­­­­­­­­Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи № 4-5

з дисципліни

*Прикладне програмування*

на тему:

**“ Комплексна лабораторна робота ”**

Виконав: студент ОІ-24

**Живоронко Роман**

Прийняв: асистент каф. АСУ

Чорненький В.Я.

**Львів – 2024**

*Лабораторна робота №* *4-5*

**Завдання лабораторної роботи**

Створити консольну програму на мові програмування Java згідно до індивідуального завдання (ваш номер варіанту, згенерований випадковим чином, можна знайти у файлі «Варіанти завдань.pdf»). Саме завдання описане досить поверхово, якщо ви претендуєте на високу оцінку – його потрібно розширити. Наприклад, у завданні може бути сказано про «створити кілька квіток, з наявних квіток сформувати букет». Цілком логічно буде додати можливість видалити квітку, поміняти параметри існуючої квітки тощо. Оцінка за лабораторну роботу залежатиме і від глибини пропрацювання предметної області.

Окрім цього, програма повинна відповідати таким додатковим вимогам:

• Використовувати можливості ООП: класи, успадкування, поліморфізм, інкапсуляцію.

• Кожний клас повинен мати назву, яка повністю описує його суть, і інформативний склад.

• Успадкування потрібно використовувати тільки тоді, коли воно має сенс.

• При записі програми потрібно використовувати домовленості щодо оформлення коду java code convention.

• Класи потрібно грамотно розкласти по пакетах.

• Для збереження параметрів ініціалізації потрібно використовувати файли.

• Повинне бути консольне меню для вибору різних дій, наприклад: «завантажити з файлу», «довідка», «вихід» тощо.

Етапи виконання завдання (кожен етап вважається окремою лабораторною роботою):

1. Уважно пропрацюйте предметну область згідно вашого варіанту завдання. Опишіть технологічне завдання (який функціонал матиме ваша програма). Створіть Use Case та Class діаграми для вашого завдання.

2. Реалізуйте консольне меню до вашого завдання з використанням паттерну «Команда». На цьому етапі не потрібно реалізовувати функціонал конкретних пунктів меню – тільки створити «кістяк» вашої програми.

3. Реалізуйте функціонал програми. Обов’язково зверніть увагу на додаткові вимоги до програми (описані вище).

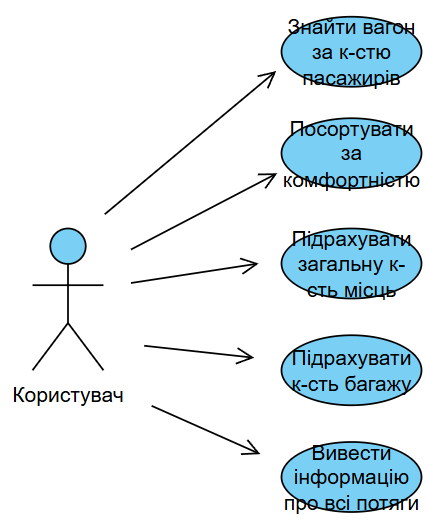
4. Покрийте вашу програму юніт тестами (coverage >=90%).

5. Додайте логгер з логуванням основних дій та виняткових ситуацій. При цьому далі логу мають зберігатися у файлі, а критичні помилки – додатково надсилатися на e-mail

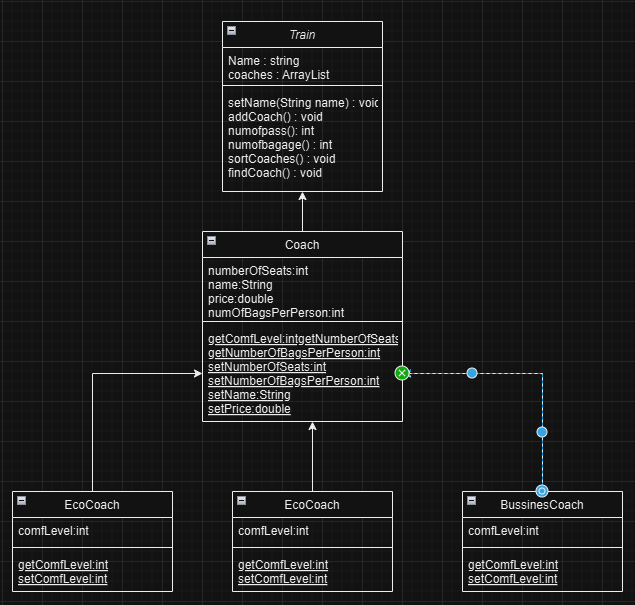
**Варіант 8**

Транспорт. Визначити ієрархію рухомого складу залізничного транспорту. Створити пасажирський потяг. Підрахувати загальну кількість пасажирів та багажу. Здійснити сортування вагонів поїзда на основі рівня комфортності. Знайти вагони в поїзді, що відповідають заданому діапазону параметрів кількості пасажирів.

