



23 年秋 JAVA 语言第二次大作业

实验名称：飞马星球飞马银行微系统设计与开发

姓名： 陈祥蔚

学院： 数学与统计学院

专业： 统计学

学号： 320190930061

2023 年 6 月 30 日

摘要

本次作业中练习应用了本学期中学习的各种有关如何用 java 语言进行面向对象编程技术，包括 GUI、数据库系统等。搭建了一个可以进行开户销户等操作的微银行系统。

关键词：面向对象编程 java 网络编程技术数据库系统 GUI

目录

1	对于源代码文件的说明	1
2	对于源代码中各个类的说明	2
2.1	Client 文件夹中的类	2
2.1.1	Main_client 类	2
2.1.2	Client 类	2
2.1.3	UI 类	2
2.1.4	Worker 类	3
2.2	Server 文件夹中的类	5
2.2.1	Main_server 类	5
2.2.2	Server 类	5
2.2.3	DB 类	5
2.2.4	Operator 类	6
3	编译及运行	7
4	结果展示	8
A	客户端 Client 部分代码	13
B	服务器端 Server 部分的代码	41

1 对于源代码文件的说明

本项目包含三个文件夹，分别是名为“lib”、“bin”和“src”的文件夹。

“lib”文件夹用于存放项目所需的第三方 jar 包，“bin”文件夹用于存放编译后的源代码字节码文件，“src”文件夹用于存放源代码。

在“src”文件夹中，分为客户端代码和服务端代码两个部分，即“Client”和“Server”文件夹。每个文件夹中包含四个 Java 文件，同时客户端和服务端各有一个主类用于运行，即“Main_client”和“Main_server”。

在“lib”文件夹中，包含两个客户端所依赖的第三方包，分别是“jxl-2.6.12.jar”和“itextpdf-5.5.8.jar”。

2 对于源代码中各个类的说明

2.1 Client 文件夹中的类

2.1.1 Main_client 类

这个类中使用了 Java 的图形用户界面 (GUI) 技术来创建一个客户端的主程序。具体来说, 它使用了 Java 的 AWT (Abstract Window Toolkit) 和 Swing 库来实现用户界面的创建和显示。

代码中的 Main_client 类的 main 方法是程序的入口点。在该方法中, 通过创建一个 UI 对象, 并传递参数“欢迎: ”、300 和 200, 来初始化一个用户界面。

UI 类的构造函数接受标题、宽度和高度作为参数, 并使用这些参数创建一个窗口。该窗口可能包含各种组件, 如标签、按钮、文本框等, 用于与用户进行交互和展示信息。

2.1.2 Client 类

Client 类只负责与服务端建立连接, 之后将收到的数据发送到服务端。它向 Worker 类提供且只提供发送数据的功能。这个类运用了 Java 中的网络编程技术, 使用了 Socket 类和相关的输入输出流类来实现客户端与服务器之间的通信。

代码中的 Client 类具有一个构造函数, 用于初始化客户端的名称和端口号。

Client 类还定义了一个 work 方法, 用于执行客户端的工作。该方法接收一个数据参数, 将数据发送给服务器并接收服务器的响应。

在 work 方法中, 通过创建一个 Socket 对象, 使用指定的名称和端口号与服务器建立连接。然后通过获取套接字的输出流, 使用 DataOutputStream 对其进行包装, 以便将数据写入服务器。接着, 通过获取套接字的输入流, 使用 DataInputStream 对其进行包装, 以便从服务器读取响应。最后, 关闭与服务器的套接字连接, 并返回从服务器接收到的响应。

2.1.3 UI 类

UI 类用来绘制图形操作界面, 是客户端的主要部分, 通过使用 Worker 类所提供的功能与用户交互。它包含了以下技术:

AWT (Abstract Window Toolkit): AWT 是 Java 的原始窗口工具包, 提供了创建图形用户界面 (GUI) 的基本组件和功能。代码中使用了 java.awt 包下的类, 如 java.awt.Frame 和 java.awt.event.ActionEvent。

Swing: Swing 是 Java 的 GUI 工具包, 提供了更多的 GUI 组件和更丰富的功能, 用于创建富客户端应用程序。代码中使用了 javax.swing 包下的类, 如 javax.swing.JFrame、javax.swing.JButton 和 javax.swing.JPanel。

Event handling: 代码中实现了 ActionListener 接口, 并重写了 actionPerformed 方法, 用于处理按钮点击事件。

Layout management: 代码中使用了不同的布局管理器来布局界面, 如 null 布局、GridLayout 和 CardLayout。

文本框和密码框: 代码中使用了 JTextArea 来显示多行文本, JPasswordField 来显示密码输入框。

通过上述技术, 代码实现了一个基于 Java 的简单银行客户端界面。界面中包含了不同的窗口和按钮, 实现了登录、注册、查询余额、取款、存款、转账、修改信息等功能。具体的界面和功能描述如下:

Frame1: 初始界面, 包含“登录”和“注册”按钮。

Frame2: 登录界面, 包含输入用户名和密码的文本框和登录按钮。

Frame3: 用户界面, 登录成功后显示, 包含查询余额、取款、存款、转账、修改信息和销户等按钮。

Frame4: 查询余额界面, 显示当前用户余额和刷新按钮。

Frame5: 取款界面, 包含输入取款数目的文本框和取款按钮。

Frame6: 存款界面, 包含输入存款数目的文本框和存款按钮。

Frame7: 转账界面, 包含输入转入用户名和金额的文本框和转账按钮。

Frame8: 修改信息界面, 包含输入新密码、修改生日和手机号的文本框和修改按钮。

Frame9: 注册界面, 包含输入用户名、生日、手机号和密码的文本框和注册按钮。

Frame10: 管理员界面, 显示管理员操作界面, 包含批量开户和年终报告按钮。

除了上述界面, 代码中还包含了相关的按钮事件处理和布局管理的逻辑。

2.1.4 Worker 类

这个类用于向 UI 类提供一些函数接口, 这些函数实现了所要求的一些功能, 并且通过发送一定格式的字符串来访问服务端数据库, Worker 类与服务端的 Operator 类相对应。其中 Worker 类用来形成命令字符串, Operator 类用来解析命令字符串。该类的编写过程主要运用了以下的 Java 编程技术:

文件操作技术: 使用 java.io.File 类和 java.io.FileOutputStream 类来进行文件读写操作。

Excel 文件读取技术：使用 `jxl.Workbook` 类和 `jxl.Sheet` 类来读取 Excel 文件中的数据。

异常处理技术：通过 `try-catch` 语句块处理可能发生的异常情况。

PDF 文档操作技术：使用 `com.itextpdf.text.Document` 类和 `com.itextpdf.text.Paragraph` 类来创建和编辑 PDF 文档，以及使用 `com.itextpdf.text.pdf.PdfWriter` 类将文档写入文件。

