package ATM;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

import javax.swing.\*;

import java.io.\*;

public class MyFrame extends JFrame implements ActionListener{

//create instance

private TextPanel txtPan = null;

private int state = 0;

private String info = null;

private BankAccount accountCheck = null;

private Checking checkingAccount = null;

private Saving savingAccount = null;

private int pinCounterControl; //if user enters pin wrong 3 times process starts all over again

private String accountType = null; //saves either checking or saving in it

private String accountTransaction = null; //saves what transaction should be made

private double accountBalance; //saves account balance, either checking or saving

//center panel

private JPanel centerPan = null;

private JTextArea text = null;

private String str = null;

private JScrollPane scroll = null;

//east panel

private JPanel eastPan = null;

private ArrayList<JButton> buttons = null;

private GridLayout layout = null;

private String[] array = {"A", "B", "C"};

//west panel

JPanel westPan = null;

private JTextField txt = null;

private ArrayList<JButton> keyPad = null;

private FlowLayout flowLayout = null;

private String[] keyArray = {"7", "8", "9", "4", "5", "6",

"1", "2", "3", "0", ".", "CE"};

public MyFrame(){

super("ATM");

//initialize objects

txtPan = new TextPanel();

this.state = 1;

this.accountCheck = new BankAccount();

this.checkingAccount = new Checking();

this.savingAccount = new Saving();

//set east panel

eastPan = new JPanel();

buttons = new ArrayList<JButton>();

layout = new GridLayout(3, 1, 0, 0);

eastPan.setPreferredSize(new Dimension(75, 30));

eastPan.setMaximumSize(new Dimension(75, 30));

eastPan.setMinimumSize(new Dimension(75, 30));

eastPan.setLayout(layout);

for(int i = 0; i < array.length ; i++){

buttons.add(new JButton(array[i]));

}

for(JButton btn: buttons){

eastPan.add(btn).setBackground(Color.LIGHT\_GRAY);

btn.addActionListener(this);

}

//set center panel

centerPan = new JPanel();

text = new JTextArea();

centerPan.setPreferredSize(new Dimension(300, 400));

text.setEditable(true);

text.setLineWrap(true);

text.setWrapStyleWord(true);

str = "Enter customer number: -->\npress A when done\nA = OK";

text.setText(str);

//jscroll pane

scroll = new JScrollPane(text,JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS,

JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

scroll.setPreferredSize(new Dimension(450, 325));

scroll.setEnabled(true);

centerPan.add(scroll);

//west panel

westPan = new JPanel();

keyPad = new ArrayList<JButton>();

flowLayout = new FlowLayout();

txt = new JTextField(17);

//set layout

flowLayout.setHgap(0);

flowLayout.setVgap(0);

//add to panel

westPan.setPreferredSize(new Dimension(200, 200));

westPan.setLayout(flowLayout);

westPan.add(txt);

//add buttons to panel

for(int i = 0; i < keyArray.length ; i++){

keyPad.add(new JButton(keyArray[i]));

}

for(JButton btn: keyPad){

btn.setPreferredSize(new Dimension(63,77));

westPan.add(btn).setBackground(Color.LIGHT\_GRAY);

btn.addActionListener(this);

}

//add panels to frame

this.add(eastPan, BorderLayout.EAST);

this.add(centerPan, BorderLayout.CENTER);

this.add(txtPan, BorderLayout.SOUTH);

this.add(westPan, BorderLayout.WEST);

//set frame

this.setSize(800, 400);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.setLocationRelativeTo(null);

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent event) {

Object source = event.getSource();

int i = 0;

//check if west panel

for(JButton btn: keyPad){

if(source != keyPad.get(11) && source == btn)

txt.setText(txt.getText() + keyArray[i]);

else if(source == keyPad.get(11))

txt.setText("");

i++;

}//end loop

//check if east panel

if(source == buttons.get(0) && this.state < 3)

eastPanelState();

else if(this.state == 3 && source == buttons.get(0)){

this.accountType = "Checking";

eastPanelState();

}

else if(this.state == 3 && source == buttons.get(1)){

this.accountType = "Savings";

eastPanelState();

}

else if(this.state == 4 && source == buttons.get(0)){

this.accountTransaction = "Withdraw";

eastPanelState();

}

else if(this.state == 4 && source == buttons.get(1)){

this.accountTransaction = "Deposit";

eastPanelState();

}

else if(this.state >= 3 && source == buttons.get(2)){

this.state = 1;

text.setText(str);

}

}

//this keeps track of the state the machine is in

//also displays needed state machine text on text area

//method checks first if user entered anything, if not breaks out of method

public void eastPanelState(){

this.info = txt.getText();

boolean isInFile = false;

this.txt.setText(""); //reset keypad text to empty

//check if machine at state one

if(this.state == 1){

//check if account exist in file

isInFile = accountCheck.isAccountInFile(this.info);

this.checkingAccount.setAccountNum(Integer.valueOf(this.info));

this.savingAccount.setAccountNum(Integer.valueOf(this.info));

if(!isInFile){

this.text.setText(text.getText() + "\n\nError: Invalid account number!\nPlease try again");

this.txt.setText("");

return;

}

this.text.setText("Enter Pin: \nA = OK");

this.state++;

}

//check if machine at state two

else if(this.state == 2){

isInFile = accountCheck.isPinCorrect(this.info);

if(!isInFile){

this.text.setText(text.getText() + "\n\nError: Invalid pin number!\nPlease try again");

this.pinCounterControl++; //increment pinCounter if user enters wrong pin

//if user enters pin wrong 3 times start process from beginning

if(this.pinCounterControl == 3){

this.state = 1;

this.text.setText(str);

}

return;

}//end if statement

this.text.setText("Select Account: \n\nA = Checking\n"

+ "B = Savings\nC = Exit");

this.state++;

}

//checks if machine at state 3

else if(this.state == 3){

//check if account is checking or savings

if(this.accountType.equals("Checking"))

this.accountBalance = this.checkingAccount.getBalance();

else

this.accountBalance = this.savingAccount.getBalance();

this.text.setText("Balance = " + this.accountBalance + "\nEnter amount and select transaction:\nA = Withdraw"

+ "\nB = Deposit\nC = cancel");

state++;

}

//check if machine at state 4 to update file data

else if(this.state == 4){

//check if amount entered

if(this.info.isEmpty()){

this.text.setText(this.text.getText() + "\n\nPlease enter amount");

return;

}

//check if its withdraw or deposit

int amount = Integer.valueOf(this.info);

if(this.accountTransaction.equals("Withdraw")){

if(this.accountType.equals("Checking"))

this.checkingAccount.withdraw(amount);

else

this.savingAccount.withdraw(amount);

}//end if statement

else{

if(this.accountType.equals("Checking"))

this.checkingAccount.deposit(amount);

else

this.savingAccount.deposit(amount);

}//end else

this.state = 1;

this.text.setText(str);

}

}//end method

}