**Use Case діаграма**



**Class діаграма**

****

**Код програми**

**Command.java**

package command;  
  
public interface Command {  
 void execute();  
}

**Menu.java**

package Menu;  
  
import Train.Train;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
import Coaches.\*;  
public class Menu {  
 private Scanner scan = new Scanner(System.*in*);  
 private ArrayList<Train> trains = new ArrayList<>();  
 private Map<Integer, Command> commands = new HashMap<>();  
  
 public Menu() {  
 createTrain();  
 initializeCommands();  
 }  
  
 private void initializeCommands() {  
 commands.put(1, new FindCoachByPassCommand(trains, scan));  
 commands.put(2, new PrintTrainsCommand(trains));  
 commands.put(3, new SortCoachByComfortCommand(trains, scan));  
 commands.put(4, new CountNumOfPassengersCommand(trains, scan));  
 commands.put(5, new CountNumOfBaggageCommand(trains, scan));  
 }  
  
 private void createTrain() {  
 Train train1 = new Train("Зимноводівський експрес");  
 for(int i = 0;i<11;i++){  
 if(i == 5 || i == 10 || i == 2){  
 train1.addCoach(new MidleCoach());  
 continue;  
 }  
 train1.addCoach(new BussinesCoach());  
 }  
 train1.addCoach(new EcoCoach());  
 Train train2= new Train("Львівський експрес");  
 for(int i = 0;i<11;i++){  
 if(i == 3 || i == 1 || i == 7){  
 train2.addCoach(new BussinesCoach());  
 continue;  
 }  
 train2.addCoach(new MidleCoach());  
 }  
 train2.addCoach(new EcoCoach());  
 Train train3 = new Train("Солонківський поїзд");  
 for(int i = 0;i<11;i++){  
 if(i == 7 || i == 9 || i == 2){  
 train3.addCoach(new MidleCoach());  
 continue;  
 }  
 train3.addCoach(new EcoCoach());  
 }  
 train3.addCoach(new BussinesCoach());  
 Train train4 = new Train("Рівненьска електричка");  
 for(int i = 0;i<11;i++){  
 if(i == 2 || i == 1 || i == 9){  
 train4.addCoach(new EcoCoach());  
 continue;  
 }  
 train4.addCoach(new MidleCoach());  
 }  
 train4.addCoach(new BussinesCoach());  
 Train train5 = new Train("Х'юндай");  
 for(int i = 0;i<11;i++){  
 if(i == 5 || i == 10 || i == 2){  
 train5.addCoach(new MidleCoach());  
 continue;  
 }  
 train5.addCoach(new BussinesCoach());  
 }  
 train5.addCoach(new EcoCoach());  
 Train train6= new Train("Лондон - Зимна Вода");  
 for(int i = 0;i<11;i++){  
 if(i == 4 || i == 6 || i == 8){  
 train6.addCoach(new MidleCoach());  
 continue;  
 }  
 train6.addCoach(new BussinesCoach());  
 }  
 train6.addCoach(new EcoCoach());  
 trains.add(train1);  
 trains.add(train2);  
 trains.add(train3);  
 trains.add(train4);  
 trains.add(train5);  
 trains.add(train6);  
  
 }  
  
 private void showMenu() {  
 System.*out*.println("\n\t\t MENU");  
 System.*out*.println("1 - Знайти вагон за к-стю пасажирів");  
 System.*out*.println("2 - Вивести інформацію про всі потяги");  
 System.*out*.println("3 - Посортувати вагони потяга за КОМФОРТНІСТЮ");  
 System.*out*.println("4 - Підрахувати загальну к-сть місць у потязі");  
 System.*out*.println("5 - Підрахувати загальну місткість багажу");  
 System.*out*.println("6 - Вихід");  
 }  
  
 public void menu() {  
 int choice;  
 while (true) {  
 showMenu();  
 System.*out*.println("Введіть опцію");  
 choice = scan.nextInt();  
 if (choice == 6) break;  
  
 Command command = commands.get(choice);  
 if (command != null) {  
 command.execute();  
 } else {  
 System.*out*.println("Неправильний вибір!");  
 }  
 }  
 }  
}

**Main.java**

package Main;  
import Menu.\*;  
public class Main {  
 public static void main(String[] args) {  
Menu menu = new Menu();  
menu.menu();  
  
 }  
}

}

**Git репозиторій: https://github.com/zhyvoronkoRoman/LPNU\_PP\_Lab\_4-8**

**Висновок**

На цій лабораторній роботі я ознайомився з Use Case діаграмами та Class діаграмами. Опанував паттерн команда