这段代码的主要目标是实现一个银行系统的客户端功能。下面是该代码实现的功能：

`Worker` 类的构造函数初始化了一个 `Client` 对象，用于与服务器建立连接。

`check_usr` 方法用于检查指定用户名是否存在于银行系统的数据库中。

`check_login` 方法用于判断指定用户名和密码是否匹配，即用于用户登录认证。

`check_money` 方法用于查询指定用户的账户余额。

`withdraw_money` 方法用于从指定用户的账户中取款，并返回操作是否成功。

`store_money` 方法用于向指定用户的账户中存款，并返回操作是否成功。

`transfer` 方法用于将一位用户的资金转账给另一位用户，并返回操作是否成功。

`signin` 方法用于注册一个新用户，并返回注册是否成功。

`bulk_signin` 方法用于批量注册多个新用户，从指定的 Excel 文件中读取用户信息，并返回成功注册的用户数量。

`logout` 方法用于注销指定用户的账户，并返回操作是否成功。

`update_info` 方法用于修改指定用户的个人信息，如密码、生日和电话号码，并返回操作是否成功。

`createpdf` 方法用于生成银行年度报告的 PDF 文件，包括总账户数、新增账户数和银行总资金，并返回操作是否成功。

但是其实代码中还存在一些潜在的缺陷。例如，异常处理部分仅仅是空的占位符，并没有实际处理可能发生的异常情况。此外，代码中的数据库操作涉及直接拼接 SQL 语句，存在安全风险和易受 SQL 注入攻击的问题。为了提高代码的健壮性和安全性，应该完善异常处理，并使用参数化查询或 ORM 框架来执行数据库操作。

2.2 Server 文件夹中的类

2.2.1 Main_server 类

该类的目标是创建一个简单的服务器，通过监听指定的端口，接受客户端的连接请求，并为每个客户端连接创建一个新的线程来处理与客户端的通信。这种架构允许服务器同时处理多个客户端请求，提高了服务器的并发性能和响应能力。通过多线程的方式，服务器能够同时处理多个客户端的请求，而不会阻塞其他客户端的连接。

该类中的代码使用了 **Socket** 编程技术和多线程技术。

Socket 编程技术：该代码创建了一个服务器端的 **Socket** 对象，并使用 **ServerSocket** 类在指定端口（6606）上监听客户端的连接请求。一旦有客户端连接成功，服务器将创建一个新的 **Socket** 对象来处理与该客户端的通信。

多线程技术：代码中的 **while** 循环用于不断接受客户端的连接请求。当有客户端连接成功后，代码通过创建新的线程来处理与该客户端的通信，从而实现了多个客户端之间的并发处理。每个客户端连接都在一个独立的线程中进行处理，以避免阻塞其他客户端的连接。

2.2.2 Server 类

该类的目标是创建一个服务器，用于处理客户端发送的命令并提供相应的响应。服务器通过 **Socket** 编程实现与客户端的通信，接收客户端的连接请求，并根据客户端发送的命令执行相应的操作。根据命令的类型，服务器通过操作数据库（**DB** 对象）执行数据库查询或更新操作，并将结果发送回客户端。多线程技术使服务器能够同时处理多个客户端连接，提高了服务器的并发性能和响应能力。

本类中也使用了 **Socket** 编程技术和多线程编程技术，除此之外还有：

输入输出流操作：代码使用 **DataInputStream** 和 **DataOutputStream** 来读取和写入数据。通过 **DataInputStream** 对象的 **readUTF()** 方法读取客户端发送的命令，并使用 **DataOutputStream** 对象的 **writeUTF()** 方法向客户端发送响应结果。

2.2.3 DB 类

该类的目标是实现与 **MySQL** 数据库的连接和交互。它提供了数据库查询和更新的方法，通过执行传入的 **SQL** 语句，从数据库中获取查询结果或执行更新操作。使用 **JDBC** 技术，它允许 **Java** 应用程序与数据库进行通信和数据交换，实现了与数据库的交互功能。使用同步方法（**synchronized**）确保多线程环境下的线程安全性。

本类运用了以下技术：

JDBC (Java Database Connectivity)：该类使用了 JDBC 技术来连接和操作数据库。它使用 JDBC 驱动程序 (`com.mysql.cj.jdbc.Driver`) 来连接 MySQL 数据库，并使用 JDBC 的 API 来执行 SQL 查询和更新操作。

数据库连接和操作：在 DB 类的构造函数中，它使用 `DriverManager` 类和 `getConnection()` 方法建立与 MySQL 数据库的连接。然后，它使用 `Statement` 对象 (通过 `conn.createStatement()` 方法获得) 执行 SQL 语句。通过调用 `executeQuery()` 方法执行查询操作，以获取结果集 (`ResultSet`)，并通过调用 `execute()` 方法执行更新操作。

异常处理：代码使用了异常处理机制来处理可能出现的 SQL 异常和其他异常。通过捕获和处理 `SQLException` 和 `Exception` 异常，它在发生错误时打印异常堆栈跟踪，提供了一定的错误处理和调试能力。

2.2.4 Operator 类

该类的目标是解析命令字符串，从中提取出不同的部分，包括标志位 (`flag`)、列名 (`column`) 和 SQL 语句。根据命令字符串的格式，它通过字符串处理和转换技术，将命令字符串拆分成相应的部分，以便后续使用。这些提取的部分可以用于执行数据库查询和更新操作，从而实现对数据库的操作。该类的方法提供了从命令字符串中获取不同部分的功能，以满足特定的操作需求。

本类运用了以下技术：

字符串处理：该类使用字符串处理技术来解析输入的命令字符串。它通过使用字符串的索引、子字符串和字符处理方法，提取出命令字符串中的各个部分。

字符串转换：代码使用了字符串转换技术来将字符转换为数字。通过调用 `parseInt()` 方法，它将字符型的数字转换为相应的整数值。

3 编译及运行

1、准备工作首先将服务端代码放入服务器（或虚拟机代替服务器），并在服务端创建相应的项目文件夹 `fmbank`，在 `fmbank` 文件夹中创建存放字节码文件文件夹 `bin` 和源码文件夹 `src`，将服务端代码 `Server` 文件夹放入 `src` 中。

2、启动服务端程序在服务端，首先进入项目文件 `fmbank`，之后输入命令：

```
javac -encoding utf8 -d bin src/Server/*.java
```

编译 java 文件，并将 class 文件放入 `bin` 文件夹中。之后进入 `bin` 文件（相应命令：`cd bin`），输入：

```
java Server.Main_server
```

启动程序，当屏幕输出如下提示表示程序运行成功。3、启动客户端程序首先进

```
[root@localhost]# cd fmbank
[root@localhost fmbank]# javac -encoding utf8 -d bin src/Server/*.java
[root@localhost fmbank]# cd bin
[root@localhost bin]# java Server.Main_server
Wait....
```

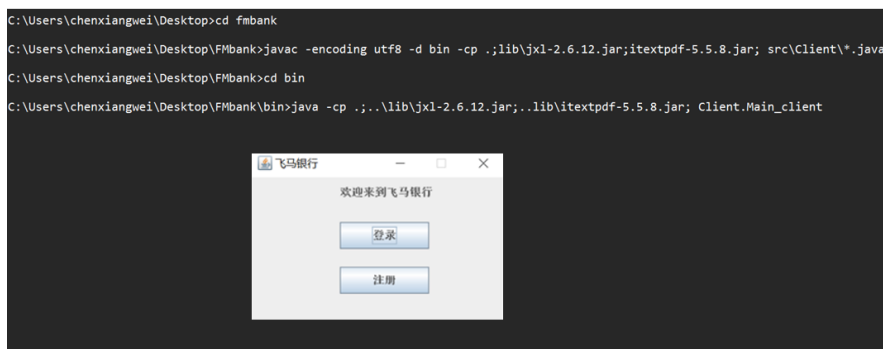
入项目文件 `fmbank`，之后输入如下命令编译代码：

```
javac -encoding utf8 -d bin -cp .;lib\jxl-2.6.12.jar;itextpdf-5.5.8.jar;
src\*Client\*.java
```

其中 `-cp` 命令表示编译时引入第三方包。之后进入 `bin` 文件，输入如下命令启动程序：

```
java -cp .;..\lib\jxl-2.6.12.jar;..\lib\itextpdf-5.5.8.jar;
Client.Main_client
```

出现如图界面表示启动成功：



4 结果展示

1、注册功能

输入用户名(不区分大小写)

测试

输入生日

2022-6-16

输入手机号

123456

输入密码

注册

返回

输入用户名(不区分大小写)

注册

注册成功!

