

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Лабораторна робота №3

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Класи та пакети»

Варіант № 2

Виконав:

ст.гр. КІ-36

Билень С.В.

Львів – 2022

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

Контрольні питання:

1. Синтаксис визначення класу.

Відповідь: [public] class НазваКласу

```
{  
    [конструктори]  
    [методи]  
    [поля]  
}
```

2. Синтаксис визначення методу.

Відповідь: [СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри])
[throws класи]

```
{  
    [Тіло методу]  
    [return [значення]];  
}
```

3. Синтаксис оголошення поля.

Відповідь: [СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];

4. Як оголосити та ініціалізувати константне поле?

Відповідь: [СпецифікаторДоступу] static final Тип НазваПоля = Значення;

- явно при оголошенні поля класу;

- у статичному блоці ініціалізації.

5. Які є способи ініціалізації полів?

Відповідь:

6. Синтаксис визначення конструктора.

Відповідь: [СпецифікаторДоступу] НазваКласу([параметри])

```
{  
    Тіло конструктора
```

}

7. Синтаксис оголошення пакету.

Відповідь: `package НазваПакету{.НазваПідпакету};`

8. Як підключити до програми класи, що визначені в зовнішніх пакетах?

Відповідь: вказуючи повне ім'я пакету перед іменем кожного класу або використовуючи оператор `import`

9. В чому суть статичного імпорту пакетів?

Відповідь: Можливість імпортувати окремі статичні методи або поля класу

- `import static`

`НазваПакету{.НазваПідпакету}.НазваКласу.НазваСтатичногоМетодуАбоПоля;`

- `import static НазваПакету{.НазваПідпакету}.*;`

10. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

Відповідь: Для уникнення конфліктів імен не зловживати імпортом пакетів.

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.

Код StarShipApp.java:

```
import java.io.FileNotFoundException;

/**
 * Class with entry point
 * @author Volokhov D.G. KI-32
 */
public class StarShipApp {
    /**
     * Static method main is an entry point in program
     *
     * @param args
     *
     */
    public static void main(String[] args) throws FileNotFoundException {
        StarShip starShip = new StarShip();
        System.out.println(starShip.getStarShipName());
        System.out.println(starShip.getStarShipColor());
        System.out.println(starShip.getEngineStatus());

        StarShip starShip1 = new StarShip("#2 Kosmo 300", StarShip.StarShipColor.BLACK);
        System.out.println(starShip1.getStarShipName());
        System.out.println(starShip1.getStarShipColor());
        starShip1.setLowEngine();
        System.out.println(starShip1.getEngineStatus());

        starShip.closeLoggerFile();
        starShip1.closeLoggerFile();
    }
}
```

Код StarShip.java:

```
import java.io.*;

/**
 * StarShip class
 */
public class StarShip {
    //Data fields
    private String starShipName;
    private StarShipColor starShipColor;
    private EngineStatus engineStatus;
```

```

private static int starShipNumber = 1;
private PrintWriter outputStream = new PrintWriter(new File(String.format("StarShipLogger%s.txt", starShipNumber)));

//Constructors
public StarShip() throws FileNotFoundException {
    starShipName = String.format("#%s StarShip", starShipNumber);
    starShipColor = StarShipColor.RED;
    engineStatus = EngineStatus.TURNED_OFF;
    outputStream.println("Creating a StarShip");
    ++starShipNumber;
}

public StarShip(String starShipName) throws FileNotFoundException {
    this.starShipName = starShipName;
    starShipColor = StarShipColor.RED;
    engineStatus = EngineStatus.TURNED_OFF;
    outputStream.println("Creating a StarShip");
    ++starShipNumber;
}

public StarShip(String starShipName, StarShipColor starShipColor) throws FileNotFoundException {
    this.starShipName = starShipName;
    this.starShipColor = starShipColor;
    engineStatus = EngineStatus.TURNED_OFF;
    outputStream.println("Creating a StarShip");
    ++starShipNumber;
}

public StarShip(String starShipName, StarShipColor starShipColor, EngineStatus engineStatus) throws FileNotFoundException {
    this.starShipName = starShipName;
    this.starShipColor = starShipColor;
    this.engineStatus = engineStatus;
    outputStream.println("Creating a StarShip");
    ++starShipNumber;
}

//Enum
private enum EngineStatus{
    TURNED_OFF, LOW, MEDIUM, HIGH
}

public enum StarShipColor{
    WHITE, BLACK, RED, PINK, YELLOW, GREEN, BLUE
}

//Getter methods
public String getStarShipName() {
    outputStream.println("getStarShipName: " + starShipName);
    return starShipName;
}

