Міністерство освіти і науки України Національний «Львівська політехніка» Кафедра ЕОМ



до лабораторної роботи № 3 з дисципліни: «Кросплатформні засоби програмування» «Спадкування та інтерфейси»Варіант - 1

Виконав: Студент групи КІ-306 Баран В. М. Прийняв: Іванов Ю. С. **Мета:** ознайомитися з спадкуванням та інтерфейсами у мові Java.

ЗАВДАННЯ

- 1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі №2, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі №2, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група . Прізвище . Lab3 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
- 2. Автоматично згенерувати документацію до розробленого пакету.
- 3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
- 4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
- 5. Дати відповідь на контрольні запитання.

Варіант завдання:

1. Спортсмен

Код програми:

File Human.java

File Sportsman.java

```
package org.example;
import java.io.FileNotFoundException;
interface ISportsman{
   void run(int sec) throws InterruptedException;
public class Sportsman extends Person implements ISportsman{
throws FileNotFoundException {
    public void run(int sec) throws InterruptedException
            Thread. sleep (1000);
```

File Person.java

```
package org.example;
import java.io.File;
import java.io.FileNotFoundException; import java.io.PrintWriter;

/**
    * Class Person
    * @version 1.0
    */
public abstract class Person {
    public String name;
    int weight;
    double height;
    int age;

    Mouth mouth;
    Brain brain;
    PrintWriter fout;

/**
    * Constructor
    *
```

```
* Oparam height person height
* Oparam weight person weight
throws FileNotFoundException {
    public void say(String msg) {
     mouth.say(msg);
       PrintMessage(name + " tells " + person.name + ": ");
       mouth.say(msg);
        PrintMessage(name + " adds two numbers: " + a + "," + b);
       PrintMessage("Result: " + result);
      PrintMessage(age);
       PrintMessage(height);
      PrintMessage(this.weight);
```

```
double idx = weight / Math.pow(height, 2);
    PrintMessage("Body mass index: " + idx);
}

/**
    * Method exits
    */
public void dispose() {
    fout.flush();
    fout.close();
}

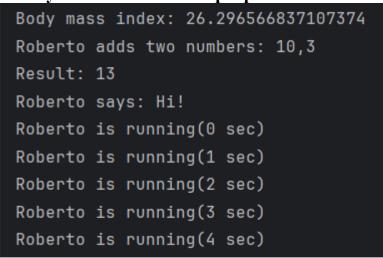
protected <T> void PrintMessage(T msg) {
    System.out.println(msg);
    fout.println(msg);
}
```

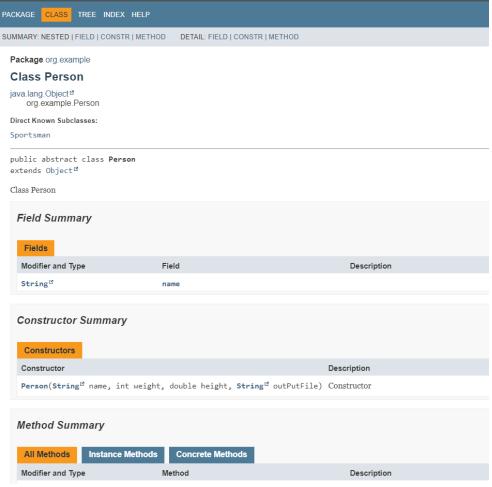
File Brain.java

File Mouth.java

```
}
   /**
   * Method to say something
   */
   public void say(String message)
   {
      String output = name + " says: " + message; System.out.println(output);
   fout.println(output);
   }
}
```

Результат виконання програми:





Відповіді на контрольні запитання

- 1. Синтаксис реалізації спадкування.
- class МійКлас implements Інтерфейс {// тіло класу }
- 2. Що таке суперклас та підклас?
- суперклас це клас, від якого інший клас успадковує властивості та методи.

Підклас - це клас, який успадковує властивості та методи від суперкласу.

- 3. Як звернутися до членів суперкласу з підкласу?
- super.назваМетоду([параметри]); // виклик методу суперкласу super.назваПоля; // звернення до поля суперкласу
- 4. Коли використовується статичне зв'язування при виклику методу?
- Статичне зв'язування використовується, коли метод є приватним, статичним,

фінальним або конструктором. В таких випадках вибір методу відбувається на

етапі компіляції.

- 5. Як відбувається динамічне зв'язування при виклику методу?
- вибір методу для виклику відбувається під час виконання програми на основі

фактичного типу об'єкта.

- 6. Що таке абстрактний клас та як його реалізувати?
- це клас, який має один або більше абстрактних методів (методів без реалізації).

Щоб створити абстрактний клас, використовується ключове слово abstract.

Приклад:

```
abstract class АбстрактнийКлас { abstract void абстрактнийМетод(); }
```

- 7. Для чого використовується ключове слово instanceof?
- для перевірки, чи об'єкт належить до певного класу або інтерфейсу.

Синтаксис:

```
if (об'єкт instanceof Клас) {
// код, який виконується, якщо об'єкт належить до класу }
```

- 8. Як перевірити чи клас ϵ підкласом іншого класу?
- В Java використовується ключове слово extends, щоб вказати, що клас ϵ підкласом іншого класу. Перевірити, чи один клас ϵ підкласом іншого класу

можна шляхом аналізу ієрархії успадкування.

- 9. Що таке інтерфейс?
- це абстрактний тип даних, який визначає набір методів, але не надає їх реалізацію. Всі методи інтерфейсу є загальнодоступними та автоматично ϵ

public. Інтерфейси використовуються для створення контрактів, які класи повинні реалізувати.

- 10. Як оголосити та застосувати інтерфейс?
- - Для оголошення інтерфейсу використовується ключове слово interface.

```
Синтаксис:
```

```
interface Інтерфейс {
// оголошення методів та констант }
```

• - Для застосування інтерфейсу в класі використовується ключове слово implements.

```
Синтаксис:
```

```
class МійКлас implements Інтерфейс {
// реалізація методів інтерфейсу }
```

Висновок: У ході виконання даної лабораторної роботи, я отримала навички роботи з концепціями спадкування та інтерфейсами в мові програмування Java. Ознайомившись з цими важливими аспектами об'єктно-орієнтованого програмування, я зрозуміла їх роль у створенні більш структурованих і гнучких програм.