Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 3

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «Спадкування та інтерфейси»

**Виконав:**

студент групи *КІ-306*

*Ярмола Юрій*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

**Мета роботи:** ознайомитися з спадкуванням та інтерфейсами у мові Java.

**Завдання ( Варіант 30 «Електричка» ):**

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі No3, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі No3, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група.Прізвище.Lab4 та 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.

1. Дати відповідь на контрольні запитання.

**Вихідний код програми**

**Файл TrainAPP**

import KI306.Yarmola.Lab3.ElectricTrain;

import java.io.FileNotFoundException;

/\*\*

\*

\* Train Application class implements main method for electric train

class possibilities demonstration

\* @author Yurii Yarmola

\* @version 1.0

\*

\*/

public class TrainApp {

public static void main(String[] args) throws FileNotFoundException {

ElectricTrain train = new ElectricTrain();

train.start\_locomotive();

train.add\_car(4);

train.open\_train\_door();

train.close\_train\_door();

train.add\_temperature(5);

train.turn\_on\_light();

train.add\_speed();

train.add\_speed();

train.stop\_locomotive();

train.dispose();

}

}

**Locomotive.java**

import KI306.Yarmola.Lab3.ElectricTrain;

import java.io.FileNotFoundException;

/\*\*

\*

\* Train Application class implements main method for electric train

class possibilities demonstration

\* @author Yurii Yarmola

\* @version 1.0

\*

\*/

public class TrainApp {

public static void main(String[] args) throws FileNotFoundException {

ElectricTrain train = new ElectricTrain();

train.start\_locomotive();

train.add\_car(4);

train.open\_train\_door();

train.close\_train\_door();

train.add\_temperature(5);

train.turn\_on\_light();

train.add\_speed();

train.add\_speed();

train.stop\_locomotive();

train.dispose();

}

}

**ElectricTrain.java**

import KI306.Yarmola.Lab3.ElectricTrain;

import java.io.FileNotFoundException;

/\*\*

\*

\* Train Application class implements main method for electric train

class possibilities demonstration

\* @author Yurii Yarmola

\* @version 1.0

\*

\*/

public class TrainApp {

public static void main(String[] args) throws FileNotFoundException {

ElectricTrain train = new ElectricTrain();

train.start\_locomotive();

train.add\_car(4);

train.open\_train\_door();

train.close\_train\_door();

train.add\_temperature(5);

train.turn\_on\_light();

train.add\_speed();

train.add\_speed();

train.stop\_locomotive();

train.dispose();

}

}

**TrainInterface.java**

package KI306.Yarmola.Lab3;

public abstract interface TrainInterface {

public void open\_train\_door();

public void close\_train\_door();

public void turn\_on\_light();

public void turn\_off\_light();

public int add\_temperature(int temp);

public int minus\_temperature(int temp);

public int get\_temperature();

