

Лабораторная работа №2
студента группы ИТ – 32
Курбатовой Софьи Андреевны

Выполнение: _____ Защита _____

CODE CONVENTION. ЛОГИРОВАНИЕ

Цель работы: изучение правил документа code convention, изучение основ логирования в Java при помощи библиотеки log4j.

Содержание работы
Вариант 10

1. Привести код из 1-й лабораторной работы к требованиям конвенции.
2. Так же необходимо в 1-ю лабораторную добавить логирование основных действий приложения с использованием библиотеки log4j. Логи должны выводиться в файл и на консоль. Логи, выводимые в файл нужно разделять по уровням INFO, DEBUG, ERROR: в один файл записывать логи INFO, DEBUG, WARNING, ERROR и FATAL во второй - только WARNING, ERROR и FATAL

3. Маска лога должна содержать следующую информацию:

<уровень> <дата> <время> (короткое имя класса) - <текст лога>

<error.printStackTrace> (для ERROR и FATAL)

Например:

```
ERROR 2010-09-01 10:01:02,525 ProcThread1 (RMIServiceServer ) - Unable to
look up client under the RMI name "//localhost/MyService_and_MPRMIServiceClient"
java.rmi.NotBoundException: MyService_and_MPRMIServiceClient
    at sun.rmi.registry.RegistryImpl.lookup(RegistryImpl.java:106)
    at sun.rmi.registry.RegistryImpl_Skel.dispatch(Unknown Source)
    at sun.rmi.server.UnicastServerRef.oldDispatch(UnicastServerRef.java:375)
    at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:240)
    at sun.rmi.transport.Transport$1.run(Transport.java:153)
    at java.security.AccessController.doPrivileged(Native Method)
    at sun.rmi.transport.Transport.serviceCall(Transport.java:149)
    at
sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:466)
    at
sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:707)
    at java.lang.Thread.run(Thread.java:595)
```

Ход работы

1. Привела код из 1-й лабораторной к требованиям конвенции.

Листинг 1. Пример оформления по правилам конвенции

```
public static double[][] TrigonometryFunction(int start, int end, int step)
{
    /**
     * <p>Описание метода "Тригонометрическая функция"</p>
     * Строит промежуточные точки между началом линии и ее концом с определенным шагом
     * @param start Начальная точка
     * @param end Конечная точка
     * @param step Шаг
     * @return array Массив значений в виде аргумент - значение функции
     */
    int rows = (int) ((end - start) / step) + 1;
    double[][] array = new double[rows][2];
    for (int i = 0; i < rows; i++)
    {
        double y = Math.sin(start) + 0.5 * Math.cos(start);
        array[i][0] = start;
        array[i][1] = y;
        start += step;
    }
    return array;
}
```

2. Добавлен класс для вывода логов на консоль и в файл.

Листинг 2. Класс для создания логов

```
public class LoggerClass {
    /**
     * <p>Описание метода setLog</p>
     * @param loggerName имя файла для записи лога
     * @return Log объекта класса Logger
     */
    @NotNull
    public static Logger setLog(String loggerName)
    {
        Logger log = Logger.getLogger(loggerName) ;
        log.setLevel(Level.FINE);
        log.setUseParentHandlers(false);

        FileHandler fileHandler = null;
        try
        {
            fileHandler = new FileHandler("./"+loggerName + ".txt",true);
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
        if (fileHandler != null)
        {
            fileHandler.setLevel(Level.FINE);
            fileHandler.setFormatter(new Format() );
        }
        log.addHandler(fileHandler);
        FileHandler fileHandler1 = null;
        try
        {
            fileHandler1 = new FileHandler("./"+loggerName + "WARNING"+ ".txt",true);
        } catch (IOException e) {
            e.printStackTrace();
        }
        if (fileHandler1 != null)
    }
```

```

    {
        fileHandler1.setLevel(Level.SEVERE);
        fileHandler1.setFormatter(new Format());
        log.addHandler(fileHandler1);
    }
    ConsoleHandler consoleHandler = new ConsoleHandler();
    consoleHandler.setLevel(Level.FINE);
    consoleHandler.setFormatter(new Format());
    log.addHandler(consoleHandler);
    return log;
}
/**
 * <p>Описание класса Format </p>
 * Является наследником класса Formatter. Методы определяют вид объекта класса LogRecord
 */
static class Format extends Formatter
{
    /**
     * <p>Описание метода format</p>
     * @param record объект класса LogRecord
     * @return output строка для записи в текстовый документ или для вывода на экран*/
    @Override
    public String format(LogRecord record)
    {
        Date currentDate = new Date();
        SimpleDateFormat timeFormat = new SimpleDateFormat("E yyyy.MM.dd hh:mm:ss a
zzz");
        String output;
        output = String.format("LEVEL: %-10s Date: %-10s Class_name: %-10s Text: %-
10s ", record.getLevel(),timeFormat.format(currentDate),
record.getSourceClassName(),record.getMessage());
        if (record.getThrown()!=null)
        {
            output += " ERROR";
            StackTraceElement[] stackTraceElements = record.getThrown().getStackTrace();
            for (StackTraceElement traceElement:stackTraceElements)
                output += traceElement.toString() + "\n";
        }
        output += "\n\n";
        return output;
    }
}
}

