**Habib Ur Rehman**

**FA23-BCS-116**

**Lab Task 1,2,3**

import java.io.\*;

import java.util.Scanner;

class Person implements Serializable {

    private String name;

    public Person(String name) {

        this.name = name;

    }

    public String getName() {

        return name;

    }

    public String toString() {

        return name;

    }

}

class Book implements Serializable {

    private String name;

    private String publisher;

    private Person author;

    public Book(String name, String publisher, Person author) {

        this.name = name;

        this.publisher = publisher;

        this.author = author;

    }

    public String getName() {

        return name;

    }

    public String getPublisher() {

        return publisher;

    }

    public Person getAuthor() {

        return author;

    }

    public String toString() {

        return "Book Name: " + name + "\nPublisher: " + publisher + "\nAuthor: " + author.getName();

    }

}

public class BookStore {

    public static void saveBooksToFile() {

        Person author1 = new Person("Ahmed Ali");

        Person author2 = new Person("Fatima Zahra");

        Person author3 = new Person("Omar Farooq");

        Person author4 = new Person("Aisha Bint Abu Bakr");

        Person author5 = new Person("Hassan Al-Basri");

        Book book1 = new Book("Introduction to Islamic Studies", "Al-Fikr", author1);

        Book book2 = new Book("The Life of Prophet Muhammad", "Islamic Book House", author2);

        Book book3 = new Book("The Fiqh of Islam", "Al-Madina Publishers", author3);

        Book book4 = new Book("Islamic Philosophy", "Al-Azhar Publications", author4);

        Book book5 = new Book("History of the Caliphate", "Al-Huda Publishers", author5);

        Book[] books = { book1, book2, book3, book4, book5 };

        try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("BookStore.ser"))) {

            for (Book book : books) {

                out.writeObject(book);

            }

            System.out.println("Books have been written to BookStore.ser.");

        } catch (IOException e) {

            System.out.println("An error occurred while writing");

            e.printStackTrace();

        }

    }

    // Method to display

    public static void displayBooks() {

        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("BookStore.ser"))) {

            System.out.println("Books available in BookStore:");

            Book book;

            while (true) {

                try {

                    book = (Book) in.readObject();  // casting

                    System.out.println(book);

                    System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                } catch (EOFException e) {

                    break;

                }

            }

        } catch (IOException | ClassNotFoundException e) {

            System.out.println("An error occurred while reading from the file.");

            e.printStackTrace();

        }

    }

    // Method to search a book

    public static void searchBookByName(String bookName) {

        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("BookStore.ser"))) {

            boolean bookFound = false;

            Book book;

            while (true) {

                try {

                    book = (Book) in.readObject();

                    if (book.getName().equalsIgnoreCase(bookName)) {

                        System.out.println(book);  // will call the to string of book

                        bookFound = true;

                        break;

                    }

                } catch (EOFException e) {

                    break;

                }

            }

            if (!bookFound) {

                System.out.println("No book found with the name: " + bookName);

            }

        } catch (IOException | ClassNotFoundException e) {

            System.out.println("An error occurred while reading from the file.");

            e.printStackTrace();

        }

    }

    public static void main(String[] args) {

        saveBooksToFile();

        displayBooks();

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the name of the book you want to search for:");

        String bookName = scanner.nextLine();

        // Search function

        searchBookByName(bookName);

        scanner.close();

    }

}

**Lab Task 04**

import java.io.\*;

import java.util.\*;

class Account implements Serializable {

    private String accountNumber;

    private String accountHolder;

    private double balance;

    public Account(String accountNumber, String accountHolder, double balance) {

        this.accountNumber = accountNumber;

        this.accountHolder = accountHolder;

        this.balance = balance;

    }

    public String getAccountNumber() {

        return accountNumber;

    }

    public String getAccountHolder() {

        return accountHolder;

    }

    public double getBalance() {

        return balance;

    }

    public void deposit(double amount) {

        balance += amount;

    }

    public boolean withdraw(double amount) {

        if (balance >= amount) {

            balance -= amount;

            return true;

        }

        return false;

    }

    public boolean transfer(Account destination, double amount) {

        if (this.withdraw(amount)) {

            destination.deposit(amount);

            return true;

        }

        return false;

