

package com.example.demo;

```
import java.io.*;
import java.util.*;
import java.util.random.*;
import java.util.Vector;
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.ListView;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.Serializable;

class bank{
    // account[] acc=new account[100];
    Vector<account> acc=new Vector<>();
    Scanner sc=new Scanner(System.in);
    int acn=0;
    Random rand = new Random();
    HashSet<Integer> set = new HashSet<>();
    public void addacc(String name){

        int aacn=rand.nextInt(1000,9999);

        while (set.contains(aacn)){
            aacn=rand.nextInt(1000,9999);
        }
        set.add(aacn);

        account newaccount=new account(aacn,name);
        acc.addElement(newaccount);
        acn++;
    }
    public void listall(){
        if (acn==0){
            System.out.println("No Account Exist");
            return;
        }
        for (int i=0;i<acn;i++){
            if (acc.get(i).name!=null){
                System.out.println(acc.get(i).acn+" "+ acc.get(i).name);
                System.out.println("~~~~~");
            }
        }
    }
    public int remove(int ii){
        for (int i=0;i<acn;i++){
            if (acc.get(i).acn==ii){
                acc.remove(i);
                acn--;
                return 0;
            }
        }
        return -1;}

    public int deposit(int no,double money){

        for (int i = 0; i < acn; i++) {
            if (acc.get(i).acn==no){
                acc.get(i).deposit(money);
                return 0;
            }
        }
        System.out.println("No account with this number");
        return -1;
    }

    public int withdraw(int no,double money){
        for (int i = 0; i < acn; i++) {
            if (acc.get(i).acn==no){
```

```
            int c=acc.get(i).withdraw(money);
            return c;
        }
    }
    System.out.println("No account with this number");
    return -1;
}
public double check(int no){

    for (int i = 0; i < acn; i++) {
        if (acc.get(i).acn==no){
            return acc.get(i).print();
        }
    }

    return -1;
}
// public void exit(){
//     writeVectorToFile(acc);
// }
// }
public void writeVectorToFile() {
    try (FileOutputStream fileOut = new FileOutputStream("bank.txt");
        ObjectOutputStream out = new ObjectOutputStream(fileOut)) {
        out.writeObject(acc);
        System.out.println("Vector has been serialized and saved");
        out.close();
    } catch (IOException e) {
        System.out.println("Error saving");
        System.out.println("Files not saved");
    }
}

public void retrieve() {

    try (FileInputStream fileIn = new FileInputStream("bank.txt");
        ObjectInputStream in = new ObjectInputStream(fileIn)) {

        List<account> accounts = (List<account>) in.readObject();

        // Add the accounts to your collection
        for (account i : accounts) {
            // Assuming `acc` is a collection in which you want to add these
accounts
            acc.addElement(i);
            acn++;
        }

    } catch (IOException | ClassNotFoundException e) {
        System.out.println("File not found");
    }
}

public List<account> getAllAccounts() {
    return acc;
}
}
class account implements Serializable {
    private static final long serialVersionUID = 9011L;
    String name;
    int acn;
    private double money=0;
    public account(int acn,String name){
        this.name=name;
        this.acn=acn;
    }
    public void deposit(double dep){
        money=dep+money;
    }
    public double print(){
        return money;
    }
    public int withdraw(double dep){
        if (dep>money){
            System.out.println("Does not have enough money");
            return 2;
        }
        money=money-dep;
        return 0;
    }
}
```

```

}
public class HelloApplication extends Application {
    bank pccoe=new bank();

    @Override
    public void start(Stage stage) {
        pccoe.retrive();
        // Create the GridPane for the main content
        GridPane homepage = new GridPane();
        homepage.setAlignment(Pos.CENTER);
        // Create a label for the title
        Label titleLabel = new Label("PCCOE Bank");
        titleLabel.setFont(new Font(30));

        // Create buttons for various actions
        Button listButton = new Button("List All");
        Button addButton = new Button("ADD Account");
        Button removeButton = new Button("Remove Account");
