

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних технологій, автоматики та метрології

Кафедра ЕОМ



Звіт
До лабораторної роботи №3
З дисципліни: «Кросплатформні засоби програмування»
На тему «Основи розробки програм мовою Java»
Варіант №9

Виконав: ст. гр. КІ-36
Нагребний Ю.С.

Прийняв:
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.

9. Рослина

Хід роботи:

Лістинг програми:

```
//Main.java
package KI36.Nahrebnyi.Lab3;

/**
 * Main class
 *
 * @author Yura
 * @version 1.0
 */
public class Main
{
    /**
     * Main method
     * @param args
     */
    public static void main(String[] args)
    {
        Plant plant = new Plant("Plant", 1, 19.2);

        plant.GrowUp();
        System.out.println("-----");
        plant.PrintInfo();
        System.out.println("-----");
        plant.Wither();
        System.out.println("-----");
    }
}
```

```

        plant.PrintInfo();
    }
}

//Leaf.java
package KI36.Nahrebnyi.Lab3;

import java.util.Random;

/**
 * Class Leaf
 * @version 1.0
 * @author Yura
 */
public class Leaf
{
    private String color;
    private double length;
    private Logger logger = Logger.getLogger();

    /**
     * Constructor
     * @param color
     */
    public Leaf(String color)
    {
        logger.log(logger.infoFlag + "Leaf constructor was called");
        this.color = color;
        Random random = new Random();
        this.length = random.nextDouble(10);
    }

    /**
     * Getter for color
     * @return color
     */
    public String getColor()
    {
        logger.log(logger.infoFlag + "Leaf getColor was called");
        return color;
    }

    /**
     * Setter for color
     * @param color
     */
    public void setColor(String color)
    {
        logger.log(logger.infoFlag + "Leaf setColor was called");
        this.color = color;
    }

    /**
     * Getter for length
     * @return length
     */
    public double getLength()
    {
        logger.log(logger.infoFlag + "Leaf setLength was called");
    }
}

```

```

        return length;
    }

    /**
     * Setter for length
     * @param length
     */
    public void setLength(double length)
    {
        logger.log(logger.infoFlag + "Leaf getLength was called");
        this.length = length;
    }

    /**
     * Method Wither
     */
    void Wither()
    {
        logger.log(logger.infoFlag + "Leaf Wither method was called");
        this.color = "Yellow";
        System.out.println("Autumn came and the leaves withered");
    }

    /**
     * Method GrowUp
     */
    void GrowUp()
    {
        logger.log(logger.infoFlag + "Leaf GrowUp method was called");
        Random random = new Random();
        double grow = random.nextDouble(10);
        this.length += grow;

        System.out.println("Leaf grow up " + grow + "s.");
    }

    /**
     * Method PrintInfo
     */
    void PrintInfo()
    {
        logger.log(logger.infoFlag + "Leaf PrintInfo was called");
        System.out.println("Leaf: { length: " + length + " s.; color: " + color +
" }");
    }
}

//Branch.java
package KI36.Nahrebnyi.Lab3;

import java.util.Random;

/**
 * Lab3. Class Branch
 *
 * @author Yura
 * @version 1.0
 */
public class Branch

```

```

{
    private double length;
    Logger logger = Logger.getLogger();

    /**
     * Constructor
     */
    public Branch()
    {
        logger.log(logger.infoFlag + "Branch constructor was called");
        Random random = new Random();
        this.length = random.nextDouble(10);
    }

    /**
     * Getter for length
     * @return
     */
    public double getLength()
    {
        logger.log(logger.infoFlag + "Branch getLength was called");
        return length;
    }

    /**
     * Setter for length
     * @param length
     */
    public void setLength(double length)
    {
        logger.log(logger.infoFlag + "Branch setLength was called");
        this.length = length;
    }

    /**
     * Method GrowUp
     */
    void GrowUp()
    {
        logger.log(logger.infoFlag + "Branch GrowUp method was called");
        Random random = new Random();
        double grow = random.nextDouble(10);
        this.length += grow;

        System.out.println("Branch grow up " + grow + "s.");
    }

    /**
     * Method to print info
     */
    void PrintInfo()
    {
        logger.log(logger.infoFlag + "Branch PrintInfo method was called");
        System.out.println("Branch { length: " + length + "s. }" );
    }
}

