

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
ИТМО»

Факультет Программной инженерии и компьютерной техники  
Дисциплина “Программирование”

Отчёт по лабораторной работе №1  
Перевод чисел между различными системами счисления  
Вариант №29029

Выполнил:

Ануфриев Андрей Сергеевич,  
Р3119

Проверил:

Ермаков Михаил Константинович

г. Санкт-Петербург

2024 год

## Оглавление

1. Текст задания.....	3
2. Исходный код программы.....	4
3. Результат работы программы. ....	5
4. Выводы по работе. ....	6

## 1. Текст задания

1. Создать одномерный массив  $z$  типа `int`. Заполнить его чётными числами от 4 до 18 включительно в порядке возрастания.
2. Создать одномерный массив  $x$  типа `double`. Заполнить его 20-ю случайными числами в диапазоне от -15.0 до 12.0.
3. Создать двумерный массив  $z$  размером  $8 \times 20$ . Вычислить его элементы по следующей

- если  $z[i] = 10$ , то

$$z[i][j] = \left( \frac{1}{4} + \sin\left(\frac{x - \frac{1}{2}}{x}\right) \right)^{\sin((0.25 \cdot x)^x)};$$

- если  $z[i] \in \{4, 6, 14, 18\}$ , то  $z[i][j] = \sin\left(\frac{2}{(x)^2}\right);$

- для остальных значений  $z[i]$ :

$$z[i][j] = e^{\left(e^{\left(\frac{2}{3} \cdot (x-1)\right)^x}\right) \left(\frac{4}{x \cdot \left(\frac{2}{3} - x\right) - 3}\right)^2}.$$

формуле (где  $x = x[j]$ ):

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

## 2. Исходный код программы.

```
import static java.lang.Math.*;
public class Lab1 {
    public static void main(String[] args) {
        // создали и заполнили массив z
        int[] z = new int[8];
        for (int i = 0; i < 8; i++) {
            z[i] = 4 + 2 * i;
        }
        // создаём и заполняем массив x
        double[] x = new double[20];
        for (int i = 0; i < 20; i++) {
            x[i] = ((double) (Math.random() * 28) - 15);
        }
        //создаём и заполняем z1
        double[][] z1 = new double[8][20];
        for (int i = 0; i < 8; i++){
            for (int j = 0; j < 20; j++) {
                if (z[i] == 10) {
                    z1[i][j] = pow((0.25 + sin((x[j] - 0.5) / x[j])), sin(pow(0.25 * x[j],
x[j]))));
                } else if (z[i] == 4 || z[i] == 6 || z[i] == 14 || z[i] == 18) {
                    z1[i][j] = sin(2.0/pow(x[j], 2));
                } else {
                    z1[i][j] = pow(E, pow(pow(E, pow(2.0/3*(x[j]-1), x[j])),
pow(4/(x[j]*(2.0/3-x[j])-3), 2)));
                }
            }
        }
        //Выводим
        System.out.print("[");
        for (int i = 0; i < 8; i++){
            System.out.print("[");
            for (int j = 0; j < 20; j++){
                System.out.printf("%8.3f\t", z1[i][j]);
            }
            if (i != 7) {
                System.out.println("],");
            } else {
                System.out.print("]]");
            }
        }
    }
}
```

### 3. Результат работы программы.

```
[[0,171 0,034 0,013 0,063 0,020 0,245 0,021 0,018 0,025 0,049 0,121 0,110 0,040 0,231 0,960 0,784 0,092 0,036 0,036 0,024 ],  
[0,171 0,034 0,013 0,063 0,020 0,245 0,021 0,018 0,025 0,049 0,121 0,110 0,040 0,231 0,960 0,784 0,092 0,036 0,036 0,024 ],  
[NaN NaN NaN Infinity NaN 4,078 Infinity NaN NaN Infinity NaN NaN NaN 4,200 NaN 3,268 7537,418 NaN NaN Infinity ],  
[NaN NaN NaN 1,020 NaN 0,994 1,001 NaN NaN 1,010 NaN NaN NaN 0,995 NaN 0,968 1,026 NaN NaN 1,049 ],  
[NaN NaN NaN Infinity NaN 4,078 Infinity NaN NaN Infinity NaN NaN NaN 4,200 NaN 3,268 7537,418 NaN NaN Infinity ],  
[0,171 0,034 0,013 0,063 0,020 0,245 0,021 0,018 0,025 0,049 0,121 0,110 0,040 0,231 0,960 0,784 0,092 0,036 0,036 0,024 ],  
[NaN NaN NaN Infinity NaN 4,078 Infinity NaN NaN Infinity NaN NaN NaN 4,200 NaN 3,268 7537,418 NaN NaN Infinity ],  
[0,171 0,034 0,013 0,063 0,020 0,245 0,021 0,018 0,025 0,049 0,121 0,110 0,040 0,231 0,960 0,784 0,092 0,036 0,036 0,024 ]]
```

