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

Звіт
з лабораторної роботи № 2
з дисципліни: «Кросплатформні засоби програмування»
на тему: «Класи та пакети»

Виконав:
студент групи КІ-306
Яцків А.Р.
Прийняв:
доцент кафедри ЕОМ
Іванов Ю. С.

Львів – 2023

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

Завдання (варіант № 1)

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту (1. Людина). Програма має задовольняти наступним вимогам:

   • програма має розміщуватися в пакеті Група.Прізвище.Lab3;

   • 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 розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання

Вихідний код програми

Файл HumanApp.java

```
package KI306.Yatskiv.Lab2;

import java.io.*;

public class HumanApp {
    public static void main(String[] args) throws FileNotFoundException {
        //Створення об'єкту людини
        Human person1 = new Human("Oleg", 30, "Male","Software Developer");
        Human person2 = new Human("Alice", 25, "Female","Unemployed");
        Human person3 = new Human("David", 58, "Male","Politician");
        Human person4 = new Human("Meg", 17, "Female","Student");
        //Вивід інформації
        person1.printInfo();
        person2.printInfo();
```

```
        //Перевірка чи людина доросла
        person2.isAdult();
        //Зміна віку
        person3.setAge(45);
        //Вивід інформаціх
        person3.printInfo();

        //Зміна імені
        person4.setName("Petro");
        //Зміна статі
        person4.setGender("Male");
        //Вивід інформації
        person4.printInfo();
        //Перевірка чи доросла людина
        person4.isAdult();

        //Запис у файл
        person1.writeToLogFile();
        person2.writeToLogFile();
        person3.writeToLogFile();
        person4.writeToLogFile();
        //Збереження записаних даних
        Human.saveLogToFile();
    }
}
```

Файл Human.java

```
package KI306.Yatskiv.Lab2;

import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
/**
 * Клас представляє собою об'єкт людини.
 */
public class Human {
    private static List<String> logData = new ArrayList<>();
    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;
    }
    /**
```

```java
     * 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 відображення інформації про людину.
     */
    public void printInfo() {
        System.out.println("\n" + "Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Gender: " + gender);
        System.out.println("Occupation: " + 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;
    }

    /**
     * Метод для запису в журнал інформаціх про людину.
     */
    public void writeToLogFile() {
        String logEntry = "Name: " + name + "\n" +
                "Age: " + age + "\n" +
                "Gender: " + gender + "\n" +
                "Occupation: " + occupation + "\n";

        logData.add(logEntry);
    }
    /**
     * Метод для збереження даних у журналі та завершення роботи з файлом.
     */
    public static void saveLogToFile()  {
        try (PrintWriter writer = new PrintWriter(new
FileWriter("human_log.txt", false))) {
            for (String logEntry : logData) {
                writer.println(logEntry);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Результат виконання програми

Консоль:

```
Name: Oleg
Age: 30
Gender: Male
Occupation: Software Developer

Name: Alice
Age: 25
Gender: Female
Occupation: Unemployed
Person is adult

Name: David
Age: 45
Gender: Male
Occupation: Politician

Name: Petro
Age: 17
Gender: Male
Occupation: Student
Person is underage
```

Human_log.txt:

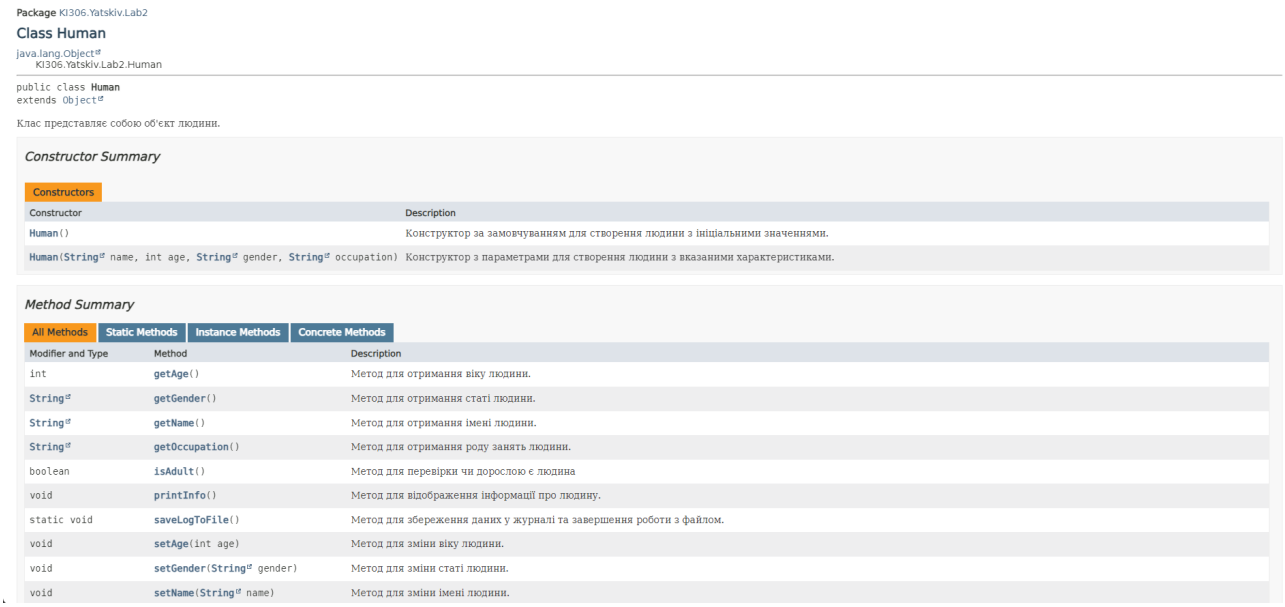
```
Name: Oleg
Age: 30
Gender: Male
Occupation: Software Developer

Name: Alice
Age: 25
Gender: Female
Occupation: Unemployed

Name: David
Age: 45
Gender: Male
Occupation: Politician

Name: Petro
Age: 17
Gender: Male
Occupation: Student
```

Фрагмент згенерованої документації



Package KI306.Yatskiv.Lab2

Class Human

java.lang.Object
KI306.Yatskiv.Lab2.Human

public class Human
extends Object

Клас представляє собою об'єкт людини.

Constructor Summary

Constructors

Constructor	Description
Human()	Конструктор за замовчуванням для створення людини з ініціальними значеннями.
Human(String name, int age, String gender, String occupation)	Конструктор з параметрами для створення людини з вказаними характеристиками.

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
int	getAge()	Метод для отримання віку людини.
String	getGender()	Метод для отримання статі людини.
String	getName()	Метод для отримання імені людини.
String	getOccupation()	Метод для отримання роду занять людини.
boolean	isAdult()	Метод для перевірки чи дорослою є людина
void	printInfo()	Метод для відображення інформації про людину.
static void	saveLogToFile()	Метод для збереження даних у журналі та завершення роботи з файлом.
void	setAge(int age)	Метод для зміни віку людини.
void	setGender(String gender)	Метод для зміни статі людини.
void	setName(String name)	Метод для зміни імені людини.

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

У ході виконання даної лабораторної роботи, я отримав цінні навички розробки класів та пакетів у мові програмування Java. Ця лабораторна робота надала мені

можливість ознайомитися з базовими конструкціями Java, такими як оголошення класів, методів та полів. Я навчився правильно структурувати свій код, визначати доступ до класів та їх членів, а також використовувати модифікатори доступу для керування видимістю.