Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 2

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «Класи та пакети»

**Виконала:**

студентка групи *КІ-306*

*Максимчук Катерина*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

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

**Завдання ( Варіант 9 «Рослина» ):**

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.

**Вихідний код програми**

import java.io.FileNotFoundException;  
  
*/\*\*  
 \* Class <code>Flower</code> implements flower  
 \* @author Kateryna Maksymchuk  
 \* @version 1.0  
 \*/*public class Flower {  
 private int size;  
  
 */\*\*  
 \* @param size  
 \*/* public void setSize(int size) {  
 this.size = size;  
 }  
  
 */\*\*  
 \* @return size  
 \*/* public int getSize() {  
 return size;  
 }  
  
 */\*\*  
 \* Constructor  
 \* @param size size of flower  
 \*/* public Flower(int size)  
 {  
 this.size = size;  
 }  
 */\*\*  
 \* Constructor  
 \*/* public Flower(){  
 this.size = 1;  
 }  
  
 */\*\*  
 \* Method prints info about flower  
 \*/* public void Print()  
 {  
 System.*out*.println("My flower is " + size);  
 Plant.*fout*.println("My flower is " + size);  
 Plant.*fout*.flush();  
  
 }  
 */\*\*  
 \* Method makes flower grow using waterpower and sunpower  
 \* @param waterpower amount of waterpower  
 \* @param sunpower amount of sunpower  
 \* @return Flower's size  
 \*/* public int GetWaterSunPower(int waterpower, int sunpower) {  
 setSize((waterpower + sunpower) \* size);  
 Plant.*fout*.println("Flower grows to " + getSize());  
 System.*out*.println("Flower grows to " + getSize());  
 Plant.*fout*.flush();  
  
 return getSize();  
 }  
 */\*\*  
 \* Method that say if flower enough size to reproduce  
 \* @return true or false  
 \*/* public boolean IfCanReproduce(){  
 if(size > 75)  
 return true;  
 return false;  
 }  
  
}

import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
*/\*\*  
 \* Class <code>Leaf</code> implements leaf of plant  
 \* @author Kateryna Maksymchuk  
 \* @version 1.0  
 \*/*public class Leaf {  
 private int size;  
  
 */\*\*  
 \* @param size  
 \*/* public void setSize(int size) {  
 this.size = size;  
 Print();  
 }  
  
 */\*\*  
 \* @return  
 \*/* public int getSize() {  
 return size;  
 }  
  
 */\*\*  
 \* @param size  
 \*/* //constructors  
 public Leaf(int size)  
 {  
 this.size = size;  
 }  
  
 */\*\*  
 \*  
 \*/* public Leaf(){  
 this.size = 1;  
 }  
  
 */\*\*  
 \*  
 \*/* public void Print()  
 {  
 System.*out*.println("My leaf is " + size);  
 Plant.*fout*.println("My leaf is " + size);  
 Plant.*fout*.flush();  
  
 }  
  
 */\*\*  
 \* @param sunpower  
 \* @return  
 \*/* public int GetSunPower(int sunpower) {  
 setSize(getSize() + sunpower/2);  
 System.*out*.println("Leaf send " + (sunpower/2) + " energy to flower");  
 Plant.*fout*.println("Leaf send " + (sunpower/2) + " energy to flower");  
 Plant.*fout*.flush();  
  
 return sunpower/2;  
 }  
  
 */\*\*  
 \* @return  
 \*/* public boolean IfCanDry(){  
 if(size > 150)  
 return true;  
 return false;  
 }  
  
}

import java.io.File;  
import java.io.IOException;  
import java.io.PrintWriter;  
  
*/\*\*  
 \*  
 \*/*public class Plant implements AutoCloseable{  
 private int age;  
 private Leaf leaf;  
 private Flower flower;  
 private Stem stem;  
 public static PrintWriter *fout*;  
  
  
 */\*\*  
 \*  
 \*/* public Plant()  
 {  
 leaf = new Leaf();  
 flower = new Flower();  
 stem = new Stem();  
 try {  
 *fout* = new PrintWriter(new File("PlantTxt.txt"));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* @param sizeLeaf  
 \* @param sizeFlower  
 \* @param sizeStem  
 \*/* public Plant(int sizeLeaf, int sizeFlower, int sizeStem){  
 try {  
 *fout* = new PrintWriter(new File("PlantTxt.txt"));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 leaf = new Leaf(sizeLeaf);  
 flower = new Flower(sizeFlower);  
 stem = new Stem(sizeStem);  
 }  
  
