

Лабораторна робота №8

ОСНОВИ ВВЕДЕННЯ/ВИВЕДЕННЯ JAVA SE

Мета: Оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

ВИМОГИ

Розробник:

- Косінов Владислав Дмитрович;
- КІТ-1206;
- Варіант №8.

Загальне завдання:

- 1) Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
- 2) Забороняється використання стандартного протокола серіалізації.
- 3) Продемонструвати використання моделі Long Term Persistence.
- 4) Забезпечити діалог з користувачем у вигляді простого текстового меню.
- 5) При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

Прикладна галузь:

Автостанція.

Запис в розкладі: номер рейсу; час відправлення; дні тижня; кількість вільних місць; маршрут - необмежений набір значень у вигляді "назва станції, час прибуття".

ОПИС ПРОГРАМИ

`BusStation[] Entries;` `// масив об'єктів, що визначає сутність записів у розкладі`

`String key;` `// змінна, яка відповідає за вибір у меню користувача`

Ієрархія та структура класів:

class Main – точка входу в програму.

class BusStation - клас, який успадковується та реалізує прикладну галузь.

ТЕКСТ ПРОГРАМИ

Текст файлу **Main**:

```
package ua.khpi.oop.kosinov08;

import ua.khpi.oop.kosinov08.BusStation;

import java.beans.XMLDecoder;
import java.beans.XMLEncoder;
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

/**
 * Entry class.
 * Contains entry point of a program.
 *
 * @author kosinov-vladyslav
 */

public class Main {
    private static final BufferedReader reader = new BufferedReader(
        new InputStreamReader(System.in));
    private static BusStation[] Entries;
    private static String key = "";

    /**
```

```

    * Main method - entry point of a program.
    * Contains user menu for library control.
    *
    * @param args - command line parameters
    * @throws IOException - if there is any unresolved input/output
    */
    public static void main(String[] args) throws IOException {

        /* Processes user's choices. */
        while (!key.equals("5")) {
            /* User menu output. */
            mainMenu();
            key = reader.readLine();
            switch (key) {
                case "1" :
                    System.out.print("\nВведите значение (кол-
во записей в расписании): ");
                    Entries = new BusStation[Integer.parseInt(reader.readLine())];
                    break;

                case "2" :
                    init();
                    break;
                case "3" :
                    save();
                    break;
                case "4" :
                    load();
                    break;
                case "5" :
                    System.out.println("Выход...");
                    break;
                default : break;
            }
        }
        reader.close();
    }

    private static void mainMenu() {
        System.out.println("\n---МЕНЮ---");
        System.out.println("1. Определить кол-во элементов массива");
        System.out.println("2. Назначить элементы массива");
        System.out.println("3. Сохранить массив на длительный срок");
        System.out.println("4. Извлечь массив из файла");
        System.out.println("5. Выход");
        System.out.print("Введите свой выбор: ");
    }
}

```

```

private static void init() throws IOException {
    String init;
    Scanner in = new Scanner(System.in);
    int number = 0;
    for (int i = 0; i < (Entries != null ? Entries.length : 0); i++) {
        System.out.print("\nЗапись в расписании #" + (i + 1) + "\n");
        Entries[i] = new BusStation();
        System.out.print("Номер рейса: ");
        number = in.nextInt();
        Entries[i].setFlightNumber(number);
        System.out.print("Время отправления: ");
        init = reader.readLine();
        Entries[i].setDepartureTime(init);
        System.out.print("День недели: ");
        init = reader.readLine();
        Entries[i].setDayOfTheWeek(init);
        System.out.print("Кол-во свободных мест: ");
        number = in.nextInt();
        Entries[i].setNumberOfFreeSeats(number);
        System.out.print("Маршрут: \n");
        System.out.print("Введите кол-во станций входящих в маршрут: ");
        number = in.nextInt();
        Entries[i].enterRoute(number);
    }
    System.out.println();
}

private static void save() throws IOException {
    ArrayList<String> hightFolder = new ArrayList<>();
    String pick;
    StringBuilder direct = new StringBuilder("X:\\");
    key = "";
    while(!key.equals("1") && !key.equals("4")) {
        System.out.println("\nТекущий каталог: " + direct.toString());
        System.out.println("1. Сохранить XML-файл в текущем каталоге.");
        System.out.println("2. Перейти в каталог.");
        System.out.println("3. Назад.");
        System.out.println("4. Отмена.");
        System.out.print("Введите ваш выбор: ");
        key = reader.readLine();
        System.out.println();
        switch (key) {
            case "1":
                String Direct = direct.toString();
                FileOutputStream fos = new FileOutputStream(Direct + "\\JavaBea
ns.xml");

