Міністерство освіти і науки України Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №3

З дисципліни: «Кросплатформенні засоби програмування» На тему: «Класи та пакети»

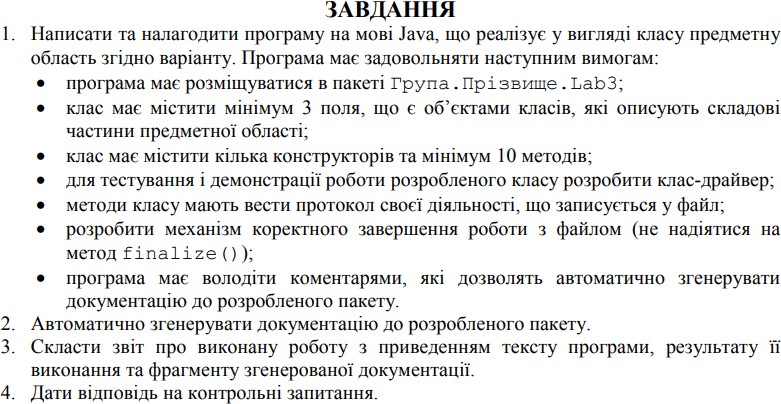
Варіант 15

Виконав: Ст. групи КІ-36 Постоюк В.Р.

Прийняв: к.т.н., доцент Іванов Ю.С.

Львів 2022

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



**Варіант завдання:** Корабель

# Код роботи

*Ship.java*

package KI34.Olkhovyk.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);

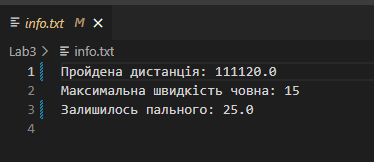
out.println("Поточний об'єм палива - " + ship.GetCurrentFuelVolume()); out.println("Пройдена відстань - " + ship.GetPassedDistance()); out.println("Поточна швидкість - " + ship.GetCurrentSpeed());

}

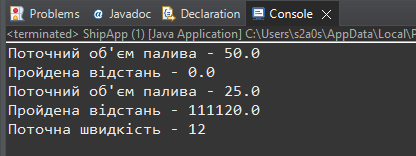
}

# Результат виконання програми

* Txt-файл з записаною інформацією:



* Консоль



**Відповіді на контрольні запитання:**

1. [public] class Назва Класу { [Методи]

[Змінні] [Поля]

[Конструктори]

}

1. [СпецифікаторДоступу] Тип назваМетоду([параметри]) [throws класи] { [Тіло методу] [return [значення]];

}

1. [СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];
2. Використати [final], тобто/наприклад private final int i;
3. Ініціалізацію полів при створенні об’єкту можна здійснювати трьома способами:
   * у конструкторі;
   * явно при оголошені поля;
   * у блоці ініціалізації (виконується перед виконанням конструктора). Якщо поле не ініціалізується жодним з цих способів, то йому присвоюється значення за замовчуванням.
4. [СпецифікаторДоступу] НазваКласу([параметри]) { Тіло конструктора

}

1. package НазваПакету{.НазваПідпакету};
2. 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:
3. вказуючи повне ім’я пакету перед іменем кожного класу
4. 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.
   1. Статичний імпорт дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.
   2. 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.*.*