Міністерство освіти і науки України Національний університет «Львівська політехніка» Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи №3

з дисципліни: “Кросплатформні засоби програмування”

на тему: “Спадкування та інтерфейси”

# Виконала:

Ст. гр. КІ-306

Покидько Юлія

# Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

**Мета:** ознайомитися зі спадкуванням та інтерфейсами у мовіJava.

# Завдання (варіант № 15) :

написати та налагодити програму на мові Java, що розширює клас, реалізований у лабораторній роботі №3, для реалізації предметної області заданої варіантом:

**Фрегат**

**Хід роботи:**

1. Запустила середовище IntelliJ IDEA та написаЛА програму згідно індивідуального завдання:

***FrigateApp.java***

*/\*\*  
 \* lab 3 package  
 \*/*package ki306.Pokydko.Lab3;  
  
import java.io.FileNotFoundException;  
  
*/\*\*  
 \* FrigateApp class implements main method for Frigate class possibilities demonstration  
 \*  
 \* @author Pokydko Yulia  
 \* @version 1.0  
 \*/*public class FrigateApp {  
 public static void main(String[] args) throws FileNotFoundException{  
 Frigate ship = new Frigate(80000, 250, 15000);  
 ship.startEngine();  
 ship.setSpeed(20);  
 ship.rudderLeft();  
 ship.rudderRight();  
 ship.rudderAlign();  
 ship.shoot();  
 ship.stopEngine();  
 ship.info();  
 ship.dispose();  
 }  
}

***Frigate.java***

*/\*\*  
 \* lab 3 package  
 \*/*package ki306.Pokydko.Lab3;  
  
import java.io.FileNotFoundException;  
  
*/\*\*  
 \* Class Frigate implements frigate  
 \*  
 \* @author Pokydko Yulia  
 \* @version 1.0  
 \*/*public class Frigate extends Ship implements Military {  
*/\*\*  
 \* Constructor  
 \*  
 \* @throws FileNotFoundException  
 \*/*public Frigate() throws FileNotFoundException{  
 super();  
}  
*/\*\*  
 \* Constructor  
 \*  
 \* @param height  
 \* @param weight  
 \* @param square  
 \* @throws FileNotFoundException  
 \*/*public Frigate(int height, int weight, int square) throws FileNotFoundException{  
 super(height,weight,square);  
}  
  
*/\*\*  
 \* Method implements shooting  
 \*/*@Override  
 public void shoot(){  
 System.*out*.println("Type of ship: Frigate");  
 fout.println("Set...Shoot...\*Boom\*!!!!");  
}  
  
*/\*\*  
 \* Method shows up the ship's info  
 \*/*@Override  
 public void info(){  
 super.info();  
 System.*out*.println("Type of ship: Frigate");  
 fout.println("Type of ship: Frigate");  
}  
}

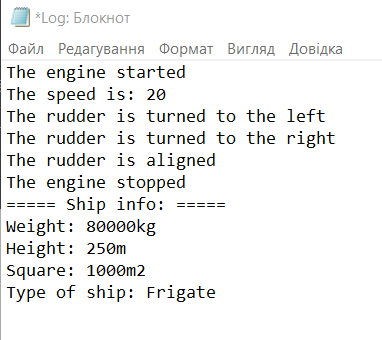
***Military.java***

*/\*\*  
 \* lab 3 package  
 \*/*package ki306.Pokydko.Lab3;  
*/\*\*  
 \* Interface Military describes military ship  
 \*  
 \* @author Pokydko Yulia  
 \* @version 1.0  
 \*/*public interface Military {  
 */\*\*  
 \* Method describes shooting  
 \*/* public void shoot();  
}

