

Міністерство освіти і науки України Національний  
університет «Львівська політехніка»



***Звіт***

до лабораторної роботи №3

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Класи та пакети»

Варіант 4

Виконав:  
Ст. групи КІ-34  
Демчик Н.О.

Прийняв:  
к.т.н., доцент  
Іванов Ю.С.

Львів 2022

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.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.

Варіант завдання: 4. Кіт

### Код роботи

*Cat.java*

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class Cat {
    private Float weight;
    private String name;
    private Integer age;
    private String color;
    private PrintWriter fout;
    private Breed classBreed;
    private FoodAddiction addiction;
    private FleaInfestation flea;

    public Cat(float weight, String breed, String name, int age) throws
FileNotFoundException {
        this.weight = weight;
        this.name = name;
        this.age = age;
        classBreed = new Breed(breed);
        addiction = new FoodAddiction();
        flea = new FleaInfestation();
        fout = new PrintWriter(new File("CatLog.txt"));
    }
}
```

```
    public Cat(float weight, String breed, String name, int age, String color)
throws FileNotFoundException {
    this(weight, breed, name, age);
    this.color = color;
}

// this classes methods
public float getWeight() {
    return weight;
}

public String getName() {
    return name;
}

public int getAge() {
    return age;
}

public String getColor() {
    return color;
}

public void myMood(String mood) {
    System.out.println("I am: " + mood + ". And this is me ");
    fout.println("I am: " + mood + ". And this is me ");
    fout.flush();
}

public void makeSound() {
    System.out.println("Meow");
    fout.println("Meow");
    fout.flush();
}

public void doNotSleepAt3AM() {
    System.out.println("I'm scraping the door");
    fout.println("I'm scraping the door");
    fout.flush();
}

public void sleeping(int time) {
    System.out.println("Sleeping for " + time + "hours");
    fout.println("Sleeping for " + time + "hours");
    fout.flush();
}

public void eat(String food) {
    System.out.println("I'm eating " + food);
    fout.println("I'm eating " + food);
    fout.flush();
}
```

```

public void watchInTheWindow() {
    System.out.println("Oh wow, there is a bird!!");
    fout.println("Oh wow, there is a bird!!");
    fout.flush();
}

public boolean isIll() {
    if (age < 8) {
        return false;
    } else {
        return true;
    }
}

// 3 classes methods

public void setFur(String fur) {
    classBreed.setFur(fur);
}

public void setNature(String nature) {
    classBreed.setNature(nature);
}

public void setFoodTime(int foodTime) {
    addiction.setFoodTimes(foodTime);
}

public void setFoodGram(int foodGram) {
    addiction.setFoodGrams(foodGram);
}

public void setFlea(boolean yes) {
    flea.setFlea(yes);
}

public void setPercentage(int per) {
    flea.setPercentageOfFlea(per);
}

public void printAllCharacteristics() {
    System.out.println("Full Properties START: -----");
    fout.println("Full Properties START: -----");

    // this class
    System.out.println("Name: " + getName());
    fout.println("Name: " + getName());

    System.out.println("Age: " + getAge());
    fout.println("Age: " + getAge());

    if (color != null) {
        System.out.println("Color: " + getColor());
        fout.println("Color: " + getColor());
    }
}

```

```

    }

    System.out.println("Weight: " + getWeight());
    fout.println("Weight: " + getWeight());

    System.out.println("Am i ill - " + (isIll() ? "Yes" : "No"));
    fout.println("Am i ill - " + (isIll() ? "Yes" : "No"));

    // 3 classes
    classBreed.printBreed();
    fout.println("Breed is : " + classBreed.getBreed());
    fout.println("Fur is : " + classBreed.getFur());
    fout.println("Nature is : " + classBreed.getNature());

    flea.printFlea();
    fout.println("Has flea ? : " + flea.getFlea());
    fout.println("Possibility to get flea : " + flea.getFlea());

    addiction.printFood();
    fout.println("How many times needs to eat ? : " +
addiction.getFoodTimes());
    fout.println("Food in grams for cat for 1 bowl: " +
addiction.getFoodGrams());

    System.out.println("Full Properties END: -----");
    fout.println("Full Properties END: -----");
    fout.flush();
}

public void dispose() {
    fout.close();
}
}

```

## *CatMain.java*

```
public class CatMain {  
    public static void main(String[] args) throws Exception {  
        Cat Simba = new Cat(5.5f, "Scottish", "Simba", 4, "Grey");  
  
        // in main class Cat.java not in  
        Simba.myMood("soo happy xD");  
        Simba.makeSound();  
        Simba.doNotSleepAt3AM();  
        Simba.sleeping(2);  
        Simba.watchInTheWindow();  
        Simba.eat("Fish");  
  
        // in 3 types of classes  
        Simba.setFoodGram(34);  
        Simba.setFlea(true);  
        Simba.setFur("thick");  
        Simba.setNature("calm");  
        Simba.setFoodTime(2);  
        // print from all classes  
        Simba.printAllCharacteristics();  
        Simba.dispose();  
  
    }  
}
```

## *FleaInfestation.java*

```
public class FleaInfestation {
    private boolean isFlea;
    private int percentageOfFlea;

    FleaInfestation() {
        isFlea = false;
        percentageOfFlea = 50;
    }

    public void printFlea() {
        System.out.println("Has flea ? : " + isFlea);
        System.out.println("Possibility to get flea : " + percentageOfFlea);
    }

    public boolean getFlea() {
        return isFlea;
    }

    public int getPercentage() {
        return percentageOfFlea;
    }

    public void setFlea(boolean fle) {
        isFlea = fle;
    }

    public void setPercentageOfFlea(int percentage) {
        percentageOfFlea = percentage;
    }
}
```

## *Breed.java*

