# Лабораторная работа № 2 студента группы ИТз-51 Черепченко Юрия Сергеевича Выполнение: \_\_\_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_

# Code Convention. Логирование

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

**Вариант-1**

# Задание к лабораторной работе

Привести код из 1-й лабораторной работы к требованиям конвенции.

Так же необходимо в 1-ю лабораторную добавить логирование основных действий приложения с использованием библиотеки log4j. Логи должны выводится 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

Маска лога должна содержать следующую информацию:

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

<error.printStackTrace> (для ERROR и FATAL)

# Ход работы

* + - 1. Привел код из первой лабораторной работы к принципам конвенции.
      2. Добавил библиотеку log4j, и пролонгировал основные действия приложения. Логи выводятся и в файл, и на консоль.

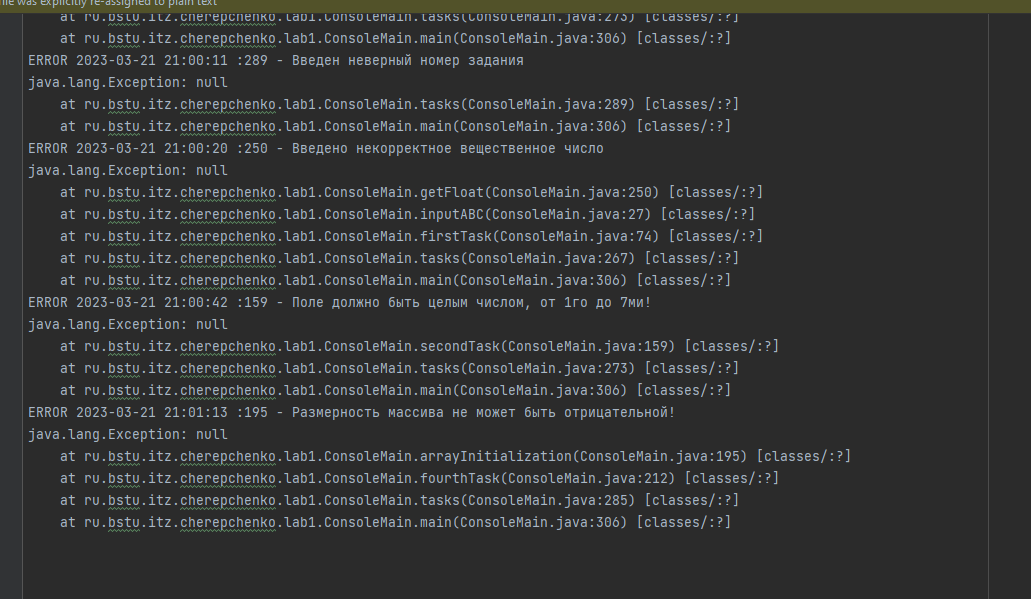


Рисунок файл warn.log

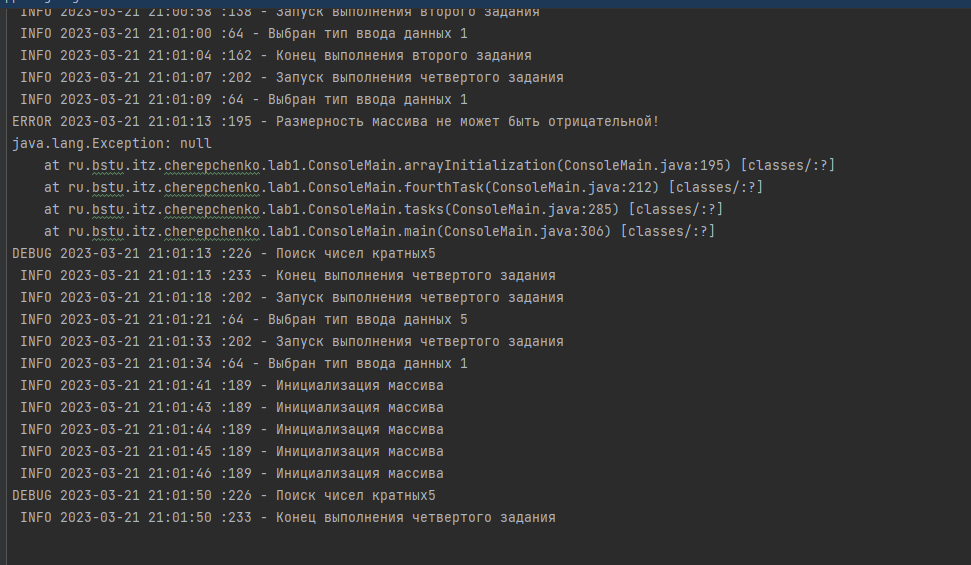


Рисунок файл info.log

* + - 1. Разделил логи по уровням.
      2. Маска лога соответствует заданию.

Текст программы.

ConsoleMain:

package ru.bstu.itz.cherepchenko.lab1;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
import org.apache.logging.log4j.core.LoggerContext;  
  
import java.io.\*;  
import java.util.Date;  
import java.util.List;  
import java.util.Scanner;  
  
public class ConsoleMain {  
  
 Logger log;  
  
 public ConsoleMain(Logger log){  
 this.log = log;  
 }  
 private float[] inputABC() {  
 log.info("Ввод последовательности чисел");  
 float a, b, c;  
 System.*out*.print("Введите 1 число: ");  
 a = getFloat();  
 System.*out*.print("Введите 2 число: ");  
 b = getFloat();  
 System.*out*.print("Введите 3 число: ");  
 c = getFloat();  
  
 return new float[]{a, b, c};  
 }  
  
 public float[] readFromFile(String path) {  
 try {  