```

```

public EngineStatus getEngineStatus() {
    outputStream.println("getEngineStatus: " + engineStatus);
    return engineStatus;
}

public StarShipColor getStarShipColor() {
    outputStream.println("getStarShipColor: " + starShipColor);
    return starShipColor;
}

//Set methods

public void turnOffEngine(){
    if(engineStatus == EngineStatus.TURNED_OFF) {
        System.out.println("Engine is already turned off.");
        outputStream.println("Engine is already turned off.");
    } else {
        System.out.println("Engine is turned off.");
        outputStream.println("Engine is turned off.");
        engineStatus = EngineStatus.TURNED_OFF;
    }
}

public void setLowEngine(){
    if(engineStatus == EngineStatus.LOW) {
        System.out.println("Engine is already in low mode.");
        outputStream.println("Engine is already in low mode.");
    } else {
        System.out.println("Engine is set in low mode.");
        outputStream.println("Engine is set in low mode.");
        engineStatus = EngineStatus.LOW;
    }
}

public void setMediumEngine(){
    if(engineStatus == EngineStatus.MEDIUM) {
        System.out.println("Engine is already in medium mode.");
        outputStream.println("Engine is already in medium mode.");
    } else {
        System.out.println("Engine is set in medium mode.");
        outputStream.println("Engine is set in medium mode.");
        engineStatus = EngineStatus.MEDIUM;
    }
}

public void setHighEngine(){
    if(engineStatus == EngineStatus.HIGH) {
        System.out.println("Engine is already in high mode.");
        outputStream.println("Engine is already in high mode.");
    } else {

```

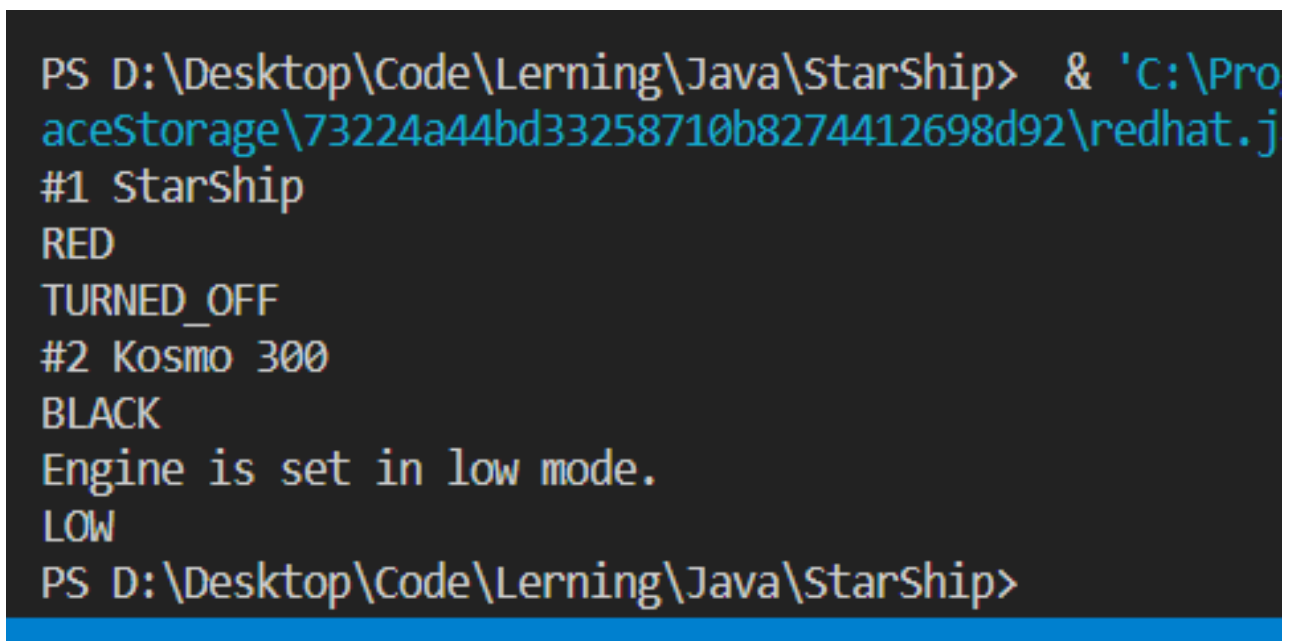
```

        System.out.println("Engine is set in high mode.");
        outputStream.println("Engine is set in high mode.");
        engineStatus = EngineStatus.HIGH;
    }
}

public void closeLoggerFile(){
    outputStream.println("Close logger file.");
    outputStream.close();
}
}

```

Скріншоти програми:

