

Лабораторна робота №7

Об'єктно-орієнтована декомпозиція

Мета: Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.

1 ВИМОГИ

1. Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно прикладної задачі - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

1.1 Розробник

- П.І.Б : Наймитенко С.І.

- Група: КІТ-119а

- Варіант 15

1.3 Задача

Автосалон

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`.

Важливі фрагменти програми

Клас `CarsShop`

```
package ua.khpi.oop.naimytenko07;

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 автомобіля - "+ 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" );
    }
}

```

3 ВИСНОВКИ

Результат роботи програми:

```

Марка автомобіля - mazda
Цена автомобиля 400000
Технічні характеристики: витрата палива в міському циклі - 25 витрата палива в заміському циклі - 20
Год выпуска автомобиля 2020
Технічний стан Хорошее состояние, требуется косметический ремонт

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

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

```

ВИСНОВКИ

У результаті виконання лабораторної роботи було набуто навичок роботи з domain-об'єктами; було набуто навичок з об'єктно-орієнтованою декомпозицією у середовищі JavaEclipse.