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

Чернышева Владислава Максимовича

Проверил:

старший преподаватель

Лифиренко Максим Вячеславович

БЕЛГОРОД 2023

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

(Вариант №20)

**Листинг класса Main**

package com.company;  
import java.io.\*;  
import java.util.Scanner;  
public class Main {  
  
 */\*\** ***@author*** *Vladislav @version 1.0 \*/  
 /\*\*  
 \* В этом методе вводится метод ввода данных из файла или с консоли  
 \*  
 \** ***@param*** *args не используется  
 \** ***@throws*** *FileNotFoundException обработчик ошибок при вводе из 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 с файла");  
 Scanner scanner = new Scanner(System.*in*);  
 int workingMode = scanner.nextInt();  
 if (workingMode == 1 || workingMode == 2) {  
 Tasks tasks = new Tasks();  
 tasks.task1(workingMode);  
 tasks.task2(workingMode);  
 tasks.task3(workingMode);  
 tasks.task4(workingMode);  
 tasks.task5(workingMode);  
 tasks.task6(workingMode);  
 tasks.task7(workingMode);  
 tasks.task8(workingMode);  
  
 } else {  
 System.*out*.println("Нет такого режима работы программы");  
 }  
 }  
}

**Листинг класса Tasks**

package com.company;  
import java.io.File;  
import java.util.Scanner;  
import org.apache.logging.log4j.\*;  
public class Tasks {  
 private static Logger *logger* = LogManager.*getLogger*(Tasks.class.getName());  
  
 */\*\*  
 \* Этот метод выводит результат выражения e^x - x - 2 + (1+x)^x при вводимой переменной x  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task1(int workingMode) {  
 double x = 0;  
 double result;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task1()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите x ");  
 x = scanner.nextInt();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\1TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 *logger*.info("Создался сканер по этому файлу");  
 x = scanner.nextInt();  
 System.*out*.println("Ввод x из файла, x=" + x);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"1TaskData.txt\" отсутствует");  
 }  
 }  
  
 result = Math.*exp*(x) - x - 2 + Math.*pow*((1+x), x);  
 *logger*.debug("~какая-то подробная ифнормация по задаче~");  
 System.*out*.println("Результат равен " + result);  
 }  
  
 */\*\*  
 \* Этот метод находит сумму арифметической прогрессии  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task2(int workingMode) {  
 int ArProg = 0; // первый член и сумма членов  
 int NumSum = 0; // знаменатель  
 int count = 0; // число членов прогрессии  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task2()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите первый член прогрессии ");  
 ArProg = scanner.nextInt();  
 System.*out*.println("Введите знаменатель прогрессии ");  
 NumSum = scanner.nextInt();  
 System.*out*.println("Введите число членов прогрессии ");  
 count = scanner.nextInt();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\2TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 ArProg = scanner.nextInt();  
 System.*out*.println("Первый член прогрессии " + ArProg);  
 NumSum = scanner.nextInt();  
 System.*out*.println("Знаменатель прогрессии " + NumSum);  
 count = scanner.nextInt();  
 System.*out*.println("Число членов прогрессии " + count);  
 }  
 catch (Exception e) {  
 *logger*.fatal("Файл \"2TaskData.txt\" отсутствует");  
 }  
 }  
  
 while (count > 0) {  
 ArProg += NumSum;  
 count--;  
 }  
 *logger*.info("~какая-то подробная ифнормация по задаче~");  
 System.*out*.println("Сумма членов арифметической прогрессии " + ArProg);  
 }  
  
 */\*\*  
 \* Этот метод перераспределяет значения переменных x и y так, чтобы в x оказалось большее из этих  
 \* значений, а в у — меньшее.  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task3(int workingMode) {  
 double x = 0;  
 double y = 0;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task3()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите x ");  
 x = scanner.nextDouble();  
 System.*out*.println("Введите y ");  
 y = scanner.nextDouble();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\3TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 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(Exception e)  
 {  
 *logger*.fatal("Файл \"3TaskData.txt\" отсутствует");  
 }  
 }  
  
 if (x < y) { // Если y больше значение переменных меняется  
 double tmp = y;  
 y = x;  
 x = tmp;  
 } else if (y == x) {  
 System.*out*.println("X и Y одинаковые");  
 }  
 System.*out*.println("X равен " + x + " Y равен " + y);  
 *logger*.warn("~какое-то предупреждение~");  
  