确定

Back arrow

Next arrow

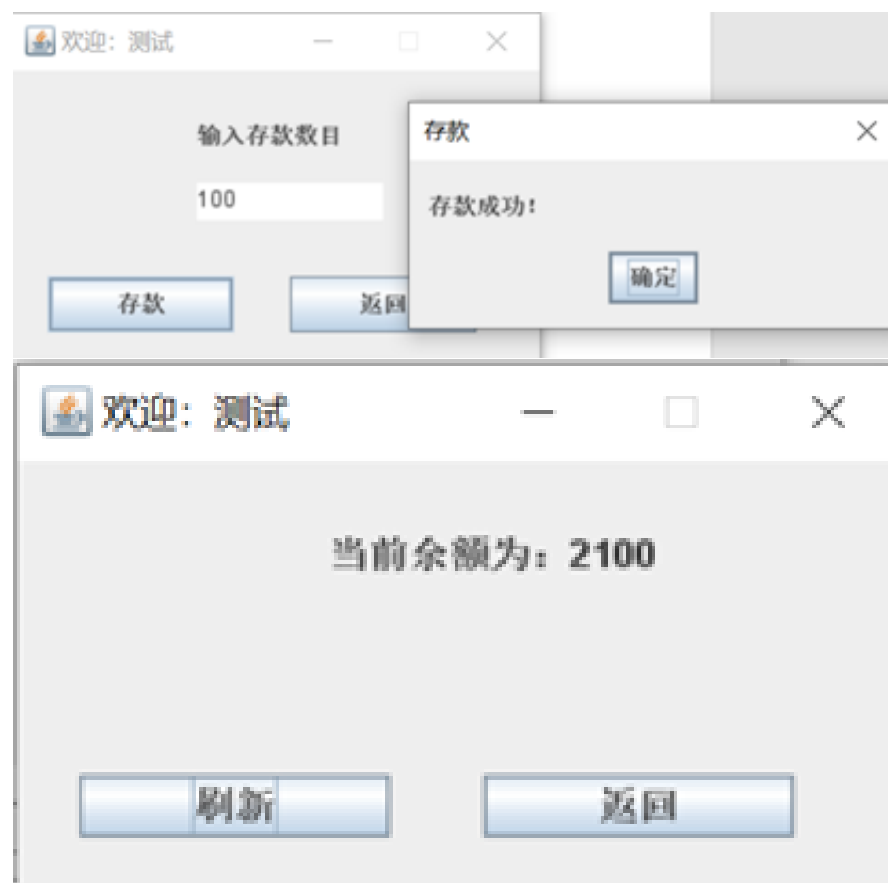
2、登录功能



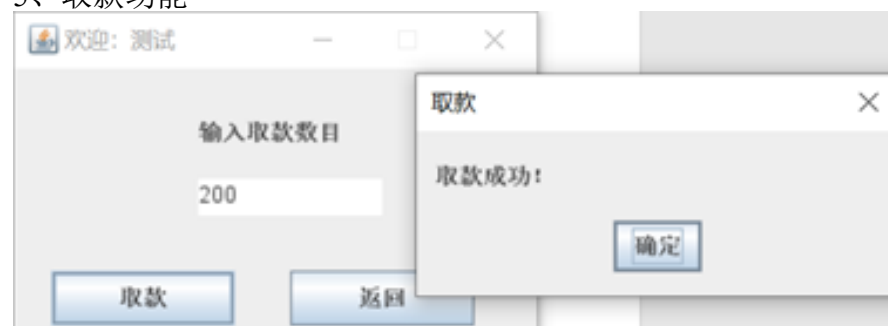
3、查询功能



4、存款功能

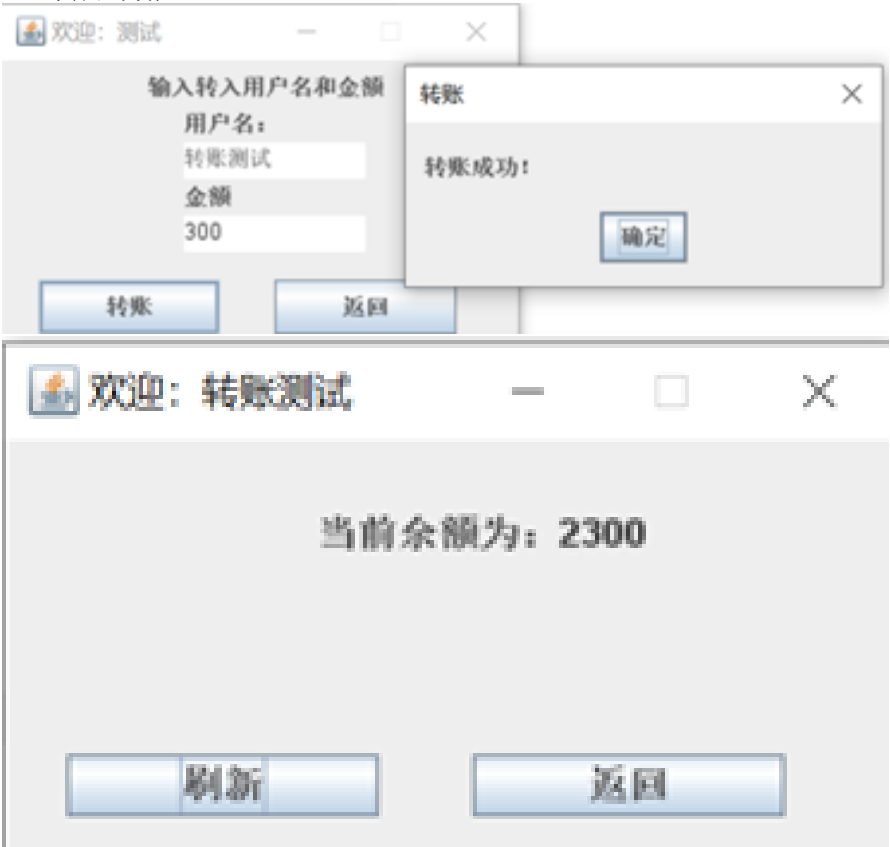


5、取款功能





6、转账功能



7、销户功能



8、报表功能

当用用户名 root（管理员默认密码为八个八）登录时，会跳转到管理员界面，在管理员界面可以使用批量注册和生成报表功能。



A 客户端 Client 部分代码

Listing 1: Main_Client.java

```
1 package Client;
2
3 public class Main_client {
4     public static void main(String[] args) {
5         new UI("欢迎: ",300,200);
6     }
7 }
```

