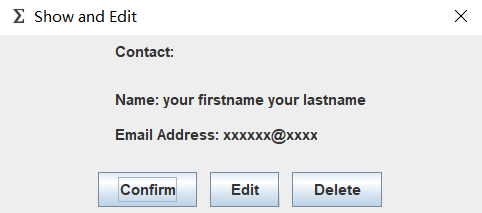
菜单栏下，左侧事件列表的实现很简单，只是一个JScrollPane内嵌一个JLabel列表。

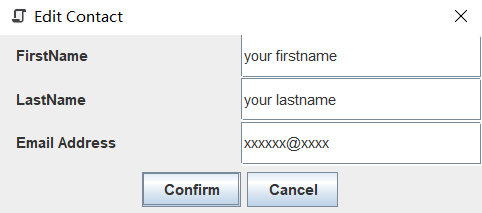


鼠标点击事件列表中的各个事项，会弹出用来展示事件信息的窗口，部分窗口如下所示。点击Comfirm按钮，可退出弹窗；点击Edit按钮，则会弹出事项编辑界面；点击Delete按钮，则会删除事项。





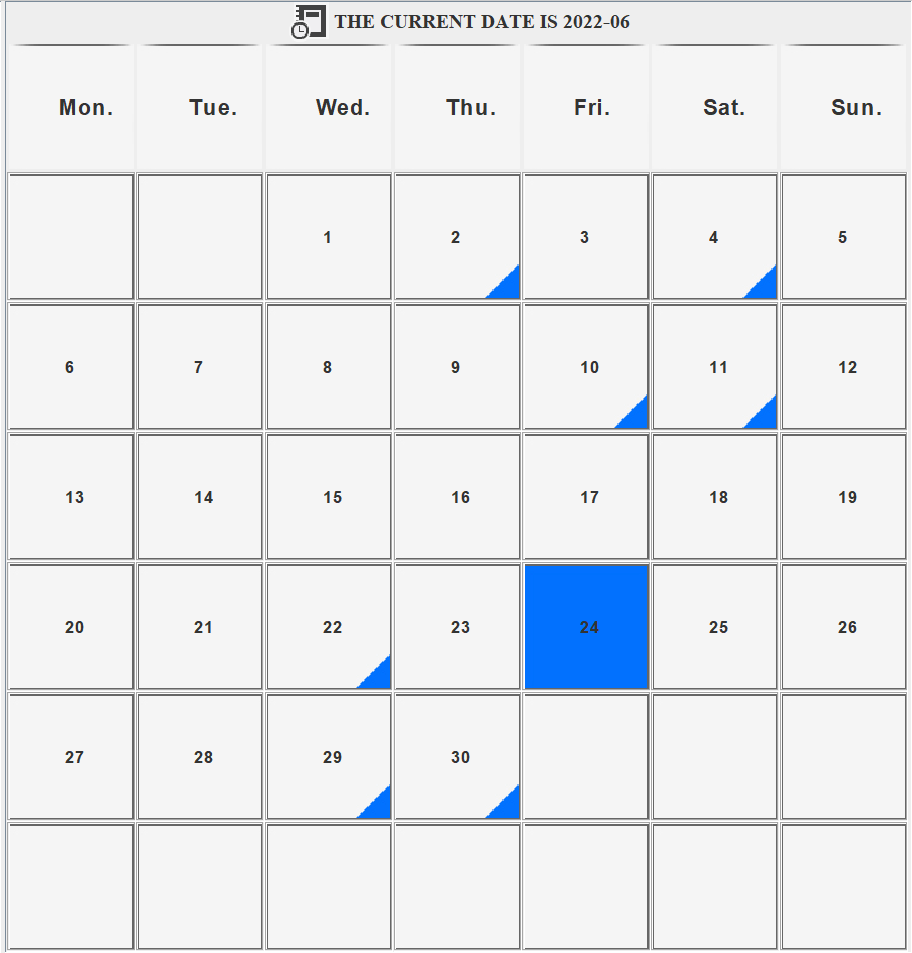




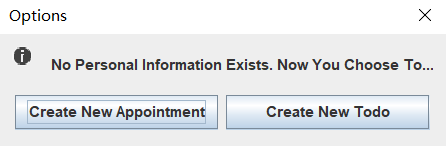
在CalendarArea类中，实现右侧日历绘制的代码如下，它创建了一个CalendarBlock对象数组，用来显示每天的日期方格。CalendarBlock类中还定义了鼠标事件的处理方法：