 */\*\*  
 \* @return  
 \*/* public Flower getFlower() {  
 return flower;  
 }  
  
 */\*\*  
 \* @param flower  
 \*/* public void setFlower(Flower flower) {  
 this.flower = flower;  
 }  
  
 */\*\*  
 \* @return  
 \*/* public Stem getStem() {  
 return stem;  
 }  
  
 */\*\*  
 \* @param stem  
 \*/* public void setStem(Stem stem) {  
 this.stem = stem;  
 }  
  
 */\*\*  
 \* @return  
 \*/* public Leaf getLeaf() {  
 return leaf;  
 }  
  
 */\*\*  
 \* @param leaf  
 \*/* public void setLeaf(Leaf leaf) {  
 this.leaf = leaf;  
 }  
  
 */\*\*  
 \*  
 \*/* public void Print(){  
 leaf.Print();  
 flower.Print();  
 stem.Print();  
 }  
  
 */\*\*  
 \* @param waterpower  
 \* @param sunpower  
 \*/* public void Grow(int waterpower, int sunpower){  
 //stem.GetWaterPower(waterpower);  
 //leaf.GetSunPower(sunpower);  
 //  
 flower.GetWaterSunPower(stem.GetWaterPower(waterpower), leaf.GetSunPower(sunpower));  
 System.*out*.println("Plant size is " + (flower.getSize() + leaf.getSize() + stem.getSize()));  
 *fout*.println("Plant size is " + (flower.getSize() + leaf.getSize() + stem.getSize()));  
 *fout*.flush();  
  
  
 }  
  
 */\*\*  
 \*  
 \*/* public void IfFlowerCanReproduce(){  
 if(flower.IfCanReproduce()) {  
 System.*out*.println("Flower can reproduce");  
 *fout*.println("Flower can reproduce");  
 }  
 else {  
 System.*out*.println("Flower can`t reproduce");  
 *fout*.println("Flower can't reproduce");  
 }  
 *fout*.flush();  
  
 }  
  
 */\*\*  
 \*  
 \*/* public void IfLeafCanDry(){  
 if(leaf.IfCanDry()) {  
 System.*out*.println("Leaf can dry");  
 Plant.*fout*.println("Leaf can dry");  
 }  
 else {  
 System.*out*.println("Leaf can`t dry");  
 *fout*.println("Leaf can't dry");  
 }  
 *fout*.flush();  
  
 }  
 */\*\*  
 \* Method closes the file  
 \* @throws Exception  
 \*/* @Override  
 public void close() throws Exception {  
 if (*fout* != null) {  
 *fout*.flush();  
 *fout*.close();  
 }  
 }  
}

import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.PrintWriter;  
  
*/\*\*  
 \*  
 \*/*public class Stem {  
 */\*\*  
 \*  
 \*/* private int size;  
  
 */\*\*  
 \* @param size  
 \*/* public void setSize(int size) {  
 this.size = size;  
 Print();  
 }  
  
 */\*\*  
 \* @return  
 \*/* public int getSize() {  
 return size;  
 }  
  
 */\*\*  
 \* @param size  
 \*/* //constructors  
 public Stem(int size)  
 {  
 this.size = size;  
 }  
  
 */\*\*  
 \*  
 \*/* public Stem(){  
  
 this.size = 2;  
 }  
  
 */\*\*  
 \*  
 \*/* public void Print()  
 {  
 System.*out*.println("My stem is " + size);  
 Plant.*fout*.println("My stem is " + size);  
 Plant.*fout*.flush();  
  
 }  
  
 */\*\*  
 \* @param waterpower  
 \* @return  
 \*/* public int GetWaterPower(int waterpower) {  
 setSize(getSize() + waterpower/2);  
 System.*out*.println("Stem send " + (waterpower/2) + " energy to flower");  
 Plant.*fout*.println("Stem send " + (waterpower/2) + " energy to flower");  
 System.*out*.println("My stem is " + size);  
 Plant.*fout*.println("My stem is " + size);  
 Plant.*fout*.flush();  
  
 return waterpower/2;  
 }  
}

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Plant Katya = new Plant(10, 7,8);  
 Plant HHHH = new Plant(6, 7,8);  
 Plant FFFF = new Plant(10, 7,8);  
 Plant JJJJ = new Plant();  
 Plant KKKK = new Plant();  
 Katya.getFlower();  
 Katya.Print();  
 Scanner in = new Scanner(System.*in*);  
 int sunpower, waterpower;  
 System.*out*.println("Enter how much water power get plant: ");  
 waterpower = in.nextInt();  
 System.*out*.println("Enter how much sun power get plant: ");  
 sunpower = in.nextInt();  
  