 log.info("Ввод из файла");  
 File file = new File(path);  
 if (!file.exists()) {  
 file.createNewFile();  
 PrintWriter printWriter = new PrintWriter(file);  
 printWriter.println("3 4 5");  
 printWriter.close();  
 }  
 Scanner scanner = new Scanner(file);  
 String string = "";  
 while (scanner.hasNextLine()) {  
 string = scanner.nextLine();  
 }  
 String[] numbersString = string.split(" ");  
 float numbers[] = new float[numbersString.length];  
 for (int i = 0; i < numbersString.length; i++) {  
 numbers[i] = Float.*parseFloat*(numbersString[i]);  
 }  
 return numbers;  
  
 } catch (IOException e) {  
 log.error("Ошибка при вводе из файла", e, new Exception());  
 return new float[0];  
 }  
 }  
  
  
 private int inputKey() {  
 System.*out*.println("Введите 1, если хотите считывать из консоли, если хотите считывать из файла 2... ");  
 int choise = getNum();  
 log.info("Выбран тип ввода данных " + choise);  
 return choise;  
 }  
  
 private void firstTask() {  
 log.info("Запуск выполнения первого задания");  
 int choice = inputKey();  
 float[] abc;  
 switch (choice) {  
 case 1: {  
 abc = inputABC();  
 }  
 break;  
 case 2: {  
 abc = readFromFile("notes1.txt");  
 }  
 break;  
 default:  
 return;  
 }  
 float a = abc[0];  
 float b = abc[1];  
 float c = abc[2];  
  
 if ((a > b) && (b > c)) {  
 log.info("Аргументы удовлетворяют a>b>c (1 задание)");  
 System.*out*.println(  
 "a: " + String.*format*("%.2f", a \* 2) + '\n' +  
 "b: " + String.*format*("%.2f", b \* 2) + '\n' +  
 "c: " + String.*format*("%.2f", c \* 2)  
 );  
 } else {  
 log.info("Аргументы не удовлетворяют a>b>c (1 задание)");  
 System.*out*.println(  
 "a: " + String.*format*("%d", (int) a) + '\n' +  
 "b: " + String.*format*("%d", (int) b) + '\n' +  
 "c: " + String.*format*("%d", (int) c)  
 );  
 }  
 log.info("Конец выполнения первого задания");  
 }  
  
