 南昌大学实验报告

学生姓名： 田宇琛 学 号： 5601115010 专业班级： 计科153

实训类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期：2018.1.3 实验成绩：

1. 实验项目名称

2017~2018 -1 《java语言程序设计实验》期末大作业

1. 实验的评分标准

实验分为A～F，A为最高，F最低。

F：在规定时间内没有完成所有的实验，而且没有及时提交实验报告，或者实验过程中出现了抄袭复制他人实验代码。

D：能完成实验，但是实验结果出现严重错误，不能体现对教学内容的理解。

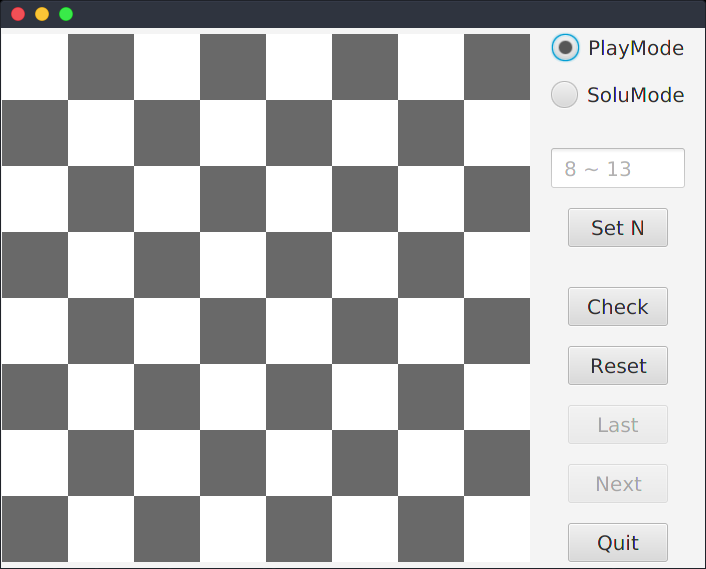
C：能基本完成实验，实验结果基本正确。但是实验内容有较少的错误，提交的实验代码质量一般。

B：能较好的完成实验，实验报告条理清楚，实验代码结构清晰，代码质量较高，及时更正试验中出现的错误，并对运行中一些异常错误进行分析，解释错误产生的原因。

A：能较好的完成实验，实验代码质量高，实验报告完成度高，能在实验完成的基础上，根据个人的理解增加实验的新功能，具有一定的创新能力。

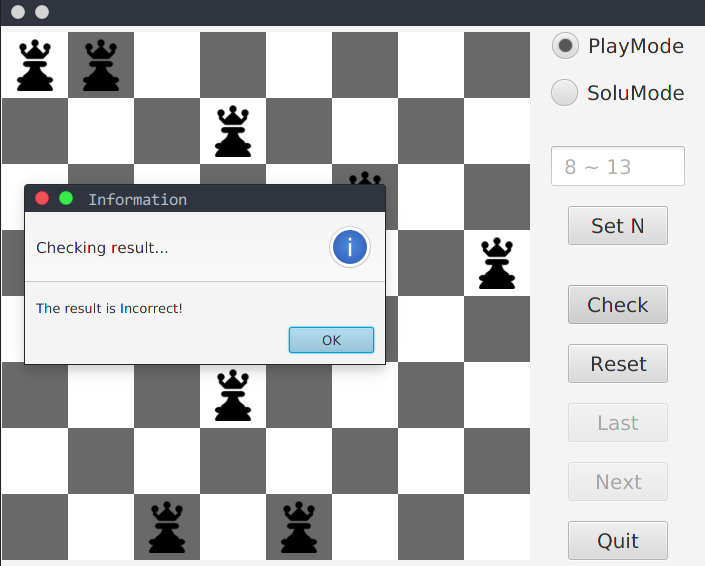
三、实验内容

**1.8皇后问题（支持手动设置，判断位置，生成解决方案，n皇后扩展，图形界面）**



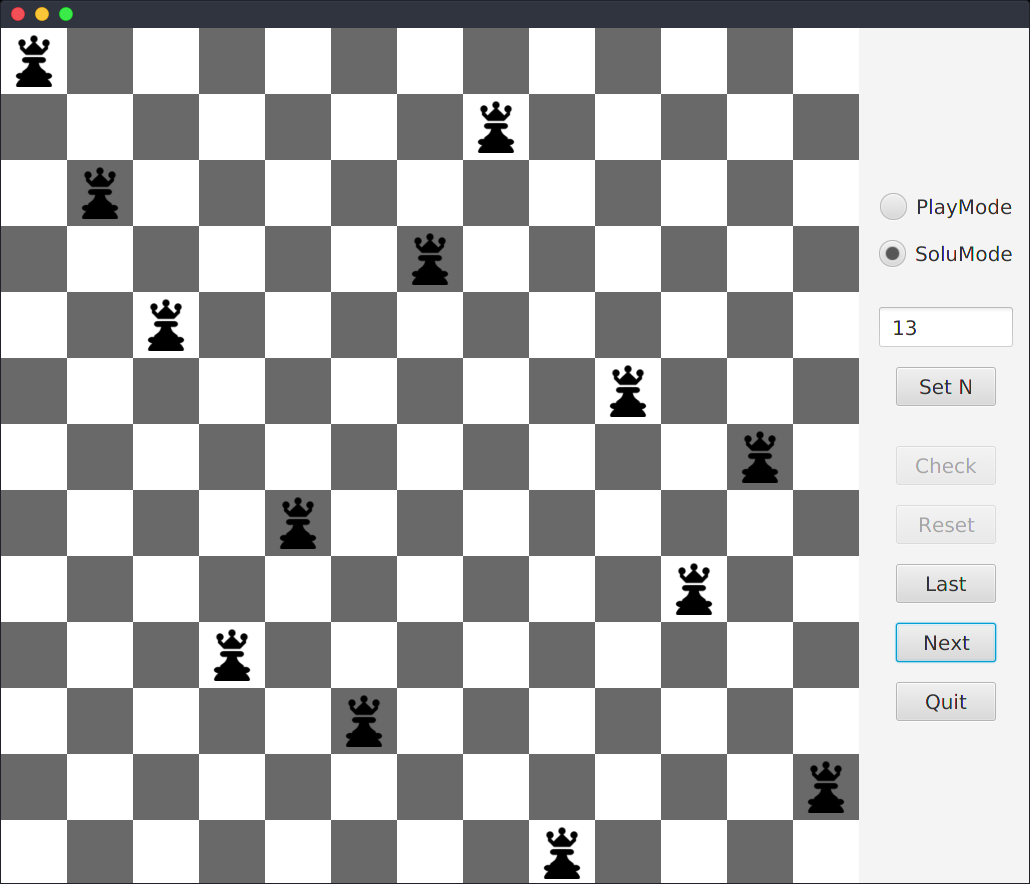
当用户没有进行皇后数量设定时，默认为8

当用户输入皇后数量太大时，将自动纠正为正确范围，因为本程序提供了n皇后所有的解法，对于太大的n，将影响计算速度。



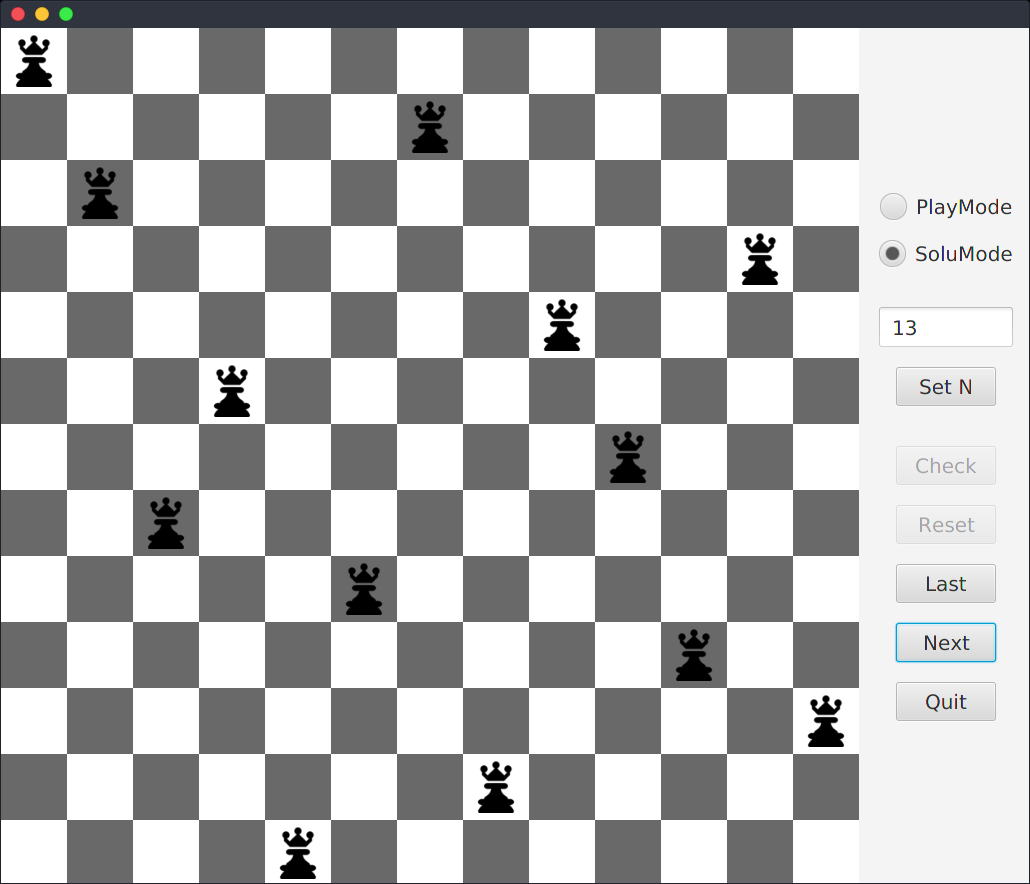
当在棋盘上点击，添加完N个皇后之后，棋盘会自动锁定，相应的事件被取消，单击Check按钮进行解法的检测，会有相应的弹出框提示解法的正确与否。

