

## **Лабораторна робота №8**

### **Основи введення/виведення Java SE**

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

#### **1 ВИМОГИ**

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.

#### **1.1Розробник**

- П.І.Б : Наймитенко С.І.

- Група: КІТ-119а

- Варіант 15

## 1.3 Задача

Автомобіль: марка, рік випуску, технічні характеристики у вигляді "параметр - значення" (серед яких витрата палива в міському і заміському циклі), технічний стан, ціна.

## 2 ОПИС ПРОГРАМИ

### 2.2 Ієрархія та структура класів

Було створено 3 класи:

- `Public class CarsShop` – клас, що містить необхідні поля та методи прикладної області “Автосалон”.
- `public class Main` – містить лише метод `main`.
- `Public class Container` – клас що містить в собі масив об’єктів класу `CarsShop`

### Важливі фрагменти програми

#### Клас `CarsShop`

```
package ua.khpi.oop.naimytenko08;

import java.util.LinkedHashMap;
import java.util.stream.Collectors;

public class CarsShop {

    private String car_brand;
    private int year_create;
    private String specifications;
    private int price;
    private LinkedHashMap<String, Integer> city_fuel_consumption = new
LinkedHashMap<String, Integer>();
    private LinkedHashMap<String, Integer> fuel_consumption_outside_the_city = new
LinkedHashMap<String, Integer>();

    public String getCar_brand() {
        return car_brand;
    }

    public void setCar_brand(String car_brand) {
```

```

        this.car_brand = car_brand;
    }

    public int getYear_create() {
        return year_create;
    }

    public void setYear_create(int year_create) {
        this.year_create = year_create;
    }

    public String getSpecifications() {
        return specifications;
    }

    public void setSpecifications(String specifications) {
        this.specifications = specifications;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public LinkedHashMap<String, Integer> getCity_fuel_consumption() {
        return city_fuel_consumption;
    }

    public void setCity_fuel_consumption(LinkedHashMap<String, Integer>
city_fuel_consumption) {
        this.city_fuel_consumption = city_fuel_consumption;
    }

    public LinkedHashMap<String, Integer> getFuel_consumption_outside_the_city() {
        return fuel_consumption_outside_the_city;
    }

    public void setFuel_consumption_outside_the_city(LinkedHashMap<String,
Integer> fuel_consumption_outside_the_city) {
        this.fuel_consumption_outside_the_city =
fuel_consumption_outside_the_city;
    }

    public String toString_gas_mileage() {
        String string = city_fuel_consumption.entrySet().stream()
            .map(e -> e.getKey() + e.getValue())
            .collect(Collectors.joining());
        string += fuel_consumption_outside_the_city.entrySet().stream()
            .map(e -> e.getKey() + e.getValue())
            .collect(Collectors.joining());
        return string;
    }

    public CarsShop(String car_brand, int year_create, String specifications, int
price, int city_fuel_consumption, int fuel_consumption_outside_the_city) {
        super();
        this.car_brand = car_brand;
        this.year_create = year_create;
    }

```

```

        this.specifications = specifications;
        this.price = price;
        this.city_fuel_consumption.put(" расход топлива по городу -
",city_fuel_consumption);
        this.fuel_consumption_outside_the_city.put(" расход топлива за городом
- ",fuel_consumption_outside_the_city);
    }

    public CarsShop() {
        super();
        // TODO Auto-generated constructor stub
    }

    public void Show()
    {
        System.out.println("Марка автомобиля - "+ car_brand +"\r\n" +
        "Цена автомобиля "+ price + "\r\n" +
        "Технічні характеристики " + toString_gas_mileage()
+
        "\nГод выпуска автомобиля "+ year_create + "\r\n" +
        "Технічний стан "+ specifications + "\r\n" );
    }
}

```

## Клас Container

```

package ua.khpi.oop.naimytenko08;

import java.util.LinkedHashMap;
import java.util.Scanner;

public class Container {

    CarsShop array[] = new CarsShop[3];

    public void add()
    {
        int size = array.length;
        Scanner scan = new Scanner(System.in);
        System.out.println("Введите марку автомобиля");
        String car_brand = scan.nextLine();
        System.out.println("Опишите техническое состояние автомобиля");
        String specifications = scan.nextLine();
        System.out.println("Введите год выпуска авто");
        int year_create = scan.nextInt();
        System.out.println("Введите цену авто");
        int price = scan.nextInt();
        System.out.println("Введите расход топлива по городу");
        int city_fuel_consumption = scan.nextInt();
        System.out.println("Введите расход топлива вне города");
        int fuel_consumption_outside_the_city = scan.nextInt();
        CarsShop new_el = new
CarsShop(car_brand,year_create,specifications,price,city_fuel_consumption,
fuel_consumption_outside_the_city);

        CarsShop new_arr[] = new CarsShop[++size];
        for(int i=0;i<size-1;i++)