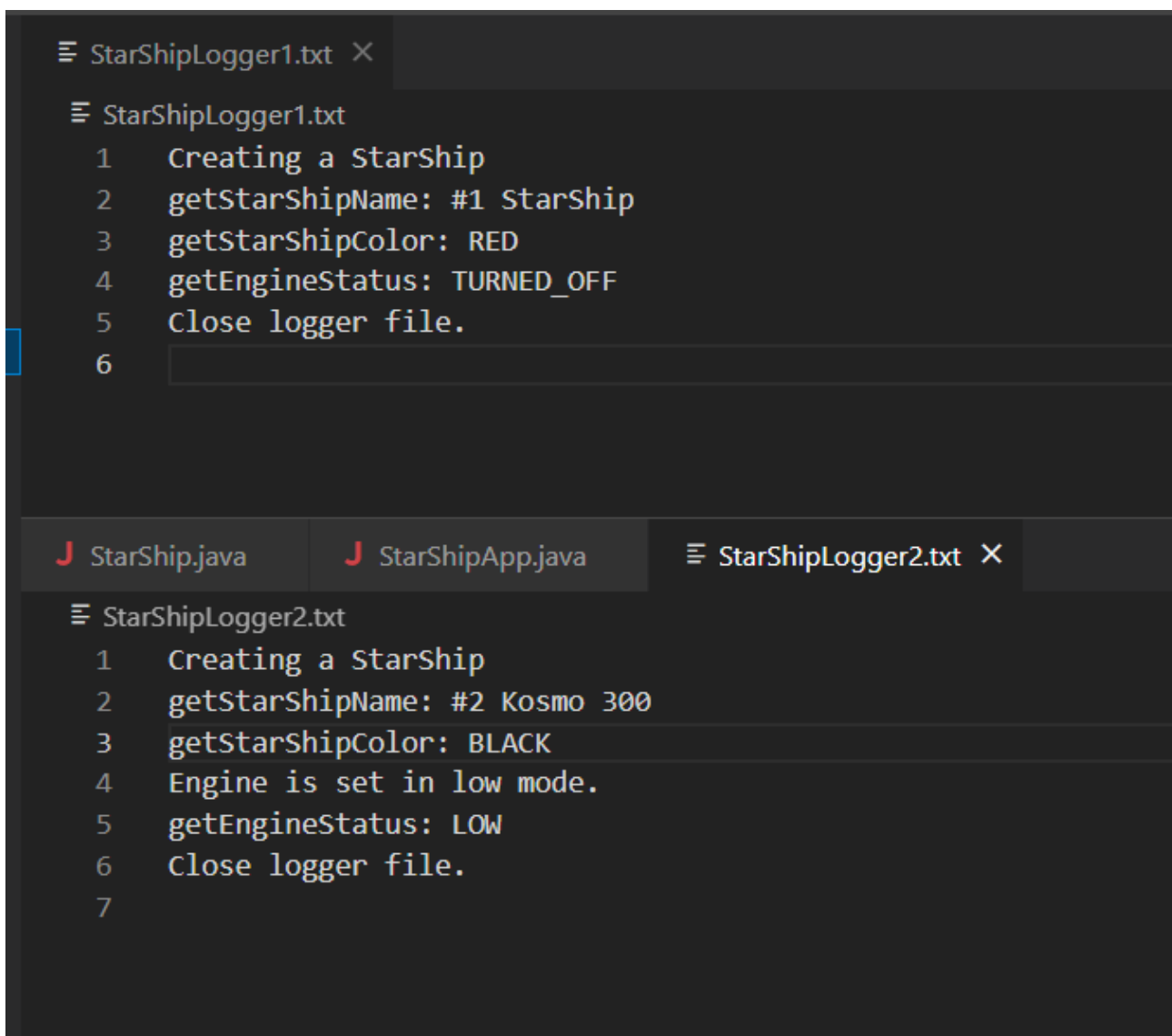


```

PS D:\Desktop\Code\Lerning\Java\StarShip> & 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Java\jdk-11.0.10\bin\java.exe' -jar 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Java\jdk-11.0.10\bin\redhat.jar'
#1 StarShip
RED
TURNED_OFF
#2 Kosmo 300
BLACK
Engine is set in low mode.
LOW
PS D:\Desktop\Code\Lerning\Java\StarShip>

```

Рис. 1. Результат роботи програми.



```
StarShipLogger1.txt
1 Creating a StarShip
2 getStarShipName: #1 StarShip
3 getStarShipColor: RED
4 getEngineStatus: TURNED_OFF
5 Close logger file.
6

StarShip.java
StarShipApp.java
StarShipLogger2.txt
StarShipLogger2.txt
1 Creating a StarShip
2 getStarShipName: #2 Kosmo 300
3 getStarShipColor: BLACK
4 Engine is set in low mode.
5 getEngineStatus: LOW
6 Close logger file.
7
```

Рис. 2. Результат роботи в файлі.

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