Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 2

з дисципліни: «Кросплатформні засоби програмування»

«Класи та пакети» Варіант - 1

Виконав:

Студент групи КІ-306

Баран В. М.

Прийняв:

Іванов Ю. С.

**Львів 2023**

**Мета:** ознайомитися з процесом розробки класів та пакетів мовою Java.

**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.Lab2;

• 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 розробленої програми.

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: людина

**Код програми:**

# File App.java

package org.example;  
  
import java.io.FileNotFoundException;  
*/\*\*  
 \* Class App  
 \* @version 1.0  
 \*/*public class App {  
 */\*\*  
 \* @param args  
 \*/* public static void main(String[] args) throws FileNotFoundException  
 {  
 Person person = new Person("Roberto", 90, 1.85, "lab3.txt");  
 person.calculateBodyMassIndex();  
 person.calculate(10,3);  
 person.say("Hi!");  
 Person person1 = new Person("Donald", 85, 2.00, "lab3.txt");  
 person.tell(person1, "Hello!");  
  
 person.getAge();  
 person.getWeight();  
 person.getHeight();  
 person.dispose();  
 }  
}

# File Person.java

package org.example;  
import java.io.File;  
import java.io.FileNotFoundException; import java.io.PrintWriter;  
*/\*\*  
 \* Class Person  
 \* @version 1.0  
 \*/*public class Person {  
 public String name;  
 int weight;  
 double height;  
 int age;  
 Mouth mouth;  
 Brain brain;  
 PrintWriter fout;  
  
 */\*\*  
 \* Constructor  
 \*  
 \* @param name person name  
 \* @param height person height  
 \* @param weight person weight  
 \* @param outPutFile output file  
 \* @throws FileNotFoundException  
 \*/* public Person(String name, int weight, double height, String outPutFile) throws FileNotFoundException {  
 this.name = name;  
 this.weight = weight;  
 this.age = age;  
 this.height = height;  
 fout = new PrintWriter(new File(outPutFile));  
 mouth = new Mouth(fout, name);  
 brain = new Brain(name);  
 }  
  
 */\*\*  
 \* this.name = name; this.weight = weight; this.age = age; this.height = height;  
 \* fout = new PrintWriter(new File(outPutFile)); mouth = new Mouth(fout, name);  
 \* brain = new Brain(name);  
 \* <p>  
 \* Method to say something  
 \*/* public void say(String msg) {  
 mouth.say(msg);  
 }  
  
 */\*\*  
 \* Method to tell someone  
 \*/* public void tell(Person person, String msg) {  
 PrintMessage(name + " tells " + person.name + ": ");  
 mouth.say(msg);  
 }  
 */\*\*  
 \* Method for calculation  
 \*/* public int calculate(int a, int b)  
 {  
 int result = brain.sum(a,b);  
 PrintMessage(name + " adds two numbers: " + a + "," + b);  
 PrintMessage("Result: " + result);  
 return result;  
 }  
 */\*\*  
 \* Method returns age  
 \*/* public void getAge()  
 {  
 PrintMessage(age);  
 }  
 */\*\*  
 \* Method returns height  
 \*/* public void getHeight()  
 {  
 PrintMessage(height);  
 }  
 */\*\*  
 \* Method returns weight  
 \*/* public void getWeight()  
 {  
 PrintMessage(this.weight);  
 }  
 */\*\*  
 \* Method for calculating BMI  
 \*/* public void calculateBodyMassIndex()  
 {  
 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