```
public class Breed {
    private String breed;
    private String fur;
    private String nature;

    Breed(String breed) {
        this.breed = breed;
        this.fur = "normal";
        this.nature = "normal";
    }

    public void printBreed() {
        System.out.println("Breed is : " + breed);
        System.out.println("Fur is : " + fur);
        System.out.println("Nature is : " + nature);
    }

    public void setFur(String fur) {
        this.fur = fur;
    }

    public void setNature(String nature) {
        this.nature = nature;
    }

    public String getBreed() {
        return breed;
    }

    public String getFur() {
        return fur;
    }

    public String getNature() {
        return nature;
    }
}
```



## *FoodAddiction.java*

```
public class FoodAddiction {
    private int foodTimesPerDay;
    private double foodInGrams;

    FoodAddiction() {
        foodTimesPerDay = 3;
        foodInGrams = 40.0f;
    }

    public void setFoodTimes(int foodTime) {
        foodTimesPerDay = foodTime;
    }

    public void setFoodGrams(int foodGrams) {
        foodInGrams = foodGrams;
    }

    public int getFoodTimes() {
        return foodTimesPerDay;
    }

    public double getFoodGrams() {
        return foodInGrams;
    }

    public void printFood() {
        System.out.println("How many times needs to eat ? : " + foodTimesPerDay);
        System.out.println("Food in grams for cat for 1 bowl : " + foodInGrams);
    }
}
```

## Результат виконання програми

- Txt-файл з записаною інформацією:

```
≡ CatLog.txt
1  I am: soo happy xD. And this is me
2  Meow
3  I'm scraping the door
4  Sleeping for 2hours
5  Oh wow, there is a bird!!
6  I'm eating Fish
7  Full Properties START: -----
8  Name: Simba
9  Age: 4
10 Color: Grey
11 Weight: 5.5
12 Am i ill - No
13 Breed is : Scottish
14 Fur is : thick
15 Nature is : calm
16 Has flea ? : true
17 Possibility to get flea : true
18 How many times needs to eat ? : 2
19 Food in grams for cat for 1 bowl: 34.0
20 Full Properties END: -----
21
```

- Консоль

```
Nazan@DESKTOP-MHVAB37 MINGW64 /d/!!!Навчання/5 Семестр/КЗП/Labs/CPPT_Demchyk_NO_KI-34/DemchykLabs/Lab3 (004-1
$ cd d:\\\\!\\!\\!Навчання\\5\\ Семестр\\КЗП\\Labs\\CPPT_Demchyk_NO_KI-34\\DemchykLabs\\Lab3 ; /usr/bin/env C:\\
\\Local\\Programs\\Eclipse\\ Adoptium\\jdk-17.0.4-hotspot\\bin\\java.exe -XX:+ShowCodeDetailsInExceptionMess
вчання\\5\\ Семестр\\КЗП\\Labs\\CPPT_Demchyk_NO_KI-34\\DemchykLabs\\Lab3\\bin CatMain
I am: soo happy xD. And this is me
Meow
I'm scraping the door
Sleeping for 2hours
Oh wow, there is a bird!!
I'm eating Fish
Full Properties START: -----
Name: Simba
Age: 4
Color: Grey
Weight: 5.5
Am i ill - No
Breed is : Scottish
Fur is : thick
Nature is : calm
Has flea ? : true
Possibility to get flea : 50
How many times needs to eat ? : 2
Food in grams for cat for 1 bowl : 34.0
Full Properties END: -----
```

## Відповіді на контрольні запитання:

1. `[public] class Назва Класу {`  
    `[Методи]`  
    `[Змінні]`  
    `[Поля]`  
    `[Конструктори]`  
    `}`
2. `[СпецифікаторДоступу] Тип назваМетоду([параметри]) [throws класи] {`  
    `[Тіло методу] [return [значення]];`  
    `}`
3. `[СпецифікаторДоступу] [static] [final] Тип НазваПоля [=`  
    `ПочатковеЗначення];`
4. Використати `[final]`, тобто/наприклад `private final int i;`
5. Ініціалізацію полів при створенні об'єкту можна здійснювати трьома способами:
  - у конструкторі;
  - явно при оголошенні поля;
  - у блоці ініціалізації (виконується перед виконанням конструктора).

Якщо поле не ініціалізується жодним з цих способів, то йому присвоюється значення за замовчуванням.

6. `[СпецифікаторДоступу] НазваКласу([параметри]) {`  
    `Тіло конструктора`  
    `}`

7. `package НазваПакету{.НазваПідпакету};`

8. Клас може використовувати всі класи з власного пакету і всі загальнодоступні класи з інших пакетів. Доступ до класів з інших пакетів можна отримати двома шляхами:

1. вказуючи повне ім'я пакету перед іменем кожного класу

2. використовуючи оператор `import`, що дозволяє підключати як один клас так і всі загальнодоступні класи пакету, позбавляючи необхідності записувати імена класів з вказуванням повної назви пакету перед ними.

9. Статичний імпорт дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.

10. Файл, каталоги повинні бути строго структурованими. Чітка ієрархія, назви пакетів та підпакетів повинні співпадати з назвами каталогів де вони розміщуються.

**Висновок:** я ознайомився з процесом розробки класів та пакетів мовою Java..