单击Reset，进行清盘，可以重新进行皇后的添加。

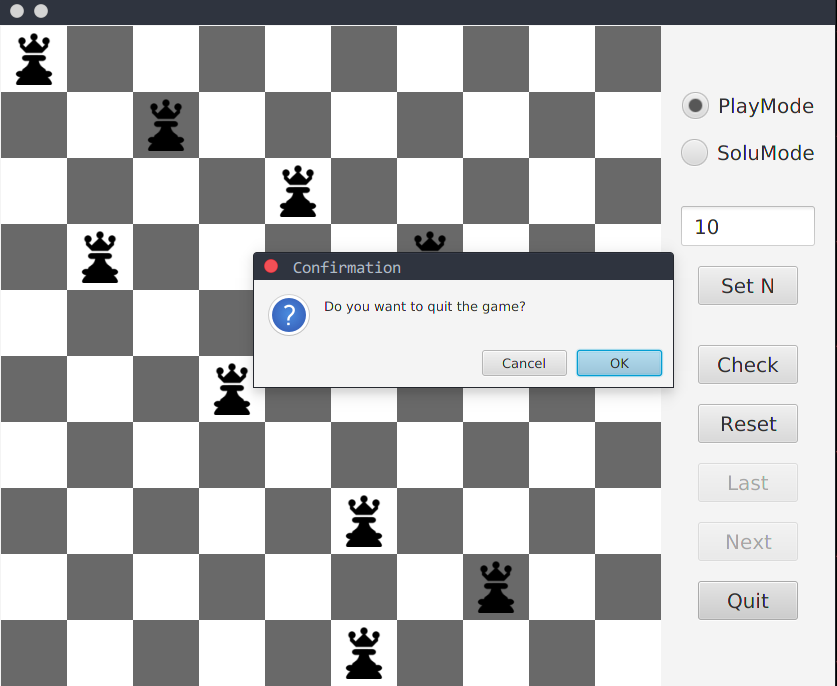


右上方有两种模式进行选择，PlayMode是玩家自己添加皇后，SoluMode是程序给出解决办法的模式。

单击Last和Next进行解法的查看。

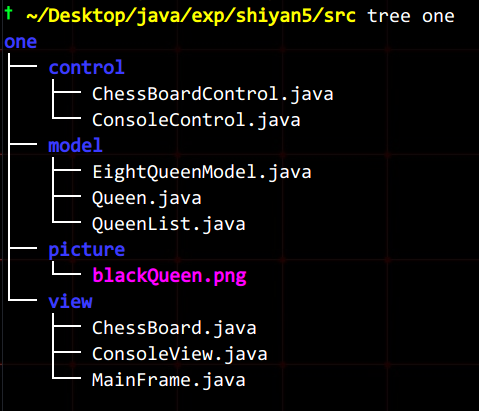


因为程序中存储了所有的解法，这里的底层棋盘采用boolean类型，解法采用了byte类型的数组，进行减少空间的占用



退出游戏的一个提醒。

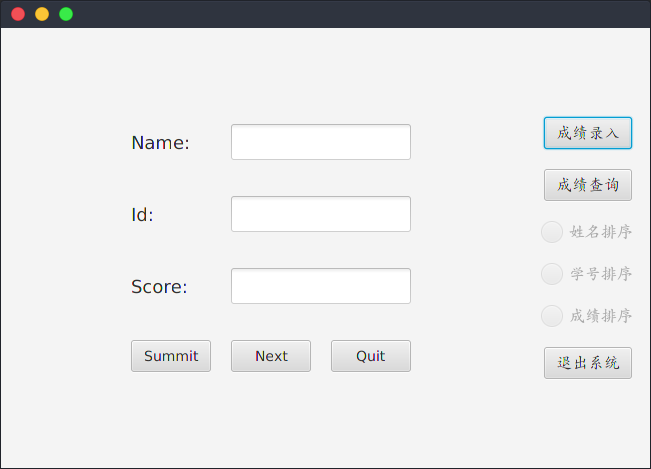
因为程序代码太多，这里给出它的一个源代码结构图（MVC模式）：



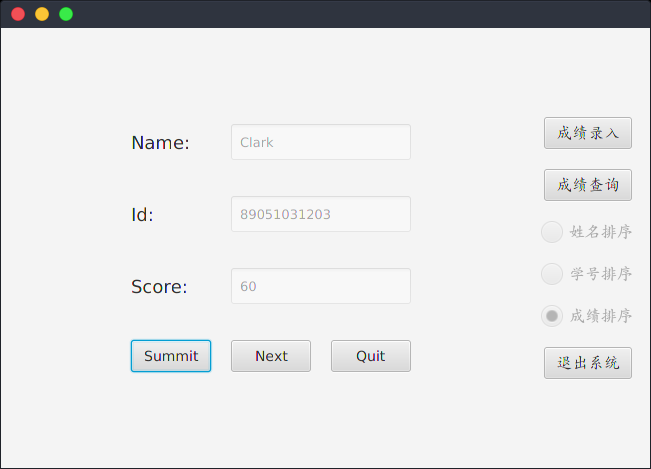
2.轻量级成绩管理系统（采用文件存储数据，支持成绩录入，成绩查询，三种排序下的成绩显示方式）

