

Міністерство освіти і науки України Національний
університет «Львівська політехніка»



Звіт

до лабораторної роботи №3

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Класи та пакети»

Варіант 15

Виконав:
Ст. групи КІ-34
Ольховик О.С.

Прийняв:
к.т.н., доцент
Іванов Ю.С.

Львів 2022

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.Lab3;
 - 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 розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

Варіант завдання: Корабель

Код роботи

Ship.java

```
package KI34.Olkhovyyk.Lab3;

import java.io.FileWriter;
import java.io.IOException;

public class Ship {
    private int maxSpeed;
    private int currentSpeed;
    private double maxFuelVolume;
    private double CurrentFuelVolume;
    private double fuelConsumption;
    private double passedDistance;
    public Ship() {
        this.currentSpeed = 0;
    }
    public Ship(int maxSpeed) {
        this.maxSpeed = maxSpeed;
    }

    public Ship(int maxSpeed, double CurrentFuelVolume) {
        this(maxSpeed);
        this.CurrentFuelVolume = CurrentFuelVolume;
    }

    public Ship(int maxSpeed, double CurrentFuelVolume, double fuelConsumption) {
```

```

        this(maxSpeed, CurrentFuelVolume);
        this.fuelConsumption = fuelConsumption;
    }

    public Ship(int maxSpeed, double CurrentFuelVolume, double fuelConsumption,
double maxFuelVolume) {
        this(maxSpeed, CurrentFuelVolume, fuelConsumption);
        this.maxFuelVolume = maxFuelVolume;
    }

    public int GetMaxSpeed() {
        return this.maxSpeed;
    }
    public int GetCurrentSpeed() {
        return this.currentSpeed;
    }
    public double GetMaxFuelVolume() {
        return this.maxFuelVolume;
    }
    public double GetCurrentFuelVolume() {
        return this.CurrentFuelVolume;
    }
    public double GetFuelConsumption() {
        return this.fuelConsumption;
    }
    public void SetCurrentSpeed(int speed) {
        if(speed > this.maxSpeed)
            this.currentSpeed = this.maxSpeed;
        this.currentSpeed = speed;
    }

    private void BurningFuel() {
        if(this.currentSpeed != this.maxSpeed)
        {
            double coef = this.maxSpeed / this.currentSpeed;
            this.fuelConsumption /= coef;
        }
        this.CurrentFuelVolume-= this.fuelConsumption;
        if(this.CurrentFuelVolume <= 0)
            this.CurrentFuelVolume = 0;
    }
    public void Run(int run) {
        while(run != 0) {
            BurningFuel();
            this.passedDistance += this.currentSpeed;
            if(this.CurrentFuelVolume == 0)
                break;
            run--;
        }
        SendData();
    }

    public String GetPassedDistance() {

```

```

        return String.valueOf(this.passedDistance*1852);
    }

    private void SendData() {

        try(FileWriter writer = new FileWriter("info.txt")){
            writer.write("Пройдена дистанція: " + GetPassedDistance() + "\n");
            writer.write("Максимальна швидкість човна: " + GetMaxSpeed() + "\n");
            writer.write("Залишилось пального: " + GetCurrentFuelVolume() + "\n");
            writer.flush();
            writer.close();
        }catch(IOException ex){

            System.out.println(ex.getMessage());
        }
    }
}

```

ShipApp.java

```

import KI34.Olkhovyk.Lab3.*;

import static java.lang.System.out;
public class ShipApp {

    public static void main(String[] args) {
        Ship ship = new Ship(15,50,5,75);
        out.println("Поточний об'єм палива - " + ship.GetCurrentFuelVolume());
        out.println("Пройдена відстань - " + ship.GetPassedDistance());

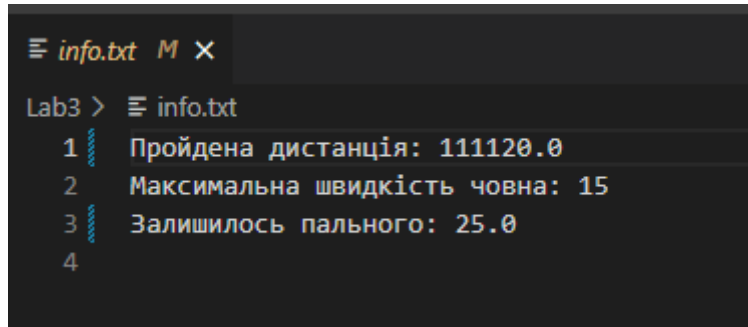
        ship.SetCurrentSpeed(12);
        ship.Run(5);

        out.println("Поточний об'єм палива - " + ship.GetCurrentFuelVolume());
        out.println("Пройдена відстань - " + ship.GetPassedDistance());
        out.println("Поточна швидкість - " + ship.GetCurrentSpeed());
    }
}