        Button balanceButton = new Button("Balance");
        Button withdrawButton = new Button("Withdraw");
        Button depositButton = new Button("Deposit");

        // Set preferred size for buttons
        double buttonWidth = 100;
        double buttonHeight = 40;
        listButton.setPrefSize(buttonWidth, buttonHeight);
        addButton.setPrefSize(buttonWidth, buttonHeight);
        removeButton.setPrefSize(buttonWidth, buttonHeight);
        balanceButton.setPrefSize(buttonWidth, buttonHeight);
        withdrawButton.setPrefSize(buttonWidth, buttonHeight);
        depositButton.setPrefSize(buttonWidth, buttonHeight);

        // Add title label and buttons to GridPane
        homepage.add(titleLabel, 0, 0, 3, 1); // Title spanning across 3 columns
        homepage.add(listButton, 0, 1);
        homepage.add(addButton, 1, 1);
        homepage.add(removeButton, 2, 1);
        homepage.add(balanceButton, 0, 2);
        homepage.add(withdrawButton, 1, 2);
        homepage.add(depositButton, 2, 2);

        // Define event handlers for buttons
        listButton.setOnAction(e -> handleListAll());
        addButton.setOnAction(e -> handleAddAccount());
        removeButton.setOnAction(e -> handleRemoveAccount());
        balanceButton.setOnAction(e -> handleBalance());
        withdrawButton.setOnAction(e -> handleWithdraw());
        depositButton.setOnAction(e -> handleDeposit());

        // Create a scene and set it to the stage
        Scene scene = new Scene(homepage, 400, 300);
        stage.setTitle("Banking Application");
        stage.setScene(scene);
        stage.show();
        stage.setOnCloseRequest(e -> {
            pccoe.writeVectorToFile();
        });
    }

    // Event handler methods
    private void handleListAll() {
        // Create a popup window to display the list of accounts
        Stage popupStage = new Stage();
        popupStage.initModality(Modality.APPLICATION_MODAL);
        popupStage.setTitle("List of Accounts");

        // Create a label to display the title
        Label titleLabel = new Label("List of Accounts");
        titleLabel.setFont(new Font(20));

        // Create a ListView to display the account information
        ListView<String> accountsListView = new ListView<>();

        // Get the list of accounts from the bank object
        List<account> accounts = pccoe.getAllAccounts(); // Assuming
        getAllAccounts() returns the list

        // Populate the ListView with account information
        for (account acc : accounts) {
            accountsListView.getItems().add(acc.acn + " " + acc.name);
        }
    }

    private void handleAddAccount() {
        Stage popupStage = new Stage();
        popupStage.initModality(Modality.APPLICATION_MODAL); // Make it
        modal to block interaction with other windows
        popupStage.setTitle("Add Account");

        Label messageLabel = new Label("Add Customer Info: ");
        messageLabel.setFont(new Font(20));
        TextField name=new TextField();
        Label n=new Label("Name: ");

        name.setMaxWidth(100*2);
        Button Submitbutton = new Button("Submit");

        Submitbutton.setOnAction(e -> {pccoe.addacc(name.getText());
        popupStage.close();});
        HBox hbo=new HBox(n,name);
        hbo.setAlignment(Pos.CENTER);
        VBox vbox = new VBox(10, messageLabel,hbo, Submitbutton);
        vbox.setAlignment(Pos.CENTER);

        Scene popupScene = new Scene(vbox, 300, 150);
        popupStage.setScene(popupScene);
        popupStage.show();
    }

    private void handleRemoveAccount() {
        Stage popupStage = new Stage();
        popupStage.initModality(Modality.APPLICATION_MODAL); // Make it
        modal

        Label messageLabel = new Label("Enter Account Number :");
        messageLabel.setFont(new Font(20));
        TextField accNumberField = new TextField();
        Label n = new Label("Acc. Number: ");

        accNumberField.setMaxWidth(100 * 2);
        Button submitButton = new Button("Submit");

        submitButton.setOnAction(e -> {
            int accNumber = Integer.parseInt(accNumberField.getText());
            int k=pccoe.remove(accNumber);
            answaning(k,popupStage);
            popupStage.close();// Call remove method to remove account
        });

        HBox hbox = new HBox(n, accNumberField);
        hbox.setAlignment(Pos.CENTER);
        VBox vbox = new VBox(10, messageLabel, hbox, submitButton);
