Міністерство освіти і науки України

Національний університет "Львівська політехніка"

Кафедра ЕОМ



Звіт

3 лабораторної роботи №2

Варіант – 3

3 дисципліни: «Кросплатформні засоби програмування»

На тему: «Класи та пакети»

Виконав: ст. гр. КІ-306

Братівник Д. А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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 класдрайвер;
- 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 ïï

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.

Варіант завдання: Пес

Код програми:

Файл DogApp.java:

```
public class DogApp {
   public static void main(String[] args) throws
FileNotFoundException {
       Dog myDog = new Dog("Рекс", 3, "Лабрадор", false);
```

```
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:

```
public class Dog {
     * @param name
     * @param age Вік собаки.
* @param breed Порода собаки.
     * @param age
     * @param isTrained Прапорець, що вказує, чи навчена собака.
     * @throws FileNotFoundException Виникає, якщо не вдається
    public Dog(String name, int age, String breed, boolean
isTrained) throws FileNotFoundException {
        this.age = age;
        this.breed = breed;
        this.isTrained = isTrained;
```

```
* @throws FileNotFoundException Виникає, якщо не вдається
public Dog() throws FileNotFoundException {
 * @param name Нове ім'я собаки.
public void setName(String name) {
   logFile.flush();
 * @return Ім'я собаки.
public String getName() {
 * @param age Новий вік собаки.
public void setAge(int age) {
   this.age = age;
   logFile.println("Змінено вік собаки на: " + age);
public int getAge() {
```

```
* @param breed Нова порода собаки.
public void setBreed(String breed) {
   this.breed = breed;
   logFile.flush();
 * @return Порода собаки.
public String getBreed() {
 * Метод для навчання собаки.
   logFile.flush();
* @return true, якщо собака навчена; false, якщо не навчена.
public boolean isTrained() {
public void closeLogFile() {
   logFile.close();
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" :
   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

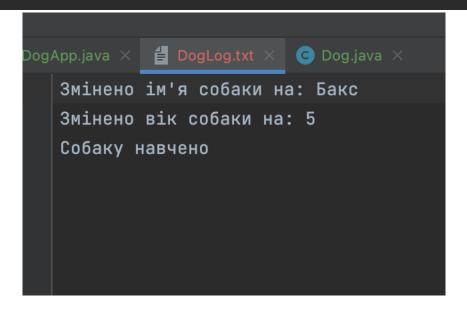
Ім'я собаки: Бакс

Вік собаки: 5

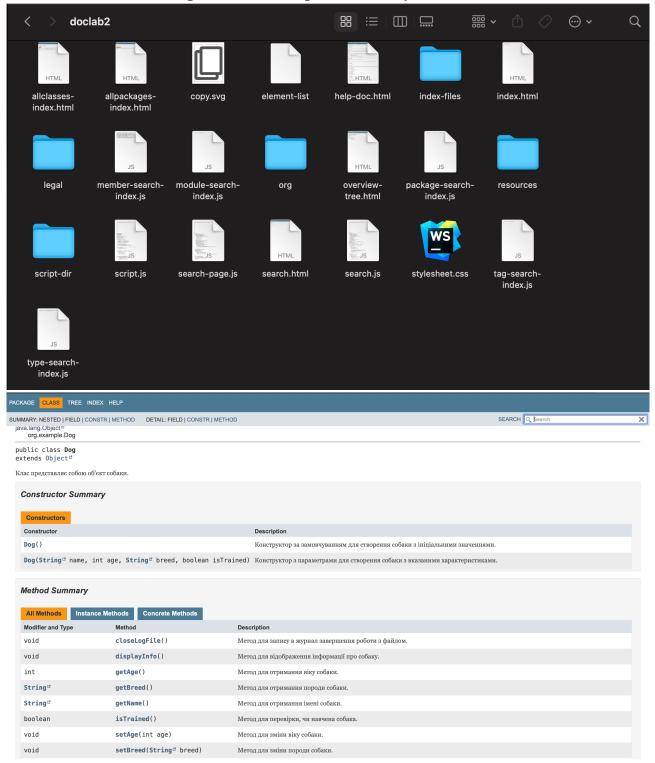
Порода собаки: Лабрадор

Навчена: Так

Process finished with exit code 0



Фрагмент згенерованої документації



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.