/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Bailey Thompson

\* Bank (1.0)

\* 16 September 2016

\* Info: This program simulates an automated banking system.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

//declaring package

package bank;

//declaring imports

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.io.BufferedOutputStream;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import static java.lang.Integer.parseInt;

import java.nio.file.FileAlreadyExistsException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import static java.nio.file.StandardOpenOption.TRUNCATE\_EXISTING;

import static java.nio.file.StandardOpenOption.WRITE;

import java.text.NumberFormat;

import javax.swing.JButton;

import javax.swing.JFormattedTextField;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.text.NumberFormatter;

//declaring main class

public class Bank {

//used for file io

Path file = Paths.get("JBBank.txt");

String saveFile;

String[] split;

//used for formatting

NumberFormat format;

NumberFormatter formatter;

//used for writing in numbers

JFormattedTextField textField;

//name of the frame

private final JFrame frame = new JFrame("JB's Bank");

//various variables used for proccessing

int username, password, balance, usernameIndex;

boolean isDepositing, isRegister, used;

//the main message shown of main screen

JLabel text = new JLabel("Welcome to JB's Bank! What action would you like to make today?");

//declaring various buttons

JButton deposit = new JButton("Deposit"), withdraw = new JButton("Withdraw"), next = new JButton("Next");

JButton exit = new JButton("Exit"), back = new JButton("Back"), login = new JButton("Login"), register = new JButton("Register");

JButton money20 = new JButton("$20"), money50 = new JButton("$50"), money100 = new JButton("$100");

JButton money200 = new JButton("$200"), money500 = new JButton("$500"), money1000 = new JButton("$1000");

//panel used for numbers

JPanel numberPanel = new JPanel();

//declaring main method

public static void main(String[] args) {

//sending to prepareGUI method

Bank Bank = new Bank();

Bank.prepareGUI();

}

//declaring mathod used for GUI

private void prepareGUI() {

//loading from file io

load();

//making program gui-friendly

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

frame.setSize(400, 400);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

startScreen();

//adding to number pad

numberPanel.add(money20);

numberPanel.add(money50);

numberPanel.add(money100);

numberPanel.add(money200);

numberPanel.add(money500);

numberPanel.add(money1000);

numberPanel.setLayout(new GridLayout(3, 2));

//when login button is clicked

login.addActionListener((ActionEvent e) -> {

isRegister = false;

operations();

});

//when register button is clicked

register.addActionListener((ActionEvent e) -> {

isRegister = true;

operations();

});

//when exit button is clicked

exit.addActionListener((ActionEvent e) -> {

if ("Return To Main Screen".equals(exit.getText())) {

startScreen();

} else {

System.exit(0);

}

});

//when back button is clicked

back.addActionListener((ActionEvent e) -> {

loginScreen();

});

//when deposit button is clicked

deposit.addActionListener((ActionEvent e) -> {

//setting the GUI

text.setText("Select amount to deposit. Amount in account: $" + balance);

frame.getContentPane().removeAll();

frame.add(text);

frame.add(numberPanel);

frame.add(back);

frame.setLayout(new GridLayout(3, 1));

frame.repaint();

isDepositing = true;

});

//when withdraw button is clicked

withdraw.addActionListener((ActionEvent e) -> {

//setting the GUI

text.setText("Select amount to withdraw. Amount in account: $" + balance);

frame.getContentPane().removeAll();

frame.add(text);

frame.add(numberPanel);

frame.add(back);

frame.setLayout(new GridLayout(3, 1));

frame.repaint();

isDepositing = false;

});

//when money20 button is clicked

money20.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 20;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 20) {

balance -= 20;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when money50 button is clicked

money50.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 50;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 50) {

balance -= 50;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when money100 button is clicked

money100.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 100;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 100) {

balance -= 100;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when money200 button is clicked

money200.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 200;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 200) {

balance -= 200;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when money500 button is clicked

money500.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 500;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 500) {

balance -= 500;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when money1000 button is clicked

money1000.addActionListener((ActionEvent e) -> {

//logic executed

if (isDepositing == true) {

if (balance <= 1000000000) {

balance += 1000;

text.setText("Select amount to deposit. Amount in account: $" + balance);

}

} else {

if (balance >= 1000) {

balance -= 1000;

} else {

balance = 0;

}

text.setText("Select amount to withdraw. Amount in account: $" + balance);

}

//resetting the balance

split[usernameIndex + 2] = String.valueOf(balance);

//saving to file io

save();

});

//when next button is clicked

next.addActionListener((ActionEvent e) -> {

//setting used variable to false

used = false;

//when login is being entered

if (username == 0) {

//if user has entered something

if (textField.getValue() != null) {

//when what user has entered is long enough

if ((int) (textField.getValue()) >= 100000) {

//setting temporary variable

int tempUsername = (int) (textField.getValue());