## 4. Ответы на вопросы

### 1) Особенности языка Java:

- **Объектно-ориентированность.** [12](#) Каждый компонент программы (объект) содержит данные и методы для их обработки. Это помогает создавать модульные и легко поддерживаемые приложения, повторно использовать код в разном ПО. [2](#)
- **Кроссплатформенность.** [12](#) Способность работать на разных операционных системах без изменений в исходном коде. Это сделала возможной Java Virtual Machine (JVM), которая интерпретирует и выполняет байт-код Java для запуска на любой платформе, поддерживающей JVM. [2](#)
- **Многопоточность.** [12](#) Можно одновременно выполнять несколько задач внутри одного приложения. Это особенно полезно для многозадачных и высоконагруженных программ: веб-серверов и баз данных. [2](#)
- **Сильная типизация.** [1](#) Это означает, что каждая переменная должна иметь определённый тип данных, и любые операции над ними проверяются на соответствие типов. Это позволяет предотвращать множество ошибок на этапе компиляции, что делает код стабильнее. [2](#)

2) **JDK (Java Development Kit)** — это набор программ для разработки на Java. Он включает в себя JRE, загрузчик кода java, компилятор javac, архиватор jar, генератор документации javadoc и другие утилиты, нужные во время разработки. [2](#)

**JRE (Java Runtime Environment)** — окружение, необходимое для запуска Java-программ. Включает в себя стандартную библиотеку. В неё входят как базовые пакеты lang, util, так и пакеты для работы с различными форматами, базами данных, пользовательским интерфейсом. [2](#)

**Компиляция и выполнение программы** осуществляются с помощью **виртуальной машины Java (JVM)**. Она отвечает за преобразование байт-кода в машинный код. [1](#)

**JAR-архив** — это ZIP-архив, в котором содержится часть программы на языке Java.

3) **Приведение типов в Java** — это процесс преобразования одного типа данных в другой тип данных с помощью оператора приведения. [5](#)

**В Java существует два типа приведения:** [1](#)

5. **Расширяющееся приведение.** Выполняется автоматически (явно) при преобразовании меньшего по размеру типа в больший: byte в short в char в int в long в float в double. [1](#)

**Сужающее приведение.** Выполняется вручную при преобразовании (не явно) большего по размеру типа в меньший: double в float в long в int в char в short в byte. [1](#)

9. **Подпрограммы (методы) в Java** — это набор операторов, который выполняет определённую задачу. Они позволяют разбить сложную задачу на более мелкие части и многократно использовать код. [3](#)

**Параметры метода** — это значение, принимаемое методом. Метод может иметь любое количество параметров. Если метод создаётся с параметрами, при вызове метода нужно передать соответствующие значения. [3](#)

**Возвращаемое значение** метод может возвращать или не возвращать. Для возврата любого значения используется оператор

return

. Если метод не возвращает значение, его возвращаемый тип —

void

. [3](#)

**Общая форма объявления метода** выглядит следующим образом: модификатор, тип, имя метода (список параметров). Метод может не иметь параметров, в этом случае используются пустые скобки

6.

4) **switch** (argument) {

```
case value1:
    statement1;
    break;
case value2:
    statement2;
```

```
    break;
case valueN:
    statementN;
    break;
default:
    default_statement;

    break;
```

## 5. Выводы по работе.

В ходе работы я научился базовым операции Java и основной синтаксис. Узнал как пользоваться сервером helios, компилировать программу с помощью javac, создавать байт-код, создавать и запускать jar-файлы.