}

**Результат виконання програми**

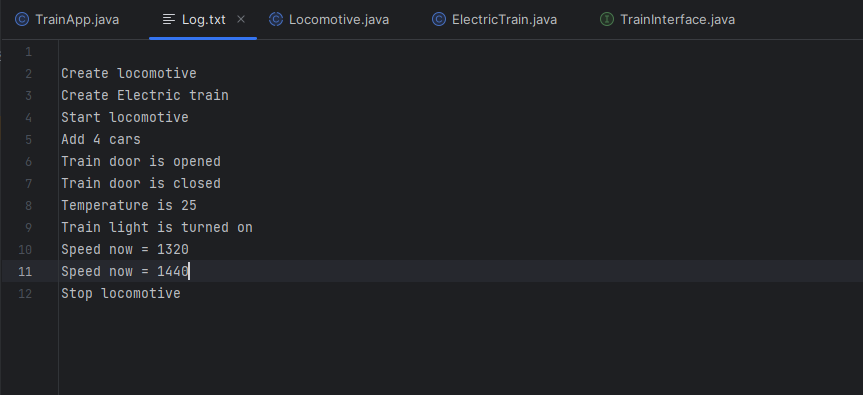
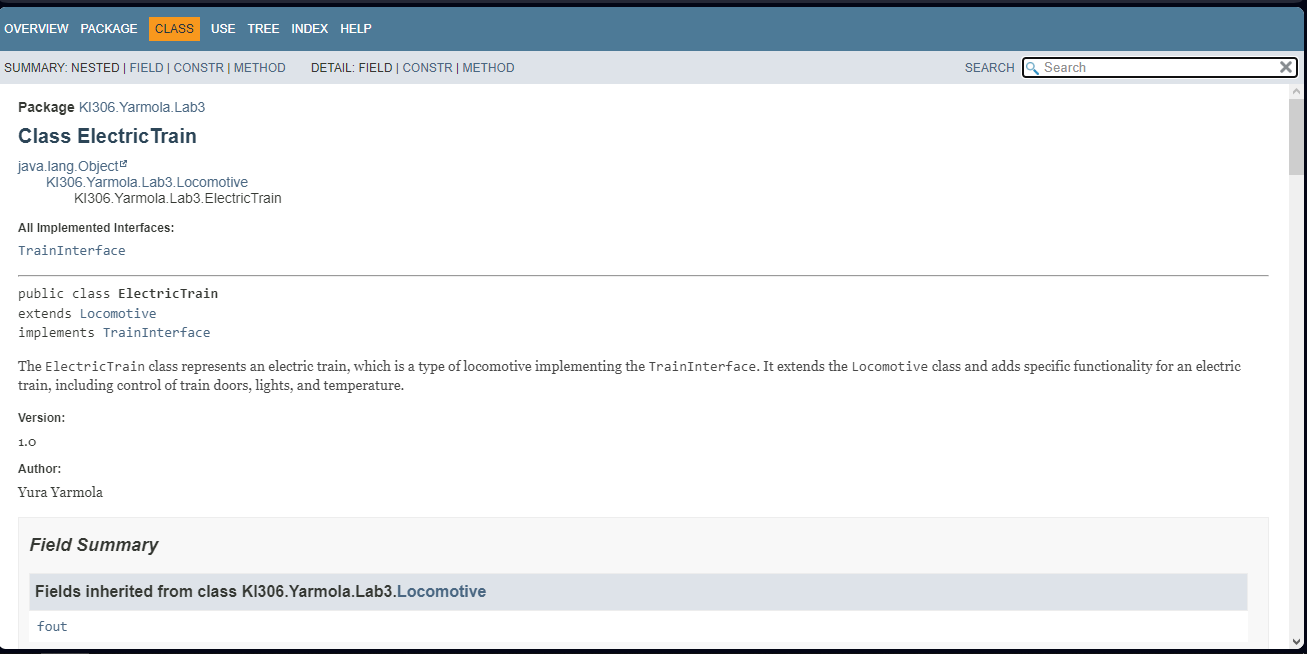
****

Рис.1 – Скріншот документу згенерованого програмою

**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

1. Синтаксис реалізації спадкування.

class ChildClass extends ParentClass {

// Код підкласу тут

}

1. Що таке суперклас та підклас?

Суперклас (або батьківський клас) - це клас, від якого успадковуються властивості та методи.

Підклас (або дочірній клас) - це клас, який успадковує властивості та методи від суперкласу і може додавати власні.

1. Як звернутися до членів суперкласу з підкласу?

За допомогою ключового слова super. Наприклад: super.methodName(); для виклику методу суперкласу.

1. Коли використовується статичне зв’язування при виклику методу?

Статичне зв'язування відбувається під час компіляції, коли відомо, який метод викликати на основі типу посилання.

1. Як відбувається динамічне зв’язування при виклику методу?

Динамічне зв'язування відбувається під час виконання програми і визначається на основі об'єкта, який викликає метод.

1. Що таке абстрактний клас та як його реалізувати?

Абстрактний клас - це клас, який не може бути інстанційований і може містити абстрактні методи.

Для створення абстрактного класу використовується ключове слово abstract.

1. Для чого використовується ключове слово instanceof?

Використовується для перевірки, чи об'єкт належить до певного класу або інтерфейсу.

1. Як перевірити чи клас є підкласом іншого класу?

Використовується ключове слово extends. Наприклад, class ChildClass extends ParentClass.

1. Що таке інтерфейс?

Інтерфейс - це контракт, який описує методи, які клас повинен реалізувати. В інтерфейсах методи не мають реалізації.

1. Як оголосити та застосувати інтерфейс?

Оголошення інтерфейсу: interface InterfaceName { ... }

Застосування інтерфейсу в класі: class ClassName implements InterfaceName { ... }

**Висновок**

Ознайомився з спадкуванням та інтерфейсами у мові Java. Розробив розширений клас ElecticTrain за областю з варіанту «Електричка».