Listing 2: Client.java

```
1 package Client;
2
3 import java.net.*;
4 import java.io.*;
5
6 public class Client {
7     private String name;
8     private int port;
9
10    Client(String name, int port) {
11        this.name = name;
12        this.port = port;
13    }
14
15    /**
16     * 执行客户端的工作，向服务器发送数据并接收响应
17     *
18     * @param data 要发送给服务器的数据
19     * @return 从服务器接收到的响应
20     */
21    public String work(String data) {
22        String res = "Null";
23        try {
24            // 创建与服务器的套接字连接，使用指定的
```



```

                                名称和端口号
25      Socket c = new Socket(name, port);
26
27      // 获取套接字的输出流，并使用
                                DataOutputStream进行包装以便写入数据
28      OutputStream outToServer = c.
                                getOutputStream();
29      DataOutputStream out = new
                                DataOutputStream(outToServer);
30
31      // 将数据写入服务器
32      out.writeUTF(data);
33
34      // 获取套接字的输入流，并使用
                                DataInputStream进行包装以便读取数据
35      InputStream inFromServer = c.
                                getInputStream();
36      DataInputStream in = new
                                DataInputStream(inFromServer);
37
38      // 从服务器读取响应
39      res = in.readUTF();
40
41      // 关闭套接字连接
42      c.close();
43      } catch (SocketException se) {
44          // 处理可能发生的套接字异常
45          // （例如，如果连接被拒绝或服务器不可
                                用）
46      } catch (IOException e) {
47          // 处理可能发生的其他输入/输出异常
48      }
49      // 返回从服务器接收到的响应
50      return res;
51  }
52 }
```