```

//Logger.java

```

package KI36.Nahrebnyi.Lab3;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 * Class Logger. Was created to log information, errors and warnings. Also there
 * was implemented Singelton
 * @author Yura
 * @version 1.0
 */
public class Logger
{
    private static Logger logger;
    private final String fileName;

    protected final String infoFlag = new String("[INFO] ");

    /**
     * Constructor
     * @param fileName
     */
    private Logger(String fileName)
    {
        this.fileName = fileName;
        File loggerFile = null;
        FileWriter fout = null;
        try
        {
            loggerFile = new File(fileName);
            fout = new FileWriter(loggerFile, true);
            SimpleDateFormat formatter= new SimpleDateFormat("yyyy-MM-dd 'at'
HH:mm:ss z");
            Date date = new Date(System.currentTimeMillis());
            fout.write "[" + formatter.format(date) + " ] " + "Logger start to
work\n");
        }
        catch (IOException e)
        {
            System.err.println("Something wrong with log file" +
e.getMessage());
            System.exit(1);
        }
        finally
        {
            try
            {
                fout.flush();
                fout.close();
            }
            catch (IOException e)
            {
                System.out.println(e.getMessage());
            }
        }
    }
}

```

```

/**
 * Method to do logging
 * @param massege
 */
public void log(String massege)
{
    File loggerFile = null;
    FileWriter fout = null;
    try
    {
        loggerFile = new File(this.fileName);
        fout = new FileWriter(loggerFile, true);
        SimpleDateFormat formatter= new SimpleDateFormat("yyyy-MM-dd 'at'
HH:mm:ss z");
        Date date = new Date(System.currentTimeMillis());
        fout.write "[" + formatter.format(date) + "]" + " " + massege + "\n";
    }
    catch (IOException e)
    {
        System.err.println("Something wrong with log file" +
e.getMessage());
        System.exit(1);
    }
    finally
    {
        try
        {
            fout.flush();
            fout.close();
        }
        catch (IOException | NullPointerException e)
        {
            System.out.println(e.getMessage());
        }
    }
}

/**
 * Singleton implementation
 * @param fileName
 * @return
 */
public static Logger getLogger(String fileName)
{
    if (logger == null)
    {
        logger = new Logger(fileName);
    }
    return logger;
}

/**
 * Getter for logger
 * @return logger
 */
public static Logger getLogger()
{
    return logger;
}

```

```

    }

}

//Plant.java
package KI36.Nahrebnyi.Lab3;

/**
 * Lab 3. Class Plant
 */
public class Plant
{
    private String name;
    private int age;
    private double length;
    private Leaf[] leaves = null;
    private Branch[] branches = null;
    private Logger logger = Logger.getLogger("logs.txt");

    /**
     * Constructor
     * @param name
     * @param age
     * @param length
     */
    public Plant(String name, int age, double length)
    {
        logger.log(logger.infoFlag + "Plant constructor was called");
        this.name = name;
        this.age = age;
        this.length = length;

        leaves = new Leaf[age * 6];
        for (int i = 0; i < age * 6; i++)
        {
            leaves[i] = new Leaf("green");
        }

        branches = new Branch[age * 3];
        for (int i = 0 ; i < age * 3; i++)
        {
            branches[i] = new Branch();
        }
    }

    /**
     * Getter for name
     * @return name
     */
    public String getName()
    {
        logger.log(logger.infoFlag + "Plant getName was called");
        return name;
    }

    /**
     * Setter for name
     * @param name
     */

