

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 № 26993

Выполнил студент:
Бых Даниил Максимович
группа: Р3109

Преподаватель:
Гаврилов А. В.,
Наумова Н. А.

ИТМО

Санкт-Петербург, 2025 г.

Содержание

Лабораторная работа № 1.	2
1. Задание варианта № 26993	2
2. Выполнение задания.	3
1. Листинги кода	3
3. Результат работы программы.	4
1. Первый запуск.	4
2. Второй запуск.	5
4. Вывод	6

Лабораторная работа № 1

1. Задание варианта № 26993

1. Создать одномерный массив n типа `int`. Заполнить его числами от 6 до 20 включительно в порядке возрастания.
2. Создать одномерный массив x типа `double`. Заполнить его 10-ю случайными числами в диапазоне от -10.0 до 10.0.
3. Создать двумерный массив w размером 15x10. Вычислить его элементы по следующей формуле (где $x = x[j]$):

- если $n[i] = 20$,
то $w[i][j] = \left(\frac{0.5}{(\pi \cdot (x)^{\frac{3-x}{4}})^2} \right)^{\tan(\ln(|x|))}$;
- если $n[i] \in \{8, 11, 13, 14, 17, 18, 19\}$,
то $w[i][j] = \left(\frac{\arcsin(\sin(x)) + 1}{1} / 2 \right)^2$;
- для остальных значений $n[i]$:
 $w[i][j] = e^{\sin(\sqrt[3]{\frac{x+1}{x}}^x)}$.

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 `lab1.jar` непосредственно на сервере.

2. 1. Листинги кода

Листинг из файла [1.1](#)

```
1 import java.util.Random;
2
3 public class Lab1 {
4     public static void main(String[] args) {
5         // Заполняем 1 массив
6         int[] l = new int[15];
7         for (int i=6; i<21; i++) {
8             l[i-6] = i;
9         }
10        // Заполняем 2 массив
11        double[] x = new double[10];
12        Random rand = new Random();
13        for (int i=0; i<10; i++) {
14            x[i] = rand.nextDouble(-10.0d, 10.0d);
15        }
16        // Заполняем 3 массив
17        double[][] w = new double[15][10];
18        for (int i=0; i<15; i++) {
19            for (int j=0; j<10; j++) {
20                w[i][j] = calcArrayElement(l[i], x[j]);
21            }
22        }
23        // Выводим результат
24        printArray(w);
25    }
26
27    // Определение элемента для 3 массива согласно заданию
28    private static double calcArrayElement(int l, double x) {
29        if (l == 20) {
30            return Math.pow(
31                0.5 / (Math.pow(Math.PI * Math.pow(x, (3/4-x / x)), 2)),
32                Math.tan(Math.log1p(Math.abs(x)))
33            );
34        }
35
36        int[] check_sums = {8, 11, 13, 14, 17, 18, 19};
37        for (int i: check_sums) {
38            if (l == i) {
39                return Math.pow(
40                    (x + 1) / 2,
41                    2
42                );
43            }
44        }
45
46        return Math.pow(
47            Math.E,
48            Math.sin(Math.pow(Math.pow((x+1) / x, x), 1/3))
49        );
50    }
51 }
```

```

50     }
51
52     // Вывод массива
53     private static void printArray(double[][] arr) {
54         for (double[] i : arr) {
55             for (double j : i) {
56                 System.out.format("%12.4f ", j);
57             }
58             System.out.println();
59         }
60     }
61 }

```

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>