Listing 3: UI.java

```
1  UI.java
2  package Client;
3
4  import java.awt.*;
5  import javax.swing.*;
6  import java.awt.event.ActionEvent;
7  import java.awt.event.ActionListener;
8
9  public class UI implements ActionListener{
10
11      // 用户参数
12      private Worker w;
13      private String usr;
14      private String pas;
15      private int count;
16
17      // 参数
18      private String Title;
19      private int size1;
20      private int size2;
21      private final int xloc = 400;
22      private final int yloc = 300;
23
24      private JFrame current;
25      private JFrame frame1;
26      private JFrame frame2;
27      private JFrame frame3;
28      private JFrame frame4;
29      private JFrame frame5;
30      private JFrame frame6;
31      private JFrame frame7;
32      private JFrame frame8;
33      private JFrame frame9;
34      private JFrame frame10;
35
36      // 回退
37      private JButton bak1;
```

```
38         private JButton bak2;
39         private JButton bak3;
40         private JButton bak4;
41         private JButton bak5;
42         private JButton bak6;
43         // frame1
44         private JButton btn1;
45         private JButton btn2;
46         // frame2
47         private JButton btn3;
48         private JButton btn4;
49         // frame3
50         private JButton btn5;
51         private JButton btn6;
52         private JButton btn7;
53         private JButton btn8;
54         private JButton btn9;
55         private JButton btn10;
56         // 查询余额
57         private JButton btn15;
58         //取款
59         private JButton btn11;
60         // 存款
61         private JButton btn12;
62         // 转账
63         private JButton btn13;
64         // 修改信息
65         private JButton btn14;
66         // 提交注册
67         private JButton btn16;
68         // 管理员操作
69         private JButton btn17;
70         private JButton btn18;
71
72         // 卡片布局
73         private CardLayout c1;
74         private JPanel cards1;
```

```
75
76     // 表单
77     private JTextArea jta1;
78     private JTextArea jta2;
79     private JTextArea jta3;
80     private JTextArea jta4;
81     private JTextArea jta5;
82     private JTextArea jta6;
83     private JTextArea jta7;
84     private JTextArea jta8;
85     // 用户名
86     private JTextArea jta9;
87     // 生日
88     private JTextArea jta10;
89     // 手机号
90     private JTextArea jta11;
91
92     // 密码框
93     private JPasswordField jpd1;
94     private JPasswordField jpd2;
95     private JPasswordField jpd3;
96     // ...
97
98     // 初始化
99     UI(String title, int size1, int size2) {
100         this.w = new Worker();
101         this.Title = title;
102         this.size1 = size1;
103         this.size2 = size2;
104         Frame1();
105     }
106
107     // 初始界面
108     private void Frame1() {
109         frame1 = new JFrame("飞马银行");
110         frame1.setBounds(xloc,yloc,size1,size2);
111         frame1.setResizable(false);
```

```
112         frame1.setDefaultCloseOperation(JFrame.  
            EXIT_ON_CLOSE);  
113         frame1.setLayout(null);  
114         frame1.setBackground(Color.gray);  
115  
116         JLabel label = new JLabel("欢迎来到飞马银行");  
117  
118         btn1 = new JButton("登录");  
119         btn2 = new JButton("注册");  
120  
121         label.setBounds(100,2,200,30);  
122         btn1.setBounds(100,50,100,30);  
123         btn2.setBounds(100,100,100,30);  
124  
125         btn1.addActionListener(this);  
126         btn2.addActionListener(this);  
127  
128         frame1.add(label);  
129         frame1.add(btn1);  
130         frame1.add(btn2);  
131  
132         current = frame1;  
133         current.setVisible(true);  
134     }  
135  
136     // 登录界面  
137     private void Frame2() {  
138         frame2 = new JFrame("飞马银行");  
139         frame2.setBounds(xloc,yloc,size1,size2);  
140         frame2.setResizable(false);  
141         frame2.setDefaultCloseOperation(JFrame.  
            EXIT_ON_CLOSE);  
142         frame2.setBackground(Color.gray);  
143  
144         JPanel p = new JPanel();  
145         cards1 = new JPanel(new CardLayout(50,10));  
146         c1 = (CardLayout)(cards1.getLayout());
```

```
147
148         JLabel label = new JLabel("输入用户名和密码");
149         jta1 = new JTextArea(1,10);
150         jpd1 = new JPasswordField(10);
151         btn3 = new JButton(" 登录 ");
152         btn4 = new JButton("忘记密码");
153
154         p.add(label);
155         p.add(jta1);
156         p.add(jpd1);
157         p.add(btn3);
158         p.add(btn4);
159
160         cards1.add(p,"ok");
161         frame2.add(cards1);
162
163         btn3.addActionListener(this);
164         btn4.addActionListener(this);
165
166     }
167
168     // 用户界面
169     private void Frame3() {
170         frame3 = new JFrame(Title+usr);
171         frame3.setBounds(xloc,yloc,size1,size2);
172         frame3.setResizable(false);
173         frame3.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
174         frame3.setBackground(Color.gray);
175
176         btn5 = new JButton("查询余额");
177         btn6 = new JButton("取款");
178         btn7 = new JButton("存款");
179         btn8 = new JButton("转账");
180         btn9 = new JButton("修改信息");
181         btn10 = new JButton("销户");
182
```

```
183         frame3.add(btn5);
184         frame3.add(btn6);
185         frame3.add(btn7);
186         frame3.add(btn8);
187         frame3.add(btn9);
188         frame3.add(btn10);
189
190         frame3.setLayout(new GridLayout(3,2,5,5));
191
192         btn5.addActionListener(this);
193         btn6.addActionListener(this);
194         btn7.addActionListener(this);
195         btn8.addActionListener(this);
196         btn9.addActionListener(this);
197         btn10.addActionListener(this);
198
199     }
200
201     // 查询余额界面
202     private void Frame4() {
203         frame4 = new JFrame(Title+usr);
204         //frame4.setSize(size1,size2);
205         frame4.setBounds(xloc,yloc,size1,size2);
206         frame4.setResizable(false);
207         frame4.setDefaultCloseOperation(JFrame.
                EXIT_ON_CLOSE);
208         frame4.setBackground(Color.gray);
209
210         frame4.setLayout(null);
211
212         JLabel label = new JLabel("当前余额为: "+count+
                "");
213         btn15 = new JButton("刷新");
214         bak1 = new JButton("返回");
215
216         label.setBounds(100,20,150,20);
217         btn15.setBounds(20,100,100,20);
```

```
218         bak1.setBounds(150,100,100,20);
219
220         frame4.add(label);
221         frame4.add(btn15);
222         frame4.add(bak1);
223
224         btn15.addActionListener(this);
225         bak1.addActionListener(this);
226     }
227
228     // 取款界面
229     private void Frame5() {
230         frame5 = new JFrame(Title+usr);
231         frame5.setBounds(xloc,yloc,size1,size2);
232         frame5.setResizable(false);
233         frame5.setDefaultCloseOperation(JFrame.
                EXIT_ON_CLOSE);
234         frame5.setBackground(Color.gray);
235         frame5.setLayout(null);
236
237         JLabel label = new JLabel("输入取款数目");
238         jta2 = new JTextArea(1,10);
239         btn11 = new JButton("取款");
240         bak2 = new JButton("返回");
241
242         label.setBounds(100,10,100,50);
243         jta2.setBounds(100,60,100,20);
244         btn11.setBounds(20,110,100,30);
245         bak2.setBounds(150,110,100,30);
246
247         frame5.add(label);
248         frame5.add(jta2);
249         frame5.add(btn11);
250         frame5.add(bak2);
251
252
253         btn11.addActionListener(this);
```



```
254         bak2.addActionListener(this);
255     }
256
257     // 存款界面
258     private void Frame6() {
259         frame6 = new JFrame(Title+usr);
260         frame6.setBounds(xloc,yloc,size1,size2);
261         frame6.setResizable(false);
262         frame6.setDefaultCloseOperation(JFrame.
                EXIT_ON_CLOSE);
263         frame6.setBackground(Color.gray);
264
265         frame6.setLayout(null);
266
267         JLabel label = new JLabel("输入存款数目");
268         jta3 = new JTextArea(1,10);
269         btn12 = new JButton("存款");
270         bak3 = new JButton("返回");
271
272         btn12.addActionListener(this);
273         bak3.addActionListener(this);
274
275         label.setBounds(100,10,100,50);
276         jta3.setBounds(100,60,100,20);
277         btn12.setBounds(20,110,100,30);
278         bak3.setBounds(150,110,100,30);
279
280         frame6.add(label);
281         frame6.add(jta3);
282         frame6.add(btn12);
283         frame6.add(bak3);
284     }
285
286     // 转账界面
287     private void Frame7() {
288         frame7 = new JFrame(Title+usr);
289         frame7.setBounds(xloc,yloc,size1,size2);
```

```
290         frame7.setResizable(false);
291         frame7.setDefaultCloseOperation(JFrame.
                EXIT_ON_CLOSE);
292         frame7.setBackground(Color.gray);
293
294         frame7.setLayout(null);
295
296         JLabel label1 = new JLabel("输入转入用户名和金
                额");
297         JLabel label2 = new JLabel("用户名: ");
298         JLabel label3 = new JLabel("金额");
299         jta4 = new JTextArea(1,10);
300         jta5 = new JTextArea(1,10);
301         btn13 = new JButton("转账");
302         bak4 = new JButton("返回");
303
304         btn13.addActionListener(this);
305         bak4.addActionListener(this);
306
307         label1.setBounds(80,5,150,20);
308         label2.setBounds(100,25,100,20);
309         jta4.setBounds(100,45,100,20);
310         label3.setBounds(100,65,100,20);
311         jta5.setBounds(100,85,100,20);
312         btn13.setBounds(20,120,100,30);
313         bak4.setBounds(150,120,100,30);
314
315         frame7.add(label1);
316         frame7.add(label2);
317         frame7.add(jta4);
318         frame7.add(label3);
319         frame7.add(jta5);
320         frame7.add(btn13);
321         frame7.add(bak4);
322     }
323
324     // 修改信息界面
```

```
325     private void Frame8() {
326         frame8 = new JFrame(Title+usr);
327         frame8.setBounds(xloc,yloc,size1,size2);
328         frame8.setResizable(false);
329         frame8.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
330         frame8.setBackground(Color.gray);
331
332         frame8.setLayout(null);
333
334         JLabel label1 = new JLabel("新密码:");
335         JLabel label2 = new JLabel("修改生日");
336         JLabel label3 = new JLabel("修改手机号");
337         Jpd2 = new JPasswordField();
338         jta7 = new JTextArea(1,10);
339         jta8 = new JTextArea(1,10);
340         btn14 = new JButton("修改");
341         bak5 = new JButton("返回");
342
343         btn14.addActionListener(this);
344         bak5.addActionListener(this);
345
346         label1.setBounds(100,5,100,20);
347         jpd2.setBounds(100,25,100,20);
348         label2.setBounds(100,45,100,20);
349         jta7.setBounds(100,65,100,20);
350         label3.setBounds(100,85,100,20);
351         jta8.setBounds(100,105,100,20);
352         btn14.setBounds(20,140,100,20);
353         bak5.setBounds(150,140,100,20);
354