```

Результат виконання програми

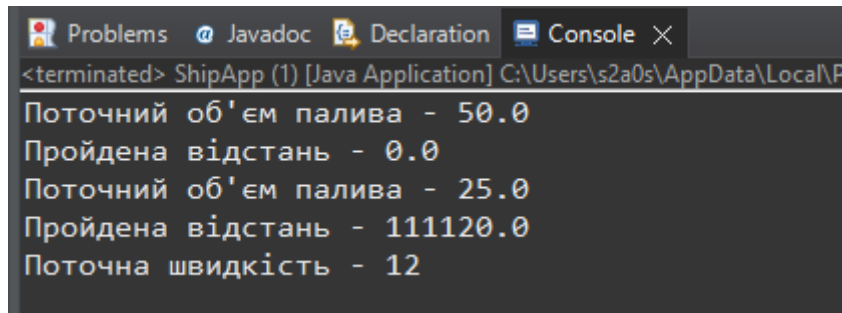
- Txt-файл з записаною інформацією:



A screenshot of a text editor window titled 'info.txt'. The editor shows a list of four items, each preceded by a blue wavy line icon. The items are: 1. Пройдена дистанція: 111120.0, 2. Максимальна швидкість човна: 15, 3. Залишилось пального: 25.0, and 4. (empty line).

```
1 Пройдена дистанція: 111120.0
2 Максимальна швидкість човна: 15
3 Залишилось пального: 25.0
4
```

- Консоль



A screenshot of a Java IDE console window. The title bar shows 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output shows the following text: <terminated> ShipApp (1) [Java Application] C:\Users\s2a0s\AppData\Local\... followed by five lines of status information in Ukrainian: 'Поточний об'єм палива - 50.0', 'Пройдена відстань - 0.0', 'Поточний об'єм палива - 25.0', 'Пройдена відстань - 111120.0', and 'Поточна швидкість - 12'.

```
<terminated> ShipApp (1) [Java Application] C:\Users\s2a0s\AppData\Local\...
Поточний об'єм палива - 50.0
Пройдена відстань - 0.0
Поточний об'єм палива - 25.0
Пройдена відстань - 111120.0
Поточна швидкість - 12
```

Відповіді на контрольні запитання:

1. `[public] class Назва Класу {`
 `[Методи]`
 `[Змінні]`
 `[Поля]`
 `[Конструктори]`
 `}`
2. `[СпецифікаторДоступу] Тип назваМетоду([параметри]) [throws класи] {`
 `[Тіло методу] [return [значення]];`
 `}`
3. `[СпецифікаторДоступу] [static] [final] Тип НазваПоля [=`
 `ПочатковеЗначення];`
4. Використати `[final]`, тобто/наприклад `private final int i;`
5. Ініціалізацію полів при створенні об'єкту можна здійснювати трьома способами:
 - у конструкторі;
 - явно при оголошенні поля;
 - у блоці ініціалізації (виконується перед виконанням конструктора).

Якщо поле не ініціалізується жодним з цих способів, то йому присвоюється значення за замовчуванням.

6. `[СпецифікаторДоступу] НазваКласу([параметри]) {`
 `Тіло конструктора`
 `}`

7. `package НазваПакету{.НазваПідпакету};`

8. Клас може використовувати всі класи з власного пакету і всі загальнодоступні класи з інших пакетів. Доступ до класів з інших пакетів можна отримати двома шляхами:

1. вказуючи повне ім'я пакету перед іменем кожного класу

2. використовуючи оператор `import`, що дозволяє підключати як один клас так і всі загальнодоступні класи пакету, позбавляючи необхідності записувати імена класів з вказуванням повної назви пакету перед ними.

9. Статичний імпорт дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.

10. Файл, каталоги повинні бути строго структурованими. Чітка ієрархія, назви пакетів та підпакетів повинні співпадати з назвами каталогів де вони розміщуються.

Висновок: я ознайомився з процесом розробки класів та пакетів мовою Java..