

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г. И. Носова»  
(ФГБОУ ВО «МГТУ им. Г.И. Носова»)

Кафедра вычислительной техники и программирования

### Лабораторная работа №1

по дисциплине «Метрология и стандартизация программного обеспечения»  
название лабораторной работы: «Оценка характеристик программы на основе  
лексического анализа»

Исполнитель: Варламов М.Н. студент 3 курса, группа АВб-19-1

Руководитель: Сибилева Н.С., ст. преподаватель каф. ВТиП

Магнитогорск, 2022

# СОДЕРЖАНИЕ

Введение.....	3
1    Постановка задания.....	4
2    Выполнения задания.....	4
Заключение.....	6

## ВВЕДЕНИЕ

В общем случае применение метрик позволяет руководителям проектов и предприятий изучить сложность разработанного или даже разрабатываемого проекта, оценить объем работ, стилистику разрабатываемой программы и усилия, потраченные каждым разработчиком для реализации того или иного решения. Однако метрики могут служить лишь рекомендательными характеристиками, ими нельзя полностью руководствоваться, так как при разработке ПО программисты, стремясь минимизировать или максимизировать ту или иную меру для своей программы, могут прибегать к хитростям вплоть до снижения эффективности работы программы.

## 1 ПОСТАНОВКА ЗАДАНИЯ

Дан двумерный массив А. Каждый элемент массива, стоящий выше главной диагонали, заменить его квадратом, а ниже диагонали – кубом его значения. Элементы главной диагонали оставить без изменения. Необходимо написать программу, решающую данную задачу. А также посчитать метрику Холстеда.

## 2 ВЫПОЛНЕНИЯ ЗАДАНИЯ

Программа была написана языке программирования “Java” и предоставлена ниже.

```
1 package Lab1;
2
3 import java.util.Arrays;
4
5 //operators 20
6 //operands 20
7 //all operators 66
8 //all operands 62
9
10 public class Main {
11     public static void main(String[] args) {
12         int[][] a = new int[6][5];
13         Arrays.stream(a).forEach(ints -> {
14             for (int i = 0; i < ints.length; i++) {
15                 ints[i] = (int) (Math.random() * 10);
16             }
17         });
18
19         prettyPrint(a);
20
21         for (int i = 0; i < a.length; i++) {
22             for (int j = 0; j < a[i].length; j++) {
23                 if (j > i)
24                     a[i][j] *= a[i][j];
25                 if (j < i)
26                     a[i][j] = (int) Math.pow(a[i][j], 3);
27             }
28         }
29
30         prettyPrint(a);
31     }
32
33     public static void prettyPrint(int[][] a) {
34         Arrays.stream(a).forEach(ints ->
35             System.err.println(Arrays.toString(ints)));
36     }
```

Результат выполнения данной программы представлен на рисунке 1.

```
[4, 3, 4, 0, 0]
[8, 7, 9, 4, 6]
[3, 1, 0, 4, 5]
[2, 8, 1, 9, 1]
[5, 6, 8, 8, 3]
[7, 1, 4, 4, 6]

[4, 9, 16, 0, 0]
[512, 7, 81, 16, 36]
[27, 1, 0, 16, 25]
[8, 512, 1, 9, 1]
[125, 216, 512, 512, 3]
[343, 1, 64, 64, 216]
```

Рисунок 1 – результат работ программы

Для расчета и анализа качества кода были собраны исходные данные, представленные в таблице 1.

Таблица 1 – Операторы, операции и операнды программы.

Операторы и операции программы			
№	Операторы и операции	Номер строки	Количество повторений
1	[]	12, 15, 22, 24, 26, 33	16
2	.	13,14,15,21,22,26,34,35	16
3	->	13, 34	2
4	*	15	1
5	>	23	1
6	<	14,21,22,25	4
7	++	14,21,22	3
8	=	12,14,15,21,22,24,26	7
9	*=	24	1
10	+	14	1
11	Concatination	34	1
12	Cast	15,26	2
13	new	12	1
14	prettyPrint()	19,30	2
15	pow()	26	1
16	toString()	34	1
17	println()	34	1
18	stream()	13,34	2
19	forEach()	13,34	2
20	random()	15	1
Всего: 66			

Операнды программы			
№	Операнд	Номер строки	Количество повторений
1	0	14,21,22	3
2	3	26	1
3	5	12	1
4	6	12	1
5	10	15	1
6	a	12,13,19,21,22,24,26,29,32,33	12
7	i	14,15,21,22,23,24,25,26	14
8	j	22,23,24,25,26	9
9	ints	13,33	2
10	ints[i]	15	1
11	ints.length	14	1
12	a[i]	22,24,26	5
13	a[i][j]	24,26	4
14	a.length	21	1
15	a[i].length	22	1
16	“\n”	33	1
17	Arrays	33	1
18	System	33	1
19	Math	15	1
20	err	33	1
<b>Всего: 62</b>			
Исходные данные			

Теперь необходимо рассчитать все необходимые характеристики, подставив имеющиеся данные. Результат расчета характеристик представлен в таблице 2.

Таблица 2 – Итоговая таблица показателей.

№	Характеристика	Обозначение и формула для вычисления	Значение
1	Число уникальных операторов и операций	$n_1$	20
2	Число уникальных операндов	$n_2$	20
3	Общее число всех операторов и операций	$N_1$	66

4	Общее число всех операндов	$N_2$	62
5	Число входных и выходных переменных (параметров)	$n_2^*$	2
6	Словарь программы	$n = n_1 + n_2$	40
7	Длина реализации программы	$N = N_1 + N_2$	128
8	Объем программы	$V = (N_1 + N_2) * \log_2(n_1 + n_2)$	681.21
9	Потенциальный объем программы	$V^* = (n_2^* + 2) * \log_2(n_2^* + 2)$	8
10	Уровень реализации программы	$L = V^* / V$	0.02
11	Уровень реализации языка	$\lambda = L * V^*$	0.16

## ЗАКЛЮЧЕНИЕ

Подводя итог, хотелось бы отметить, что ни одной универсальной метрики не существует. Любые контролируемые метрические характеристики программы должны контролироваться либо в зависимости друг от друга, либо в зависимости от конкретной задачи, кроме того, можно применять гибридные меры, однако они так же зависят от более простых метрик и также не могут быть универсальными. Строго говоря, любая метрика — это лишь показатель, который сильно зависит от языка и стиля программирования, поэтому ни одну меру нельзя возводить в абсолют и принимать какие-либо решения, основываясь только на ней.