**Лабораторна робота №7**

[**Об'єктно-орієнтована декомпозиція**](https://oop-khpi.gitlab.io/#tasks)

**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](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/#_4) - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

**1.1Розробник**

- 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> сity\_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> getСity\_fuel\_consumption() {**

**return сity\_fuel\_consumption;**

**}**

**public void setСity\_fuel\_consumption(LinkedHashMap<String, Integer> сity\_fuel\_consumption) {**

**this.сity\_fuel\_consumption = сity\_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 = сity\_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.сity\_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" +**

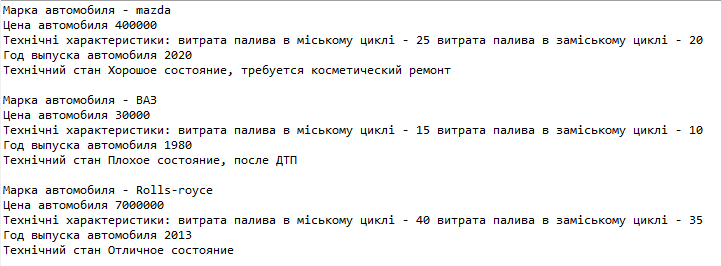
**"Технічний стан "+ specifications + "\r\n" );**

**}**

**}**

**3 ВИСНОВКИ**

Результат роботи програми:



**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.