    }

    @Override

    public String toString() {

        return "Account Number: " + accountNumber + "\nAccount Holder: " + accountHolder + "\nBalance: " + balance;

    }

}

public class AtmSystem {

    private static List<Account> accounts = new ArrayList<>();

    public static void saveAccountsToFile() {

        try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("Accounts.ser"))) {

            for (Account account : accounts) {

                out.writeObject(account);

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    public static void getAccountsFromFile() {

        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("Accounts.ser"))) {

            Account account;

            while (true) {

                try {

                    account = (Account) in.readObject();

                    accounts.add(account);

                } catch (EOFException e) {

                    break;

                }

            }

        } catch (IOException | ClassNotFoundException e) {

            e.printStackTrace();

        }

    }

    public static Account getAccountByNumber(String accountNumber) {

        for (Account account : accounts) {

            if (account.getAccountNumber().equals(accountNumber)) {

                return account;

            }

        }

        return null;

    }

    public static void withdraw() {

        Scanner scanner = new Scanner(System.in);

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

        String accountNumber = scanner.nextLine();

        Account account = getAccountByNumber(accountNumber);

        if (account != null) {

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

            double amount = scanner.nextDouble();

            if (account.withdraw(amount)) {

                System.out.println("Withdrawal successful.");

                saveAccountsToFile();

            } else {

                System.out.println("Insufficient balance.");

            }

        } else {

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

        }

    }

    public static void deposit() {

        Scanner scanner = new Scanner(System.in);

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

        String accountNumber = scanner.nextLine();

        Account account = getAccountByNumber(accountNumber);

        if (account != null) {

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

            double amount = scanner.nextDouble();

            account.deposit(amount);

            System.out.println("Deposit successful.");

            saveAccountsToFile();

        } else {

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

        }

    }

    public static void transfer() {

        Scanner scanner = new Scanner(System.in);

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

        String sourceAccountNumber = scanner.nextLine();

        Account sourceAccount = getAccountByNumber(sourceAccountNumber);

        if (sourceAccount != null) {

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

            String destinationAccountNumber = scanner.nextLine();

            Account destinationAccount = getAccountByNumber(destinationAccountNumber);

            if (destinationAccount != null) {

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

                double amount = scanner.nextDouble();

                if (sourceAccount.transfer(destinationAccount, amount)) {

                    System.out.println("Transfer successful.");

                    saveAccountsToFile();

                } else {

                    System.out.println("Insufficient balance.");

                }

            } else {

                System.out.println("receivers account not found.");

            }

        } else {

            System.out.println("Source account not found.");

        }

    }

    public static void balanceInquiry() {

        Scanner scanner = new Scanner(System.in);

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

        String accountNumber = scanner.nextLine();

        Account account = getAccountByNumber(accountNumber);

        if (account != null) {

            System.out.println("Balance: " + account.getBalance());

        } else {

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

        }

    }

    public static void main(String[] args) {

        getAccountsFromFile();

        if (accounts.isEmpty()) {

            accounts.add(new Account("1031", "Ahmed Ali", 1000));

            accounts.add(new Account("10562", "Fatima ", 15040));

            accounts.add(new Account("1073", "Omar Farooq", 2000));

            accounts.add(new Account("1074", "Aisha Bint Abu Bakr", 25200));

            accounts.add(new Account("1056", "Hassan Al-Basri", 3000));

            accounts.add(new Account("6106", "Ali Ibn Abi Talib", 3500));

            accounts.add(new Account("1407", "Umm Salama", 4000));

            accounts.add(new Account("108", "Abu Huraira", 4500));

            accounts.add(new Account("1093", "Bilal ", 5000));

            accounts.add(new Account("1170", "ana", 25500));

            saveAccountsToFile();

        }

        Scanner scanner = new Scanner(System.in);

        int choice;

        do {

            System.out.println("\nATM System");

            System.out.println("1. Withdraw");

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

            System.out.println("3. Transfer");

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

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

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

            choice = scanner.nextInt();

            switch (choice) {

                case 1:

                    withdraw();

                    break;

                case 2:

                    deposit();

                    break;

                case 3:

                    transfer();

                    break;

                case 4:

                    balanceInquiry();

                    break;

                case 5:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice, try again.");

            }

        } while (choice != 5);

    }

}