 private int inputDay() {  
 System.*out*.print("Введите день недели: ");  
 return getNum();  
 }  
  
  
 private int getSchedule(int dayNumber) {  
 switch (dayNumber) {  
 case 1:  
 case 7: {  
 return 0;  
 }  
 case 2: {  
 return 1;  
 }  
 case 3: {  
 return 2;  
 }  
 case 4:  
 case 6: {  
 return 3;  
 }  
 case 5: {  
 return 4;  
 }  
 default:  
 return 0;  
 }  
 }  
  
  
 private void secondTask() {  
 log.info("Запуск выполнения второго задания");  
 int choice = inputKey();  
 int day;  
 switch (choice) {  
 case 1: {  
 day = inputDay();  
 }  
 break;  
 case 2: {  
 day = (int) readFromFile("notes2.txt")[0];  
 }  
 break;  
 default:  
 return;  
 }  
 if ((1 <= day) && (day <= 7)) {  
 int countClasses = getSchedule(day);  
 System.*out*.printf(  
 "Количество пар: " + countClasses + "\n"  
 );  
 } else {  
 log.error("Поле должно быть целым числом, от 1го до 7ми!", new Exception());  
 System.*out*.print("Поле должно быть целым числом, от 1го до 7ми!\n");  
 }  
 log.info("Конец выполнения второго задания");  
 }  
  
 private void thirdTask() {  
 log.info("Запуск выполнения третьего задания");  
 Expression expr = new Expression(0, 15);  
 final String[] result = {"Результаты расчета функции: "};  
  
 log.info("Запуск расчета значений функции (3 задание)");  
 List<Integer> numbers = expr.getAllYs();  
 numbers.forEach(y -> result[0] += y + " ");  
  
 log.info("Проверка чисел на простоту (3 задание)");  
 if (expr.isSimpleList(numbers)) {  
 result[0] += "\n Все числа простые\n";  
 } else {  
 result[0] += "\n Не все числа простые\n";  
 }  
  
 System.*out*.printf(result[0]);  
 log.info("Конец выполнения третьего задания");  
 }  
  
 private float[] arrayInitialization(int n) {  
 if (n > 0) {  
 float nums[] = new float[n];  
 for (int i = 0; i < nums.length; i++) {  
 log.info("Инициализация массива");  
 System.*out*.print("Введите " + i + " элемент: ");  
 nums[i] = getFloat();  
 }  
 return nums;  
 } else {  
 log.error("Размерность массива не может быть отрицательной!", new Exception());  
 System.*out*.println("Размерность массива не может быть отрицательной!");  
 return new float[0];  
 }  
 }  
  
 private void fourthTask() {  
 log.info("Запуск выполнения четвертого задания");  
 int choice = inputKey();  
 int k;  
 float [] arr;  
 switch (choice) {  
 case 1: {  
 System.*out*.print("Введите K: ");  
 k = getNum();  
 System.*out*.print("Введите размерность массива: ");  
 int n = getNum();  
 arr = arrayInitialization(n);  
 }  
 break;  
 case 2: {  
 System.*out*.print("Введите K: ");  
 k = getNum();  
 arr = readFromFile("notes4.txt");  
 }  
 break;  
 default:  
 return;  
 }  
 int summary = 0;  
  
 log.debug("Поиск чисел кратных" + k);  
 for (float number : arr) {  
 if (number % k == 0)  
 summary += number;  
 }  
  
 System.*out*.printf("Сумма чисел кратных: " + summary + '\n');  
 log.info("Конец выполнения четвертого задания");  
 }  
  
 public int getNum() {  
 Scanner sc = new Scanner(System.*in*);  
 while (!sc.hasNextInt()) {  
 log.error("Введено некорректное целое число", new Exception());  
 System.*out*.println("Введенно некоректное значение!");  
 System.*out*.print("Введите значение повторно: ");  
 sc.next();  
 }  
 return sc.nextInt();  
 }  
  