 }  
 */\*\*  
 \* Этот метод определяет по двум углам существует ли треугольник и прямоугольный ли он  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task4(int workingMode) {  
 double a = 0, b = 0;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task4()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите угол a ");  
 a = scanner.nextDouble();  
 System.*out*.println("Введите угол b ");  
 b = scanner.nextDouble();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\4TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 a = scanner.nextDouble();  
 System.*out*.println("Ввод a из файла, a=" + a);  
 b = scanner.nextDouble();  
 System.*out*.println("Ввод b из файла, b=" + b);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"4TaskData.txt\" отсутствует");  
 }  
 }  
  
 *logger*.info("~какая-то подробная ифнормация по задаче~");  
 if (a <= 90 && b <= 90 ) {  
 System.*out*.println("Треугольник существует");  
 if (a == 90 || b == 90 || (a + b == 90)) System.*out*.println("Треугльник прямоугольный");  
  
 }  
 }  
 */\*\*  
 \* Этот метод выводит вес где ед. измерения зависит от введенного числа  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task5(int workingMode) {  
 int number = 0;  
 int weight = 0;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task5()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("0-Килограмм");  
 System.*out*.println("1-Миллиграмм");  
 System.*out*.println("2-Грамм");  
 System.*out*.println("3-Тонна");  
 System.*out*.println("4-Центнер");  
 System.*out*.println("Введите число ");  
 number = scanner.nextInt();  
 System.*out*.println("Введите вес ");  
 weight = scanner.nextInt();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\5TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 number = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa " + number);  
 weight = scanner.nextInt();  
 System.*out*.println("Ввод из файла веса " + weight);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"5TaskData.txt\" отсутствует");  
 }  
 }  
  
 *logger*.debug("~какая-то подробная ифнормация по задаче~");  
 System.*out*.print(weight);  
 if (number >= 0 && number < 5) {  
 switch (number) {  
 case 0:  
 System.*out*.println(" Килограмм");  
 break;  
 case 1:  
 System.*out*.println(" Миллиграмм");  
 break;  
 case 2:  
 System.*out*.println(" Грамм");  
 break;  
 case 3:  
 System.*out*.println(" Тонна");  
 break;  
 case 4:  
 System.*out*.println(" Центнер");  
 break;  
 }  
 }  
 }  
 */\*\*  
 \* Этот метод выводит все двухначные числа, сумма квадратов цифр которых кратна M.  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task6(int workingMode) {  
 int m = 0;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task6()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число m ");  
 m = scanner.nextInt();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\6TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 m = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa " + m);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"6TaskData.txt\" отсутствует");  
 }  
 }  
  
 *logger*.debug("~какая-то подробная ифнормация по задаче~");  
 for (int i = 1; i < 10; i++) //Первая цифра двухзначного числа  
 for (int j = 0; j < 10; j++){ //Вторая цифра двухзначного числа  
 if (i\*i + j\*j == m) System.*out*.println(i + "" + j);  
 }  
 }  
  
 */\*\*  
 \*Этот метод возврашает факториал числа n  
 \** ***@param*** *n число которое будет возводистя в факториал  
 \** ***@return*** *на возврат факториал числа  
 \*/* public long factorialUsingForLoop(int n) {  
 long fact = 1;  
 for (int i = 2; i <= n; i++) {  
 fact = fact \* i;  
 }  
 return fact;  
 }  
 */\*\*  
 \* Этот метод выводит сумму sum членов ряда n  
 \* sum = 1 + 3x^2 + ... + 2n + 1 / n! \* x^(2n)  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task7(int workingMode) {  
 int n = 0;  
 double x = 0;  
 int sum = 0;  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task7()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число n ");  
 n = scanner.nextInt();  
 System.*out*.println("Введите число x ");  
 x = scanner.nextInt();  
 }  
 catch (Exception e) {  
 *logger*.fatal("Ошибка считывания данных");  
 }  
 } else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\7TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 n = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa n " + n);  
 x = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa x " + x);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"7TaskData.txt\" отсутствует");  
 }  
 }  
  
 for (int i = 0; i < n; i++) {  
 sum += 2 \* n + 1 / factorialUsingForLoop(n) + 3 \* Math.*pow*(x,3 \* n);  
 }  
  