***Ship.java***

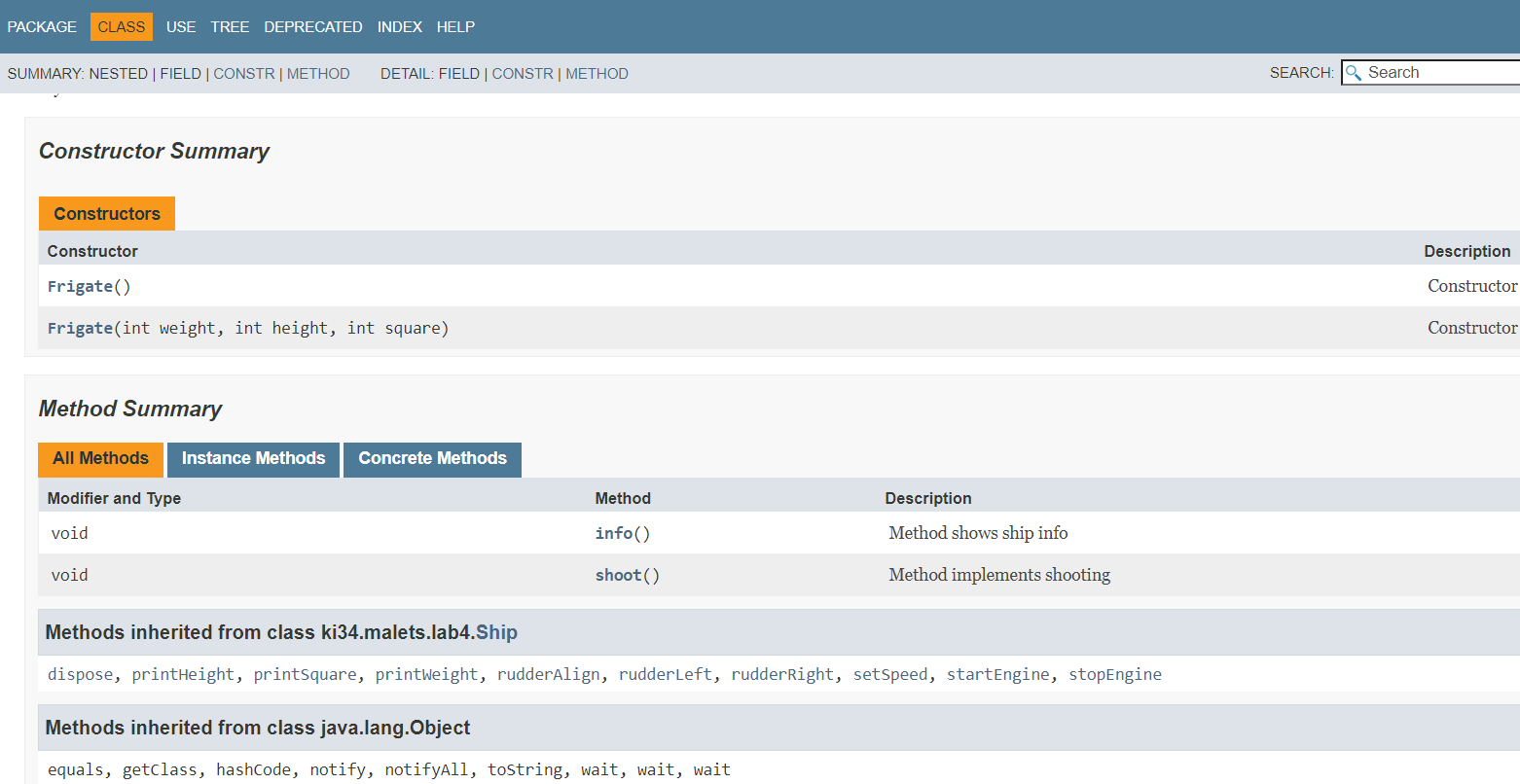
*/\*\*  
 \* lab 3 package  
 \*/*package ki306.Pokydko.Lab3;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
*/\*\*  
 \* Class Ship implements ship  
 \*  
 \* @author Pokydko Yulia  
 \* @version 1.0  
 \*/*public abstract class Ship {  
  
 */\*\*  
 \* Constructor  
 \* @throws FileNotFoundException  
 \*/* public Ship() throws FileNotFoundException {  
 engine = new Engine();  
 rudder = new Rudder();  
 deck = new Deck();  
 weight = 50000;  
 height = 100;  
 fout = new PrintWriter(new File("Log.txt"));  
 }  
  
 */\*\*  
 \* Constructor  
 \*  
 \* @param weight  
 \* @param height  
 \* @param square  
 \* @throws FileNotFoundException  
 \*/* public Ship(int weight, int height, int square) throws FileNotFoundException {  
 this.weight = weight;  
 this.height = height;  
 engine = new Engine();  
 rudder = new Rudder();  
 deck = new Deck();  
 fout = new PrintWriter(new File("Log.txt"));  
 }  
  
 private int weight;  
 private int height;  
 private Engine engine;  
 private Rudder rudder;  
 private Deck deck;  
 protected PrintWriter fout;  
  
 */\*\*  
 \* Method implements the engine's starting  
 \*/* public void startEngine() {  
 engine.start();  
 }  
  