//if login is being proccessed

if (isRegister == false) {

for (int counter = 0; counter < split.length; counter += 3) {

if (parseInt(split[counter]) == tempUsername) {

used = true;

usernameIndex = counter;

}

}

if (used == true) {

username = tempUsername;

} else {

text.setText("This bank code does not exist!");

}

//if register is being proccessed

} else {

for (int counter = 0; counter < split.length; counter += 3) {

if (parseInt(split[counter]) == tempUsername) {

used = true;

}

}

if (used == false) {

username = tempUsername;

usernameIndex = split.length;

} else {

text.setText("That bank code has already been used.");

}

}

//if username entered passes all checks

if (username != 0) {

text.setText("Please insert your four digit passcode!");

formatter.setMaximum(9999);

textField.setValue(null);

save();

}

} else {

text.setText("Your bank code must not start with a zero and must be six digits!");

}

}

//when password is being entered

} else if (username != 0) {

//if user has entered something

if (textField.getValue() != null) {

//when what user has entered is long enough

if ((int) (textField.getValue()) >= 1000) {

//setting temporary variable

int tempPassword = (int) (textField.getValue());

//if login is being proccessed

if (isRegister == false) {

if (tempPassword == parseInt(split[usernameIndex + 1])) {

password = tempPassword;

} else {

text.setText("Incorrect password!");

}

balance = parseInt(split[usernameIndex + 2]);

//if register is being proccessed

} else {

password = (int) (textField.getValue());

saveFile = "";

for (int counter = 0; counter < split.length; counter += 1) {

saveFile += split[counter] + " ";

}

saveFile += username + " " + password + " 0 ";

split = saveFile.split("\\s+");

save();

}

//if password entered passes all checks

if (password != 0) {

loginScreen();

}

} else {

text.setText("Your passcode must not start with a zero and must be four digits!");

}

}

}

});

}

//declaring method used for menu

private void startScreen() {

//resetting variables

username = password = balance = 0;

//setting text

text.setText("Welcome to JB's Bank! What action would you like to make today?");

exit.setText("Exit Application");

//resetting GUI

frame.getContentPane().removeAll();

frame.add(text);

frame.add(login);

frame.add(register);

frame.add(exit);

frame.setLayout(new GridLayout(4, 1));

frame.repaint();

}

//declaring method used for when user is logged in

private void loginScreen() {

//setting text

text.setText("Please select your operation!");

//resetting GUI

frame.getContentPane().removeAll();

frame.add(text);

frame.add(deposit);

frame.add(withdraw);

frame.add(exit);

frame.setLayout(new GridLayout(4, 1));

frame.repaint();

}

//method used for logging in

private void operations() {

//setting text

text.setText("Please insert your six digit bank code!");

exit.setText("Return To Main Screen");

//formatting for test field

format = NumberFormat.getInstance();

formatter = new NumberFormatter(format);

formatter.setValueClass(Integer.class);

formatter.setMinimum(1);

formatter.setMaximum(999999);

formatter.setAllowsInvalid(false);

formatter.setCommitsOnValidEdit(true);

textField = new JFormattedTextField(formatter);

//resetting GUI

frame.getContentPane().removeAll();

frame.add(text);

frame.add(textField);

frame.add(next);

frame.add(exit);

frame.setLayout(new GridLayout(4, 1));

frame.repaint();

}

//method used for loading from file

private void load() {

try {

//trying to create file

Files.createFile(file);

//executed if file already exists

} catch (FileAlreadyExistsException x) {

//file is read from and saved to variable saveFile is file already exists

try (InputStream in = Files.newInputStream(file);

BufferedReader reader = new BufferedReader(new InputStreamReader(in))) {

String line;

while ((line = reader.readLine()) != null) {

//content of file is saved to saveFile

saveFile = line;

}

} catch (IOException y) {

System.err.println(y);

}

} catch (IOException x) {

System.err.println(x);

}

//if the file does not contain anything since it was just created, default variables are used for save file

if (saveFile == null) {

saveFile = "301942 1234 0 ";

}

//a String array is created and each part of the array is saved to from saveFile seperated by spaces

split = saveFile.split("\\s+");

}

//method used for saving to file

private void save() {

saveFile = "";

//weird loop used for saving to file

for (int counter = 0; counter < split.length; counter += 1) {

saveFile += split[counter] + " ";

}

//saveFile is converted to byte data

byte data[] = saveFile.getBytes();

//byte data is saved to file using file io

try (OutputStream out = new BufferedOutputStream(

Files.newOutputStream(file, WRITE, TRUNCATE\_EXISTING))) {

out.write(data, 0, data.length);

} catch (IOException x) {

System.err.println(x);

}

}

}