ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

**ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Отчет**

по дисциплине «Разработка программных приложений на языках высокого уровня»

специальности 09.03.03 «Прикладная информатика»

**Лабораторная работа № 3**

студента очного отделения

3 курса 12002005 группы

Лысенко Натальи Александровны

Проверил:

старший преподаватель

Лифиренко Максим Вячеславович

БЕЛГОРОД 2023

**Цель работы:** изучить Java Code Convention, получить практические навыки по логированию с помощью Log4J.

(Вариант № 13)

**Листинг класса Main**

package com.company;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
*/\*\* Программа реализует выполнение 13 варианта 2 лабораторной  
 \*/*public class Main {  
 */\*\* Метод main предназначен для выбора режима ввода данных,  
 \* выбора задачи и вызова соответствующего метода  
 \** ***@author*** *Natalia @version 3.2  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public static void main(String[] args) throws FileNotFoundException {  
 System.*out*.println("Введите 1 для режима работы ввода параметров задач" +  
 " с клавиатуры или 2 для ввода параметров из файла");  
 Scanner scanner = new Scanner(System.*in*);  
 int inputMode = scanner.nextInt();  
 if (inputMode == 1 || inputMode == 2) {  
 Tasks tasks = new Tasks();  
 tasks.task1(inputMode);  
 tasks.task2(inputMode);  
 tasks.task3(inputMode);  
 tasks.task4(inputMode);  
 tasks.task5(inputMode);  
 tasks.task6(inputMode);  
 tasks.task7(inputMode);  
 tasks.task8(inputMode);  
 }  
 }  
}

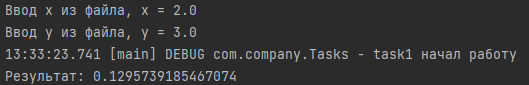
**Листинг класса Tasks**

package com.company;  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
import java.io.File;  
import java.util.Scanner;  
import java.io.IOException;  
*/\*\* Класс Tasks описывает основные задания 2 лабораторной работы.  
 \*/*public class Tasks {  
 private static Logger *logger* = LogManager.*getLogger*(Tasks.class.getName());  
  
 */\*\* Метод task1 вычисляет значение выражения по формуле 13 варианта.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task1(int inputMode) {  
 double x=0;  
 double y=0;  
 double result;  
  
 if (inputMode == 1) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введите x ");  
 x = scanner.nextDouble();  
 System.*out*.print("Введите y ");  
 y = scanner.nextDouble();  
 } else {  
 try {  
 String path = "src/1taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 x = scanner.nextDouble();  
 System.*out*.println("Ввод x из файла, x = " + x);  
 y = scanner.nextDouble();  
 System.*out*.println("Ввод y из файла, y = " + y);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"1taskData\" отсутствует");  
 }  
 }  
 *logger*.debug("task1 начал работу");  
 result = Math.*pow*(2, -x) - Math.*cos*(x) + Math.*sin*(2 \* x \* y);  
 System.*out*.println("Результат: " + result);  
 }  
  
 */\*\*Метод task2 находит площадь трапеции по заданым  
 \* основаниям и углу при большем основании а.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task2(int inputMode) {  
 double sideA=0;  
 double sideB=0;  
 double alpha=0;  
 double square;  
  
 if (inputMode == 1) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите большее основание трапеции a: ");  
 sideA = scan.nextDouble();  
 System.*out*.print("Введите меньшее основание трапеции b: ");  
 sideB = scan.nextDouble();  
 System.*out*.print("Введите угол при большем основании: ");  
 alpha = scan.nextDouble();  
 }  
 else {  
 try {  
 String path = "src/2taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 sideA = scanner.nextDouble();  
 System.*out*.println("Ввод из файла, a = " + sideA);  
 sideB = scanner.nextDouble();  
 System.*out*.println("Ввод из файла, b = " + sideB);  
 alpha = scanner.nextDouble();  
 System.*out*.println("Ввод из файла, альфа = " + alpha);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"2taskData\" отсутствует или не найден");  
 }  
 }  
 square=(0.5 \* (Math.*pow*(sideA, 2) - Math.*pow*(sideB, 2)) \* Math.*tan*(Math.*toRadians*(alpha)));  
 System.*out*.println("Площадь трапеции = " + square);  
 *logger*.info("Если площадь трапеции больше 0, то все ок");  
 }  
  
