

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

Лабораторная работа № 1

Принципы ООП

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 `neovim`, позже собрано с помощью `javac` в `jar` файл `itmo_proga_lab1.jar`.

2. 1. Листинги кода

Листинг из файла [1.1](#)

```
1 import java.util.random.RandomGenerator;
2
3 public class Main {
4     public static double[][] w;
5
6     private static void generateRX(short[] s, double[] x) {
7         for (int i = 1; i <= 16; i++) {
8             s[16 - i] = (short) i;
9         }
10        for (int i = 0; i < 11; i++) {
11            x[i] = RandomGenerator.getDefault().nextDouble(-14.0, 11.0);
12        }
13    }
14
15    private static void calculateW(double[][] w, short[] s, double[] x,
16    int i, int j) {
17        switch (s[i]) {
18            case 15:
19                w[i][j] = Math.log10(Math.sqrt(Math.pow((Math.abs(x[j]) +
20                1) / 2, 2)));
21                break;
22            case 1, 3, 7, 8, 11, 12, 13, 16:
23                w[i][j] = Math.pow(((double) 3 / 4 * ((Math.pow((Math.cbrt
24                (x[j])), 3) / ((double) 1 / 3 + Math.pow(x[j], x[j] + Math.PI))) + 1)),
25                3);
26                break;
27            default:
28                w[i][j] = Math.tan(Math.pow(Math.tan(((double) 1 / 2) / x[
29                j]), Math.pow((double) 2 / 3 * Math.tan(x[j]), 2)));
30        }
31    }
32
33    private static void printFormatted(double[][] w) {
34        for (double[] row : w) {
35            for (double e : row) {
36                System.out.format("%8.4f ", e);
37            }
38            System.out.println();
39        }
40    }
41
42    public static void main(String[] args) {
43        short[] s = new short[16];
44        double[] x = new double[11];
45        w = new double[16][11];
46
47        generateRX(s, x);
48
49        for (int i = 0; i < 16; i++) {
```

```

45         for (int j = 0; j < 11; j++) {
46             calculateW(w, s, x, i, j);
47         }
48     }
49
50     printFormatted(w);
51 }
52 }

```

Листинг 1.1: Исходный код программы

3. Результат работы программы.

3. 1. Первый запуск.

0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.4465	0.0044	0.5442	0.2940	0.5439	0.2140	0.2207	0.8667	0.1828	0.1650	0.5961
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>
0.0000	0.5157	<i>NaN</i>	1.4478	<i>NaN</i>	<i>NaN</i>	0.5003	<i>NaN</i>	<i>NaN</i>	0.0133	<i>NaN</i>
0.4219	2.1391	<i>NaN</i>	0.4272	<i>NaN</i>	<i>NaN</i>	0.4517	<i>NaN</i>	<i>NaN</i>	0.5154	<i>NaN</i>

3. 2. Второй запуск.

0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
0.2727	0.8196	0.6911	0.6421	0.2623	0.7320	0.5301	0.4477	0.6763	0.7570	0.2665
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>
1.2042	<i>NaN</i>	0.6068	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.1006	<i>NaN</i>	<i>NaN</i>
0.4310	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	0.4219	<i>NaN</i>	<i>NaN</i>

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], а также ознакомился с базовыми командами *NIX[4] и Git[3].

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 Git с базовыми командами для работы с системами контроля версий файлов: <https://git-scm.com/docs/giteveryday>

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