```

```

        BufferedOutputStream bos = new BufferedOutputStream(fos);
        XMLEncoder xmlEncoder = new XMLEncoder(bos);
        xmlEncoder.writeObject(Entries);
        xmlEncoder.close();
        break;
    case "2":
        int helpCount = 1;
        File directory = new File(direct.toString());
        File[] array = directory.listFiles();
        ArrayList<File> folders = new ArrayList<>();
        if (array != null) {
            for (File it : array) {
                if (it.isDirectory()) {
                    folders.add(it);
                }
            }
            for (File it : folders) {
                System.out.println(helpCount++ + ". " + it.getName());
            }
            System.out.println();
            System.out.print("Выберите каталог: ");
            pick = reader.readLine();
            hightFolder.add(direct.toString());
            direct.append(folders.get(Integer.parseInt(pick) - 1).getNa
me()).append("\\");
        }
        break;
    case "3":
        if (direct.length() <= 3) {
            System.out.println("Вы уже в корневом каталоге!");
            break;
        }
        direct.delete(hightFolder.get(hightFolder.size() - 1).length(),
            direct.length());
        hightFolder.remove(hightFolder.size() - 1);
        break;
    default: break;
}
}
}

private static void load() throws IOException {
    System.out.println("Введите каталог для извлечения массива в кодировке XML:");
    String dirToExtract = reader.readLine();
    FileInputStream fis = new FileInputStream(dirToExtract + ".xml");
    BufferedInputStream bis = new BufferedInputStream(fis);

```

```

XMLDecoder xmlDecoder = new XMLDecoder(bis);
BusStation[] getEntries = (BusStation[]) xmlDecoder.readObject();
for (int i = 0; i < getEntries.length; i++) {
    System.out.format("Рейс №%d\n", i + 1);
    System.out.println(getEntries[i].toString() + "\n");
}
xmlDecoder.close();
}
}

```

Текст файлу **BusStation**

```

package ua.khpi.oop.kosinov08;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

/**
 * Bus Station class.
 * Class defines the entity of a special task.
 * Schedule entries that contains of flightNumber,departureTime etc.
 */
public class BusStation {
    private int flightNumber;    // A flight number in Schedule
    private String departureTime; // departure time of the bus
    private String dayOfTheWeek; // day of the week when the bus travels
    private int numberOfFreeSeats; // count of free seats in bus
    private Flight[] route;      // The bus route (name of station, arrival time)
    int size = 0;

    /**
     * The setters of our information variables
     */
}

```

```

public void setFlightNumber(int flightNumber) {
    this.flightNumber = flightNumber;
}

public void setDepartureTime(String departureTime) {
    this.departureTime = departureTime;
}

public void setDayOfTheWeek(String dayOfTheWeek) {
    this.dayOfTheWeek = dayOfTheWeek;
}

public void setNumberOfFreeSeats(int numberOfFreeSeats) {
    this.numberOfFreeSeats = numberOfFreeSeats;
}

public void setRoute(Flight[] route) {
    this.route = route;
}

/**
 * Adding stations in the route.
 * @param stationNum - the number of stations
 * @throws IOException - if there is any unresolved input/output
 */
public void enterRoute(int stationNum) throws IOException {
    BufferedReader reader = new BufferedReader(
        new InputStreamReader(System.in));
    this.route = new Flight[stationNum];
    System.out.println("Введите " + stationNum + " станций");
    String station;
    String time;
    for (int i = 0; i < stationNum; i++) {
        System.out.print((i + 1) + ".\n");
        System.out.print("Введите название: ");
        station = reader.readLine();
        System.out.print("Введите время прибытия: ");
        time = reader.readLine();
        this.route[i] = new Flight(station, time);
    }
}

/**
 * The getters of our information variables
 */
public int getFlightNumber() {
    return flightNumber;
}

