

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



**ЗВІТ**  
**з лабораторної роботи № 3**  
**з дисципліни: «Кросплатформенні засоби програмування»**  
**на тему: «КЛАСИ ТА ПАКЕТИ»**

Виконав: ст. гр. КІ-35

Дністряк Я. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю.С

Львів-2022

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

### ЗАВДАННЯ (Варіант - 4)

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту.

## 4. Кіт

Рис.1. Предметна область

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.

### Текст програми:

#### Лістинг Cat.java

```
/**
 * lab 3 package
 */
package KI35.Dnistrian.Lab3;
import java.io.*;
import java.util.*;
/**
 * Class <code>Cat</code> implements cat
 * @author Slaweeek
 * @version 1.0
```

```

*
*/
public class Cat {
    private Eyes eyes;
    private Paws paws;
    private Fur fur;
    private PrintWriter fout;

    /**
     * Constructor
     * @param <code>vision</code> Vision mode
     * @param <code>colorEye</code> The color of our cat's eye
     * @param <code>distanceToPrey</code> Distance to prey
     * @param <code>speed</code> The speed of our cat
     * @param <code>condition</code> The position claw of our cat
     * @param <code>colorFur</code> The color fur of our cat
     * @param <code>lengthFur</code> The length of our cat's fur
     * @throws FileNotFoundException
     */
    public Cat(Eyes.Regime vision,String colorEye,float distanceToPrey,float speed,
Paws.clawCondition condition,String colorFur,int lengthFur) throws
FileNotFoundException
    {
        eyes = new Eyes(vision,colorEye,distanceToPrey);
        paws=new Paws(speed,condition);
        fur = new Fur(colorFur,lengthFur);
        fout = new PrintWriter(new File("Cat.txt"));
    }

    /**
     * Constructor
     * @throws FileNotFoundException
     */
    public Cat() throws FileNotFoundException {
        eyes = new Eyes(Eyes.Regime.dilated,"Brown",120.0F);
        paws=new Paws(16.5F, Paws.clawCondition.within);
        fur = new Fur("Black",2);
        fout = new PrintWriter(new File("Cat.txt"));
    }

    /**
     * Method returns the vision mode of our cat
     * @return vision mode
     */
    public Eyes.Regime getVision(){
        return eyes.getRegime();
    }

    /**
     * Method returns the cat's claw condition
     * @return claw condition
     */
    public Paws.clawCondition getClawCondition(){
        return paws.getClaw();
    }

    /**
     * Method returns the speed of our cat
     * @return speed of our cat
     */
    public float getSpeedCat() {
        return paws.getSpeed();
    }
}

```

```

/**
 * Method returns the distance to prey
 * @return distance to prey
 */
public float getDistanceToPreyCat() {
    return eyes.getDistanceToPrey();
}

/**
 * Method returns the length of cat's fur
 * @return length of cat's fur
 */
public int getLengthFurCat() {
    return fur.getLegth();
}

/**
 * Method returns the color of cat's eyes
 * @return color of cat's eyes
 */
public String getColorEyeCat() {
    return eyes.getColorEye();
}

/**
 * Method returns the color of cat's eyes
 * @return color of cat's fur
 */
public String getColorFurCat() {
    return fur.getColorFur();
}

/**
 * Method reports the color of the cat's eyes and fur
 * @param <code>eye</code> color of cat's eyes
 * @param <code>fur</code> color of cat's fur
 */
public void colorCat(String eye, String fur) {
    System.out.println("The color of the cat's eyes: " + eye + "\nThe color of
the cat's fur: "+ fur );
    fout.println("The color of the cat's eyes: " + eye + "\nThe color of the
cat's fur: "+ fur);
}

/**
 * Method reports the behavior of the cat
 * @param <code>vision</code> Vision mode
 * @param <code>condition</code>The position claw of our cat
 */
public void pet(Eyes.Regime vision,Paws.clawCondition condition)
{
    if(vision==Eyes.Regime.dilated &&condition==Paws.clawCondition.out) {
        System.out.println("Do not touch, the cat focused on the prey!!!");
        fout.println("Do not touch, the cat focused on the prey!!!");
    }else if(vision==Eyes.Regime.narrow &&condition==Paws.clawCondition.out) {
        System.out.println("You can pet the cat, but there will be
scratches");
        fout.println("You can pet the cat, but there will be scratches");
    }else {
        System.out.println("You can pet the cat");
        fout.println("You can pet the cat");
    }
}