在右边选择不同的功能，主界面会跳转到相应的功能页面。

只有当查询功能被选中时，下面提供的三种排序选择方式才会显示可用。

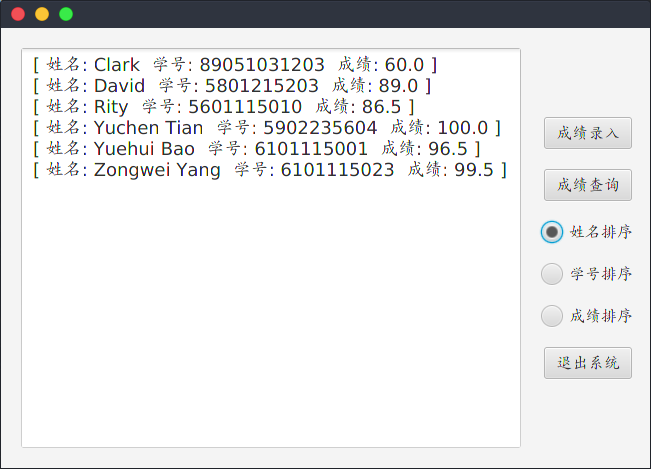


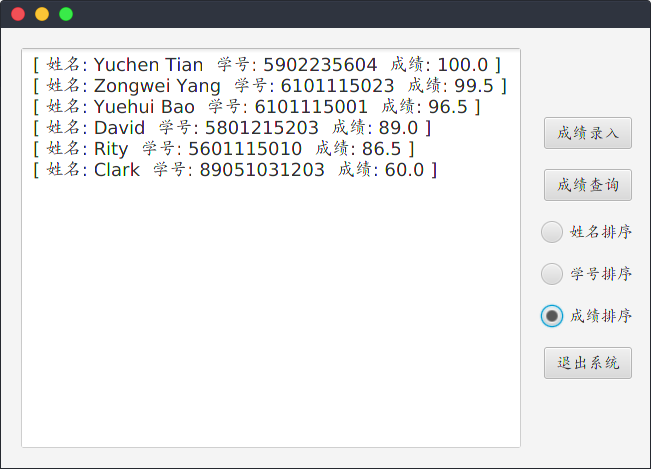
简单的录入界面，将信息输入后，单击Submit，后台将记录存入文件。



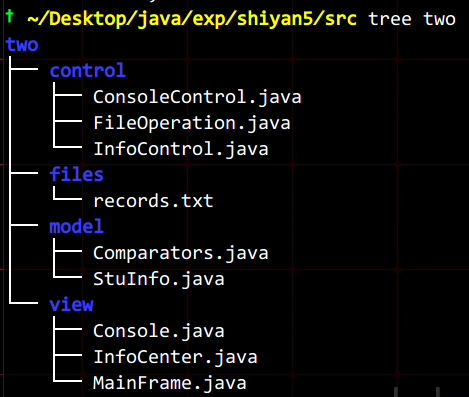
单击Next进行下一条记录的输入。输入完成后，单击Quit，返回主界面

单击成绩查询，刚才被锁定的三个排序选项可以使用了，点击不同的排序方式，中心文本域将出现不同排序方式的内容。





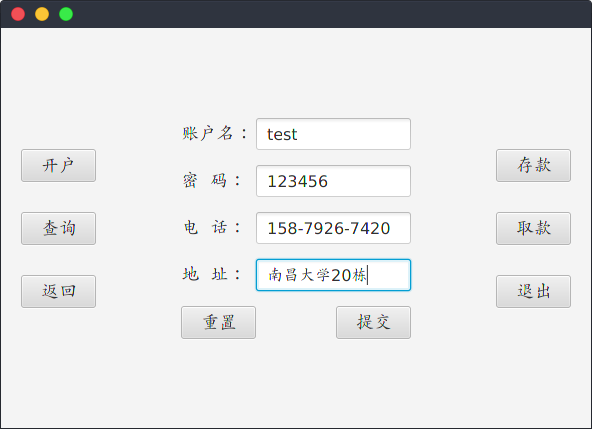
这里给出程序的源代码结构图（MVC模式）：



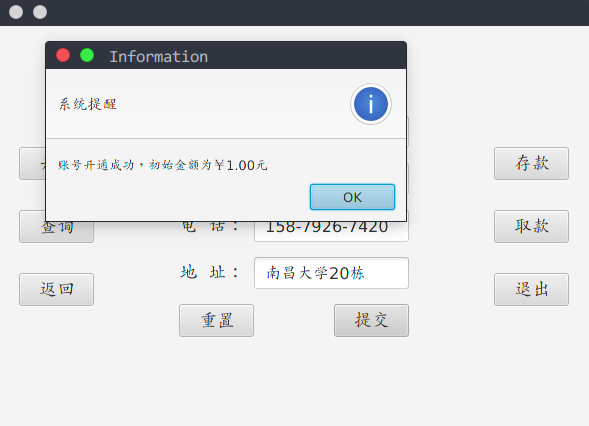
**3.储蓄管理系统（面向用户！支持开户，存取款功能，查询记录）**



主界面提供不同的功能，同样的，单击不同的功能，主界面会进行跳转，然后到相应的界面完成功能即可。



**开户**：正确填写信息后，单击提交。

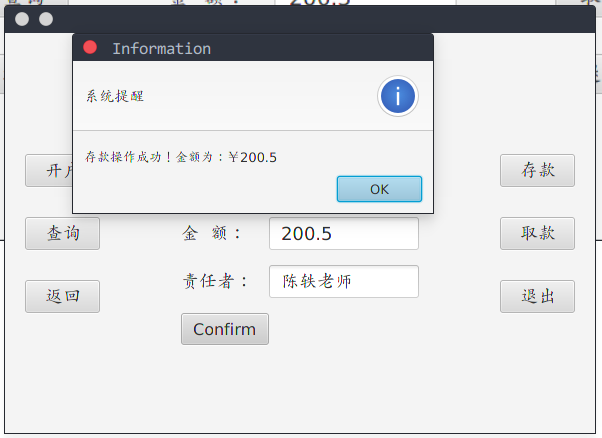


开户成功，**客户文件**中会增加一条记录，同时有弹窗提醒。

这里给账户中设置一个默认金额1元。

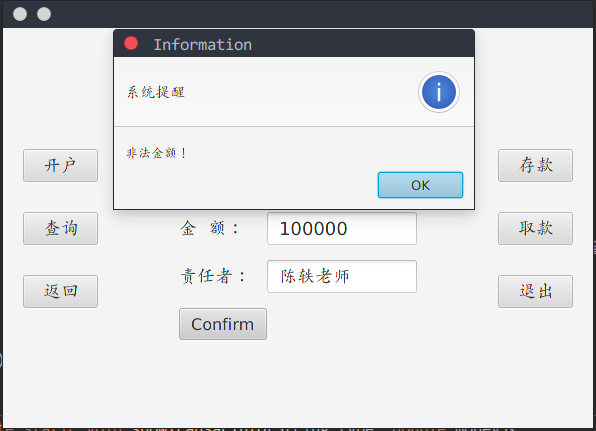


存款界面：正确填写信息，经办人签名，然后单击Comfirm，进行存款业务的提交。

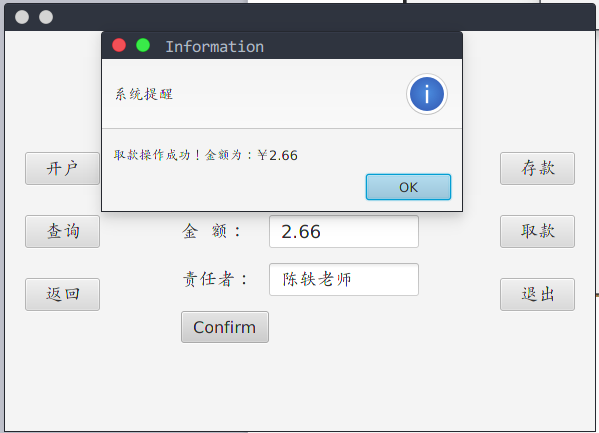


存款成功后弹窗消息提醒。

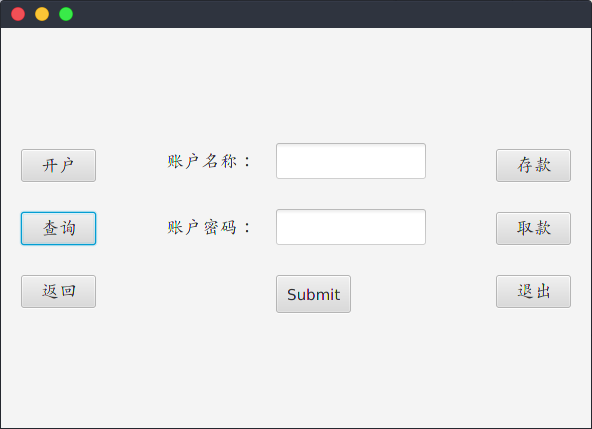




取款操作时，会有一个检测的步骤，当取款金额大于了账户拥有的金额，会提示金额非法，不给予取款。



取款成功后也会有相应提醒。同时在**交易文件**中会写入一条记录。

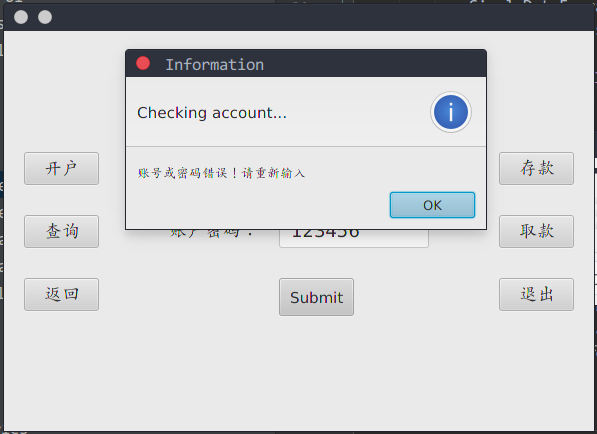


**进行账户查询时，首先要求用户登录**

因为系统面向用户，所以合法用户才能查看**自己的**交易记录和账户信息。

**没有提供查看所有账户信息的接口，但是所有记录是存放在一起的！**





如果用户不存在，或者密码输入错误，会禁止进入查询页面，同时弹窗提醒。



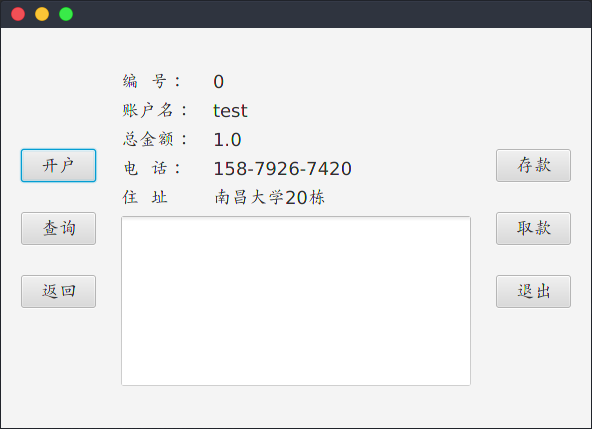


当身份信息验证合格后，跳转到信息界面，展示了该用户的所有信息。

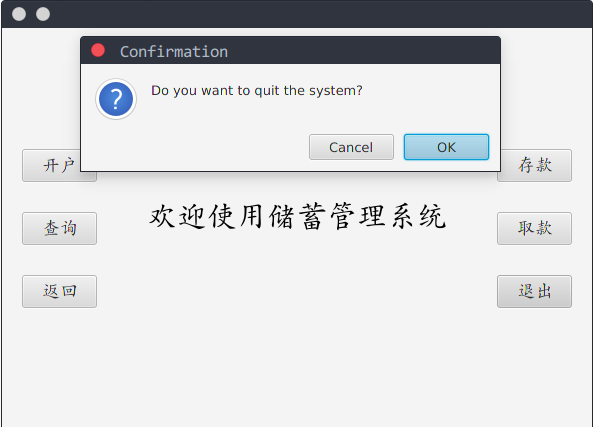
其中：上半部分信息来自**客户文件**。

下半部分信息来自**交易记录文件内容的筛选**。

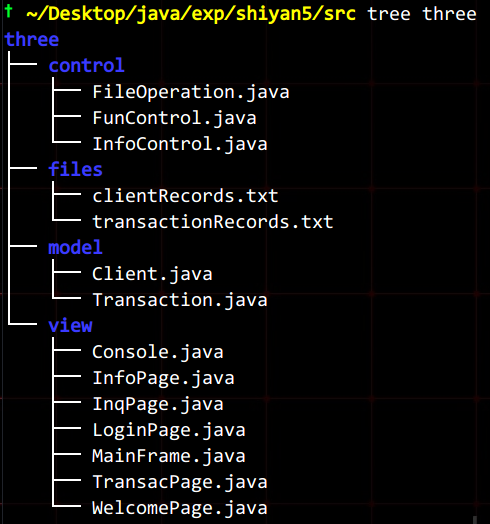




这里是编号为0 的一个测试账户，没有进行任何操作，所以记录为空。



这里给出源代码的结构图（MVC模式）：



1. 实验小结

略

1. 实验源代码
2. 八皇后问题

package one.model;

import java.util.ArrayList;

public class **EightQueenModel** {

public static int N = 8;

private boolean[][] arr;

private ArrayList<byte[]> solutionList;

private int pointer;

private static EightQueenModel eightQueenModel = null;

public static EightQueenModel getInstance(){

if(eightQueenModel == null)

eightQueenModel = new EightQueenModel();

return eightQueenModel;

}

public static void setN(int n){

N = n;

}

private EightQueenModel(){

arr = new boolean[N][N];

solutionList = new ArrayList<>();

pointer = 0;

}

public ArrayList<byte[]> getSolutionList() {

return solutionList;

}

public int getPointer(){

return pointer;

}

public void addPointer(){

pointer = (pointer + 1) % solutionList.size();

}

public void subPointer(){

pointer = (pointer - 1 + solutionList.size()) % solutionList.size();

}

public boolean addQueen(int i, int j){

if(i<N && j<N && !arr[i][j]){

arr[i][j] = true;

return true;

}

else

return false;

}

public boolean isValid(int k,int j){

for(int i =0;i<N;++i){

if(arr[i][j]) //conflict with row

return false;

}

for(int i=k-1,m=j-1;i>=0&&m>=0;i--,m--){

if(arr[i][m]) //left-up

return false;

}

for(int i=k-1,m=j+1;i>=0&&m<=N-1;i--,m++){

if(arr[i][m]) //right-up

return false;

}

return true;

}

public void findSolution(int i){

if(i==N){

byte[] solution = new byte[N];

for(byte q = 0; q<N; ++q){

for(byte t = 0; t<N; ++t){

if(arr[q][t]){

solution[q] = t;

break;

}

}

}

solutionList.add(solution);

return;

}

for(int m=0;m<N;m++){ //recursion

if(isValid(i,m)){

arr[i][m]=true;

findSolution(i+1);

arr[i][m]=false; //value of last level

}

}

}

public void clear(){

arr = new boolean[N][N];

pointer = 0;

}

}

package one.model;

import javafx.scene.image.Image;

public class **Queen** {

private int x;

private int y;

private Image image;

public Queen(int x,int y){

this.x = x;

this.y = y;

this.image = new Image("one/picture/blackQueen.png");

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public Image getImage() {

return image;

}

}

package one.model;

import java.util.ArrayList;

public class **QueenList** {

private ArrayList<Queen> list;

private static QueenList queenList = null;

public static QueenList getInstance(){

if(queenList == null)

queenList = new QueenList();

return queenList;

}

private QueenList(){

list = new ArrayList<>();

}

public void add(Queen q){

list.add(q);

}

public void clear(){

list.clear();

}

public int size(){

return list.size();

}

public Queen get(int i){

return list.get(i);

}

}

package one.view;

import javafx.scene.canvas.Canvas;

import javafx.scene.canvas.GraphicsContext;