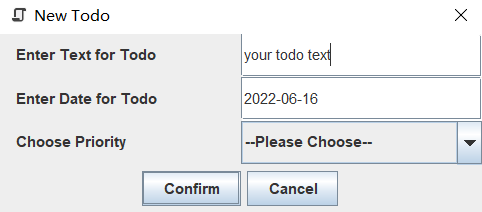
最后绘制出的2022年6月的日历如下所示，蓝色右下角的方格代表该天存在事项，全蓝色方格代表鼠标光标移到此处。通过Navigate菜单的Jump To Date菜单项，我们可以跳到任意年月的日历：



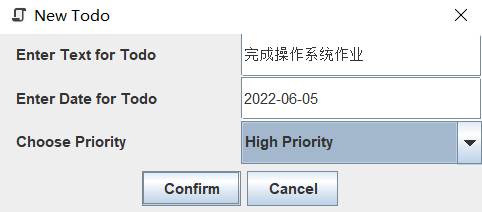
点击已经存在事项的方格，会在左侧事项列表中显示所有具有相关日期的事件，此时可以通过Navigate菜单中的Show All菜单项，重新显示所有事件。点击不存在事项的白色方格，则会弹出如下窗口，要求创建与该日期相关的事项：

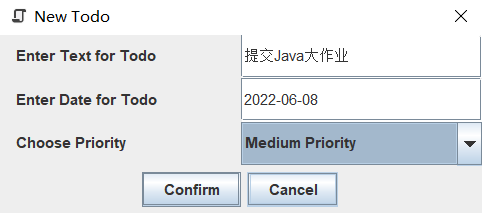


