Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

ЗВІТ

Про виконання лабораторної роботи №7

“Об`єктно-орієнтована декомпозиція”

Виконавець:

студент гр. КІТ-120В

Олексієнко Микита

Харків 2021

Лабораторна робота №7.

**Об`єктно-орієнтована декомпозиція**

**Мета**

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

**1.Вимоги**

**1.1 Розробник:**

Олексієнко Микита Віталійович

студент групи КІТ-120В

15 варіант

**1.2 Загальне завдання**

Реалізувати програму відповідно до індивідуального завдання.

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.

**1.3 Прикладні задачі**

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**

**2.1 Засоби ООП:**

У лабороторного роботі 7 , я використовую клас Program, який э головним класом та в якому безпосередньо оголошую приватні поля і публічні статичні методи. Також я оголосив клас Auto, який має всі перераховані вище поля та методи обробки полей.

**2.2 Ієрархія та структура класів:**

* Program.java
* Auto.java

**2.3 Важливі фрагменти програми:**

package ua.khpi.oop.alekseenko07;  
  
public class Auto implements java.io.Serializable {  
 private String model;  
 private int releaseYear;  
  
 private int urbanFuel;  
 private int suburbanFuel;  
  
 private boolean technicalCondition;  
 private int price;  
  
 // КОНСТРУКТОР ------------  
  
 public Auto () {  
 model = "";  
 releaseYear = 0;  
 urbanFuel = 0;  
 suburbanFuel = 0;  
 technicalCondition = false;  
 price = 0;  
 }  
  
 // МЕТОДЫ ДОСТУПА ---------  
  
 public String getModel () {  
 return model;  
 }  
 public void setModel ( String model ) {  
 this.model = model;  
 }  
  
 // ------------------------  
  
 public int getReleaseYear () {  
 return releaseYear;  
 }  
 public void setReleaseYear ( int releaseYear ) {  
 this.releaseYear = releaseYear;  
 }  
  
 // ------------------------  
  
 public int getUrbanFuel () {  
 return urbanFuel;  
 }  
 public void setUrbanFuel ( int urbanFuel ) {  
 this.urbanFuel = urbanFuel;  
 }  
  
 // ------------------------  
  
 public int getSubUrbanFuel () {  
 return suburbanFuel;  
 }  
 public void setSubUrbanFuel ( int suburbanFuel ) {  
 this.suburbanFuel = suburbanFuel;  
 }  
  
 // ------------------------  
  
 public boolean getTechnicalCondition () {  
 return technicalCondition;  
 }  
 public void setTechnicalCondition ( boolean technicalCondition ) {  
 this.technicalCondition = technicalCondition;  
 }  
  
 // ------------------------  
  
 public int getPrice () {  
 return price;  
 }  
 public void setPrice ( int price ) {  
 this.price = price;  
 }  
  
 // ------------------------  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) {  
 return true;  
 }  
 if (o == null || getClass() != o.getClass()) {  
 return false;  
 }  
  
 Auto that = (Auto) o; // приводим объект к классу  
  
 return model.equals(that.model) &&  
 releaseYear == that.releaseYear &&  
 urbanFuel == that.urbanFuel &&  
 suburbanFuel == that.suburbanFuel &&  
 technicalCondition == that.technicalCondition &&  
 price == that.price;  
 }  
  
 @Override  
 public String toString() {  
 return "Auto{" +  
 "model='" + model + '\'' +  
 ", releaseYear=" + releaseYear +  
 ", urbanFuel=" + urbanFuel +  
 ", suburbanFuel=" + suburbanFuel +  
 ", technicalCondition=" + technicalCondition +  
 ", price=" + price +  
 '}';  
 }  
  
 // JavaBeans - cтандарт написания классов, предусматривающий  
 // создание полей, методов доступа к ним и перезапись 3 стандратных методов  
 // обработки ( equals(), hashCode() и toString() )  
}

**3. Висновок:**

* Навчився використовувати об'єктно-орієнтований підхід для розробки об'єкта предметної (прикладної) галузі.
* Розробив прикладну галузь.