 */\*\* Метод task3 считает количество отрицательных чисел среди a, b, c  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task3(int inputMode) {  
 float[] array = new float[3];  
 int kol=0;  
 if (inputMode == 1) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Введите числа a b c через пробел: ");  
 for (int i = 0; i < 3; i++) {  
 array[i] = scan.nextFloat(); // Заполняем массив элементами, введёнными с клавиатуры  
 }  
 }  
 else{  
 try {  
 String path = "src/3taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 System.*out*.println("Ввод a b c из файла: ");  
 for (int i = 0; i < 3; i++) {  
 array[i] = scanner.nextFloat(); // Заполняем массив элементами, введёнными из файла  
 System.*out*.println(array[i]);  
 }  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"3taskData\" отсутствует или не найден");  
 }  
 }  
  
 for (int i = 0; i < 3; i++)  
 {  
 if (array[i]<0)  
 kol++;  
 }  
 System.*out*.println("Колличество отрицательных чисел = "+ kol);  
 *logger*.warn("Предупреждение!!! Сегодня хороший день)");  
 }  
  
 */\*\* Метод task4 определяет, принадлежит ли точка A(x, y) треугольнику с вершинами в точках (x1, у1), (х2, у2), (х3, у3).  
 \** ***@param*** *inputMode параметр передает режим ввода данных (1 - ввод данных с клавиатуры, любое другое число - ввод данных из файла).  
 \*/* public void task4(int inputMode) {  
 // Создание массива, который содержит координаты вершин треугольника и точки A  
 float[] tr = new float[8];  
 // Ввод координат треугольника и точки А  
 if (inputMode == 1) {  
 Scanner scan = new Scanner(System.*in*);  
  
 System.*out*.println("Введите координаты вершин треугольника (x1 у1 х2 у2 х3 у3): ");  
 for (int i = 0; i < 6; i++) {  
 tr[i] = scan.nextFloat();  
 }  
 System.*out*.println("Введите координаты точки А (x, у): ");  
 for (int i = 6; i < 8; i++) {  
 tr[i] = scan.nextFloat();  
 }  
 }  
 // Ввод координат треугольника и точки А из файла  
 else {  
 try {  
 String path = "src/4taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 System.*out*.println("Ввод координат вершин треугольника (x1 у1 х2 у2 х3 у3) из файла: ");  
 for (int i = 0; i < 8; i++) {  
 tr[i] = scanner.nextFloat();  
 if (i==6) {  
 System.*out*.println("A (x,y): ");  
 }  
 System.*out*.println(tr[i]);  
 }  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"4taskData\" отсутствует или не найден");  
 }  
 }  
 // a, b, c - вспомогательные, предназначены для вычисления площади треугольника ABC  
 float a = (tr[6] - tr[0]) \* (tr[3] - tr[1]) - (tr[7] - tr[1]) \* (tr[2] - tr[0]);  
 float b = (tr[6] - tr[2]) \* (tr[5] - tr[3]) - (tr[7] - tr[3]) \* (tr[4] - tr[2]);  
 float c = (tr[6] - tr[4]) \* (tr[1] - tr[5]) - (tr[7] - tr[5]) \* (tr[0] - tr[4]);  
 // Проверка принадлежности точки А треугольнику  
 // Если все три знака площадей совпадают, то точка лежит внутри треугольника  
 if ((a >= 0 && b >= 0 && c >= 0) || (a <= 0 && b <= 0 && c <= 0)) {  
 System.*out*.println("Точка А (" + tr[6] + ", " + tr[7] + ") принадлежит треугольнику");  
 } else {  
 System.*out*.println("Точка А (" + tr[6] + ", " + tr[7] + ") не принадлежит треугольнику");  
 }  
 *logger*.info("Да, тут длинный if, не бейте");  
 }  
  