点击按钮后，会弹出相应的事件创建窗口，比如点击16号的方格、选择创建TODO后弹出的事件创建窗口如下，对应的日期已经**自动填入**：



# 实验结果分析

完成程序编写后，在Eclipse中点击Run。在GUI界面中使用Edit菜单项，依次输入如下信息： 





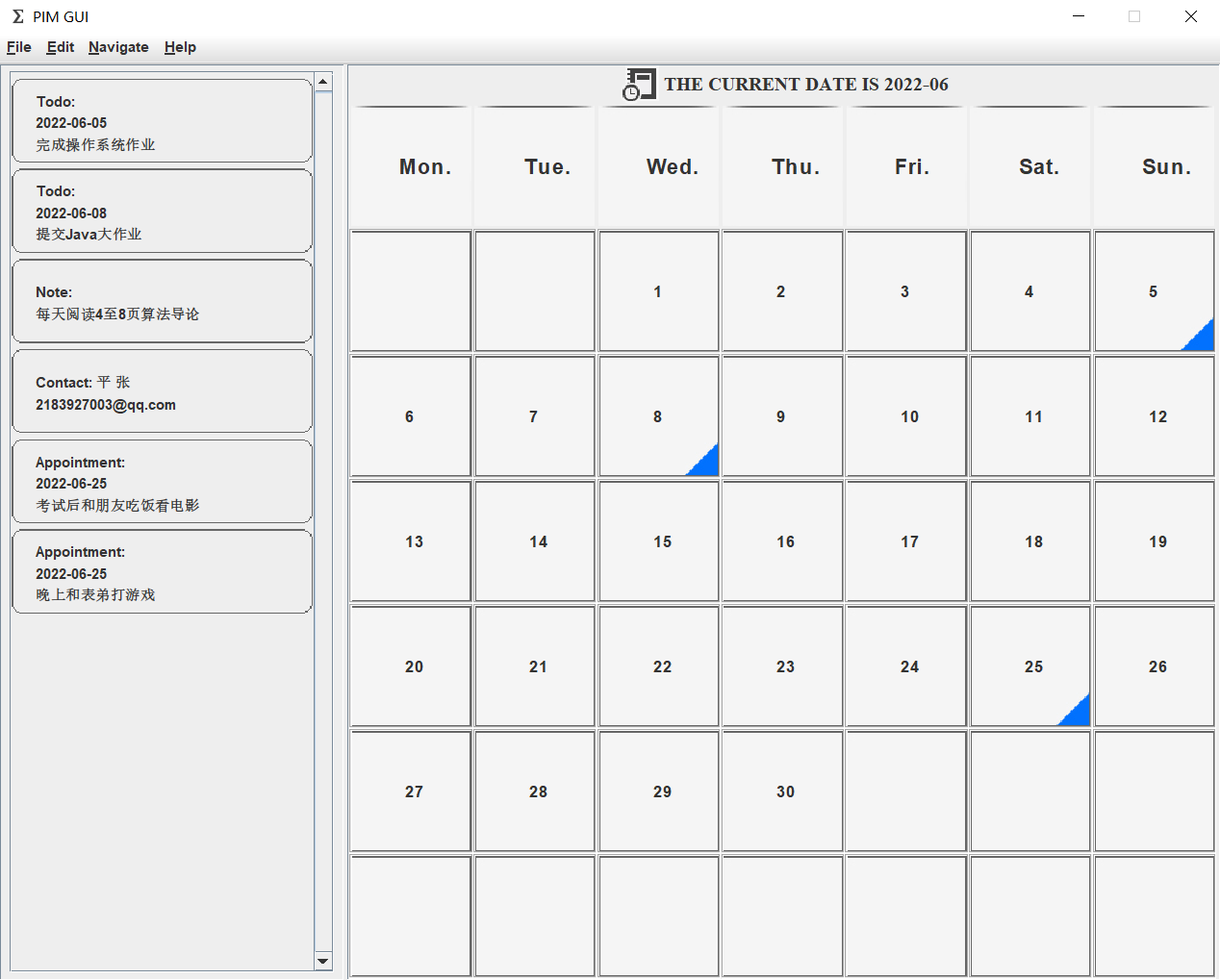




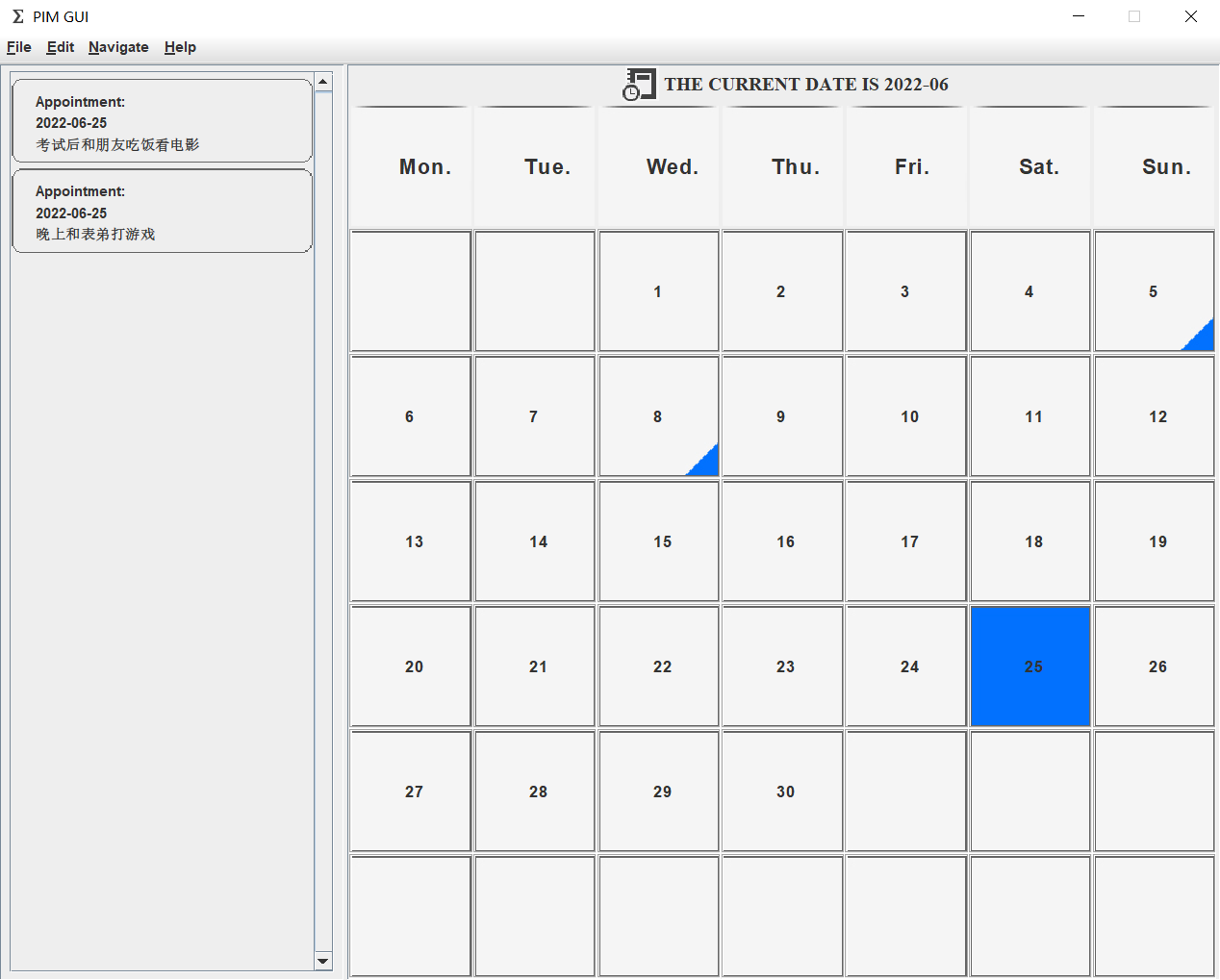


接着用快捷键CTRL+S保存数据到默认文件：

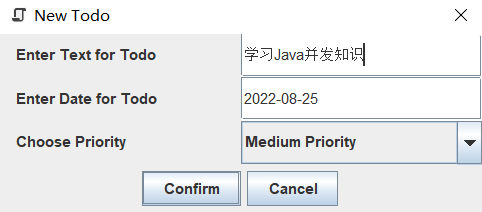


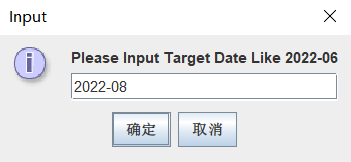
现在的界面如下所示：  
 

