Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №2

З дисципліни: «Кросплатформні засоби програмування»

На тему: «КЛАСИ ТА ПАКЕТИ»

Варіант – 2

Виконав: ст. гр. КІ-303

Гірняк О.Р.

Прийняв:

Олексів М.В.

Львів-2023

**Мета:** ознайомитися з процесом розробки класів та пакетів мовою Java.

ЗАВДАННЯ

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну

область згідно варіанту. Програма має задовольняти наступним вимогам:

• програма має розміщуватися в пакеті Група.Прізвище.Lab2;

• клас має містити мінімум 3 поля, що є об’єктами класів, які описують складові

частини предметної області;

• клас має містити кілька конструкторів та мінімум 10 методів;

• для тестування і демонстрації роботи розробленого класу розробити клас-драйвер;

• методи класу мають вести протокол своєї діяльності, що записується у файл;

• розробити механізм коректного завершення роботи з файлом (не надіятися на

метод finalize());

• програма має володіти коментарями, які дозволять автоматично згенерувати

документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленої програми.

3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.

**Варіант завдання**

2. Космічний корабель

**Хід роботи:**

Код:

**SpaceShip.java**

package lab2Hirniak304;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.time.LocalDateTime;

import java.io.BufferedReader;

import java.io.FileReader;

public class SpaceShip {

private String name;

private int speed;

private int passengerCount;

public SpaceShip(String name, int speed, int passengerCount) {

this.name = name;

this.speed = speed;

this.passengerCount = passengerCount;

}

public SpaceShip(String name, int speed) {

this(name, speed, 25);

}

public void sendToSpace() {

// Logic for sending the spaceship to space

logActivity("Spaceship '" + name + "' with speed " + speed + " has been sent to space.");

}

public void startManeuver(String maneuverName) {

logActivity("Spaceship '" + name + "' has started maneuver: " + maneuverName);

}

public void endManeuver(String maneuverName) {

logActivity("Spaceship '" + name + "' has completed maneuver: " + maneuverName);

}

public void changeName(String newName) {

logActivity("Spaceship's name has been changed to '" + newName + "'");

name = newName;

}

public void land() {

// Logic for landing the spaceship

logActivity("Spaceship '" + name + "' has landed.");

}

public void addPassengers(int count) {

passengerCount += count;

logActivity("Added " + count + " passengers to spaceship '" + name + "'.");

}

public void removePassengers(int count) {

if (count <= passengerCount) {

passengerCount -= count;

logActivity("Removed " + count + " passengers from spaceship '" + name + "'.");

} else {

logActivity("Error: Cannot remove more passengers than are on board.");

}

}

private void logActivity(String message) {

String logMessage = LocalDateTime.now() + " - " + message;

try (FileWriter file = new FileWriter("SpaceshipLog.txt", true);

PrintWriter writer = new PrintWriter(file)) {

writer.println(logMessage);

} catch (IOException e) {

System.err.println("Error writing to log file: " + e.getMessage());

}

}

public void checkFileWrite() {

try {

BufferedReader reader = new BufferedReader(new FileReader("SpaceshipLog.txt"));

reader.close();

} catch (IOException e) {

System.err.println("Error reading the file: " + e.getMessage());

}

}

}

**SpaceShipDriver.java**

**package** lab2Hirniak304;

**public** **class** SpaceShipDriver {

**public** **static** **void** main(String[] args) {

SpaceShip spaceship1 = **new** SpaceShip("Milyarder-1", 5000, 10);

SpaceShip spaceship2 = **new** SpaceShip("SuperNavigator", 8000);

spaceship1.sendToSpace();

spaceship2.addPassengers(5);

spaceship2.land();

spaceship2.removePassengers(2);

// Check file write for spaceship1

spaceship1.checkFileWrite();

// Check file write for spaceship2

spaceship2.checkFileWrite();

// Register start of a maneuver for spaceship1

spaceship1.startManeuver("Orbit Adjustment");

// Change the name of spaceship2

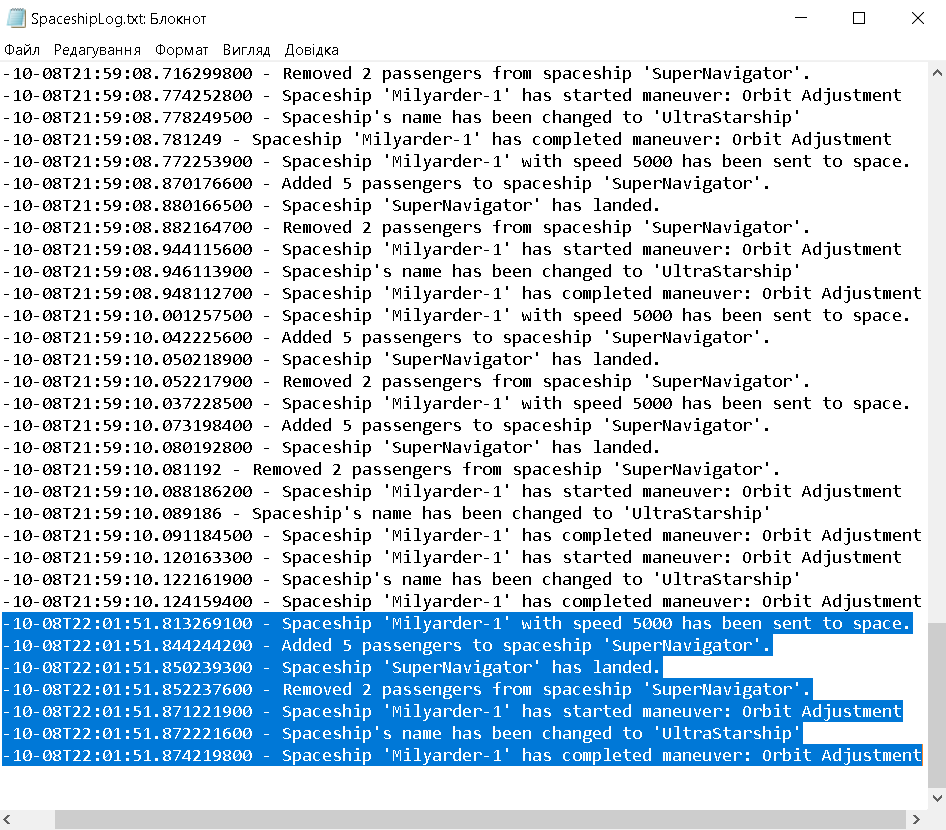
spaceship2.changeName("UltraStarship");

// Register end of a maneuver for spaceship1

spaceship1.endManeuver("Orbit Adjustment");

}

}

**Скрін виконання програми:** ****