355         frame8.add(label1);
356         frame8.add(label2);
357         frame8.add(label3);
358         frame8.add(jpd2);
359         frame8.add(jta7);
360         frame8.add(jta8);
```

```
361         frame8.add(btn14);
362         frame8.add(bak5);
363     }
364
365     // 注册界面
366     private void Frame9() {
367         frame9 = new JFrame(Title);
368         frame9.setBounds(xloc,yloc,size1,size2+50);
369         frame9.setResizable(false);
370         frame9.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
371         frame9.setBackground(Color.gray);
372
373         frame9.setLayout(null);
374
375         JLabel label1 = new JLabel("输入用户名(不区分大
            小写)");
376         JLabel label2 = new JLabel("输入生日");
377         JLabel label3 = new JLabel("输入手机号");
378         JLabel label4 = new JLabel("输入密码");
379         jta9 = new JTextArea(1,10);
380         jta10 = new JTextArea(1,10);
381         jta11 = new JTextArea(1,10);
382         jpd3 = new JPasswordField();
383         btn16 = new JButton("注册");
384         bak6 = new JButton("返回");
385
386         label1.setBounds(80,5,200,20);
387         jta9.setBounds(100,25,100,20);
388         label2.setBounds(100,45,100,20);
389         jta10.setBounds(100,65,100,20);
390         label3.setBounds(100,85,100,20);
391         jta11.setBounds(100,105,100,20);
392         label4.setBounds(100,125,100,20);
393         jpd3.setBounds(100,145,100,20);
394         btn16.setBounds(20,170,100,20);
395         bak6.setBounds(150,170,100,20);
```

```
396
397         btn16.addActionListener(this);
398         bak6.addActionListener(this);
399
400         frame9.add(label11);
401         frame9.add(label12);
402         frame9.add(label13);
403         frame9.add(label14);
404         frame9.add(jta9);
405         frame9.add(jta10);
406         frame9.add(jta11);
407         frame9.add(jpd3);
408         frame9.add(btn16);
409         frame9.add(bak6);
410     }
411
412     // 管理员界面
413     private void Frame10() {
414         frame10 = new JFrame(Title+"管理员"+usr);
415         frame10.setBounds(xloc,yloc,size1,size2);
416         frame10.setResizable(false);
417         frame10.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
418         frame10.setLayout(null);
419         frame10.setBackground(Color.gray);
420
421         JLabel label = new JLabel("管理员操作界面");
422
423         btn17 = new JButton("批量开户");
424         btn18 = new JButton("年终报告");
425
426         label.setBounds(100,2,200,30);
427         btn17.setBounds(100,50,100,30);
428         btn18.setBounds(100,100,100,30);
429
430         btn17.addActionListener(this);
431         btn18.addActionListener(this);
```

```
432
433         frame10.add(label);
434         frame10.add(btn17);
435         frame10.add(btn18);
436
437     }
438
439     // ...
440
441
442
443     // 事件监听
444     public void actionPerformed(ActionEvent e) {
445         if(e.getSource() == bak6) bak0();
446         else if(e.getSource() == bak1 || e.getSource()
447             == bak2 || e.getSource() == bak3 || e.
448             getSource() == bak4 || e.getSource() == bak5
449             ) bak();
450         else if(e.getSource() == btn1) btn_1();
451         else if(e.getSource() == btn2) btn_2();
452         else if(e.getSource() == btn3) btn_3();
453         else if(e.getSource() == btn4) btn_4();
454         else if(e.getSource() == btn5) btn_5();
455         else if(e.getSource() == btn6) btn_6();
456         else if(e.getSource() == btn7) btn_7();
457         else if(e.getSource() == btn8) btn_8();
458         else if(e.getSource() == btn9) btn_9();
459         else if(e.getSource() == btn10) btn_10();
460         else if(e.getSource() == btn11) btn_11();
461         else if(e.getSource() == btn12) btn_12();
462         else if(e.getSource() == btn13) btn_13();
463         else if(e.getSource() == btn14) btn_14();
464         else if(e.getSource() == btn15) btn_15();
465         else if(e.getSource() == btn16) btn_16();
466         else if(e.getSource() == btn17) btn_17();
467         else if(e.getSource() == btn18) btn_18();
468     }
```

```
466
467     // 返回按钮
468     private void bak() {
469         current.setVisible(false);
470         frame3.setVisible(true);
471         current = frame3;
472     }
473
474     private void bak0() {
475         current.setVisible(false);
476         frame1.setVisible(true);
477         current = frame1;
478     }
479
480
481     // 跳转登录界面
482     private void btn_1() {
483         Frame2();
484         current.setVisible(false);
485         frame2.setVisible(true);
486         c1.show(cards1, "ok");
487         current = frame2;
488     }
489
490     // 跳转创建账户页面
491     private void btn_2() {
492         Frame9();
493         current.setVisible(false);
494         frame9.setVisible(true);
495         current = frame9;
496     }
497
498     // 跳转用户页面
499     private void btn_3() {
500         usr = jta1.getText();
501         pas = new String(jpd1.getPassword());
502         if(w.check_login(usr,pas) == false) {
```

```
503         JOptionPane.showMessageDialog(frame2, "
           密码或用户名错误!", "错误", 2);
504     } else {
505         if(usr.equals("root")) {
506             Frame10();
507             current.setVisible(false);
508             frame10.setVisible(true);
509             current = frame10;
510         } else {
511             Frame3();
512             current.setVisible(false);
513             frame3.setVisible(true);
514             current = frame3;
515         }
516     }
517 }
518
519 // 忘记密码
520 private void btn_4() {
521     JOptionPane.showMessageDialog(frame2, "仔细回忆
           ", "忘记密码", 1);
522 }
523
524 // 跳转查询余额页面
525 private void btn_5() {
526     Frame4();
527     count = w.check_money(usr);
528     Frame4();
529     current.setVisible(false);
530     frame4.setVisible(true);
531     current = frame4;
532 }
533
534 // 跳转取款页面
535 private void btn_6() {
536     Frame5();
537     current.setVisible(false);
```



```
538         frame5.setVisible(true);
539         current = frame5;
540     }
541
542     // 跳转存款页面
543     private void btn_7() {
544         Frame6();
545         current.setVisible(false);
546         frame6.setVisible(true);
547         current = frame6;
548     }
549
550     // 跳转转账页面
551     private void btn_8() {
552         Frame7();
553         current.setVisible(false);
554         frame7.setVisible(true);
555         current = frame7;
556     }
557
558     // 跳转修改信息页面
559     private void btn_9() {
560         Frame8();
561         current.setVisible(false);
562         frame8.setVisible(true);
563         current = frame8;
564     }
565
566     // 销户
567     private void btn_10() {
568         if(w.logout(usr)) {
569             JOptionPane.showMessageDialog(frame3, "
                    注销成功!", "销户", 2);
570             btn_1();
571         }
572     }
573
```

```
574 // 取款提交
575 private void btn_11() {
576     try {
577         int withdraw = Integer.parseInt(jta2.
578             getText());
579         // System.out.println(withdraw);
580         w.withdraw_money(usr, withdraw);
581         JOptionPane.showMessageDialog(frame5, "
582             取款成功!", "取款", JOptionPane.
583             PLAIN_MESSAGE);
584     } catch (Exception e) {JOptionPane.
585         showMessageDialog(frame5, "请输入数字!", "格
586             式错误", 2);}
587 }
588 // 存款提交
589 private void btn_12() {
590     try {
591         int store = Integer.parseInt(jta3.
592             getText());
593         // System.out.println(store);
594         w.store_money(usr, store);
595         JOptionPane.showMessageDialog(frame6, "
596             存款成功!", "存款", JOptionPane.
597             PLAIN_MESSAGE);
598     } catch (Exception e) {JOptionPane.
599         showMessageDialog(frame6, "请输入数字!", "格
600             式错误", 2);}
601 }
602 // 转账提交
603 private void btn_13() {
604     try {
605         String tousr = jta4.getText();
606         int transfer = Integer.parseInt(jta5.
607             getText());
```

```
600         if(touser.equals("root")) {
601             JOptionPane.showMessageDialog(
                frame7, "禁止向管理员转账！ ",
                "错误",2);
602         } else {
603             if(w.transfer(usr, tousr,
                transfer) == false)
                JOptionPane.
                showMessageDialog(frame7, "
                转账失败！用户不存在或余额不
                足！","转账",JOptionPane.
                PLAIN_MESSAGE);
604             else JOptionPane.
                showMessageDialog(frame7, "
                转账成功！","转账",
                JOptionPane.PLAIN_MESSAGE);
605         }
606     } catch (Exception e) {JOptionPane.
        showMessageDialog(frame7, "请输入数字！","格
        式错误",2);}
607
608 }
609
610 // 修改信息提交
611 private void btn_14() {
612     String pas = new String(jpd2.getPassword());
613     String birth = jta7.getText();
614     String phone = jta8.getText();
615     if(w.update_info(usr, pas, birth, phone)) {
616         JOptionPane.showMessageDialog(frame8, "
            修改成功！","修改",JOptionPane.
            PLAIN_MESSAGE);
617     }
618 }
619
620 // 余额刷新
621 private void btn_15() {
```

```
622         btn_5();
623     }
624
625     // 注册提交
626     private void btn_16() {
627         String usr_name = jta9.getText();
628         String birth = jta10.getText();
629         String phone = jta11.getText();
630         String pass = new String(jpd3.getPassword());
631         if(w.signin(usr_name, birth, phone, pass, 1,
632             9941, 2000) == false) {
633             JOptionPane.showMessageDialog(frame9, "
634                 用户名重复!", "注册失败", 2);
635         } else {
636             JOptionPane.showMessageDialog(frame9, "
637                 注册成功!", "注册", JOptionPane.
638                 PLAIN_MESSAGE);
639             btn_1();
640         }
641     }
642
643     // 批量开户
644     private void btn_17() {
645         JFileChooser fc = new JFileChooser("C:\\");
646         int val = fc.showOpenDialog(frame10);
647         if(val == fc.APPROVE_OPTION) {
648             System.out.println(fc.getSelectedFile()
649                 .toString());
650             int res = w.bulk_signin(fc.
651                 getSelectedFile().toString());
652             if(res == -1) {
653                 JOptionPane.showMessageDialog(
654                     frame10, "文件格式错误或不支
655                         持xlsx格式, 请使用xls格式! "
656                         , "文件格式错误", 2);
657             } else {
658                 JOptionPane.showMessageDialog(
```

```
        frame10, "成功注册"+res+"个  
        用户! ", "成功", JOptionPane.  
        PLAIN_MESSAGE);  
650    }  
651    }  
652    }  
653  
654    // 年终报告  
655    private void btn_18() {  
656        if(w.createpdf(".\\REPORT.pdf")) {  
657            JOptionPane.showMessageDialog(frame10,  
            "生成报告成功! 文件在当前目录中。", "  
            生成报告", JOptionPane.PLAIN_MESSAGE)  
            ;  
658        } else {  
659            JOptionPane.showMessageDialog(frame10,  
            "生成报告失败! ", "生成报告", 2);  
660        }  
661    }  
662  
663 }
```