import javafx.scene.paint.Paint;

import one.model.EightQueenModel;

public class **ChessBoard** {

public static final int SIZE = 66;

public static final int PSIZE = 52;

private Canvas canvas;

private GraphicsContext gc;

private static ChessBoard chessBoard = null;

public static ChessBoard getInstance(){

if(chessBoard == null)

chessBoard = new ChessBoard();

return chessBoard;

}

private ChessBoard(){

canvas = new Canvas(SIZE\* EightQueenModel.N, SIZE\* EightQueenModel.N);

gc = canvas.getGraphicsContext2D();

drawBoard();

}

public void updateChessBoard(){

gc.clearRect(0,0,SIZE\* EightQueenModel.N,SIZE\* EightQueenModel.N);

canvas.setHeight(SIZE\* EightQueenModel.N);

canvas.setWidth(SIZE\* EightQueenModel.N);

}

public GraphicsContext getGc() {

return gc;

}

public Canvas getCanvas() {

return canvas;

}

public void drawBoard(){

for(int i = 0;i<EightQueenModel.N;i++){

for(int j = 0;j<EightQueenModel.N;j++){

String color = (i+j)%2 == 0 ? "white" : "dimgray";

gc.setFill(Paint.valueOf(color));

gc.fillRect(i\*SIZE,j\*SIZE,SIZE,SIZE);

}

}

}

}

package one.view;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.layout.VBox;

import javafx.scene.text.Font;

public class **ConsoleView** {

private TextField textField;

private Button set;

private RadioButton[] modeSelection;

private ToggleGroup tgroup;

private Button[] function;

private VBox console;

private Font font;

private static ConsoleView consoleView = null;

public static ConsoleView getInstance(){

if(consoleView == null)

consoleView = new ConsoleView();

return consoleView;

}

private ConsoleView(){

font = Font.font(20);

tgroup = new ToggleGroup();

textField = new TextField();

textField.setPrefSize(120, 40);

textField.setFont(font);

set = new Button("Set N");

set.setPrefSize(100,28);

set.setFont(font);

modeSelection = new RadioButton[2];

for(int i = 0; i < 2; ++i){

modeSelection[i] = new RadioButton();

modeSelection[i].setFont(font);

modeSelection[i].setToggleGroup(tgroup);

}

function = new Button[5];

for(int i = 0; i < 5; i++){

function[i] = new Button();

function[i].setFont(font);

function[i].setPrefSize(100,28);

}

function[3].setDisable(true);

function[2].setDisable(true);

console = new VBox(20);

setModule();

loadModule();

}

public Button getSet(){

return set;

}

public TextField getTextField() {

return textField;

}

public RadioButton getMode(int i){

return modeSelection[i];

}

public Button getFunction(int i) {

return function[i];

}

public VBox getConsole() {

return console;

}

private void setModule(){

textField.setPromptText("8 ~ 13");

modeSelection[0].setText("PlayMode");

modeSelection[0].setSelected(true);

modeSelection[1].setText("SoluMode");

modeSelection[1].setSelected(false);

function[0].setText("Check");

function[1].setText("Reset");

function[2].setText("Last");

function[3].setText("Next");

function[4].setText("Quit");

}

private void loadModule(){

console.getChildren().addAll(modeSelection);

console.getChildren().addAll(textField, set);

console.getChildren().addAll(function);

VBox.setMargin(textField, new Insets(20,0,0,0));

VBox.setMargin(function[0],new Insets(20,0,0,0));

console.setAlignment(Pos.CENTER);

}

}

package one.view;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

import one.control.ChessBoardControl;

import one.control.ConsoleControl;

public class **MainFrame** extends Application {

public void start(Stage stage){

BorderPane borderPane = new BorderPane();

//get components

ChessBoard board = ChessBoard.getInstance();

ConsoleView consoleView = ConsoleView.getInstance();

//put components into right place

borderPane.setCenter(board.getCanvas());

borderPane.setRight(consoleView.getConsole());

//set margin

BorderPane.setMargin(consoleView.getConsole(), new Insets(0,20,0,20));

//add all the events

ChessBoardControl.addChessBoardEvents();

ConsoleControl.addConsoleEvents();

//run

Scene scene = new Scene(borderPane);

stage.setScene(scene);

stage.show();