```

```

/**
 * Method calculates the cat's hunting speed
 * @param speed Speed of our cat
 * @param distanceToPrey Distance to prey
 */
public void timeToEat(float speed,float distanceToPrey) {
    float res;
    res= distanceToPrey/speed;
    String time=String.format("Time to hunt for prey: %.2f s. \n" ,res);
    System.out.print(time);
    fout.print(time);
}

/**
 * Method counts the number of scratches by the cat
 */
public void numberOfScratches() {
    Random numOfScratches = new Random();
    int number = numOfScratches.nextInt(10);
    System.out.println("The number of scratches caused by the cat: " +
number);
    fout.println("The number of scratches caused by the cat:" + number );
}

/**
 * Method counts the time of cat petting
 */
public void catPettingTime() {
    Random time = new Random();
    int number = time.nextInt(20);
    System.out.println("Cat petting time : " + number + " minute");
    fout.println("Cat petting time : " + number + " minute");
}

/**
 * Method sets the cat's fur type
 * @param lengthFur The length of our cat's fur
 */
public void typeOfCatFur(int lengthFur) {
    if(lengthFur>=5) {
        System.out.println("Type of cat fur: Long-haired");
        fout.println("Type of cat fur: Long-haired");
    }
    else {
        System.out.println("Type of cat fur: Short-haired");
        fout.println("Type of cat fur: Short-haired");
    }
}

/**
 * Method releases used recourses
 */
public void dispose()
{
    fout.close();
}

}

class Eyes{
    enum Regime{dilated,narrow};
    private Regime vision;

```

```

private String colorEye;
private float distanceToPrey;

/**
 * Constructor
 * @param<code>vision</code> Vision mode
 * @param<code>colorEye</code> The color of our cat's eye
 * @param<code>distanceToPrey</code> Distance to prey
 */
public Eyes(Regime vision, String colorEye, float distanceToPrey ) {
    this.vision=vision;
    this.colorEye=colorEye;
    this.distanceToPrey=distanceToPrey;
}

```

```

/**
 * Method sets the vision mode
 * @param vision Vision mode
 */
public void setRegime(Regime vision) {
    this.vision=vision;
}

/**
 * Method returns the vision mode
 * @return vision
 */
public Regime getRegime() {
    return vision;
}

/**
 * Method returns the color eyes of our cat
 * @return color eyes of our cat
 */
public String getColorEye() {
    return colorEye;
}

/**
 * Method sets the distance to prey
 * @param distance Distance to prey
 */
public void setDistanceToPrey(float distance) {
    this.distanceToPrey=distance;
}

/**
 * Method returns the distance to prey
 * @return Distance to prey
 */
public float getDistanceToPrey() {
    return distanceToPrey;
}

```

```

class Paws{
    enum clawCondition{out,within};

    private float speed;
    private clawCondition clawCondition;

    /**

```

```

    * Constructor
    * @param <code>speed</code> The speed of our cat
    * @param <code>condition</code> The position claw of our cat
    */
    public Paws(float speed, clawCondition condition){
        this.speed= speed;
        this.clawCondition=condition;
    }

    /**
     * Method sets the speed of our cat
     * @param speed Speed of our cat
     */
    public void setSpeed(float speed) {
        this.speed=speed;
    }

    /**
     * Method returns the speed of our cat
     * @return speed
     */
    public float getSpeed(){
        return speed;
    }

    /**
     * Method sets the cat's claw condition
     * @param condition The position claw of our cat
     */
    public void setClaw(clawCondition condition ) {
        this.clawCondition=condition;
    }

    /**
     * Method returns the cat's claw condition
     * @return cat's claw condition
     */
    public clawCondition getClaw() {
        return clawCondition;
    }
}