```

```

    }

    public int getNumberOfFreeSeats() {
        return numberOfFreeSeats;
    }

    public String getDepartureTime() {
        return departureTime;
    }

    public String getDayOfTheWeek() {
        return dayOfTheWeek;
    }

    public Flight getRouteOnIndex(int index) {
        return route[index];
    }

    public Flight[] getRoute (){
        return route;
    }

    /**
     Default constructor
    */
    public BusStation() {
        flightNumber = 0;
        departureTime = null;
        dayOfTheWeek = null;
        numberOfFreeSeats = 0;
        route = null;
    }

    @Override
    public String toString() {
        return "Номер рейса: " + flightNumber + "\n" +
            "Время отправления: " + departureTime + '\n' +
            "День недели: " + dayOfTheWeek + '\n' +
            "Кол-во свободных мест: " + numberOfFreeSeats + '\n' +
            "Маршрут: " + Arrays.toString(route);
    }
}

```

ВАРІАНТИ ВИКОРИСТАННЯ

Перше зображення представляє ініціалізацію масиву об'єктів домену (рис. 8.1а). Другий показує використання моделі Long Term Persistence (одиниці довгострокової стійкості) (рис. 8.1б), переміщення по каталогах (рис.8.1в), (рис. 8.2г), (рис. 8.1д) та (рис. 8.1е). Відновлення масиву об'єктів за заданою директорією (рис. 8.1є).

```
---МЕНЮ---
1. Определить кол-во элементов массива
2. Назначить элементы массива
3. Сохранить массив на длительный срок
4. Извлечь массив из файла
5. Выход
Введите свой выбор: 2

Запись в расписании #1
Номер рейса: 339
Время отправления: 12:00
День недели: Суббота
Кол-во свободных мест: 12
Маршрут:
Введите кол-во станций входящих в маршрут: 2
Введите 2 станций
1.
Введите название: Харьков
Введите время прибытия: 12:00
2.
Введите название: Киев
Введите время прибытия: 16:00
```

а)

```

---МЕНЮ---
1. Определить кол-во элементов массива
2. Назначить элементы массива
3. Сохранить массив на длительный срок
4. Извлечь массив из файла
5. Выход
Введите свой выбор: 3

Текущий каталог: X:\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: |

```

б)

```

Текущий каталог: X:\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 2

1. $RECYCLE.BIN
2. AMAZING
3. APheIp
4. BD
5. CashVisual
6. Counter-Strike 1.6 Original
7. eclipse
8. FlowChart
9. Games
10. GitBash
11. GitRepos
12. IntelliJ IDEA Community Edition 2020.2.1
13. Java Projects
14. Microsoft VS Code
15. MSOCache
16. OFFICE
17. Photoshop

```

в)

```

30. СП

Выберите каталог: 13

Текущий каталог: X:\Java Projects\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: |

```

г)

```

Текущий каталог: X:\Java Projects\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 2

1. .git
2. .idea
3. .settings
4. doc
5. out
6. src

Выберите каталог: 4

Текущий каталог: X:\Java Projects\src\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 2

1. ua

Выберите каталог: |

```

д)

```

Текущий каталог: X:\Java Projects\src\va\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 3

Текущий каталог: X:\Java Projects\src\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 3

Текущий каталог: X:\Java Projects\
1. Сохранить XML-файл в текущем каталоге.
2. Перейти в каталог.
3. Назад.
4. Отмена.
Введите ваш выбор: 1

```

е)

```

---МЕНЮ---
1. Определить кол-во элементов массива
2. Назначить элементы массива
3. Сохранить массив на длительный срок
4. Извлечь массив из файла
5. Выход
Введите свой выбор: 4
Введите каталог для извлечения массива в кодировке XML:
JavaBeans.xml
Рейс №1
Номер рейса: 339
Время отправления: 12:00
День недели: Суббота
Кол-во свободных мест: 12
Маршрут: [Харьков 12:00, Киев 16:00]

```

є)

Рисунок 8.1 – Результат виконання завдання

ВИСНОВОК

Під час виконання лабораторної роботи було набуто навичок управління введенням/виведенням даних з використанням класів платформи Java SE. Забезпечено можливість збереження і відновлення масиву об'єктів. Реалізовано, використання Long Term Persistence, який дає новий спосіб довгострокового збереження об'єктів домену та отримання їх у будь-який час. Масив об'єктів домену збережено та повністю отримано з XML-файлу.