import java.util.HashMap;

import java.util.Scanner;

// BankAccount class with basic operations

class BankAccount {

private String accountHolderName;

private String accountNumber;

private double balance;

// Constructor

public BankAccount(String accountHolderName, String accountNumber) {

this.accountHolderName = accountHolderName;

this.accountNumber = accountNumber;

this.balance = 0.0;

}

// Get the account number

public String getAccountNumber() {

return accountNumber;

}

// Deposit money into account

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

} else {

System.out.println("Invalid deposit amount.");

}

}

// Withdraw money from account

public void withdraw(double amount) throws InsufficientFundsException {

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

balance -= amount;

System.out.println("Withdrawn: " + amount);

} else if (amount > balance) {

throw new InsufficientFundsException("Insufficient balance. Current balance: " + balance);

} else {

System.out.println("Invalid withdrawal amount.");

}

}

// Check account balance

public double checkBalance() {

return balance;

}

@Override

public String toString() {

return "Account Holder: " + accountHolderName + ", Account Number: " + accountNumber + ", Balance: " + balance;

}

}

// Custom exception for insufficient funds

class InsufficientFundsException extends Exception {

public InsufficientFundsException(String message) {

super(message);

}

}

// Bank class to manage multiple accounts

public class Bank {

private HashMap<String, BankAccount> accounts;

private Scanner scanner;

// Constructor

public Bank() {

accounts = new HashMap<>();

scanner = new Scanner(System.in);

}

// Create a new bank account

public void createAccount() {

System.out.print("Enter account holder's name: ");

String name = scanner.nextLine();

System.out.print("Enter a new account number: ");

String accountNumber = scanner.nextLine();

if (accounts.containsKey(accountNumber)) {

System.out.println("Account number already exists. Please choose another.");

} else {

BankAccount newAccount = new BankAccount(name, accountNumber);

accounts.put(accountNumber, newAccount);

System.out.println("Account created successfully!");

}

}

// Perform a deposit

public void deposit() {

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

BankAccount account = accounts.get(accountNumber);

if (account != null) {

System.out.print("Enter amount to deposit: ");

double amount = scanner.nextDouble();

scanner.nextLine(); // consume the newline

account.deposit(amount);

} else {

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

}

}

// Perform a withdrawal

public void withdraw() {

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

BankAccount account = accounts.get(accountNumber);

if (account != null) {

System.out.print("Enter amount to withdraw: ");

double amount = scanner.nextDouble();

scanner.nextLine(); // consume the newline

try {

account.withdraw(amount);

} catch (InsufficientFundsException e) {

System.out.println(e.getMessage());

}

} else {

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

}

}

// Check account balance

public void checkBalance() {

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

BankAccount account = accounts.get(accountNumber);

if (account != null) {

System.out.println("Current balance: " + account.checkBalance());

} else {

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

}

}

// Show options menu

public void showMenu() {

int option;

do {

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

System.out.println("1. Create Account");

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

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

System.out.println("4. Check Balance");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

option = scanner.nextInt();

scanner.nextLine(); // consume the newline

switch (option) {

case 1:

createAccount();

break;

case 2:

deposit();

break;

case 3:

withdraw();

break;

case 4:

checkBalance();

break;

case 5:

System.out.println("Exiting... Thank you for using the Bank Account Management System.");

break;

default:

System.out.println("Invalid option. Please try again.");

}

} while (option != 5);

}

// Main method

public static void main(String[] args) {

Bank bank = new Bank();

bank.showMenu();

}

}