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»

Виконав:

студент групи КІ-306

Яцків А.Р.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання ( Варіант 1 «Спортсмен» ):**

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**

**Вихідний код програми**

**Файл SportsmanApp.java**

package KI306.Yatskiv.Lab3;  
  
public class SportsmanApp {  
 public static void main(String[] args) {  
 // Створюємо спортсмена з певним ім'ям, віком, стажем та видом спорту  
 Sportsman sportsman1 = new Sportsman("John", 37, "Male", "Athlete", "Swimming",7,8,4, 20);  
 sportsman1.logInfo();  
 sportsman1.training(6); // Моделювати тренування на 6 годин  
 sportsman1.compete(); // Моделювати змагання  
  
 // Створюємо іншого спортсмена  
 Sportsman sportsman2 = new Sportsman("Mary", 32, "Female", "Athlete", "Running",3,4,6, 13);  
 sportsman2.logInfo();  
 sportsman2.training(4); // Моделювати тренування на 4 годин  
 sportsman2.compete(); // Моделювати змагання  
 Sportsman sportsman3 = new Sportsman("Taras", 32, "Male", "Athlete", "Running",3,4,6, 13);  
 Sportsman sportsman4 = new Sportsman("Ivan", 32, "Male", "Athlete", "Running",3,4,6, 13);  
 Sportsman sportsman5 = new Sportsman("Marie", 32, "Female", "Athlete", "Running",3,4,6, 13);  
 Sportsman sportsman6 = new Sportsman("David", 32, "Male", "Athlete", "Running",3,4,6, 13);  
 Sportsman sportsman7 = new Sportsman("Osen", 32, "Female", "Athlete", "Running",3,4,6, 13);  
  
  
 System.*out*.println("Male sportsmen number: " + Sportsman.*getMaleCount*());  
 System.*out*.println("Female sportsmen number: " + Sportsman.*getFemaleCount*());  
 }  
}

**Файл Human.java**

package KI306.Yatskiv.Lab3;  
  
import java.io.PrintWriter;  
  
*/\*\*  
 \* Клас представляє собою об'єкт людини.  
 \*/*public abstract class Human {  
  
 private String name;  
 private int age;  
 private String gender;  
 private String occupation;  
  
 */\*\*  
 \* Конструктор з параметрами для створення людини з вказаними характеристиками.  
 \*  
 \** ***@param*** *name Ім'я людинм.  
 \** ***@param*** *age Вік людини.  
 \** ***@param*** *gender Стать лююдини.  
 \** ***@param*** *occupation Рід занятості.  
 \*/* public Human(String name, int age, String gender, String occupation) {  
 this.name = name;  
 this.age = age;  
 this.gender = gender;  
 this.occupation = occupation;  
 }  
  
 */\*\*  
 \* Конструктор за замовчуванням для створення людини з ініціальними значеннями.  
 \*/* public Human(){  
 this.name = "Unknown";  
 this.age = 0;  
 this.gender = "Unknown";  
 this.occupation = "Unknown";  
 }  
  
 */\*\*  
 \* Метод для зміни імені людини.  
 \*  
 \** ***@param*** *name Нове ім'я людини.  
 \*/* public void setName(String name) {  
 this.name = name;  
 }  
  
 */\*\*  
 \* Метод для зміни віку людини.  
 \*  
 \** ***@param*** *age Новий вік людини.  
 \*/* public void setAge(int age) {  
 if (age >= 0) {  
 this.age = age;  
 }  
 }  
  
 */\*\*  
 \* Метод для зміни статі людини.  
 \*  
 \** ***@param*** *gender нова стать людини.  
 \*/* public void setGender(String gender) {  
 this.gender = gender;  
 }  
  
 */\*\*  
 \* Метод для зміни роду занять людини.  
 \*  
 \** ***@param*** *occupation новий рід занять людини.  
 \*/* public void setOccupation(String occupation) {  
 this.occupation = occupation;  
 }  
  
 */\*\*  
 \* Метод для отримання імені людини.  
 \*  
 \** ***@return*** *Ім'я людини.  
 \*/* public String getName() {  
 return name;  
 }  
 */\*\*  
 \* Метод для отримання віку людини.  
 \*  
 \** ***@return*** *Вік людини.  
 \*/* public int getAge() {  
 return age;  
 }  
 */\*\*  
 \* Метод для отримання статі людини.  
 \*  
 \** ***@return*** *Стать людини.  
 \*/* public String getGender() {  
 return gender;  
 }  
 */\*\*  
 \* Метод для отримання роду занять людини.  
 \*  
 \** ***@return*** *Рід занять людини.  
 \*/* public String getOccupation() {  
 return occupation;  
 }  
  
 */\*\*  
 \* Метод для перевірки чи дорослою є людина  
 \*  
 \** ***@return*** *true якщо людина доросла, false якщо ні  
 \*/* public boolean isAdult() {  
 boolean isAdult = age > 18;  
 if (isAdult) {  
 System.*out*.println("Person is adult");  
 } else {  
 System.*out*.println("Person is underage");  
 }  
 return isAdult;  
 }  
  
 protected void logInfo() {  
 Logger.*getInstance*().log("\n" + "Name: " + name);  
 Logger.*getInstance*().log("Age: " + age);  
 Logger.*getInstance*().log("Gender: " + gender);  
 Logger.*getInstance*().log("Occupation: " + occupation);  
 }  
  
  
  