  
  
 System.*out*.println("with: ");  
 Katya.Grow(waterpower, sunpower);  
 Katya.Print();  
 Katya.IfFlowerCanReproduce();  
  
  
  
 //Katya.IfLeafCanDieRightNow();  
 }  
 }

**Результат виконання програми**

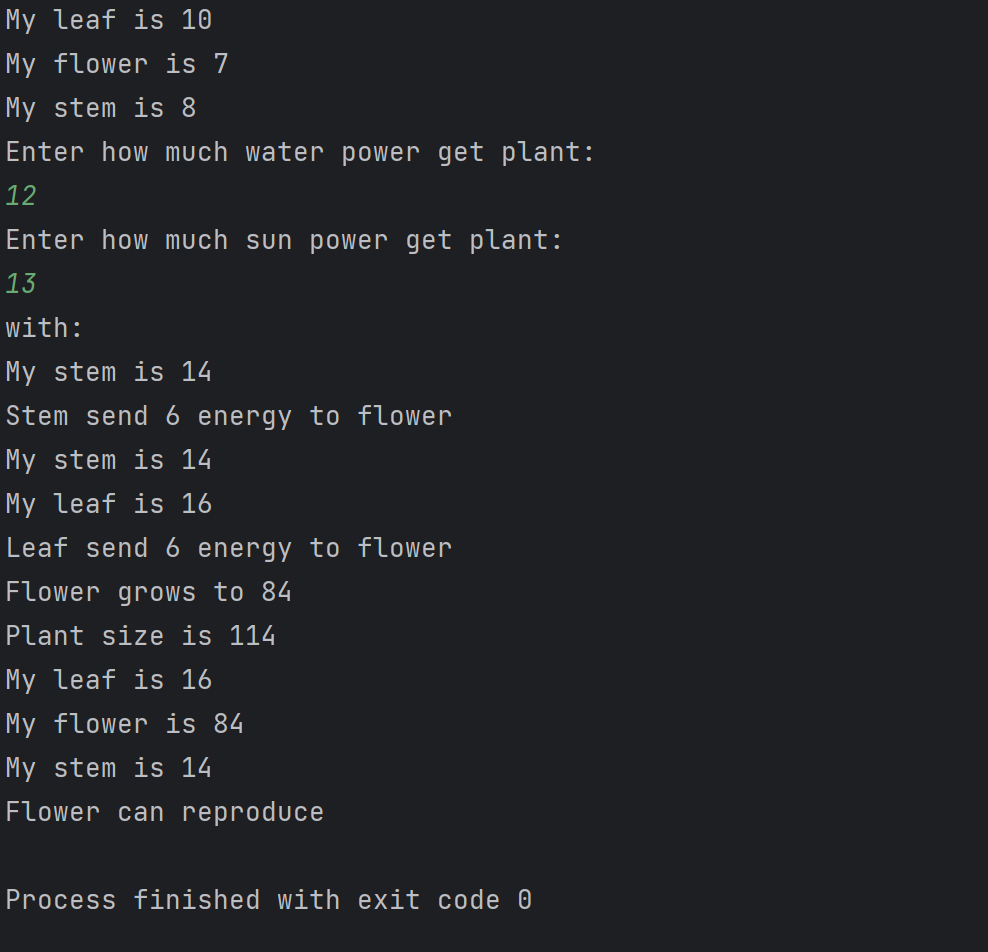
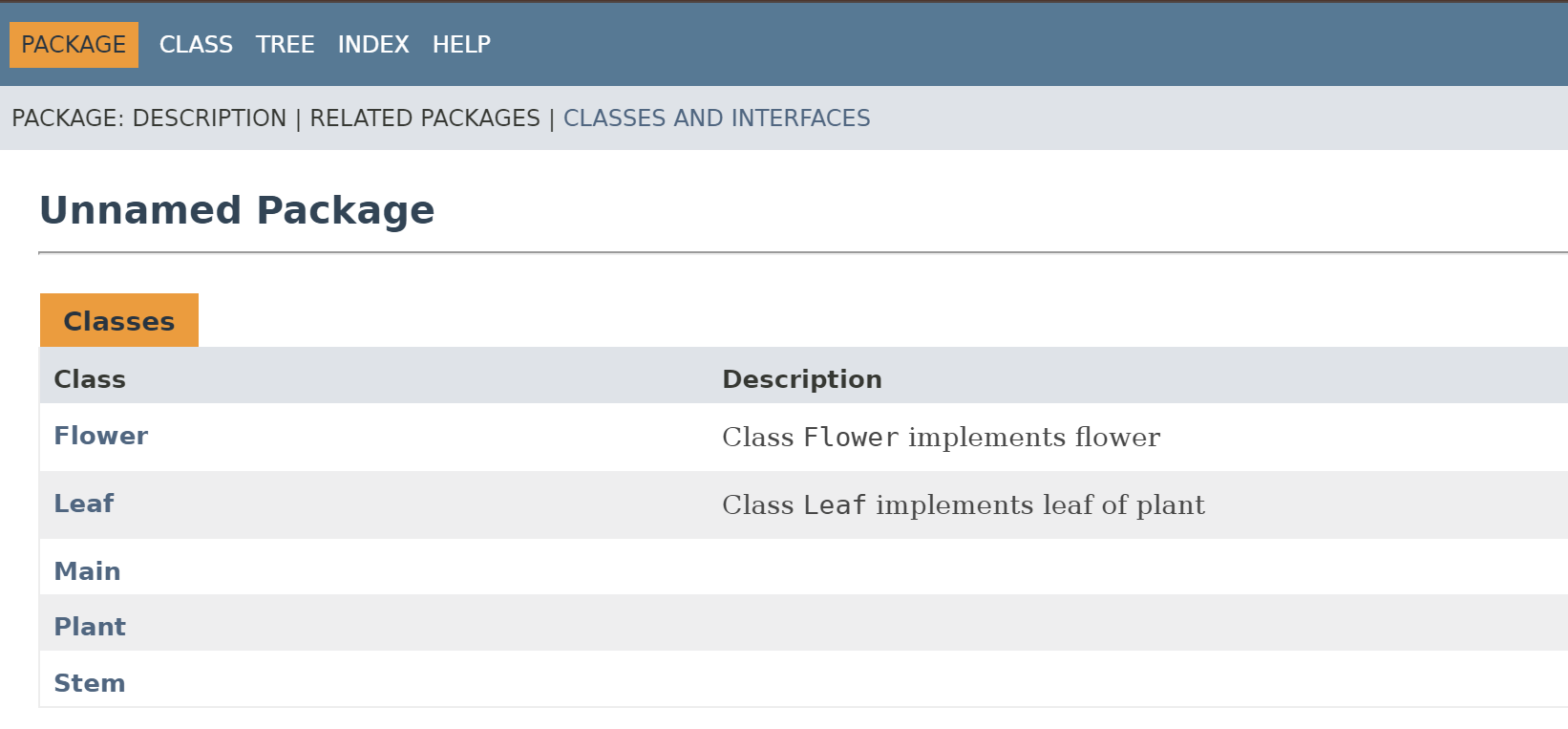
****