 */\*\*  
 \* Method implements the engine's stopping  
 \*/* public void stopEngine() {  
 engine.stop();  
 }  
  
 */\*\*  
 \* Method sets the speed of the ship  
 \*/* public void setSpeed(int speed) {  
 engine.setSpeed(speed);  
 }  
  
 */\*\*  
 \* Method implements going forward  
 \*/* public void rudderAlign() {  
 rudder.align();  
 }  
  
 */\*\*  
 \* Method implements turning to the left  
 \*/* public void rudderLeft() {  
 rudder.turnLeft();  
 }  
  
 */\*\*  
 \* Method implements turning to the right  
 \*/* public void rudderRight() {  
 rudder.turnRight();  
 }  
  
 */\*\*  
 \* Method prints the ship's weight  
 \*/* public void printWeight() {  
 System.*out*.println("Weight: " + weight + "kg");  
 fout.println("Weight: " + weight + "kg");  
 }  
  
 */\*\*  
 \* Method prints the ship's height  
 \*/* public void printHeight() {  
 System.*out*.println("Height: " + height + "m");  
 fout.println("Height: " + height + "m");  
 }  
  
 */\*\*  
 \* Method prints the ship's square  
 \*/* public void printSquare() {  
 System.*out*.println("Square: " + deck.getSquare() + "m2");  
 fout.println("Square: " + deck.getSquare() + "m2");  
 }  
  
 */\*\*  
 \* Method shows the ship's info  
 \*/* public void info() {  
 System.*out*.println("===== Ship info: =====");  
 fout.println("===== Ship info: =====");  
 printWeight();  
 printHeight();  
 printSquare();  
 }  
  
 */\*\*  
 \* Method releases used recourses  
 \*/* public void dispose() {  
 fout.close();  
 }  
  
 class Engine {  
  
 */\*\*  
 \* Constructor  
 \*/* public Engine() {  
 speed = 0;  
 isStarted = false;  
 }  
  
 boolean isStarted;  
 int speed;  
  
 */\*\*  
 \* Method implements the engine's starting  
 \*/* public void start() {  
 if (isStarted) {  
 System.*out*.println("The engine already started!");  
 fout.println("The engine already started!");  
 } else {  
 System.*out*.println("The engine started");  
 fout.println("The engine started");  
 isStarted = true;  
 }  
 }  
  
 */\*\*  
 \* Method implements the engine's stopping  
 \*/* public void stop() {  
 if (isStarted) {  
 System.*out*.println("The engine stopped");  
 fout.println("The engine stopped");  
 isStarted = false;  
 } else {  
 System.*out*.println("The engine is not started!");  
 fout.println("The engine is not started!");  
 }  
 }  
  
 */\*\*  
 \* Method sets the speed  
 \*/* public void setSpeed(int speed) {  
 this.speed = speed;  
 System.*out*.println("The speed is: " + speed);  
 fout.println("The speed is: " + speed);  
 }  
 }  
  
 class Rudder {  
  
 */\*\*  
 \* Method implements the aligning of the rudder  
 \*/* public void align() {  
 System.*out*.println("The rudder is aligned");  
 fout.println("The rudder is aligned");  
 }  
  
 */\*\*  
 \* Method implements turning to the left  
 \*/* public void turnLeft() {  
 System.*out*.println("The rudder is turned to the left");  
 fout.println("The rudder is turned to the left");  
 }  
  
 */\*\*  
 \* Method implements turning to the right  
 \*/* public void turnRight() {  
 System.*out*.println("The rudder is turned to the right");  
 fout.println("The rudder is turned to the right");  
 }  
 }  
  
 class Deck {  
  
 */\*\*  
 \* Constructor  
 \*/* Deck() {  
 square = 1000;  
 }  
  
 */\*\*  
 \* Constructor  
 \*  
 \* @param square  
 \*/* Deck(int square) {  
 this.square = square;  
 }  
  
 int square;  
  
 */\*\*  
 \* @return the square  
 \*/* public int getSquare() {  
 return square;  
 }  
  
 */\*\*  
 \* @param square the square to set  
 \*/* public void setSquare(int square) {  
 this.square = square;  
 }  
 }  
}

1. Після виконання програми переглянула створений файл Log.txt:



*Рис.1. Результат виконання програми*

1. Згенерувала документацію



*Рис.2. Згенерований html-файл*

**Висновок:**

На даній лабораторній роботі ознайомилася зі спадкуванням та інтерфейсами у мові Java.