        vbox.setAlignment(Pos.CENTER);

        Scene popupScene = new Scene(vbox, 300, 150);
        popupStage.setScene(popupScene);
        popupStage.show();
    }

    private void handleBalance() {
        Stage popupStage = new Stage();
        popupStage.initModality(Modality.APPLICATION_MODAL); // Make it
        modal to block interaction with other windows
        popupStage.setTitle("Account Balance");

        Label messageLabel = new Label("Enter Account Number :");
    }
}

```

```

messageLabel.setFont(new Font(20));
TextField name=new TextField();
Label n=new Label("Acc. Number: ");

name.setMaxWidth(100*2);
Button Submitbutton = new Button("Submit");
Label Bal=new Label();

Submitbutton.setOnAction(e ->{
    int nn=Integer.parseInt( name.getText());
    double balance=pccoe.check(nn);
    int ret= (int) balance;

    Bal.setText(String.valueOf(balance));
    answaning(ret,popupStage);
});
HBox hbo=new HBox(n,name);
hbo.setAlignment(Pos.CENTER);
VBox vbox = new VBox(10, messageLabel,hbo, Submitbutton,Bal);
vbox.setAlignment(Pos.CENTER);

Scene popupScene = new Scene(vbox, 300, 150);
popupStage.setScene(popupScene);
popupStage.show();
}

private void handleWithdraw() {
    Stage popupStage = new Stage();
    popupStage.initModality(Modality.APPLICATION_MODAL); // Make it
modal

    Label messageLabel = new Label("Enter Account Number :");
    messageLabel.setFont(new Font(20));
    TextField accNumberField = new TextField();
    Label n = new Label("Acc. Number: ");
    Label amountLabel = new Label("Amount :");
    TextField amountField = new TextField();

    accNumberField.setMaxWidth(100 * 2);
    amountField.setMaxWidth(100 * 2);
    Button submitButton = new Button("Submit");

    submitButton.setOnAction(e -> {
        int accNumber = Integer.parseInt(accNumberField.getText());
        double amount = Double.parseDouble(amountField.getText());
        int returnans=pccoe.withdraw(accNumber, amount); // Call withdraw
method
        answaning(returnans,popupStage);
        popupStage.close();

    });

    HBox hbox1 = new HBox(n, accNumberField);
    hbox1.setAlignment(Pos.CENTER);
    HBox hbox2 = new HBox(amountLabel, amountField);
    hbox2.setAlignment(Pos.CENTER);
    VBox vbox = new VBox(10, messageLabel, hbox1, hbox2, submitButton);
    vbox.setAlignment(Pos.CENTER);

    Scene popupScene = new Scene(vbox, 300, 180);
    popupStage.setScene(popupScene);
    popupStage.show();
}

private void answaning(int returnans, Stage popupStage) {
    if (returnans!=0){
        if (returnans ==-1){
            Stage s=new Stage();
            Label warning=new Label("Accooount Not Found");
            VBox waningbox=new VBox(warning);
            Scene warnscene=new Scene(waningbox);
            s.setScene(warnscene);
            popupStage.close();
            s.show();
        }else if (returnans==2){
            Stage s=new Stage();
            Label warning=new Label("Not Enough Moeny");
            VBox waningbox=new VBox(warning);
            Scene warnscene=new Scene(waningbox);
            s.setScene(warnscene);

```

```

        popupStage.close();
        s.show();
    }
}

private void handleDeposit() {
    Stage popupStage = new Stage();
    popupStage.initModality(Modality.APPLICATION_MODAL); // Make it
modal

    Label messageLabel = new Label("Enter Account Number :");
    messageLabel.setFont(new Font(20));
    TextField accNumberField = new TextField();
    Label n = new Label("Acc. Number: ");
    Label amountLabel = new Label("Amount :");
    TextField amountField = new TextField();

    accNumberField.setMaxWidth(100 * 2);
    amountField.setMaxWidth(100 * 2);
    Button submitButton = new Button("Submit");

    submitButton.setOnAction(e -> {
        int accNumber = Integer.parseInt(accNumberField.getText());
        double amount = Double.parseDouble(amountField.getText());
        int k=pccoe.deposit(accNumber, amount); // Call deposit method
        answaning(k,popupStage);
        popupStage.close();
    });

    HBox hbox1 = new HBox(n, accNumberField);
    hbox1.setAlignment(Pos.CENTER);
    HBox hbox2 = new HBox(amountLabel, amountField);
    hbox2.setAlignment(Pos.CENTER);
    VBox vbox = new VBox(10, messageLabel, hbox1, hbox2, submitButton);
    vbox.setAlignment(Pos.CENTER);
    Scene popupScene = new Scene(vbox, 300, 180);
    popupStage.setScene(popupScene);
    popupStage.show();
}

public static void main(String[] args) {
    launch();
}
}

```