}

}

package one.control;

import javafx.scene.image.Image;

import one.model.EightQueenModel;

import one.model.Queen;

import one.model.QueenList;

import one.view.ChessBoard;

public class **ChessBoardControl** {

private static ChessBoard chessBoard = ChessBoard.getInstance();

private static EightQueenModel eightQueenModel = EightQueenModel.getInstance();

private static QueenList qlist = QueenList.getInstance();

private static int numOfQueen = 0;

private ChessBoardControl(){}

public static void addChessBoardEvents(){

addQueenEvent();

}

public static void addQueenEvent(){

chessBoard.getCanvas().setOnMouseClicked(event -> {

if(numOfQueen < EightQueenModel.N){

double tempX = event.getX();

double tempY = event.getY();

int x = (int)(tempX/ ChessBoard.SIZE);

int y = (int)(tempY/ ChessBoard.SIZE);

//position is valid

if(eightQueenModel.addQueen(x,y)){

Queen queen = new Queen(x,y);

qlist.add(queen);

drawQueen(queen);

numOfQueen++;//make sure only 8 queens are available

}

}

});

}

public static void cancelAddQueenEvent(){

chessBoard.getCanvas().setOnMouseClicked(event -> {});

}

public static void clearBoard(){

chessBoard.updateChessBoard();

chessBoard.drawBoard();

numOfQueen = 0;

}

private static void drawQueen(Queen queen){

double startX = queen.getX()\* ChessBoard.SIZE + (ChessBoard.SIZE - ChessBoard.PSIZE) / 2;

double startY = queen.getY()\* ChessBoard.SIZE + (ChessBoard.SIZE - ChessBoard.PSIZE) / 2;

chessBoard.getGc().drawImage(queen.getImage(),startX,startY, ChessBoard.PSIZE, ChessBoard.PSIZE);

}

public static void drawSolution(){

clearBoard();

byte[] pos = eightQueenModel.getSolutionList().get(eightQueenModel.getPointer());

for(int i = 0; i<EightQueenModel.N; ++i){

double startX = i\* ChessBoard.SIZE + (ChessBoard.SIZE - ChessBoard.PSIZE) / 2;

double startY = pos[i]\* ChessBoard.SIZE + (ChessBoard.SIZE - ChessBoard.PSIZE) / 2;

chessBoard.getGc().drawImage(new Image("one/picture/blackQueen.png"), startX, startY, ChessBoard.PSIZE, ChessBoard.PSIZE );

}

}

}

package one.control;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import one.model.EightQueenModel;

import one.model.QueenList;

import one.view.ConsoleView;

import java.util.Optional;

public class **ConsoleControl** {

private static QueenList qlist = QueenList.getInstance();

private static EightQueenModel eightQueenModel = EightQueenModel.getInstance();

private static ConsoleView consoleView = ConsoleView.getInstance();

public static void addConsoleEvents(){

modeSelectionEvent();

setScaleEvent();

checkEvent();

resetEvent();

stepEvent();

quitEvent();

}

private static void modeSelectionEvent(){

consoleView.getMode(0).setOnAction(event -> {

ChessBoardControl.addChessBoardEvents();

clearAll();

consoleView.getFunction(0).setDisable(false);

consoleView.getFunction(1).setDisable(false);

consoleView.getFunction(2).setDisable(true);

consoleView.getFunction(3).setDisable(true);

});

consoleView.getMode(1).setOnAction(event -> {

generateSolution();

ChessBoardControl.cancelAddQueenEvent();

clearAll();

consoleView.getFunction(2).setDisable(false);

consoleView.getFunction(3).setDisable(false);

consoleView.getFunction(0).setDisable(true);

consoleView.getFunction(1).setDisable(true);

});

}

private static void setScaleEvent(){

consoleView.getSet().setOnAction(event -> {

getScale();//change the scale of the game

clearAll();

});

}

private static void resetEvent(){

consoleView.getFunction(1).setOnAction(event -> {

clearAll();

});

}

private static void checkEvent(){

consoleView.getFunction(0).setOnAction(event -> {

boolean valid = false;

if(qlist.size() == 8) {

for (int i = 0; i < qlist.size(); ++i) {

if (eightQueenModel.isValid(qlist.get(i).getX(), qlist.get(i).getY()))

valid = true;

}

}

showInfo(valid);

});

}

private static void quitEvent(){

Alert alert = new Alert(Alert.AlertType.CONFIRMATION);

alert.setTitle("Confirmation");

alert.setHeaderText("");

alert.setContentText("Do you want to quit the game?");

consoleView.getFunction(4).setOnAction(event -> {

Optional<ButtonType> result = alert.showAndWait();

if(result.isPresent() && result.get() == ButtonType.OK){

System.exit(0);

}

});

}

private static void clearAll(){

ChessBoardControl.clearBoard();//draw a new chess board

qlist.clear();

eightQueenModel.clear();

}

private static void showInfo(boolean valid){

String result = "The result is " + (valid ? "Correct!" : "Incorrect!");

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("Information");

a.setHeaderText("Checking result...");

a.setContentText(result);

a.showAndWait();

}

private static void generateSolution(){

eightQueenModel.findSolution(0);

}

private static void stepEvent(){

consoleView.getFunction(2).setOnAction(event -> {

eightQueenModel.subPointer();

ChessBoardControl.drawSolution();

});

consoleView.getFunction(3).setOnAction(event -> {

eightQueenModel.addPointer();

ChessBoardControl.drawSolution();

});

}

private static void getScale(){

int scale = 8;

String s = consoleView.getTextField().getText();

if(!s.equals("") && Integer.parseInt(s) > 7 && Integer.parseInt(s) < 14)

scale = Integer.parseInt(s);

EightQueenModel.setN(scale);

}

}

==============================================================================================

2.成绩管理系统

package two.model;

import java.io.Serializable;

public class **StuInfo** implements Serializable{

private String name;

private String id;

private double score;

public StuInfo(String name, String id){

this.name = name;

this.id = id;

}

public StuInfo(String name, String id, Double score){

this(name, id);

this.score = score;

}

public String getName() {

return name;

}

public String getId() {

return id;

}

public Double getScore() {

return score;

}

@Override

public String toString() {

return "[ 姓名: " + this.name + " 学号: " + this.id + " 成绩: " + this.score + " ]\n";

}

}

package two.model;

import java.util.Comparator;

public class **Comparators** {

public static class nameComparator implements Comparator<StuInfo> {

@Override

public int compare(StuInfo s1, StuInfo s2){

return s1.getName().compareTo(s2.getName());

}

};

public static class idComparator implements Comparator<StuInfo> {

@Override

public int compare(StuInfo s1, StuInfo s2) {

return s1.getId().compareTo(s2.getId());

}

};

public static class scoreComparator implements Comparator<StuInfo> {

@Override

public int compare(StuInfo s1, StuInfo s2) {

if(s1.getScore() == s2.getScore())

return 0;

else

return (s1.getScore() < s2.getScore()) ? 1 : -1;

}

};

}

package two.view;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.control.RadioButton;

import javafx.scene.control.ToggleGroup;

import javafx.scene.layout.VBox;

import javafx.scene.text.Font;

public class **Console** {

private VBox vBox;

private Button[] funcs;

private RadioButton[] selecs;

private ToggleGroup toggleGroup;

private Font font;

private static Console console = null;

public static Console getConsole(){

if(console == null)

console = new Console();

return console;

}

private Console(){

vBox = new VBox(20);

funcs = new Button[3];

selecs = new RadioButton[3];

toggleGroup = new ToggleGroup();

font = Font.font(16);

setModule();

loadModule();

}

public VBox getvBox() {

return vBox;

}

public Button getFunc(int i){

return funcs[i];

}

public RadioButton getSelecs(int i){

return selecs[i];

}

private void setModule(){

for(int i = 0; i<3; ++i){

funcs[i] = new Button();

funcs[i].setPrefSize(88,32);

funcs[i].setFont(font);

selecs[i] = new RadioButton();

selecs[i].setToggleGroup(toggleGroup);

selecs[i].setFont(font);

selecs[i].setDisable(true);

}

funcs[0].setText("成绩录入");

funcs[1].setText("成绩查询");

funcs[2].setText("退出系统");

selecs[0].setText("姓名排序");

selecs[1].setText("学号排序");

selecs[2].setText("成绩排序");

}

private void loadModule(){

vBox.getChildren().add(funcs[0]);

vBox.getChildren().add(funcs[1]);

vBox.getChildren().addAll(selecs);

vBox.getChildren().add(funcs[2]);

vBox.setAlignment(Pos.CENTER);;

}

}

package two.view;

import javafx.geometry.Pos;

import javafx.scene.canvas.Canvas;

import javafx.scene.canvas.GraphicsContext;

import javafx.scene.control.\*;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

public class **InfoCenter** {

//scene 1

private Canvas canvas;

private GraphicsContext gc;

//scene 2

private GridPane gridPane;

private Label[] labels;

private TextField[] textFields;

private Button[] buttons;