  
}

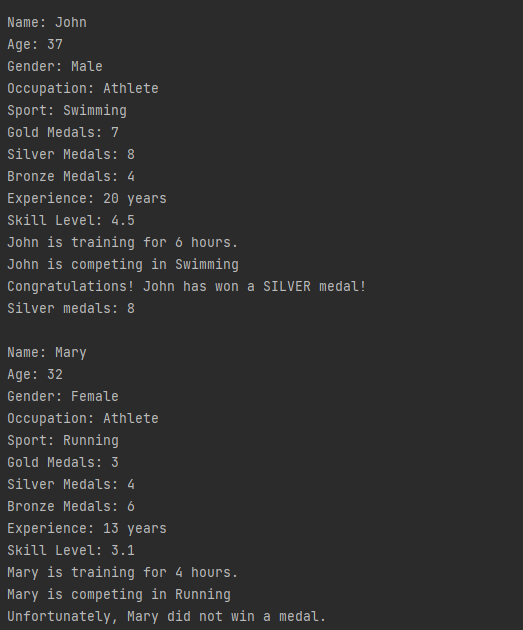
}

**Файл Logger.java**

package KI306.Yatskiv.Lab3;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.PrintWriter;  
  
  
public class Logger {  
 private static final String *LOG\_FILE* = "Sportsman.txt";  
 private static PrintWriter *logWriter*;  
  
 private static Logger *instance* = null;  
  
 private Logger() {  
 try {  
 *logWriter* = new PrintWriter(new FileWriter(*LOG\_FILE*, false));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static Logger getInstance() {  
 if (*instance* == null) {  
 *instance* = new Logger();  
 }  
 return *instance*;  
 }  
  
 public void log(String message) {  
 System.*out*.println(message); // Print to console  
 *logWriter*.println(message); // Write to log file  
 *logWriter*.flush(); // Ensure the message is written immediately  
 }  
  
 public void close() {  
 *logWriter*.close();  
 }  
}

**Результат виконання програми**

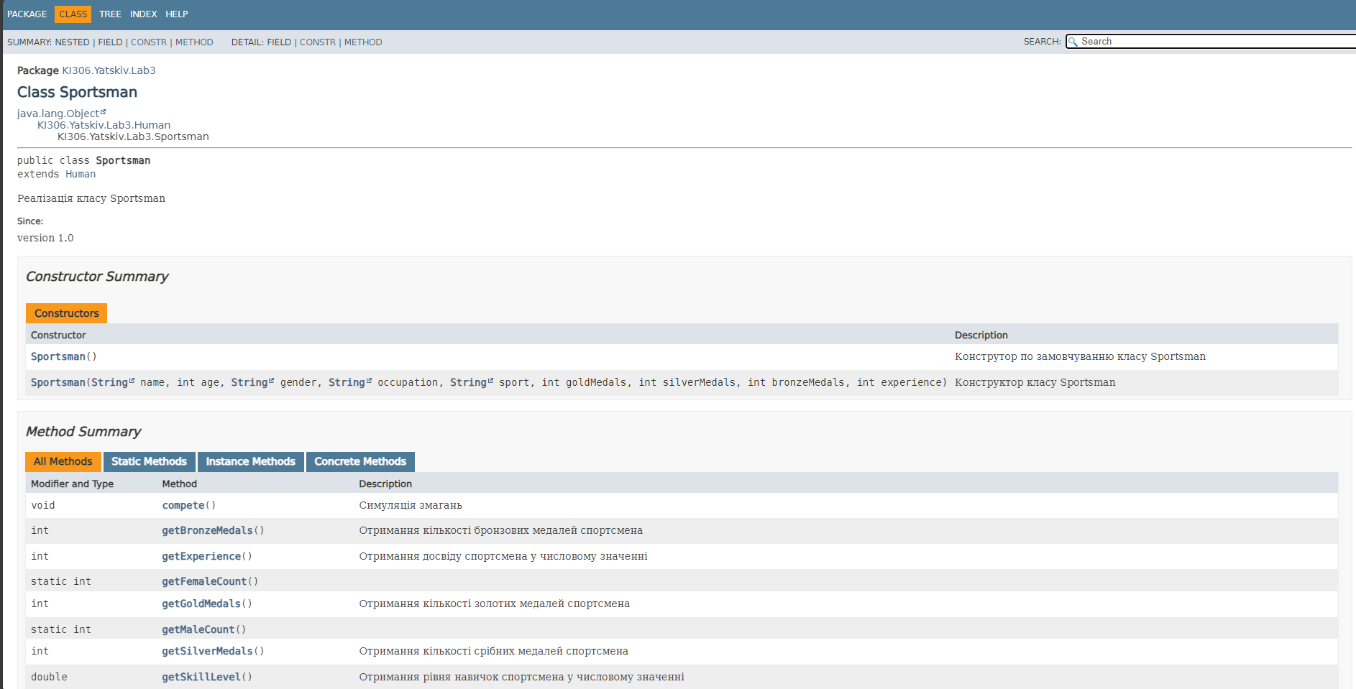
Консоль:



Sportsman.txt:

Name: John  
Age: 37  
Gender: Male  
Occupation: Athlete  
Sport: Swimming  
Gold Medals: 7  
Silver Medals: 8  
Bronze Medals: 4  
Experience: 20 years  
Skill Level: 4.5  
John is training for 6 hours.  
John is competing in Swimming  
Congratulations! John has won a SILVER medal!  
Silver medals: 8  
  
Name: Mary  
Age: 32  
Gender: Female  
Occupation: Athlete  
Sport: Running  
Gold Medals: 3  
Silver Medals: 4  
Bronze Medals: 6  
Experience: 13 years  
Skill Level: 3.1  
Mary is training for 4 hours.  
Mary is competing in Running  
Unfortunately, Mary did not win a medal.

**Фрагмент згенерованої документації**

**

**Відповіді на контрольні запитання**

**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 Sportsman за областю з варіанту «Людина».