

Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



## **Звіт**

З лабораторної роботи №2

Варіант – 3

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Класи та пакети»

Виконав: ст. гр. КІ-306

Братівник Д. А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

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

### **Завдання(Варіант 3)**

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:** Пес

**Код програми:**

**Файл DogApp.java:**

```
package org.example;

import java.io.*;

public class DogApp {
    public static void main(String[] args) throws
FileNotFoundException {
        // Створення об'єкту собаки
        Dog myDog = new Dog("Рекс", 3, "Лабрадор", false);
        // Виведення інформації про собаку
```

```

        myDog.displayInfo();
        // Зміна імені та віку собаки
        myDog.setName("Бакс");
        myDog.setAge(5);
        // Виведення інформації про собаку
        myDog.displayInfo();
        // Навчання собаки
        myDog.train();

        // Виведення інформації про собаку
        myDog.displayInfo();
        // Закриття файлу журналу
        myDog.closeLogFile();
    }
}

```

### Файл Dog.java:

```

package org.example;

import java.io.*;

/**
 * Клас представляє собою об'єкт собаки.
 */
public class Dog {
    private String name;
    private int age;
    private String breed;
    private boolean isTrained;
    private PrintWriter logFile;

    /**
     * Конструктор з параметрами для створення собаки з вказаними
     * характеристиками.
     *
     * @param name      Ім'я собаки.
     * @param age       Вік собаки.
     * @param breed     Порода собаки.
     * @param isTrained Прапорець, що вказує, чи навчена собака.
     * @throws FileNotFoundException Виникає, якщо не вдається
     * створити файл журналу.
     */
    public Dog(String name, int age, String breed, boolean
isTrained) throws FileNotFoundException {
        this.name = name;
        this.age = age;
        this.breed = breed;
        this.isTrained = isTrained;
        logFile = new PrintWriter(new File("DogLog.txt"));
    }

    /**
     * Конструктор за замовчуванням для створення собаки з
     * ініціальними значеннями.
     */
}

```

```

    *
    * @throws FileNotFoundException Виникає, якщо не вдається
створити файл журналу.
    */
    public Dog() throws FileNotFoundException {
        this.name = "Невідомо";
        this.age = 0;
        this.breed = "Невідома";
        this.isTrained = false;
        logFile = new PrintWriter(new File("DogLog.txt"));
    }

    /**
     * Метод для зміни імені собаки.
     *
     * @param name Нове ім'я собаки.
     */
    public void setName(String name) {
        this.name = name;
        logFile.println("Змінено ім'я собаки на: " + name);
        logFile.flush();
    }

    /**
     * Метод для отримання імені собаки.
     *
     * @return Ім'я собаки.
     */
    public String getName() {
        return name;
    }

    /**
     * Метод для зміни віку собаки.
     *
     * @param age Новий вік собаки.
     */
    public void setAge(int age) {
        this.age = age;
        logFile.println("Змінено вік собаки на: " + age);
        logFile.flush();
    }

    /**
     * Метод для отримання віку собаки.
     *
     * @return Вік собаки.
     */
    public int getAge() {
        return age;
    }

    /**
     * Метод для зміни породи собаки.

```

```

    *
    * @param breed Нова порода собаки.
    */
    public void setBreed(String breed) {
        this.breed = breed;
        logFile.println("Змінено породу собаки на: " + breed);
        logFile.flush();
    }

    /**
     * Метод для отримання породи собаки.
     *
     * @return Порода собаки.
     */
    public String getBreed() {
        return breed;
    }

    /**
     * Метод для навчання собаки.
     */
    public void train() {
        isTrained = true;
        logFile.println("Собаку навчено");
        logFile.flush();
    }

    /**
     * Метод для перевірки, чи навчена собака.
     *
     * @return true, якщо собака навчена; false, якщо не навчена.
     */
    public boolean isTrained() {
        return isTrained;
    }

    /**
     * Метод для запису в журнал завершення роботи з файлом.
     */
    public void closeLogFile() {
        logFile.close();
    }

    /**
     * Метод для відображення інформації про собаку.
     */
    public void displayInfo() {
        System.out.println("Ім'я собаки: " + name);
        System.out.println("Вік собаки: " + age);
        System.out.println("Порода собаки: " + breed);
        System.out.println("Навчена: " + (isTrained ? "Так" :
"Ні"));
        System.out.print("\n");
    }

```

```
}  
}
```

### Результати роботи програми:

```
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java -  
Ім'я собаки: Рекс  
Вік собаки: 3  
Порода собаки: Лабрадор  
Навчена: Ні  
  
Ім'я собаки: Бакс  
Вік собаки: 5  
Порода собаки: Лабрадор  
Навчена: Ні  
  
Ім'я собаки: Бакс  
Вік собаки: 5  
Порода собаки: Лабрадор  
Навчена: Так  
  
Process finished with exit code 0
```

DogApp.java × DogLog.txt × Dog.java ×

Змінено ім'я собаки на: Бакс

Змінено вік собаки на: 5

Собаку навчено

## Фрагмент згенерованої документації

doclab2

HTML

allclasses-index.html

HTML

allpackages-index.html

copy.svg

copy.svg

element-list

HTML

help-doc.html

index-files

HTML

index.html

legal

JS

member-search-index.js

JS

module-search-index.js

org

HTML

overview-tree.html

JS

package-search-index.js

resources

script-dir

JS

script.js

JS

search-page.js

HTML

search.html

JS

search.js

stylesheet.css

stylesheet.css

JS

tag-search-index.js

JS

type-search-index.js

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH

java.lang.Object<sup>Ⓢ</sup>

org.example.Dog

public class Dog

extends Object<sup>Ⓢ</sup>

Клас представляє собою об'єкт собаки.

Constructor Summary

Constructors

Constructor	Description
Dog()	Конструктор за замовчуванням для створення собаки з ініціальними значеннями.
Dog(String <sup>Ⓢ</sup> name, int age, String <sup>Ⓢ</sup> breed, boolean isTrained)	Конструктор з параметрами для створення собаки з вказаними характеристиками.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
void	closeLogFile()	Метод для запису в журнал завершення роботи з файлом.
void	displayInfo()	Метод для відображення інформації про собаку.
int	getAge()	Метод для отримання віку собаки.
String <sup>Ⓢ</sup>	getBreed()	Метод для отримання породи собаки.
String <sup>Ⓢ</sup>	getName()	Метод для отримання імені собаки.
boolean	isTrained()	Метод для перевірки, чи навчена собака.
void	setAge(int age)	Метод для зміни віку собаки.
void	setBreed(String <sup>Ⓢ</sup> breed)	Метод для зміни породи собаки.

## 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 додаючи 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.