 */\*\*Метод task5 по вводимому числу от 1 до 11 (номеру класса) выдает  
 \* соответствующее сообщение «Привет, k-классник».  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task5(int inputMode) {  
 int number=0;  
 if (inputMode == 1) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число ");  
 number = scanner.nextInt();  
 } else {  
 try {  
 String path = "src/5taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 number = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa " + number);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"5taskData\" отсутствует или не найден");  
 }  
 }  
  
 if (number > 0 && number < 12) {  
 switch (number) {  
 case 1 -> System.*out*.println("Привет, Первоклассник");  
 case 2 -> System.*out*.println("Привет, Второклассник");  
 case 3 -> System.*out*.println("Привет, Третьеклассник");  
 case 4 -> System.*out*.println("Привет, Четвероклассник");  
 case 5 -> System.*out*.println("Привет, Пятиклассник");  
 case 6 -> System.*out*.println("Привет, Шестиклассник");  
 case 7 -> System.*out*.println("Привет, Семиклассник");  
 case 8 -> System.*out*.println("Привет, Восьмиклассник");  
 case 9 -> System.*out*.println("Привет, Девятиклассник");  
 case 10 -> System.*out*.println("Привет, Десятиклассник");  
 case 11 -> System.*out*.println("Привет, Одиннадцатиклассник");  
 }  
 }  
 else *logger*.error("Ты точно школьник?");  
 }  
  
 */\*\* Метод task6 находит все делители натурального числа n.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task6(int inputMode) {  
 int n=0;  
 if (inputMode == 1) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число ");  
 n = scanner.nextInt();  
 } else {  
 try {  
 String path = "src/6taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 n = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa " + n);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"6taskData\" отсутствует или не найден");  
 }  
 }  
 System.*out*.println("Делители числa " + n);  
 for (int i = 1; i <= n; i++) {  
 if (n % i == 0) {  
 System.*out*.print(i + " ");  
 }  
 }  
 }  
 */\*\* Метод task7 находит сумму n членов ряда  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task7(int inputMode) {  
 int n=0;  
 float x=0;  
 float sum = 0;  
  
 if (inputMode == 1) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число n ");  
 n = scanner.nextInt();  
 System.*out*.println("Введите число x ");  
 x = scanner.nextFloat();  
 } else {  
 try {  
 String path = "src/7taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 n = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa n " + n);  
 x = scanner.nextFloat();  
 System.*out*.println("Ввод из файла числa x " + x);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"7taskData\" отсутствует или не найден");  
 }  
 }  
  
 for (int i = 1; i <= n; i++) {  
 sum += Math.*cos*(2\*i\*x)/i;  
 }  
 System.*out*.println("Сумма n членов ряда = " + sum);  
 }  
 */\*\* Метод task8 вычисляет сумму целых положительных чисел,  
 \* больших M, меньших N и кратных k. Полученное число выводится на экран.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \*/* public void task8(int inputMode) {  
 int M=0;  
 int N=0;  
 int k=0;  
 int sum = 0;  
  
 if (inputMode == 1) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введите число M: ");  
 M = scanner.nextInt();  
 System.*out*.print("Введите число N: ");  
 N = scanner.nextInt();  
 System.*out*.print("Введите число k: ");  
 k = scanner.nextInt();  
 } else {  
 try {  
 String path = "src/8taskData.txt";  
 Scanner scanner = new Scanner(new File(path));  
 M = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa M " + M);  
 N = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa N " + N);  
 k = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa k " + k);  
 } catch (IOException e) {  
 *logger*.fatal("Файл \"7taskData\" отсутствует или не найден");  
 }  
 }  
  
 for (int i = M + 1; i < N; i++) {  
 if (i % k == 0) {  
 sum += i;  
 }  
 }  
 System.*out*.println("Сумма чисел, больших " + M + ", меньших " + N + " и кратных " + k + ": " + sum);  
 }  
}

**Листинг файла log4j2.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<Configuration status="warn">  
 <Appenders>  
 <Console name="LogToConsole" target="SYSTEM\_OUT">  
 <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>  
 </Console>  
  