//scene 3

private TextArea textArea;

private static InfoCenter infoCenter= null;

public static InfoCenter getInfoCenter(){

if(infoCenter == null)

infoCenter = new InfoCenter();

return infoCenter;

}

private InfoCenter() {

canvas = new Canvas();

gc = canvas.getGraphicsContext2D();

gridPane = new GridPane();

labels = new Label[3];

textFields = new TextField[3];

buttons = new Button[3];

textArea = new TextArea();

setModule();

loadModule();

}

public Canvas getCanvas() {

return canvas;

}

public TextArea getTextArea() {

return textArea;

}

public GridPane getGridPane() {

return gridPane;

}

public Button getButton(int i){

return buttons[i];

}

public TextField getTextField(int i){

return textFields[i];

}

private void setModule(){

for(int i = 0; i<3; ++i){

labels[i] = new Label();

labels[i].setPrefSize(80,36);

labels[i].setFont(Font.font(18));

}

labels[0].setText("Name:");

labels[1].setText("Id:");

labels[2].setText("Score:");

for(int i = 0; i<3; ++i){

textFields[i] = new TextField();

textFields[i].setPrefSize(150,36);

}

for(int i = 0; i<3; ++i){

buttons[i] = new Button();

buttons[i].setPrefSize(80,32);

buttons[i].setFont(Font.font(14));

}

buttons[0].setText("Summit");

buttons[1].setText("Next");

buttons[2].setText("Quit");

canvas.setWidth(500);

canvas.setHeight(400);

gridPane.setPrefSize(500,400);

textArea.setPrefSize(500,400);

textArea.setFont(Font.font(18));

}

private void loadModule(){

gc.setFont(Font.font(50));

gc.fillText("Java成绩管理系统",60,190);

gc.setFont(Font.font(20));

gc.fillText("V-1.0",410,220);

gridPane.setVgap(36);

gridPane.setHgap(20);

gridPane.setAlignment(Pos.CENTER);

for(int i = 0; i<3; i++){

gridPane.add(labels[i],0,i);

gridPane.add(textFields[i],1,i,2,1);

gridPane.add(buttons[i],i,3);

}

}

}

package two.view;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

import two.control.ConsoleControl;

import two.control.InfoControl;

public class **MainFrame** extends Application{

private static BorderPane borderPane = new BorderPane();

public static BorderPane getBorderPane(){

return borderPane;

}

public void start(Stage stage){

InfoCenter infoCenter = InfoCenter.getInfoCenter();

Console console = Console.getConsole();

borderPane.setCenter(infoCenter.getCanvas());

borderPane.setRight(console.getvBox());

BorderPane.setMargin(infoCenter.getCanvas(), new Insets(20,20,20,20));

BorderPane.setMargin(console.getvBox(), new Insets(0,16,0,0));

ConsoleControl.addFunctionEvents();

InfoControl.addInfoEvents();

Scene scene = new Scene(borderPane);

stage.setScene(scene);

stage.show();

}

}

package two.control;

import two.model.StuInfo;

import two.view.InfoCenter;

public class **InfoControl** {

private static InfoCenter infoCenter = InfoCenter.getInfoCenter();

public static void addInfoEvents(){

summit();

next();

quit();

}

private static void addRecords(){

String name = infoCenter.getTextField(0).getText();

String id = infoCenter.getTextField(1).getText();

String score = infoCenter.getTextField(2).getText();

StuInfo stuInfo;

if(score.equals(""))

stuInfo = new StuInfo(name, id);

else

stuInfo = new StuInfo(name, id, Double.parseDouble(score));

FileOperation.AddRecords(stuInfo);

}

private static void summit(){

infoCenter.getButton(0).setOnAction(event -> {

addRecords();

for(int i = 0; i<3; ++i){

infoCenter.getTextField(i).setDisable(true);

}

});

}

private static void next(){

infoCenter.getButton(1).setOnAction(event -> {

for(int i = 0; i<3; ++i){

infoCenter.getTextField(i).setDisable(false);

infoCenter.getTextField(i).clear();

}

});

}

private static void quit(){

infoCenter.getButton(2).setOnAction(event -> {

ConsoleControl.changeSceneToCanvas();

});

}

}

package two.control;

import two.model.StuInfo;

import java.io.\*;

import java.util.ArrayList;

public class **FileOperation** {

private static File file = new File("src/two/files/records.txt");

public static void AddRecords(StuInfo stuInfo){

if(!file.exists()){

try {

file.createNewFile();

} catch (IOException e) {

System.out.println("Fail to create new files");

e.printStackTrace();

}

}

try(FileOutputStream fos = new FileOutputStream(file,true);

ObjectOutputStream oos = new ObjectOutputStream(fos)

){

oos.writeObject(stuInfo);

}catch (FileNotFoundException e){

e.printStackTrace();

}catch (IOException e){

e.printStackTrace();

}

}

public static ArrayList<StuInfo> getRecords(){

if(!file.exists()){

System.out.println("No records yet");

return null;

}

else{

ArrayList<StuInfo> list = new ArrayList<>();

StuInfo temp = null;

try(FileInputStream fis = new FileInputStream(file)

){

ObjectInputStream ois = null;

while (fis.available() > 0){

try {

ois = new ObjectInputStream(fis);

temp = (StuInfo) ois.readObject();

list.add(temp);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}catch (IOException e){

e.printStackTrace();

}

return list;

}

}

}

package two.control;

import javafx.geometry.Insets;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import javafx.scene.layout.BorderPane;

import two.model.Comparators;

import two.model.StuInfo;

import two.view.Console;

import two.view.InfoCenter;

import two.view.MainFrame;

import java.util.ArrayList;

import java.util.Optional;

public class **ConsoleControl** {

private static InfoCenter infoCenter = InfoCenter.getInfoCenter();

private static Console console = Console.getConsole();

public static void addFunctionEvents(){

addRecord();

inquireRc();

quit();

selecEvents();

}

public static void changeSceneToCanvas(){

MainFrame.getBorderPane().setCenter(infoCenter.getCanvas());

BorderPane.setMargin(infoCenter.getCanvas(), new Insets(20,20,20,20));

}

private static void changeSceneToGrid(){

MainFrame.getBorderPane().setCenter(infoCenter.getGridPane());

BorderPane.setMargin(infoCenter.getGridPane(), new Insets(20,20,20,20));

}

private static void changeSceneToText(){

MainFrame.getBorderPane().setCenter(infoCenter.getTextArea());

BorderPane.setMargin(infoCenter.getTextArea(), new Insets(20,20,20,20));

}

private static void addRecord(){

console.getFunc(0).setOnAction(event -> {

setSelecsDisable(true);

changeSceneToGrid();

});

}

private static void inquireRc(){

console.getFunc(1).setOnAction(event -> {

setSelecsDisable(false);

changeSceneToText();

});

}

private static void quit(){

Alert alert = new Alert(Alert.AlertType.CONFIRMATION);

alert.setTitle("Confirmation");

alert.setHeaderText("");

alert.setContentText("Do you want to quit the system?");

console.getFunc(2).setOnAction(event -> {

Optional<ButtonType> result = alert.showAndWait();

if(result.isPresent() && result.get() == ButtonType.OK){

System.exit(0);

}

});

}

private static void setSelecsDisable(boolean choice){

for(int i = 0; i<3; ++i){

console.getSelecs(i).setDisable(choice);

}

}

private static void selecEvents(){

console.getSelecs(0).setOnAction(event -> {

ArrayList<StuInfo> list = FileOperation.getRecords();

list.sort(new Comparators.nameComparator());

showInfo(list);

});

console.getSelecs(1).setOnAction(event -> {

ArrayList<StuInfo> list = FileOperation.getRecords();

list.sort(new Comparators.idComparator());

showInfo(list);

});

console.getSelecs(2).setOnAction(event -> {

ArrayList<StuInfo> list = FileOperation.getRecords();

list.sort(new Comparators.scoreComparator());

showInfo(list);

});

}

private static void showInfo(ArrayList<StuInfo> list){

infoCenter.getTextArea().clear();

for(StuInfo stuInfo : list){

infoCenter.getTextArea().appendText(stuInfo.toString());

}

}

}

====================================================================================================

1. 储蓄管理系统

package three.model;

import java.io.Serializable;

public class **Client** implements Serializable{

private static int num = 0;

private int id;

private String name;

private String passWord;

private String address;

private String tel;

private double money;

public Client(String name, String passWord, String address, String tel){

this.name = name;

this.passWord = passWord;

this.address = address;

this.tel = tel;

this.money = 1;

this.id = num++;

}

@Override

public String toString(){

return String.format("编号：%d\n姓名：%s\n密码：%s\n地址：%s\n电话：%s\n总金额：￥%.2f\n",

id, name, passWord, address, tel, money);

}

public String getName() {

return name;

}

public String getPassWord() {

return passWord;

}

public int getId() {

return id;

}

public String getAddress() {

return address;

}

public String getTel() {

return tel;

}

public double getMoney() {

return money;

}

public void deposit(double money){

this.money += money;