```

```

class Fur{
    private String colorFur;
    private int lengthFur;

    /**
     * Constructor
     * @param <code>colorFur</code> The color fur of our cat
     * @param <code>lengthFur</code> The length of our cat's fur
     */
    public Fur(String colorFur,int lengthFur) {
        this.colorFur=colorFur;
        this.lengthFur=lengthFur;
    }

    /**
     * Method sets the color fur of our cat
     * @param colorFur color fur of our cat
     */
    public void setColorFur(String colorFur) {

```

```

        this.colorFur=colorFur;
    }

    /**
     * Method returns color fur of our cat
     * @return color fur of our cat
     */
    public String getColorFur() {
        return colorFur;
    }

    /**
     * Method sets the length of our cat's fur
     * @param lengthFur The length of our cat's fur
     */
    public void setLengthFur(int lengthFur) {
        this.lengthFur=lengthFur;
    }

    /**
     * Method returns the length of our cat's fur
     * @return The length of our cat's fur
     */
    public int getLegth() {
        return lengthFur;
    }
}

```

## Лістинг CatApp.java

```

/**
 * lab 3 package
 */
package KI35.Dnistrian.Lab3;
import java.io.*;

/**
 * Cat Application class implements main method for Cat class possibilities
demonstration
 * @author Slaweeek
 * @version 1.0
 */
public class CatApp {

    /**
     *
     * @param args
     * @throws FileNotFoundException
     */
    public static void main(String[] args) throws FileNotFoundException{

        Cat tom=new
Cat(Eyes.Regime.dilated,"Blue",100.0F,13.5F,Paws.clawCondition.within,"Gray",4);

        tom.colorCat(tom.getColorEyeCat(),tom.getColorFurCat());
        tom.typeOfCatFur(tom.getLengthFurCat());
        tom.pet(tom.getVision(),tom.getClawCondition());

        if(tom.getVision()==Eyes.Regime.dilated &&
tom.getClawCondition()==Paws.clawCondition.out)

```

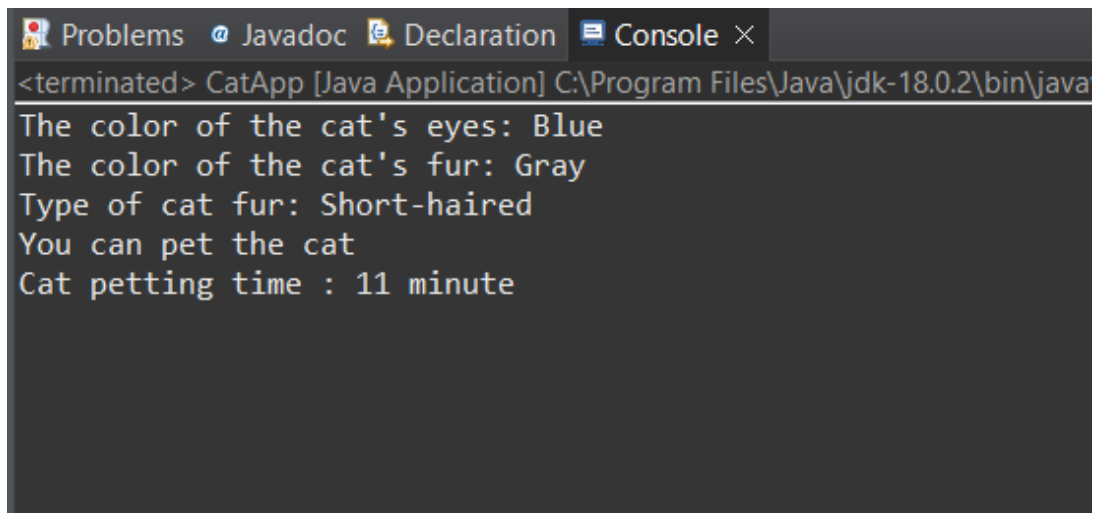


```

        tom.timeToEat(tom.getSpeedCat(), tom.getDistanceToPreyCat());
    else if(tom.getVision()==Eyes.Regime.narrow &&
tom.getClawCondition()==Paws.clawCondition.out)
        tom.numberOfScratches();
    else
        tom.catPettingTime();
        tom.dispose();
    }
}