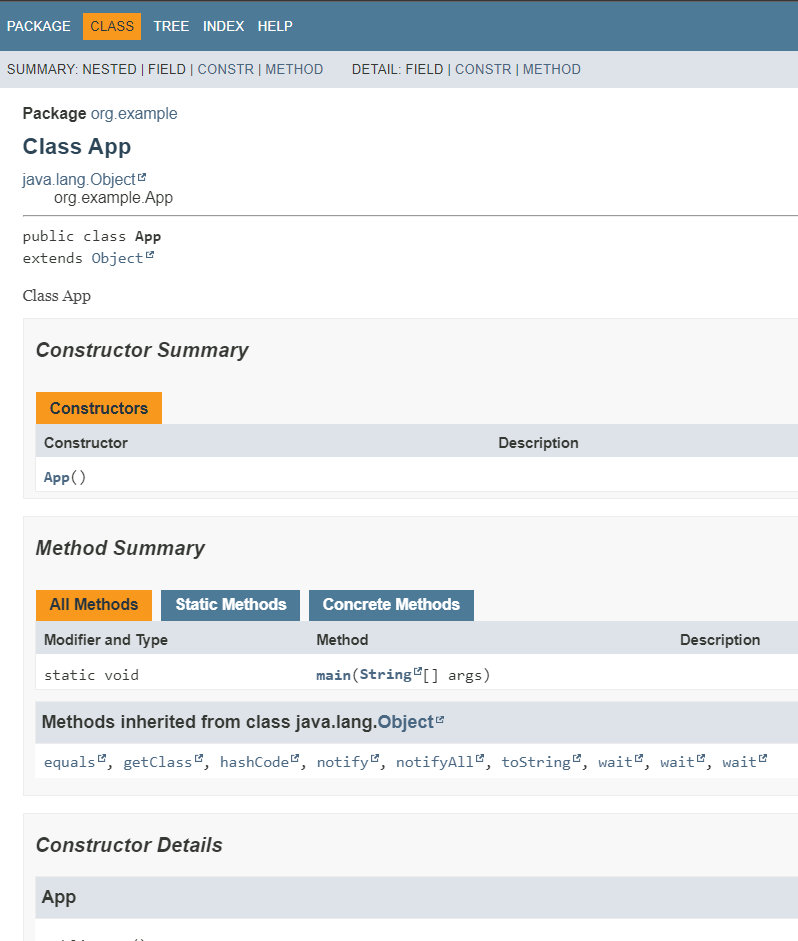
package org.example;  
  
public class Brain {  
 String name;  
 */\*\*  
 \* Constructor  
 \* @param name person name  
 \*/* public Brain(String name)  
 {  
 this.name = name;  
 }  
 */\*\*  
 \* Method returns the sum  
 \*/* public int sum(int a, int b)  
 {  
 return a + b;  
 }  
 }

**File Mouth.java**

package org.example;  
import java.io.PrintWriter;  
*/\*\*  
 \* Class Mouth  
 \* @version 1.0  
 \*/*public class Mouth {  
 private String name;  
 PrintWriter fout;  
 */\*\*  
 \* Constructor  
 \* @param name person name  
 \* @param fout PrintWriter  
 \*/* public Mouth(PrintWriter fout, String name)  
 {  
 this.name = name;  
 this.fout = fout;  
 }  
 */\*\*  
 \* Method to say something  
 \*/* public void say(String message)  
 {  
 String output = name + " says: " + message; System.*out*.println(output); fout.println(output);  
 }  
}

# Результат виконання прогами:

# 

****

**Відповіді на контрольні запитання**

1. Синтаксис визначення класу.

- public class ClassName {

// Class members (fields, methods, constructors) }

2. Синтаксис визначення методу.

- public returnType methodName(parameters) { // Method body

}

3. Синтаксис оголошення поля.

- accessModifier dataType fieldName;

4. Як оголосити та ініціалізувати константне поле?

- public static final dataType CONSTANT\_NAME = initial\_value;

5. Які є способи ініціалізації полів?

- Явна ініціалізація при оголошенні поля.

- Ініціалізація у конструкторі класу.

- Ініціалізація у блоку ініціалізації (конструкторі, статичному або

звичайному).

6. Синтаксис визначення конструктора.

- public ClassName(parameters) {

// Constructor body }

7. Синтаксис оголошення пакету.

- package packageName.subpackage;

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

- Вказати повне ім'я класу перед використанням (наприклад,

java.util.Date today = new java.util.Date();).

- Використовувати оператор import для підключення класів з інших

пакетів, щоб уникнути повторення повного імені класу.

9. В чому суть статичного імпорту пакетів?

- Статичний імпорт дозволяє підключити статичні методи і поля

класів без повного імені класу.

- Завдяки статичному імпорту, можна використовувати статичні члени класу, не

додаваючи перед ними ім'я класу.

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

- Назви пакетів повинні відповідати структурі каталогів.

- Назви загальнодоступних класів повинні співпадати з назвами

файлів, де вони розміщені.

- Після компіляції ієрархія каталогів проекту повинна відповідати

ієрархії пакетів.

- Для компіляції та запуску програми слід використовувати шляхи

до файлів та пакетів.

**Висновок:** під час виконання даної лабораторної роботи я ознайомилась з процесом розробки класів та пакетів мовою Java.