}

public boolean withDraw(double money){

boolean withDrawRes;

if(this.money < money)

withDrawRes = false;

else{

this.money -= money;

withDrawRes = true;

}

return withDrawRes;

}

}

package three.model;

import java.io.Serializable;

public class **Transaction** implements Serializable{

private static int num;

private int id;

private String account;

private String date;

private String type;

private double money;

private String charger;

public Transaction(String account, String date, String type, double money, String charger) {

this.account = account;

this.date = date;

this.type = type;

this.money = money;

this.charger = charger;

this.id = num++;

}

public String getAccount(){

return account;

}

public String toString(){

return String.format("编号：%d\n账户：%s\n金额：%.2f\n类型：%s\n经办人：%s\n日期：%s\n"

, id, account, money, type, charger, date);

}

}

package three.view;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.layout.VBox;

import javafx.scene.text.Font;

public class **Console** {

private VBox vBox;

private Button[] funcs;

private static Console consoleL = null;

private static Console consoleR = null;

public static Console getConsoleL(){

if(consoleL == null){

consoleL = new Console();

}

consoleL.getButtons(0).setText("开户");

consoleL.getButtons(1).setText("查询");

consoleL.getButtons(2).setText("返回");

return consoleL;

}

public static Console getConsoleR(){

if(consoleR == null){

consoleR = new Console();

}

consoleR.getButtons(0).setText("存款");

consoleR.getButtons(1).setText("取款");

consoleR.getButtons(2).setText("退出");

return consoleR;

}

private Console(){

vBox = new VBox(30);

funcs = new Button[3];

setModule();

loadModule();

}

public Button getButtons(int i){

return funcs[i];

}

public VBox getvBox(){

return vBox;

}

private void setModule(){

for(int i = 0; i<3; i++){

funcs[i] = new Button();

funcs[i].setPrefSize(75,32);

funcs[i].setFont(Font.font(18));

}

}

private void loadModule(){

vBox.getChildren().addAll(funcs);

vBox.setAlignment(Pos.CENTER);

}

}

package three.view;

import javafx.geometry.Pos;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

public class **InfoPage** {

private GridPane infoPane;

private TextArea records;

private Label[] items;//账户编号，账户名，总金额，电话，住址，

private Label[] values;

public InfoPage(){

infoPane = new GridPane();

records = new TextArea();

items = new Label[5];

values = new Label[5];

setItems();

loadModules();

}

public TextArea getRecords() {

return records;

}

public GridPane getInfoPane(){

return infoPane;

}

private void setItems(){

for(int i = 0; i<5; ++i){

items[i] = new Label();

values[i] = new Label();

items[i].setFont(Font.font(18));

values[i].setFont(Font.font(18));

}

items[0].setText("编 号：");

items[1].setText("账户名：");

items[2].setText("总金额：");

items[3].setText("电 话：");

items[4].setText("住 址");

records.setFont(Font.font(20));

records.setPrefSize(350,170);

}

public void setValues(int id, String name, double money, String tel, String address){

values[0].setText(String.valueOf(id));

values[1].setText(name);

values[2].setText(String.valueOf(money));

values[3].setText(tel);

values[4].setText(address);

}

private void loadModules(){

for(int i = 0; i<5; ++i){

infoPane.add(items[i],0,i);

infoPane.add(values[i],1,i);

}

infoPane.add(records,0,5,2,2);

infoPane.setAlignment(Pos.CENTER);

infoPane.setPrefSize(400, 400);

infoPane.setHgap(20);

infoPane.setVgap(8);

}

}

package three.view;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

public class **InqPage** {

private GridPane inqPane;

private Label[] prompt;

private TextField[] content;

private Button submit;

private static InqPage inqPage = new InqPage();

public static InqPage getInqPage (){

return inqPage;

}

private InqPage(){

inqPane = new GridPane();

prompt = new Label[2];

content = new TextField[2];

submit = new Button();

setModule();

loadModule();

}

public TextField getContent(int i) {

return content[i];

}

public Button getSubmit() {

return submit;

}

public GridPane getInqPane() {

return inqPane;

}

private void setModule(){

for(int i = 0; i<2; ++i){

prompt[i] = new Label();

content[i] = new TextField();

prompt[i].setFont(Font.font(18));

content[i].setFont(Font.font(18));

content[i].setPrefSize(150,36);

}

prompt[0].setText("账户名称：");

prompt[1].setText("账户密码：");

submit.setText("Submit");

submit.setPrefSize(75,38);

submit.setFont(Font.font(15));

}

private void loadModule(){

for(int i = 0; i<2; ++i){

inqPane.add(prompt[i], 0, i);

inqPane.add(content[i], 1, i);

}

inqPane.add(submit,1,2);

inqPane.setAlignment(Pos.CENTER);

inqPane.setPrefSize(400,400);

inqPane.setVgap(30);

inqPane.setHgap(20);

}

public void clear(){

for(int i = 0; i<2; ++i){

content[i].clear();

}

}

}

package three.view;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

public class **LoginPage** {

private GridPane loginPane;

private Label[] items;

private TextField[] values;

private Button[] selection;

private static LoginPage loginPage = new LoginPage();

public static LoginPage getLoginPage(){

return loginPage;

}

private LoginPage(){

loginPane = new GridPane();

items = new Label[4];

values = new TextField[4];

selection = new Button[2];

setModule();

loadModule();

}

public GridPane getLoginPane() {

return loginPane;

}

public Button getButton(int i){

return selection[i];

}

public TextField getTextField(int i){

return values[i];

}

private void setModule(){

for(int i = 0; i<4; i++){

items[i] = new Label();

items[i].setFont(Font.font(18));

values[i] = new TextField();

values[i].setFont(Font.font(16));

values[i].setPrefSize(150,30);

}

items[0].setText("账户名：");

items[1].setText("密 码：");

items[2].setText("电 话：");

items[3].setText("地 址：");

for(int i = 0; i<2; ++i){

selection[i] = new Button();

selection[i].setPrefSize(75, 32);

selection[i].setFont(Font.font(18));

}

selection[0].setText("重置");

selection[1].setText("提交");

}

private void loadModule(){

for(int i = 0; i<4; i++){

loginPane.add(items[i], 0, i);

loginPane.add(values[i], 1, i, 2, 1);

}

loginPane.add(selection[0], 0, 4);

loginPane.add(selection[1], 2, 4);

loginPane.setVgap(15);

loginPane.setAlignment(Pos.CENTER);

loginPane.setPrefSize(400,400);

GridPane.setMargin(selection[1], new Insets(0, 0, 0, 80));

}

public void clear(){

for(int i = 0; i < 4; i++){

loginPage.getTextField(i).clear();

}

}

}

package three.view;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

import three.control.FunControl;

import three.control.InfoControl;

public class **MainFrame** extends Application{

public static BorderPane borderPane = new BorderPane();

public void start(Stage stage){

borderPane.setCenter(WelcomePage.getInstance().getCanvas());

borderPane.setLeft(Console.getConsoleL().getvBox());

borderPane.setRight(Console.getConsoleR().getvBox());

BorderPane.setMargin(borderPane.getLeft(), new Insets(0, 0, 0, 20));

BorderPane.setMargin(borderPane.getRight(), new Insets(0, 20, 0, 0));

InfoControl.addAllPageEvents();

InfoControl.addmenuEvents();

FunControl.addFunEvents();

Scene scene = new Scene(borderPane);

stage.setScene(scene);

stage.show();

}

}

package three.view;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

public class **TransacPage** {

private GridPane transacPane;

private Label[] items;

private TextField[] values;

private Button confirm;

private String type;

public TransacPage(String type){

transacPane = new GridPane();

items = new Label[4];

values = new TextField[4];

confirm = new Button();

this.type = type;

setModule();

loadModule();

}

public TextField getTextField(int i){

return values[i];

}

public Button getConfirm() {

return confirm;

}

public GridPane getTransacPane() {

return transacPane;

}

public String getType(){

return type;

}

private void setModule(){

for(int i = 0; i < 4; i++){

items[i] = new Label();

items[i].setFont(Font.font(18));

values[i] = new TextField();

values[i].setFont(Font.font(18));

values[i].setPrefSize(150,30);

}

items[0].setText("账户名：");

items[1].setText("密 码：");

items[2].setText("金 额：");

items[3].setText("责任者：");

confirm.setFont(Font.font(16));

confirm.setText("Confirm");

confirm.setPrefSize(88,32);

}

private void loadModule(){

for(int i = 0; i < 4; ++i){

transacPane.add(items[i], 0, i);

transacPane.add(values[i], 1, i);

}

transacPane.add(confirm, 0, 4);

transacPane.setAlignment(Pos.CENTER);

transacPane.setVgap(15);

transacPane.setPrefSize(400,400);

}

}

package three.view;

import javafx.scene.canvas.Canvas;

import javafx.scene.canvas.GraphicsContext;

import javafx.scene.text.Font;

public class **WelcomePage** {

private Canvas canvas;

private GraphicsContext gc;

private static WelcomePage welcomePage = new WelcomePage();

public static WelcomePage getInstance(){

return welcomePage;

