

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1
курса «Программирование»

Вариант № 108699

Выполнил студент:
Мухамедьяров Артур Альбертович
группа: Р3109

Преподаватель:
Гаврилов А. В.,
Мустафаева А. В.

ИТМО

Санкт-Петербург, 2024 г.

Содержание

Лабораторная работа № 1.	2
1. Задание варианта № 108699	2
2. Выполнение задания.	3
1. Листинги кода	3
3. Результат работы программы.	5
1. Первый запуск.	5
2. Второй запуск.	5
4. Вывод	6

Лабораторная работа № 1

1. Задание варианта № 108699

, , ,

1. Создать одномерный массив s типа `short`. Заполнить его числами от 1 до 16 включительно в порядке убывания.
2. Создать одномерный массив x типа `double`. Заполнить его 11-ю случайными числами в диапазоне от -14.0 до 11.0.
3. Создать двумерный массив w размером 16x11. Вычислить его элементы по следующей формуле (где $x = x[j]$):

- если $s[i] = 15$,

$$\text{то } w[i][j] = \ln \left(\sqrt{\left(\frac{|x| + 1}{2} \right)^2} \right);$$

- если $s[i] \in \{1, 3, 7, 8, 11, 12, 13, 16\}$,

$$\text{то } w[i][j] = \left(\frac{3}{4} \cdot \left(\left(\frac{\sqrt[3]{x}}{\frac{1}{3} + (x)^{x+\pi}} \right)^3 + 1 \right) \right)^3;$$

- для остальных значений $s[i]$:

$$w[i][j] = \tan \left(\left(\tan \left(\frac{1}{2/x} \right) \right)^{\left(\frac{2}{3} \cdot \tan(x) \right)^2} \right).$$

4. Напечатать полученный в результате массив в формате с четырьмя знаками после запятой.

, , ,

2. Выполнение задания.

Задание было выполнено в редакторе кода, позже собрано с помощью `javac` в `jar` файл `itmo_proga_lab1.jar` непосредственно на сервере.

2. 1. Листинги кода

Листинг из файла [1.1](#)

```
1 import java.util.random.RandomGenerator;
2
3 public class Main {
4     /*
5      * Create a public variable w for (possibly) future external use.
6      * Although for security reasons we should only use a function that only
7      * returns the value of the variable, otherwise it may cause bugs, but in
8      * this case we will ignore it.
9      */
10    public static double[][] w;
11
12    /*
13     * We will use the private type only for those functions that are used
14     * for internal calculations and must be inaccessible outside the class.
15     * This is a necessity to ensure class security and proper debugging and
16     * testing, otherwise it will be hard to catch places where data changes
17     * occur.
18     */
19
20    private static void generateSX(short[] s, double[] x) {
21        for (int i = 1; i <= 16; i++) {
22            // Generate reversed array (0=16..15=1)
23            s[16 - i] = (short) i;
24        }
25        for (int i = 0; i < 11; i++) {
26            // Generate random number (type Int) in range -14..11, then
27            // add it to double array
28            // (It's automatically converts to double (-14 => -14.0))
29            x[i] = RandomGenerator.getDefault().nextInt(-14, 11);
30        }
31    }
32
33    private static void calculateW(double[][] w, short[] s, double[] x,
34    int i, int j) {
35        switch (s[i]) {
36            case 15:
37                w[i][j] = Math.log10(Math.sqrt(Math.pow((Math.abs(x[j]) +
38                1) / 2, 2)));
39                break;
40            case 1, 3, 7, 8, 11, 12, 13, 16:
41                w[i][j] = Math.pow(((double) 3 / 4 * ((Math.pow((Math.cbrt
42                (x[j])), 3) / ((double) 1 / 3 + Math.pow(x[j], x[j] + Math.PI))) + 1)),
43                3);
44                break;
45            default:
46                w[i][j] = Math.tan(Math.pow(Math.tan(((double) 1 / 2) / x[
47                j]), Math.pow((double) 2 / 3 * Math.tan(x[j]), 2)));
48        }
49    }
50 }
```

```

39
40 // Make the output readable and suitable for the requirements
41 private static void printFormatted(double[][] w) {
42     // Go through rows
43     for (double[] row : w) {
44         // Go through elements of row
45         for (double e : row) {
46             System.out.format("%8.4f ", e); // Set format width (8
symbols), and 4 symbols after dot
47         }
48         System.out.println(); // newline
49     }
50 }
51
52 public static void main(String[] args) {
53     // Initialize variables
54     short[] s = new short[16];
55     double[] x = new double[11];
56     w = new double[16][11];
57
58     // Generate array elements
59     generateSX(s, x);
60
61     for (int i = 0; i < 16; i++) {
62         for (int j = 0; j < 11; j++) {
63             calculateW(w, s, x, i, j);
64         }
65     }
66
67     printFormatted(w); // Output final elements
68 }
69 }

```

Листинг 1.1: Исходный код программы

3. Результат работы программы.

3. 1. Первый запуск.

0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.1761	0.5441	0.6532	0.7404	0.7782	0.8451	0.3979	0.8129	0.6021	0.4771	0.5441
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>
0.0553	1.2885	0.0000	0.6430	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.0000	<i>NaN</i>
0.4970	0.4219	0.4219	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>

3. 2. Второй запуск.

2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
0.0000	0.0000	0.3010	0.0000	0.7404	0.5441	0.8751	0.0000	0.3010	0.5441	-0.3010
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219
0.5741	<i>NaN</i>	1.5040	<i>NaN</i>	0.6430	1.2885	<i>NaN</i>	0.5741	<i>NaN</i>	1.2885	1.5574
2.2610	<i>NaN</i>	0.4263	<i>NaN</i>	0.4219	0.4219	<i>NaN</i>	2.2610	<i>NaN</i>	0.4219	0.4219

4. Вывод

Во время выполнения лабораторной работы я изучил синтаксис языка **Java**, встроенную библиотеку **Math**, научился работать со средством разработки Java (JDK). Также в процессе выполнения я научился работать с типами данных, классами, функциями, массивами и циклами. Полученные мною знания являются необходимой базой для дальнейшего изучения языка и разработки уже более комплексных проектов.

Также во время работы над лабораторной, я научился работать с официальной документацией Oracle по встроенной библиотеке **Math**[\[2\]](#), **RandomGenerator**[\[3\]](#), а также ознакомился с базовыми командами ***NIX**[\[5\]](#) и **Git**[\[4\]](#).

Литература

- [1] Ссылка на личный репозиторий GitHub: <https://github.com/pozitp/itmo-labs/tree/main/prog/lab1>

- [2] Ссылка на официальную документацию Oracle для JDK 17 по встроенной библиотеке Math: <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html>

- [3] Ссылка на официальную документацию Oracle для JDK 17 по встроенной библиотеке RandomGenerator: <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/random/RandomGenerator.html>

- [4] Ссылка на официальную документацию Git с базовыми командами для работы с системами контроля версий файлов: <https://git-scm.com/docs/giteveryday>

- [5] Ссылка на официальную документацию GNU по coreutils (базовые команды *NIX): <https://www.gnu.org/software/coreutils/manual/coreutils.html>