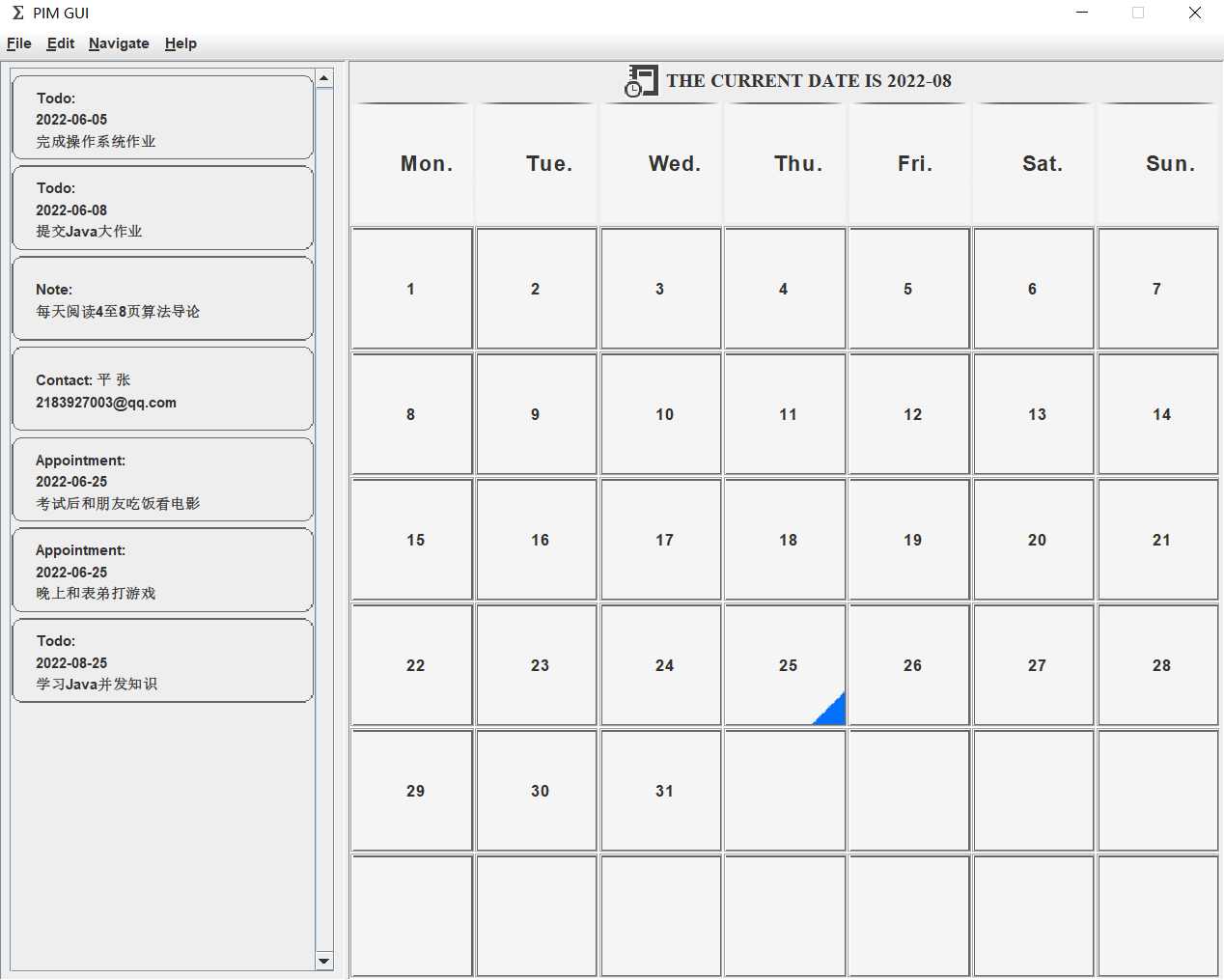
点击25号方格，左侧事项列表显示当天的事件：



如果用Jump To Date跳到其他月份，左边会重新显示所有事件，右边也会重新绘制。比如，我们先创建一个2022年8月25日的TODO，再跳到该月：







可见，实验结果是正确的。

# 实验小结

为了完成本次实验的Java GUI，我特意把《Java核心技术卷I》第10章图形用户界面程序设计、第11章Swing用户界面组件和《Java核心技术卷II》第2章输入与输出仔细看了一遍，虽然有所收获，但还是感觉细节太多、难以掌握。

因为我们第一次（合作）实现这种GUI+后端处理的项目，整个项目代码较长，相对比较混乱。这也让我们认识到编写代码之前画好类图、做好UI设计等的重要性。如果不画好类图，各个类之间的关系就一团乱麻、难以掌握。如果没有简单的UI界面示例，实现起界面来就难以确定各个组件的位置。协同工作也是一个问题，我们两人的代码风格有许多不一致的地方，因此代码看起来比较凌乱。

当然，为了完成本次的实验，我们还把之前写的PIMManager类、PIMEntity类（及其子类）做了一些修改，之前的实现还是比较合适的。不过，本次的代码感觉还有很大的改善空间，个人认为有些重复代码需要优化，只是狗咬刺猬——不知从何处下手，以后还需学习设计模式思想，多向优秀开源项目学习。

此外，由于只是初步学习了Java AWT和Swing的基础知识，不会对样式进行美化，也没有时间挑选好看的图标，整个用户界面相对来说比较原始和简陋。而且对GUI的调试有些繁琐，可能有些隐藏的Bug没有找出。