Рис.1 – Скріншот документу згенерованого програмою

**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

1. Синтаксис визначення класу.

public class ClassName {

// Код класу тут

}

1. Синтаксис визначення методу.

returnType methodName(parameterType parameterName) {

// Тіло методу

}

1. Синтаксис оголошення поля.

dataType fieldName;

1. Як оголосити та ініціалізувати константне поле?

final dataType fieldName = value;

1. Які є способи ініціалізації полів?

конструктор, ініціалізація при оголошенні, блоки ініціалізації.

1. Синтаксис визначення конструктора.

public ClassName(constructorParameters) {

// Тіло конструктора

}

1. Синтаксис оголошення пакету.

package packageName;

1. Як підключити до програми класи, що визначені в зовнішніх пакетах?

import packageName.ClassName;

1. В чому суть статичного імпорту пакетів?

Статичний імпорт пакетів дозволяє використовувати статичні члени класу без префіксу

1. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

Вимоги до файлів і каталогів у використанні пакетів: файл класу повинен зберігатися в каталозі, що відповідає імені пакету, і шлях до нього повинен відповідати його імені простору імен.

**Висновок**

Ознайомилася з роботою класів та пакетів у мові програмування Java. Розробивла клас який реалізує предметну область «Рослина», та програму драйвер яка перевіряє роботу класу.