 *logger*.warn("~какое-то предупреждение~");  
 System.*out*.println("Сумма равна " + sum);  
 }  
 */\*\*  
 \* Этот метод выводит вычисляет сумму первых n целых положительных целых чисел.  
 \** ***@param*** *workingMode используется для выбора ввода переменных  
 \*/* public void task8(int workingMode) {  
 int n = 0; // Кол-во суммируемых чисел  
 int sum = 0;  
  
  
 if (workingMode == 1) {  
 *logger*.info("Вызов функция Task8()");  
 try {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число n ");  
 n = scanner.nextInt();  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"8TaskData.txt\" отсутствует");  
 }  
 }  
 else {  
 String path = "C:\\JavaProjects\\Lab2\\Lab2\_1\\src\\8TaskData.txt";  
 *logger*.info("Получена ссылка на файл");  
 try {  
 Scanner scanner = new Scanner(new File(path));  
 n = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa a " + n);  
 }  
 catch(Exception e)  
 {  
 *logger*.fatal("Файл \"8TaskData.txt\" отсутствует");  
 }  
 }  
  
 for (int i = 1; i < n; i++) {  
 sum += n;  
 }  
  
 *logger*.info("~какая-то подробная ифнормация по задаче~");  
 System.*out*.println("Сумма первых " + n + " целых положительных чисел равна " + 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.

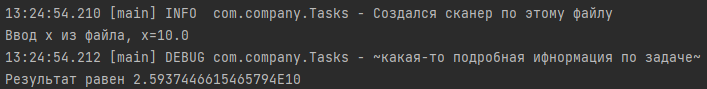


Рисунок 1 – Первая задача

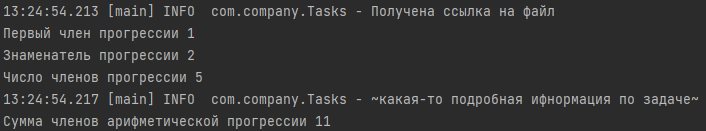


Рисунок 2 –Вторая задача

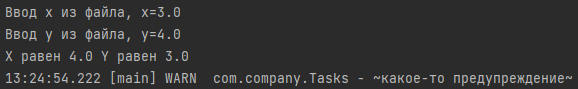


Рисунок 3 – Третья задача

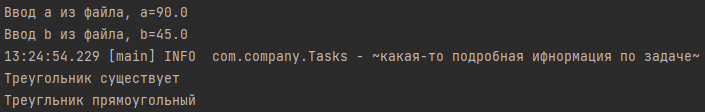


Рисунок 4 – Четвертого задача

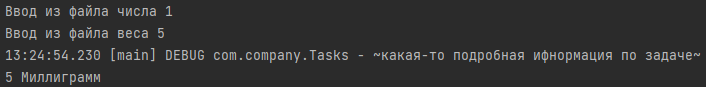


Рисунок 5 – Пятая задача

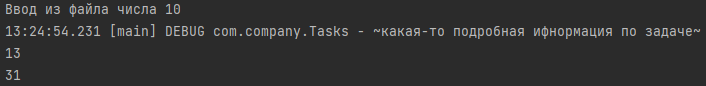


Рисунок 6 – Шестая задача

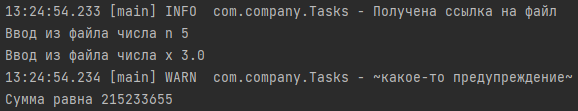


Рисунок 7 – Седьмая задача

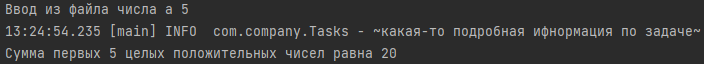


Рисунок 8 – Восьмая задача

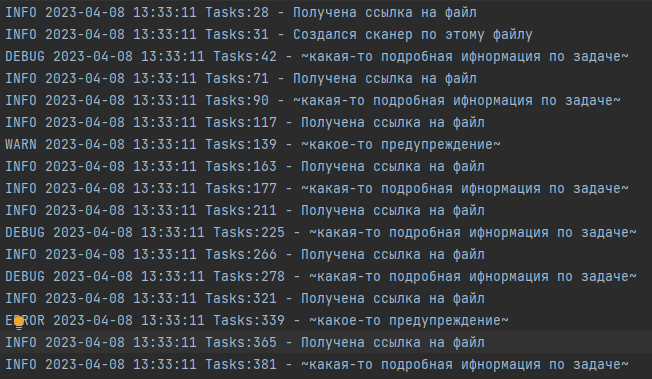


Рисунок 9 – Файл логирования уровня debug и выше



Рисунок 10 – Файл логирования уровня warn и выше

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