 public float getFloat() {  
 Scanner sc = new Scanner(System.*in*);  
 while (!sc.hasNextFloat()) {  
 log.error("Введено некорректное вещественное число", new Exception());  
 System.*out*.println("Введенно некоректное значение!");  
 System.*out*.print("Введите значение повторно: ");  
 sc.next();  
 }  
 return sc.nextFloat();  
 }  
  
 private void tasks() {  
 System.*out*.print("Задача №1 - 1 \nЗадача №2 - 2 \nЗадача №3 - 3 \nЗадача №4 - 4 \nдля выхода нажмите 0 \n");  
 boolean run = true;  
 while (run) {  
 System.*out*.print("Введите номер задания: ");  
 int choice = getNum();  
 switch (choice) {  
 case 1: {  
 System.*out*.println("Первое задание.");  
 firstTask();  
 }  
 break;  
  
 case 2: {  
 System.*out*.println("Второе задание.");  
 secondTask();  
 }  
 break;  
  
 case 3: {  
 System.*out*.println("Третье задание.");  
 thirdTask();  
 }  
 break;  
  
 case 4: {  
 System.*out*.println("Четвертое задание.");  
 fourthTask();  
 }  
 break;  
 default: {  
 log.error("Введен неверный номер задания", new Exception());  
 System.*out*.println("Введенно неверное значение!");  
 }  
 break;  
  
 case 0: {  
 run = false;  
 }  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 LoggerContext context = (LoggerContext) LogManager.*getContext*(false);  
 File file = new File("log4j2.xml");  
  
 context.setConfigLocation(file.toURI());  
 new ConsoleMain(LogManager.*getRootLogger*()).tasks();  
 }  
}

Expression

package ru.bstu.itz.cherepchenko.lab1;  
  
  
import java.util.ArrayList;  
import java.util.List;  
  
import static java.lang.Math.*sqrt*;  
  
public class Expression {  
 int start;  
 int end;  
  
 public Expression(int start, int end) {  
 this.start = start;  
 this.end = end;  
 }  
  
 public List<Integer> getAllYs(){  
 List<Integer> Ys = new ArrayList<>();  
 for (int x=start; x<end; x++)  
 Ys.add(x\*x+x+17);  
 return Ys;  
 }  
  
 protected boolean isSimple(int ANum) {  
 if (ANum < 2)  
 return false;  
 double s = *sqrt*(ANum);  
 for (int i = 2; i <= s; i++) {  
 if (ANum % i == 0)  
 return false;  
 }  
 return true;  
 }  
  
 public boolean isSimpleList(List<Integer> numbers) {  
 for (Integer number : numbers)  
 if (!isSimple(number))  
 return false;  
 return true;  
 }  
}

log4g2:

<?xml version="1.0" encoding="UTF-8"?>  
<Configuration status="WARN">  
 <!-- Секция аппендеров -->  
 <Appenders>  
 <!-- Консольный аппендер -->  
 <Console name="Console" target="SYSTEM\_OUT">  
 <PatternLayout pattern="%5p %d{yyyy-MM-dd HH:mm:ss} %c{1}:%L - %m %n"/>  
 </Console>  
 <!-- Файловые аппендеры -->  
 <File name="InfoFile" fileName="info.log">  
 <PatternLayout pattern="%5p %d{yyyy-MM-dd HH:mm:ss} %c{1}:%L - %m %n"/>  
 </File>  
 <File name="WarnFile" fileName="warn.log">  
 <PatternLayout pattern="%5p %d{yyyy-MM-dd HH:mm:ss} %c{1}:%L - %m %n"/>  
 </File>  
 </Appenders>  
 <!-- Секция логгеров -->  
 <Loggers>  
 <Logger name="sample" level="debug"/>  
 <Root level="all">  
 <AppenderRef ref="Console"/>  
 <AppenderRef ref="InfoFile" level="debug"/>  
 <AppenderRef ref="WarnFile" level="warn"/>  
 </Root>  
 </Loggers>  
</Configuration>

Вывод: изучил правила документа code convention, изучил основф логирования в Java при помощи библиотеки log4j.