Listing 4: Worker.java

```
1 package Client;  
2  
3 import java.io.File;  
4 import java.io.FileOutputStream;  
5 import java.io.IOException;  
6 import jxl.*;  
7 import jxl.read.biff.BiffException;  
8 import com.itextpdf.text.Document;  
9 import com.itextpdf.text.DocumentException;  
10 import com.itextpdf.text.Paragraph;  
11 import com.itextpdf.text.pdf.PdfWriter;  
12  
13  
14 public class Worker {
```

```
15     private Client c;
16
17     // 格式: flag(有返回值为0) + number(int为1) + length(
18         column) + column + sql
19     // 表: id(int2), usr(string3), pass(string4), schn(int4
20         ), phone(string5), sex(int3), birth(string5), money(
21         int5)
22
23     // 构造, 初始化
24     Worker() {
25         this.c = new Client("192.168.66.128",6606);
26     }
27
28     // 查询用户是否存在 不存在返回true, 存在返回false
29     private boolean check_usr(String usr) {
30         return send_msg("012idSELECT id FROM fmbank
31             WHERE usr='"+usr+"';").equals("Null");
32     }
33
34     // 判断是否登录成功
35     public boolean check_login(String usr, String pass) {
36         if(check_usr(usr) == true) return false;
37         return send_msg("004passSELECT pass FROM fmbank
38             WHERE usr='"+usr+"';").equals(pass);
39     }
40
41     // 发送数据
42     private String send_msg(String data) {
43         return c.work(data);
44     }
45
46     // 余额查询, 返回余额
47     public int check_money(String usr) {
48         return Integer.parseInt(send_msg("015
49             moneySELECT money FROM fmbank WHERE usr='"+
50             usr+"';"));
51     }
52 }
```

```
45
46 // 取款, 返回是否成功
47 public boolean withdraw_money(String usr, int money) {
48     int count = check_money(usr);
49     if(count < money) return false;
50     count -= money;
51     send_msg("1UPDATE fmbank SET money="+count+"
52             WHERE usr='"+usr+"'");
53     return true;
54 }
55
56 // 存款, 返回是否存款成功
57 public boolean store_money(String usr, int money) {
58     int count = check_money(usr);
59     count += money;
60     send_msg("1UPDATE fmbank SET money="+count+"
61             WHERE usr='"+usr+"'");
62     return true;
63 }
64
65 // 转账, 返回是否转账成功
66 public boolean transfer(String fromusr, String tousr,
67     int money) {
68     if(check_usr(tousr) == true) return false;
69     boolean res = withdraw_money(fromusr, money);
70     if(res == false) return res;
71     return store_money(tousr, money);
72 }
73
74 // 注册, 返回是否注册成功
75 public boolean signin(String usr_name, String birth,
76     String phone, String pass, int sex, int sch, int
77     money) {
78     if(check_usr(usr_name) == false) return false;
79     send_msg("1INSERT INTO fmbank (usr,pass,sch,
80             phone,sex,birth,money) VALUES ('"+usr_name+"
81             ','"+pass+"','"+sch+"','"+phone+"','"+sex+"','"+
```

```
        +birth+"'," +money+");");
75         return true;
76     }
77
78     // 批量注册，返回是否成功
79     public int bulk_signin(String file_name) {
80         int cnt = 0;
81         try {
82             Workbook book = Workbook.getWorkbook(
83                 new File(file_name));
84             Sheet sheet = book.getSheet(0);
85             int cols = sheet.getColumns();
86             int rows = sheet.getRows();
87             for(int i = 0; i < rows; i++) {
88                 String usr,birth,phone,pass,sex
89                     ,schn;
90                 usr = birth = phone = pass =
91                     sex = schn = null;
92                 for(int j = 0; j < cols; j++) {
93                     switch(j) {
94                         case 0: usr =
95                             sheet.
96                             getCell(j, i
97                             ).
98                             getContents
99                             (); break;
100                        case 1: birth =
101                            sheet.
102                            getCell(j, i
103                            ).
104                            getContents
105                            (); break;
106                        case 2: phone =
107                            sheet.
108                            getCell(j, i
109                            ).
110                            getContents
```



```
94         (); break;
          case 3: pass =
              sheet.
                  getCell(j, i
                      ).
                      getContents
                          (); break;
95         case 4: sex =
              sheet.
                  getCell(j, i
                      ).
                      getContents
                          (); break;
96         case 5: schn =
              sheet.
                  getCell(j, i
                      ).
                      getContents
                          (); break;
97     }
98 }
99 int sex_, schn_;
100 try {
101     sex_ = Integer.parseInt
        (sex);
102 } catch (NumberFormatException
        e) {sex_ = -1;}
103 try {
104     schn_ = Integer.
        parseInt(schn);
105 } catch (NumberFormatException
        e) {schn_ = -1;}
106 signin(usr,birth,phone,pass,
        sex_,schn_,2000);
107 cnt++;
108 }
109 } catch (IOException e) {
```

```
110         return -1;
111     } catch (BiffException e) {
112         return -1;
113     }
114     return cnt;
115 }
116
117 // 注销用户，返回是否注销成功
118 public boolean logout(String usr) {
119     send_msg("1DELETE FROM fmbank WHERE usr='"+usr+
120             "';");
121     return true;
122 }
123
124 // 修改信息，返回是否修改成功
125 public boolean update_info(String usr, String pass,
126     String birth, String phone) {
127     send_msg("1UPDATE fmbank SET pass='"+pass+"',
128             birth='"+birth+"',phone='"+phone+"' WHERE
129             usr='"+usr+"';");
130     return true;
131 }
132
133 // 年终报告
134 public boolean createpdf(String file_name) {
135     try {
136         Document document = new Document();
137         PdfWriter.getInstance(document, new
138             FileOutputStream(file_name));
139         int account = Integer.parseInt(send_msg(
140             "014numsSELECT COUNT(*) AS nums
141             FROM fmbank;"));
142         int count = Integer.parseInt(send_msg("
143             014numsSELECT SUM(money) AS nums
144             FROM fmbank;"));
145         document.open();
146         document.add(new Paragraph("Fei Ma Bank
```

```
        annual report"));
138     document.add(new Paragraph("The total
        number of the account is: "+account)
        );
139     document.add(new Paragraph("The number
        of the new account is: "+account));
140     document.add(new Paragraph("The total
        cash in the bank are: "+count));
141     document.close();
142     return true;
143 } catch (DocumentException e) {
144     return false;
145 } catch (IOException e) {
146     return false;
147 }
148 }
149 }
```