 <File name="LogToFile1" fileName="logs/app1.log" append="false">  
 <PatternLayout>  
 <Pattern>%-1p %d{yyyy-MM-dd HH:mm:ss} %c{1}:%L - %m%n</Pattern>  
 </PatternLayout>  
 </File>  
  
 <File name="LogToFile2" fileName="logs/app2.log" append="false">  
 <PatternLayout>  
 <Pattern>%-1p %d{yyyy-MM-dd HH:mm:ss} %c{1}:%L - %m%n</Pattern>  
 </PatternLayout>  
 </File>  
  
 </Appenders>  
 <Loggers>  
 <!-- avoid duplicated logs with additivity=false -->  
 <Logger name="com.company.Tasks" level="debug" additivity="false">  
 <AppenderRef ref="LogToFile1" level="debug"/>  
 <AppenderRef ref="LogToFile2" level="warn"/>  
 <AppenderRef ref="LogToConsole"/>  
 </Logger>  
 <Root level="error">  
  
 </Root>  
 </Loggers>  
</Configuration>

Результаты работы программы представлены на рисунках 1-8.



Можем посмотреть лог fatal если поломаем» название файла



Рисунок 1 – Первая задача

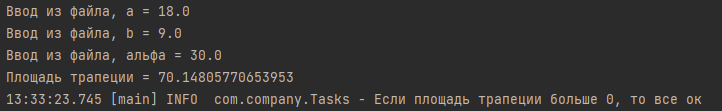


Рисунок 2 –Вторая задача

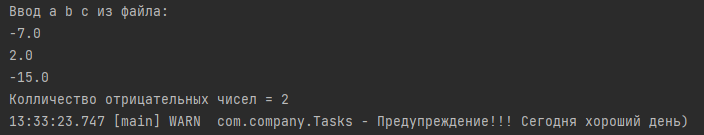


Рисунок 3 – Третья задача

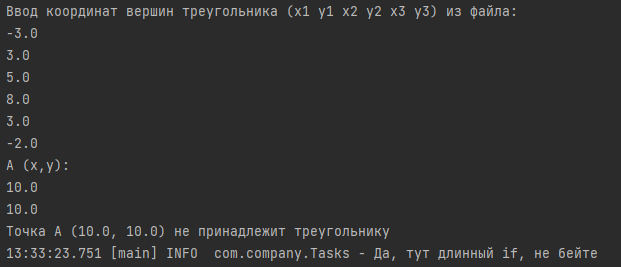


Рисунок 4 – Четвертая задача



Рисунок 5 – Пятая задача

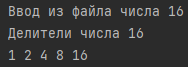


Рисунок 6 – Шестая задача

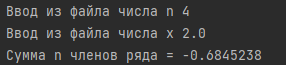


Рисунок 7 – Седьмая задача

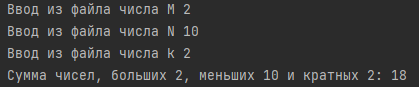


Рисунок 8 – Восьмая задача

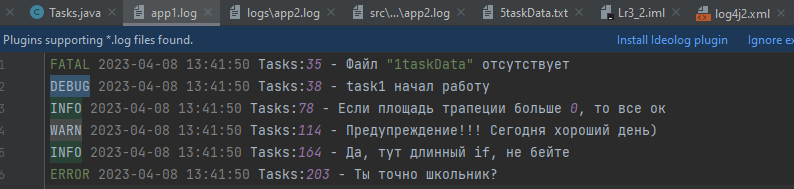


Рисунок 9 – Файл логирования уровня debug и выше

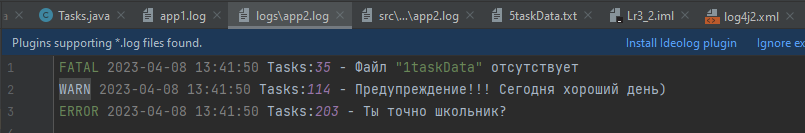


Рисунок 10 – Файл логирования уровня warn и выше

Вывод: были изучены Java Code Convention, и получены практические навыки по логированию с помощью Log4J.