```

3. Тестирование:

Имя	Дата изменения	Тип	Размер	LEVEL: SEVERE Date: нт 2021.03.05 02:33:34 PM MSK Class_name: guiapp.Controller Text: / by zero
.idea	05.03.2021 14:58	Папка с файлами		
doc	05.03.2021 14:51	Папка с файлами		
out	16.02.2021 1:38	Папка с файлами		
src	05.03.2021 13:34	Папка с файлами		
GUIapp.iml	28.02.2021 10:32	Файл "IML"	1 КБ	
logger_app.txt	05.03.2021 14:34	Текстовый докум...	1 КБ	
logger_appWARNING.txt	05.03.2021 14:33	Текстовый докум...	1 КБ	

Рис. 2.1. Содержимое файла с логами

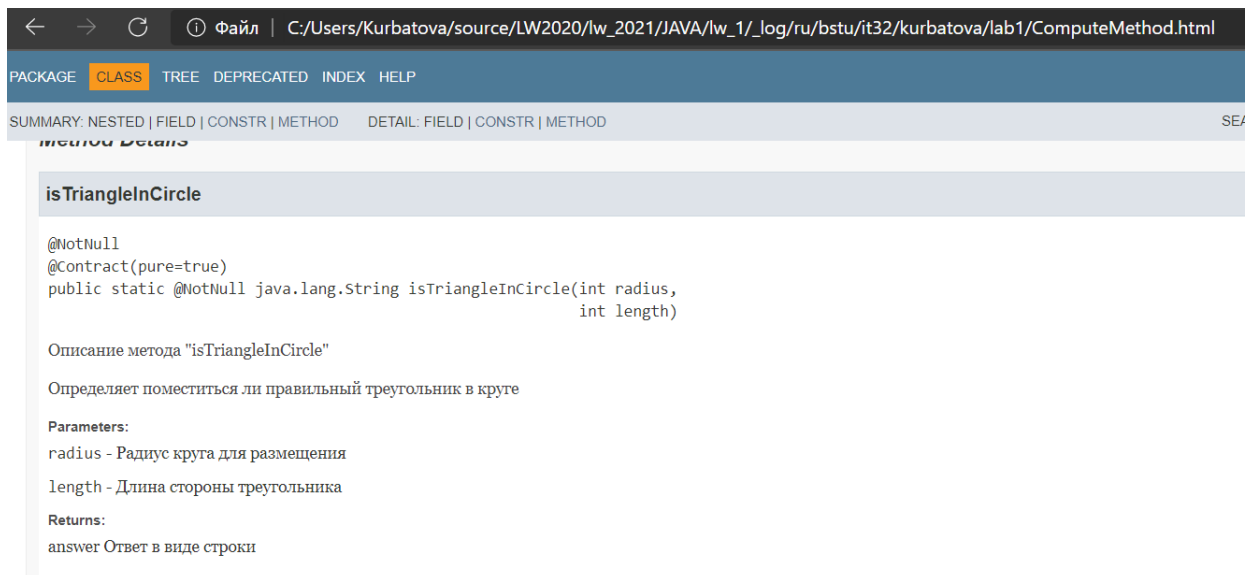


Рис. 2.2. html файл документации



Рис. 2.3. Вывод в графическом приложении

Вывод: В ходе выполнения лабораторной работы были изучены правила оформления документации по code convention и изучены основ логирования в Java при помощи библиотеки logging. Результатом работы стала оформленная документация и сформированные логи, отражающие работу в разработанном приложении.