}

public Canvas getCanvas() {

return canvas;

}

private WelcomePage(){

canvas = new Canvas(400,400);

gc = canvas.getGraphicsContext2D();

printWelcomInfo();

}

private void printWelcomInfo(){

gc.setFont(Font.font(30));

gc.fillText("欢迎使用储蓄管理系统",50,200);

}

}

package three.control;

import three.model.Client;

import three.model.Transaction;

import java.io.\*;

import java.util.ArrayList;

public class **FileOperation** {

private static File clientFile = new File("src/three/files/clientRecords.txt");

private static File transactionFile = new File("src/three/files/transactionRecords.txt");

public static void AddClient(Client c){

if(!clientFile.exists()){

try {

clientFile.createNewFile();

} catch (IOException e) {

System.out.println("Fail to create new files");

e.printStackTrace();

}

}

try(FileOutputStream fos = new FileOutputStream(clientFile,true);

ObjectOutputStream oos = new ObjectOutputStream(fos)

){

oos.writeObject(c);

}catch (FileNotFoundException e){

e.printStackTrace();

}catch (IOException e){

e.printStackTrace();

}

}

public static ArrayList<Client> getAllClients(){

if(!clientFile.exists()){

System.out.println("No records yet");

return null;

} else {

ArrayList<Client> cList = new ArrayList<>();

Client temp = null;

try(FileInputStream fis = new FileInputStream(clientFile)

){

ObjectInputStream ois = null;

while (fis.available() > 0){

try{

ois = new ObjectInputStream(fis);

temp = (Client)ois.readObject();

cList.add(temp);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}catch (IOException e){

e.printStackTrace();

}

return cList;

}

}

public static void writeBack(ArrayList<Client> clientList){

if(!clientFile.exists()){

try {

clientFile.createNewFile();

} catch (IOException e) {

System.out.println("Fail to create new files");

e.printStackTrace();

}

} else {

try(FileOutputStream fos = new FileOutputStream(clientFile)

){

ObjectOutputStream oos = null;

for(Client c : clientList){

oos = new ObjectOutputStream(fos);

oos.writeObject(c);

}

} catch (IOException e){

e.printStackTrace();

}

}

}

public static void addTscRecord(Transaction transaction){

if(!transactionFile.exists()){

try {

transactionFile.createNewFile();

} catch (IOException e) {

System.out.println("Fail to create new files");

e.printStackTrace();

}

}

try(FileOutputStream fos = new FileOutputStream(transactionFile,true);

ObjectOutputStream oos = new ObjectOutputStream(fos)

){

oos.writeObject(transaction);

}catch (FileNotFoundException e){

e.printStackTrace();

}catch (IOException e){

e.printStackTrace();

}

}

public static ArrayList<Transaction> getAllTscRecords(){

if(!transactionFile.exists()){

System.out.println("No records yet");

return null;

} else {

ArrayList<Transaction> tList = new ArrayList<>();

Transaction temp = null;

try(FileInputStream fis = new FileInputStream(transactionFile)

){

ObjectInputStream ois = null;

while (fis.available() > 0){

try{

ois = new ObjectInputStream(fis);

temp = (Transaction)ois.readObject();

tList.add(temp);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}catch (IOException e){

e.printStackTrace();

}

return tList;

}

}

}

package three.control;

import javafx.scene.control.Alert;

import three.model.Client;

import three.model.Transaction;

import three.view.\*;

import java.util.ArrayList;

public class **InfoControl** {

private static InqPage inqPage = InqPage.getInqPage();

private static LoginPage loginPage = LoginPage.getLoginPage();

private static Console consoleL = Console.getConsoleL();

public static void addAllPageEvents(){

loginPagevents();

inqPageEvents();

}

private static void loginPagevents(){

loginPage.getButton(0).setOnAction(event -> {

loginPage.clear();

});

loginPage.getButton(1).setOnAction(event -> {

String name = loginPage.getTextField(0).getText();

String passWord = loginPage.getTextField(1).getText();

String tel = loginPage.getTextField(2).getText();

String address = loginPage.getTextField(3).getText();

Client client = new Client(name, passWord, address, tel);

FileOperation.AddClient(client);

showSignUpInfo();

});

}

private static void inqPageEvents(){

inqPage.getSubmit().setOnAction(event -> {

String name = inqPage.getContent(0).getText();

String passWord = inqPage.getContent(1).getText();

ArrayList<Client> cList = FileOperation.getAllClients();

ArrayList<Transaction> tList = FileOperation.getAllTscRecords();

for(Client c : cList){

if(c.getName().equals(name) && c.getPassWord().equals(passWord)){

InfoPage infoPage = new InfoPage();

for(Transaction t : tList){

if(t.getAccount().equals(name)){

infoPage.getRecords().appendText(t.toString());

infoPage.getRecords().appendText("\n");

}

}

infoPage.setValues(c.getId(), c.getName(), c.getMoney(), c.getTel(), c.getAddress());

changeToInfoPage(infoPage);

return;

}

}

showValidInfo();

});

}

private static void changeToWelcomePage(){

MainFrame.borderPane.setCenter(WelcomePage.getInstance().getCanvas());

}

public static void changeToInqPage(){

inqPage.clear();

MainFrame.borderPane.setCenter(inqPage.getInqPane());

}

private static void changeToInfoPage(InfoPage infoPage){

MainFrame.borderPane.setCenter(infoPage.getInfoPane());

}

public static void changeToLoginPage(){

loginPage.clear();

MainFrame.borderPane.setCenter(loginPage.getLoginPane());

}

private static void showValidInfo(){

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("Information");

a.setHeaderText("Checking account...");

a.setContentText("账号或密码错误！请重新输入");

a.showAndWait();

}

private static void showSignUpInfo(){

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("Information");

a.setHeaderText("系统提醒");

a.setContentText("账号开通成功，初始金额为￥1.00元");

a.showAndWait();

}

public static void addmenuEvents(){

consoleL.getButtons(0).setOnAction(event -> {

changeToLoginPage();

});

consoleL.getButtons(1).setOnAction(event -> {

changeToInqPage();

});

consoleL.getButtons(2).setOnAction(event -> {

changeToWelcomePage();

});

}

}

package three.control;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import three.model.Client;

import three.model.Transaction;

import three.view.Console;

import three.view.MainFrame;

import three.view.TransacPage;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Optional;

public class **FunControl** {

private static Console consoleR = Console.getConsoleR();

public static void addFunEvents(){

consoleR.getButtons(0).setOnAction(event -> {

TransacPage transacPage = new TransacPage("存款");

changeToTransacPage(transacPage);

TransacPageEvent(transacPage);

});

consoleR.getButtons(1).setOnAction(event -> {

TransacPage transacPage = new TransacPage("取款");

changeToTransacPage(transacPage);

TransacPageEvent(transacPage);

});

consoleR.getButtons(2).setOnAction(event -> {

Alert alert = new Alert(Alert.AlertType.CONFIRMATION);

alert.setTitle("Confirmation");

alert.setHeaderText("");

alert.setContentText("Do you want to quit the system?");

Optional<ButtonType> result = alert.showAndWait();

if(result.isPresent() && result.get() == ButtonType.OK){

System.exit(0);

}

});

}

private static void changeToTransacPage(TransacPage transacPage){

MainFrame.borderPane.setCenter(transacPage.getTransacPane());

}

private static void TransacPageEvent(TransacPage transacPage){

transacPage.getConfirm().setOnAction(event -> {

//1.find the object client

ArrayList<Client> clients = FileOperation.getAllClients();

Client client = getClient(clients, transacPage.getTextField(0).getText());

double money = Double.parseDouble(transacPage.getTextField(2).getText());

if(transacPage.getType().equals("取款") && money > client.getMoney()){

notEnoughMoneyInfo();

return;

}

Date currentDate = new Date();

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd-HH:mm:ss");

String date = simpleDateFormat.format(currentDate);

String name = transacPage.getTextField(0).getText();

String charger = transacPage.getTextField(3).getText();

if(transacPage.getType().equals("取款")){

client.withDraw(-1 \* money);

} else {

client.deposit(money);

}

Transaction transaction = new Transaction(name, date, transacPage.getType(), money,charger);

FileOperation.writeBack(clients);

FileOperation.addTscRecord(transaction);

showTransacInfo(transacPage.getType(),money);

});

}

private static void showTransacInfo(String type, double money){

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("Information");

a.setHeaderText("系统提醒");

a.setContentText(type + "操作成功！" + "金额为：￥" + money);

a.showAndWait();

}

private static void notEnoughMoneyInfo(){

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("Information");

a.setHeaderText("系统提醒");

a.setContentText("非法金额！");

a.showAndWait();

}

private static Client getClient(ArrayList<Client> clientArrayList, String name){

for(Client c : clientArrayList){

if(c.getName().equals(name))

return c;

}

return null;

}

}