Name – Aamir

Roll No. - 2401730238

Course – B.tech CSE (AI/ML)

Submitted to – Meenu Mam

import java.util.Scanner;

// Account class to store account details and perform operations

class Account {

private int accountNumber;

private String accountHolderName;

private double balance;

private String email;

private String phoneNumber;

// Constructor to initialize account

public Account(int accountNumber, String name, double initialDeposit, String email, String phone) {

this.accountNumber = accountNumber;

this.accountHolderName = name;

this.balance = initialDeposit;

this.email = email;

this.phoneNumber = phone;

}

// Method to deposit money (amount must be positive)

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount + " | New Balance: " + balance);

} else {

System.out.println("Deposit must be positive.");

}

}

// Method to withdraw money (only if sufficient balance available)

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

System.out.println("Withdrawn: " + amount + " | Remaining Balance: " + balance);

} else {

System.out.println("Invalid withdrawal! Check amount or balance.");

}

}

// Method to display all account details

public void displayAccountDetails() {

System.out.println("\n--- Account Details ---");

System.out.println("Account Number : " + accountNumber);

System.out.println("Holder Name : " + accountHolderName);

System.out.println("Balance : " + balance);

System.out.println("Email : " + email);

System.out.println("Phone : " + phoneNumber);

}

// Method to update email and phone number

public void updateContactDetails(String email, String phone) {

this.email = email;

this.phoneNumber = phone;

System.out.println("Contact details updated successfully!");

}

// Getter method to return account number

public int getAccountNumber() {

return accountNumber;

}

}

// UserInterface class to handle menu and user interaction

class UserInterface {

private Account[] accounts;

private int accountCount;

private Scanner sc;

// Constructor to initialize array and scanner

public UserInterface(int size) {

accounts = new Account[size];

accountCount = 0;

sc = new Scanner(System.in);

}

// Method to create a new account

public void createAccount() {

System.out.print("Enter Account Holder Name: ");

String name = sc.nextLine();

System.out.print("Enter Initial Deposit: ");

double deposit = sc.nextDouble();

sc.nextLine(); // consume newline

// Simple validation for email

String email;

while (true) {

System.out.print("Enter Email: ");

email = sc.nextLine();

if (email.contains("@")) break;

System.out.println("Invalid email! Must contain @");

}

// Simple validation for phone

String phone;

while (true) {

System.out.print("Enter Phone Number (10 digits): ");

phone = sc.nextLine();

if (phone.length() == 10) break;

System.out.println("Invalid phone number!");

}

// Auto-generate account number

int accNo = 1001 + accountCount;

accounts[accountCount] = new Account(accNo, name, deposit, email, phone);

accountCount++;

System.out.println("Account created successfully! Account Number: " + accNo);

}

// Helper method to find account using account number

private Account findAccount(int accNo) {

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

if (accounts[i].getAccountNumber() == accNo) {

return accounts[i];

}

}

return null;

}

// Method to deposit money in a given account

public void performDeposit() {

System.out.print("Enter Account Number: ");

int accNo = sc.nextInt();

System.out.print("Enter Amount to Deposit: ");

double amt = sc.nextDouble();

sc.nextLine(); // consume newline

Account acc = findAccount(accNo);

if (acc != null) acc.deposit(amt);

else System.out.println("Account not found!");

}

// Method to withdraw money from a given account

public void performWithdrawal() {

System.out.print("Enter Account Number: ");

int accNo = sc.nextInt();

System.out.print("Enter Amount to Withdraw: ");

double amt = sc.nextDouble();

sc.nextLine();

Account acc = findAccount(accNo);

if (acc != null) acc.withdraw(amt);

else System.out.println("Account not found!");

}

// Method to show details of a given account

public void showAccountDetails() {

System.out.print("Enter Account Number: ");

int accNo = sc.nextInt();

sc.nextLine();

Account acc = findAccount(accNo);

if (acc != null) acc.displayAccountDetails();

else System.out.println("Account not found!");

}

// Method to update contact details of a given account

public void updateContact() {

System.out.print("Enter Account Number: ");

int accNo = sc.nextInt();

sc.nextLine();

Account acc = findAccount(accNo);

if (acc != null) {

System.out.print("Enter New Email: ");

String email = sc.nextLine();

System.out.print("Enter New Phone Number: ");

String phone = sc.nextLine();

acc.updateContactDetails(email, phone);

} else {

System.out.println("Account not found!");

}

}

// Main menu to display options and perform operations

public void mainMenu() {

while (true) {

System.out.println("\n--- Banking Application ---");

System.out.println("1. Create a new account");

System.out.println("2. Deposit money");

System.out.println("3. Withdraw money");

System.out.println("4. View account details");

System.out.println("5. Update contact details");

System.out.println("6. Exit");

System.out.print("Enter your choice: ");

int choice = sc.nextInt();

sc.nextLine(); // consume newline

// switch-case for user choice

switch (choice) {

case 1: createAccount(); break;

case 2: performDeposit(); break;

case 3: performWithdrawal(); break;

case 4: showAccountDetails(); break;

case 5: updateContact(); break;

case 6:

System.out.println("Thank you for using Banking Application!");

return;

default: System.out.println("Invalid choice! Try again.");

}

}

}

}

// Main class with main() method

public class BankingApp {

public static void main(String[] args) {

UserInterface ui = new UserInterface(50); // up to 50 accounts

ui.mainMenu();

}

}