```

```

        {
            new_arr[i]=array[i];
        }
        new_arr[size-1] = new_el;
        array = new_arr;
    }

```

```

public void remove()
{
    int size = array.length;
    if(size!=0)
    {
        CarsShop new_arr[] = new CarsShop[--size];
        for(int i=0;i<size;i++)
        {
            new_arr[i]=array[i];
        }
        array = new_arr;
    }
    else
    {
        System.out.println("Ваш массив пуст, удалять нечего");
    }
}

public void Show()
{
    for (CarsShop carsShop : array) {
        carsShop.Show();
    }
}

}

```

### 3 ВИСНОВКИ

#### Результат роботи програми:

Доступні команди

- 1 - серіалізація -
- 2 - десіалізація -
- 3 - виведення масиву на екран -
- 4 - додавання елементу в кінець -
- 5 - видалення елементу з початку -
- 0 - завершення програми -

Введіть команду:

3

Марка автомобіля - mazda

Цена автомобіля 400000

Технічні характеристики: витрата палива в міському циклі - 25 витрата палива в заміському циклі - 20

Год выпуска автомобіля 2020

Технічний стан Хорошее состояние, требуется косметический ремонт

Марка автомобіля - ВАЗ

Цена автомобіля 30000

Технічні характеристики: витрата палива в міському циклі - 15 витрата палива в заміському циклі - 10

Год выпуска автомобіля 1980

Технічний стан Плохое состояние, после ДТП

Марка автомобіля - Rolls-royce

Цена автомобіля 700000

Технічні характеристики: витрата палива в міському циклі - 40 витрата палива в заміському циклі - 35

Год выпуска автомобіля 2013

Технічний стан Отличное состояние

Доступні команди

- 1 - серіалізація -
- 2 - десіалізація -
- 3 - виведення масиву на екран -
- 4 - додавання елементу в кінець -
- 5 - видалення елементу з початку -
- 0 - завершення програми -

Введіть команду:

1

Введите название для вашего XML файла для сохранения данных

xml

Название вашего XML файла - xml.xml

Текущий путь: C:\Users\fox\eclipse-workspace\Java-oop

Список всех файлов и каталогов в указанном каталоге:

- 1) .git
- 2) .gitignore
- 3) naimytenko sergiy
- 4) xml.xml
- 5) Выберите этот вариант, чтобы сохранить файл в этом каталоге.
- 6) выберите этот вариант, если хотите подняться по тропе

Введите номер каталога, в который вы хотите сохранить файл, или выберите этот каталог:

5

Serialization successful

Марка автомобіля - ВАЗ  
Цена автомобіля 30000  
Технічні характеристики: витрата палива в міському циклі - 15 витрата палива в заміському циклі - 10  
Год выпуска автомобіля 1980  
Технічний стан Плохое состояние, после ДТП

Марка автомобіля - Rolls-royce  
Цена автомобіля 7000000  
Технічні характеристики: витрата палива в міському циклі - 40 витрата палива в заміському циклі - 35  
Год выпуска автомобіля 2013  
Технічний стан Отличное состояние

Марка автомобіля - 1  
Цена автомобіля 4  
Технічні характеристики: витрата палива в міському циклі - 5 витрата палива в заміському циклі - 6  
Год выпуска автомобіля 3  
Технічний стан 2

Доступні команди  
1 - серіалізація -  
2 - десіалізація -  
3 - виведення масиву на екран -  
4 - додавання елементу в кінець -  
5 - видалення елементу з початку -  
0 - завершення програми -

Введіть команду:

2

Имя текущего файла XML - xml.xml  
Текущий путь: C:\Users\fox\eclipse-workspace\Java-oop  
Список всех файлов и каталогов в указанном каталоге:

- 1) .git
- 2) .gitignore
- 3) naumytenko sergiy
- 4) xml.xml

5) Выберите этот вариант, если хотите подняться по тропе

Введите номер каталога для перемещения туда или XML-файл, который вы хотите прочитать:

4

Марка автомобіля - mazda  
Цена автомобіля 400000  
Технічні характеристики: витрата палива в міському циклі - 20 витрата палива в заміському циклі - 20  
Год выпуска автомобіля 2020  
Технічний стан Хорошее состояние, требуется косметический ремонт

Марка автомобіля - ВАЗ  
Цена автомобіля 30000  
Технічні характеристики: витрата палива в міському циклі - 10 витрата палива в заміському циклі - 10  
Год выпуска автомобіля 1980  
Технічний стан Плохое состояние, после ДТП

Марка автомобіля - Rolls-royce  
Цена автомобіля 7000000  
Технічні характеристики: витрата палива в міському циклі - 35 витрата палива в заміському циклі - 35  
Год выпуска автомобіля 2013  
Технічний стан Отличное состояние

## ВИСНОВКИ

У результаті виконання лабораторної роботи було набуто навичок роботи з основами введення/виведення у середовищі Java Eclipse.