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-305

Вознюк О. 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 HumanApp.java

*/\*\*  
 \* lab 2 package  
 \*/*package KI305.Vozniuk.Lab2;  
import java.io.FileNotFoundException;  
*/\*\*  
 \* Class App  
 \* @version 1.0  
 \*/*public class HumanApp {  
  
 */\*\*  
 \* @param args  
 \* @throws FileNotFoundException  
 \*/* public static void main(String[] args) throws FileNotFoundException {  
 Human person1 = new Human("Yura Martyniuk",24,180.5,70.5, "healthy", "Lab2.txt");  
 System.*out*.println("Your name is: " + person1.getName());  
 System.*out*.println("Your age is: " + person1.getAge());  
 System.*out*.println("Your weight is: " + person1.getWeight());  
 System.*out*.println("Your height is: " + person1.getHeight());  
 person1.ShowIndex();  
 person1.TestResultsHealth();  
 System.*out*.println("Your card is using for salary: " + person1.getBankInfo());  
 System.*out*.println("Your salary is: " + person1.getSalary());  
 System.*out*.println("Your salary with taxes is: " + person1.getSalaryWithTaxes());  
 person1.dispose();  
 }  
  
}

# File Human.java

*/\*\*  
 \* lab 2 package  
 \*/*package KI305.Vozniuk.Lab2;  
import java.io.\*;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
*/\*\*  
 \* Class Human  
 \* @version 1.0  
 \*/*public class Human {  
 private String name;  
 private AthleteDetails athletics;  
 private HealthRecords health;  
 private FinancialDetails finances;  
 private PrintWriter fout;  
  
  
  
  
 */\*\*  
 \* Constructor  
 \*  
 \* @param name person name  
 \* @param height person height  
 \* @param weight person weight  
 \* @param weight person HealthResults  
 \* @param outPutFile output file  
 \* @throws FileNotFoundException  
 \*/* public Human(String name ,int age, double height, double weight, String HealthResults, String outPutFile) throws FileNotFoundException{  
 this.name = name;  
 athletics = new AthleteDetails(age, height, weight);  
 health = new HealthRecords(HealthResults);  
 finances = new FinancialDetails();  
 fout = new PrintWriter(new File(outPutFile));  
   
 }  
 */\*\*  
 \* this.name = name; athletics = new AthleteDetails(age, height, weight);  
 \* health = new HealthRecords(HealthResults); finances = new FinancialDetails();  
 \* fout = new PrintWriter(new File(outPutFile));  
 \* <p>  
 \* Method to return name  
 \*/* public String getName(){  
 return name;  
 }  
 */\*\*  
 \* Method returns age  
 \*/* public int getAge(){  
 return athletics.getAge();  
 }  
 */\*\*  
 \* Method returns height  
 \*/* public double getHeight(){  
 return athletics.getHeight();  
 }  
 */\*\*  
 \* Method returns weight  
 \*/* public double getWeight(){  
 return athletics.getWeight();  
 }  
 */\*\*  
 \* Method for calculation weight index  
 \*/* public void ShowIndex(){  
 athletics.idx();  
 }  
 */\*\*  
 \* Method for calculation is person need to get vaccinated  
 \*/* public void TestResultsHealth(){  
 if (health.getHealthResults().equals("healthy")) {  
 fout.println("You don't need get vaccinated");  
 fout.flush();  
 }else if(health.getHealthResults().equals("unhealthy")){  
 fout.println("You should get vaccinated");  
 fout.flush();  
 }else{  
 fout.println("You entered incorrect values");  
 fout.flush();  
 }  
 }  
 */\*\*  
 \* Method returns random salary from 4000 to 5000  
 \*/* public int getSalary(){  
 return finances.CalculateSalary();  
 }  
 */\*\*  
 \* Method returns salary with 19.5% taxes  
 \*/* public int getSalaryWithTaxes(){  
 return finances.getSalaryWithTaxes();  
 }  
 */\*\*  
 \* Method returns BankInfo  
 \*/* public String getBankInfo(){  
 return finances.getBankInfo();  
 }  
  
 */\*\*  
 \* Method exits  
 \*/* public void dispose(){  
 fout.flush();  
 fout.close();  
 }  
}

}

# File AthleteDetails.java

package KI305.Vozniuk.Lab2;  
  
public class AthleteDetails {  
 private String name;  
 private int age;  
 private double height;  
 private double weight;  
  
 */\*\*  
 \* Constructor  
 \* @param age person name  
 \* @param height person height  
 \* @param weight person weight  
 \*/* public AthleteDetails(int age, double height, double weight)  
 {  
  
  
 this.height = height;  
 this.weight = weight;  
 }  
  
 */\*\*  
 \* Method returns age  
 \*/* public int getAge(){  
  
 return age;  
 }  
 */\*\*  
 \* Method returns the height  
 \*/* public double getHeight() {  
 return height;  
 }  
 */\*\*  
 \* Method returns the weight  
 \*/* public double getWeight() {  
 return weight;  
 }  
 */\*\*  
 \* Method calculations weight index  
 \*/* public void idx(){  
 double y = height - weight + 10;  
 if(y <= weight + 10){  
 System.*out*.println("Your weight index is not normal");  
 }else {  
 System.*out*.println("Your weight index is normal");  
 }  
 }  
}

**File FinancialDetails.java**

package KI305.Vozniuk.Lab2;  
  
public class FinancialDetails{  
 private int salary;  
 private String bankInfo;  
 */\*\*  
 \* Constructor without parametres  
 \*/* public FinancialDetails() {  
 salary = 0;  
 bankInfo = "4444 5555 6666 7777";  
 }  
 */\*\*  
 \* Method calculate and return salary  
 \*/* public int CalculateSalary(){  
 salary = (int)(Math.*random*()\*1000 + 4000);  
 return salary;  
 }  
 */\*\*  
 \* Method returns salary with taxes  
 \*/* public int getSalaryWithTaxes(){  
 return (int) (0.805 \* salary);  
 }  
 */\*\*  
 \* Method returns bankInfo  
 \*/* public String getBankInfo(){  
 return bankInfo;  
 }  
}

**File HealthRecords.java**

package KI305.Vozniuk.Lab2;  
  
public class HealthRecords {  
 private String testResults;  
 */\*\*  
 \* Constructor  
 \* @param testResults person testResults  
 \*/* public HealthRecords(String testResults) {  
 this.testResults = testResults;  
 }  
 */\*\*  
 \* Method returns HealthResults  
 \*/* public String getHealthResults() {  
 return testResults;  
 }  
}

# 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.