```

### Результат виконання програми

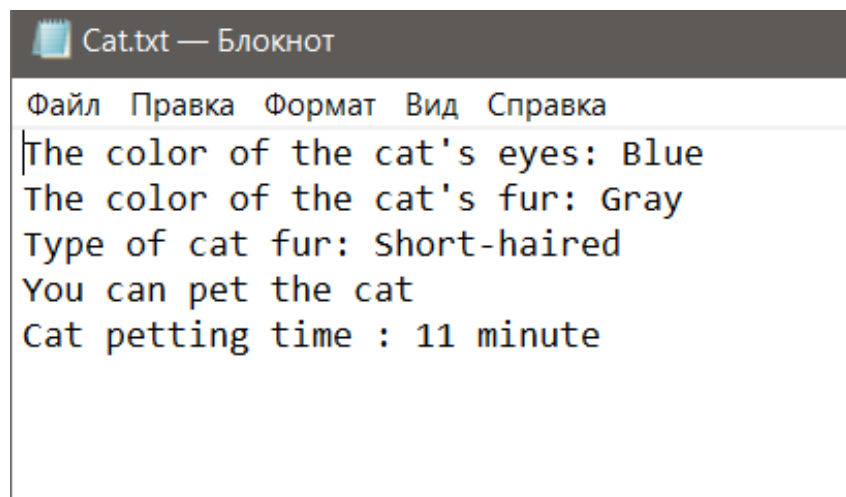


```

Problems Javadoc Declaration Console ×
<terminated> CatApp [Java Application] C:\Program Files\Java\jdk-18.0.2\bin\java
The color of the cat's eyes: Blue
The color of the cat's fur: Gray
Type of cat fur: Short-haired
You can pet the cat
Cat petting time : 11 minute

```

Рис.2.Результати роботи програми



```

Cat.txt — Блокнот
Файл Правка Формат Вид Справка
The color of the cat's eyes: Blue
The color of the cat's fur: Gray
Type of cat fur: Short-haired
You can pet the cat
Cat petting time : 11 minute

```

Рис. 3.Результат виводу у файл

## Фрагмент згенерованої документації

The screenshot shows a web-based Java documentation interface. At the top, there is a navigation bar with links: PACKAGE, CLASS (highlighted), USE, TREE, INDEX, and HELP. Below this is a secondary navigation bar with links: SUMMARY: NESTED | FIELD | CONSTR | METHOD, and a search bar labeled 'SEARCH' with a magnifying glass icon. The main content area displays the following information:

- Package:** Ki35.Dnistrian.Lab3
- Class Cat**
- java.lang.Object** (with a small icon)
- Ki35.Dnistrian.Lab3.Cat**

---

`public class Cat`  
`extends Object`

Class Cat implements cat

Version:  
1.0

Author:  
Slaweeek

---

**Constructor Summary**

**Constructors**

Constructor	Description
<code>Cat()</code>	Constructor
<code>Cat(Eyes.Regime vision, String colorEye, float distanceToPrey, float speed, Paws.clawCondition condition, String colorFur, int lengthFur)</code>	Constructor

---

**Method Summary**

**All Methods** | **Instance Methods** | **Concrete Methods**

Рис .4. Результат автоматичної генерації документації

### Відповіді на контрольні запитання:

1. Як оголосити та ініціалізувати константне поле?

Приклад оголошення константного поля:

```
private final int i;
```

Ініціалізацію полів при створенні об'єкту можна здійснювати трьома способами:

- у конструкторі;
- явно при оголошенні поля;
- у блоці ініціалізації (виконується перед виконанням конструктора).

2. Синтаксис оголошення пакету.

Синтаксис оператора package:

```
package НазваПакету{.НазваПідпакету};
```

**Висновок:** виконуючи лабораторну роботу №3 я ознайомився з процесом розробки класів та пакетів мовою Java.