```



```
public void setName(String name)
{
    logger.log(logger.infoFlag + "Plant setName was called");
    this.name = name;
}

/**
 * Getter for Age
 * @return age
 */
public int getAge()
{
    logger.log(logger.infoFlag + "Plant getAge was called");
    return age;
}

/**
 * Setter for Age
 * @param age
 */
public void setAge(int age)
{
    logger.log(logger.infoFlag + "Plant setAge was called");
    this.age = age;
}

/**
 * Getter for leaves
 * @return leaves
 */
public Leaf[] getLeaves()
{
    logger.log(logger.infoFlag + "Plant getLeaves was called");
    return leaves;
}

/**
 * Setter for leaves
 * @param leaves
 */
public void setLeaves(Leaf[] leaves)
{
    logger.log(logger.infoFlag + "Plant setLeaves was called");
    this.leaves = leaves;
}

/**
 * Getter for branches
 * @return branches
 */
public Branch[] getBranches()
{
    logger.log(logger.infoFlag + "Plant getBranches was called");
    return branches;
}

/**
 * Setter for branches
 * @param branches
 */
```

```

    */
    public void setBranches(Branch[] branches)
    {
        logger.log(logger.infoFlag + "Plant setBranches was called");
        this.branches = branches;
    }

    /**
     * Getter for length
     * @return length
     */
    public double getLength()
    {
        logger.log(logger.infoFlag + "Plant getLength was called");
        return length;
    }

    /**
     * Setter for length
     * @param length
     */
    public void setLength(double length)
    {
        logger.log(logger.infoFlag + "Plant setLength was called");
        this.length = length;
    }

    /**
     * Method to grow up
     */
    public void GrowUp()
    {
        logger.log(logger.infoFlag + "Plant GrowUp method was called was
called");
        for (int i = 0; i < age*3; i++)
        {
            branches[i].GrowUp();
        }
        for (int i = 0; i < age*6; i++)
        {
            leaves[i].GrowUp();
        }
    }

    /**
     * Method to print info
     */
    public void PrintInfo()
    {
        logger.log(logger.infoFlag + "Plant PrintInfo method was called");
        System.out.println("Plant: { name: " + name + "; length: " + length +
"s.; age: " + age + " }");
        for (int i = 0; i < age*3; i++)
        {
            branches[i].PrintInfo();
        }
        for (int i = 0; i < age*6; i++)
        {
            leaves[i].PrintInfo();
        }
    }

```

```

    }
}

/**
 * Method to Wither
 */
public void Wither()
{
    logger.log(logger.infoFlag + "Plant Wither was called");
    for (int i = 0; i < age * 6; i++)
    {
        leaves[i].Wither();
    }
}
}

```

Результат:

```

Plant: { name: Plant; length: 19.2s.; age: 1 }
Branch { length: 8.703197070033806s. }
Branch { length: 5.14255471949351s. }
Branch { length: 7.780559465481126s. }
Leaf: { length: 12.524510050269011 s.; color: green }
Leaf: { length: 13.044835557921957 s.; color: green }
Leaf: { length: 8.386221491568739 s.; color: green }
Leaf: { length: 4.778718801983711 s.; color: green }
Leaf: { length: 9.990080226460082 s.; color: green }
Leaf: { length: 6.808248858019224 s.; color: green }
-----
Autumn came and the leaves withered
Autumn came and the leaves withered
Autumn came and the leaves withered
Autumn came and the leaves withered
Autumn came and the leaves withered
Autumn came and the leaves withered
-----
Plant: { name: Plant; length: 19.2s.; age: 1 }
Branch { length: 8.703197070033806s. }
Branch { length: 5.14255471949351s. }
Branch { length: 7.780559465481126s. }
Leaf: { length: 12.524510050269011 s.; color: Yellow }
Leaf: { length: 13.044835557921957 s.; color: Yellow }
Leaf: { length: 8.386221491568739 s.; color: Yellow }
Leaf: { length: 4.778718801983711 s.; color: Yellow }
Leaf: { length: 9.990080226460082 s.; color: Yellow }
Leaf: { length: 6.808248858019224 s.; color: Yellow }

```

Висновок: у ході данної лабораторної роботи я ознайомився з процесом розробки класів та пакетів мовою Java.