B 服务器端 Server 部分的代码

Listing 5: Main_Server.java

```
1 package Server;
2
3 import java.net.*;
4 import java.io.*;
5
6 public class Main_server {
7     public static void main(String[] args) {
8         ServerSocket socket = null;
9         Socket s = null;
10        try {
11            // 创建服务器Socket, 监听指定端口
12            socket = new ServerSocket(6606);
13        } catch (IOException e) {
14            e.printStackTrace();
15        }
16        System.out.println("等待连接....");
17        while (true) {
18            try {
19                // 等待客户端连接
20                s = socket.accept();
21                // 创建新的线程处理客户端请求
22                new Thread(new Server(s)).start();
23            } catch (IOException e) {
24                e.printStackTrace();
25            }
26        }
27    }
28 }
```

Listing 6: Server.java

```
1 package Server;
2
```

```
3 import java.net.*;
4 import java.io.*;
5
6 public class Server implements Runnable {
7     private Operator opr;
8     private DB data_base;
9     private Socket s;
10
11     Server(Socket s) {
12         this.s = s;
13         this.data_base = new DB();
14         this.opr = new Operator();
15     }
16
17     // 线程执行的主体方法
18     public void run() {
19         System.out.println("远程主机连接, 地址为: " +
20             this.s.getRemoteSocketAddress());
21         try {
22             DataInputStream in = new
23                 DataInputStream(this.s.
24                     getInputStream());
25             String rec = in.readUTF();
26             System.out.println("接收命令: " + rec);
27             DataOutputStream out = new
28                 DataOutputStream(this.s.
29                     getOutputStream());
30             if (opr.get_flag(rec).equals("1")) {
31                 // 执行数据库更新操作
32                 data_base.db_exe(opr.get_sql(
33                     rec));
34                 out.writeUTF("done");
35             } else {
36                 // 执行数据库查询操作
37                 out.writeUTF(data_base.db_query
38                     (opr.get_sql(rec), opr.
39                         get_col(rec), opr.get_f(rec))

```

```
        ));  
32        }  
33        } catch (IOException e) {  
34            // 处理异常  
35        }  
36    }  
37  
38    public static void main(String[] args) {  
39        // 主函数，用于启动服务器  
40    }  
41 }
```

Listing 7: DB.java

```
1 package Server;  
2  
3 import java.sql.*;  
4  
5 public class DB {  
6     static final String JDBC_DRIVER = "com.mysql.cj.jdbc.  
7         Driver";  
8     static final String DB_URL = "jdbc:mysql://localhost/  
9         testgb";  
10  
11     static final String USER = "root";  
12     static final String PASS = "2%duLg7dWiND";  
13  
14     private Connection conn;  
15     private Statement stmt;  
16  
17     // 构造函数，初始化数据库连接  
18     DB() {  
19         try {  
20             Class.forName(JDBC_DRIVER);  
21  
22             System.out.println("连接到数据库...");  
23             conn = DriverManager.getConnection(  
24                 DB_URL, USER, PASS);  
25         } catch (Exception e) {  
26             e.printStackTrace();  
27         }  
28     }  
29 }  
30  
31 }
```

```
22         stmt = conn.createStatement();
23     } catch (SQLException se) {
24         se.printStackTrace();
25     } catch (Exception e) {
26         e.printStackTrace();
27     }
28     System.out.println("连接成功。");
29 }
30
31 // 执行数据库查询并返回结果
32 public synchronized String db_query(String sql, String
33     col, int flag) {
34     String res = "Null";
35     try {
36         ResultSet rs = stmt.executeQuery(sql);
37         if (flag == 1)
38             while (rs.next())
39                 res = rs.getInt(col) + "";
40         else
41             while (rs.next())
42                 res = rs.getString(col);
43     } catch (SQLException se) {
44         se.printStackTrace();
45     } catch (Exception e) {
46         e.printStackTrace();
47     }
48     return res;
49 }
50
51 // 执行数据库操作
52 public synchronized void db_exe(String sql) {
53     try {
54         stmt.execute(sql);
55     } catch (SQLException se) {
56         se.printStackTrace();
57     } catch (Exception e) {
58         e.printStackTrace();
59     }
```

```
58         }
59     }
60 }
```

Listing 8: Operator.java

```
1 package Server;
2
3 public class Operator {
4     // 114abcd sql
5     // 格式: flag + number + length(column) + column + sql
6
7     public String get_flag(String s) {
8         // 获取命令标志位 (flag)
9         return s.charAt(0) + "";
10    }
11
12    public String get_col(String s) {
13        if ((s.charAt(0) + "").equals("0")) {
14            int st = Integer.parseInt(s.charAt(2) +
15                                     "");
16            // 获取列名 (column)
17            return s.substring(3, st + 3);
18        } else {
19            return "";
20        }
21    }
22
23    public int get_f(String s) {
24        if ((s.charAt(0) + "").equals("0")) {
25            // 获取查询标志位 (flag)
26            return Integer.parseInt(s.charAt(1) +
27                                   "");
28        } else {
29            return -1;
30        }
31    }
32}
```



```
31     public String get_sql(String s) {
32         if ((s.charAt(0) + "").equals("0")) {
33             int st = Integer.parseInt(s.charAt(2) +
34                                     "");
35             // 获取SQL语句
36             return s.substring(st + 3);
37         } else {
38             return s.substring(1);
39         }
40     }
```