



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»
КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №2

«Оценка качества программного продукта»

ДИСЦИПЛИНА: «Основы программной инженерии»

Выполнил: студент гр. ИУК4-21Б


(подпись)

(Суриков Н.С.)
(Ф.И.О.)

Проверил:


(подпись)

(Амеличев Г. Э.)
(Ф.И.О.)

Дата сдачи (защиты): 04.05.2024

Результаты сдачи (защиты):

- Балльная оценка: (6)

- Оценка:

Калуга, 2024

Цель: изучение основных методов и подходов оценки качества программного продукта.

Задачи:

1. изучить основы метрической теории Холстеда;
2. для написанных ранее программ произвести расчет количественных характеристик программ;
3. сравнить полученные результаты.

Листинг программы:

```
1  #include <iostream>
2  #include <cstdlib>
3
4  const int MATRIX_SIZE = 17;
5
6  int matrix[MATRIX_SIZE][MATRIX_SIZE];
7
8  void generateRandomMatrix()
9  {
10     for (auto &i : matrix)
11     {
12         for (int &j : i)
13         {
14             j = std::rand() % 25 - 12;
15         }
16     }
17 }
18
19 void printMatrix()
20 {
21     for (auto &i : matrix)
22     {
23         for (int j : i)
24         {
25             std::cout << j << " ";
26         }
27         std::cout << std::endl;
28     }
29 }
30
31 int calculateDigitSum(int number)
32 {
33     number = abs(number);
34     return number / 10 + number % 10;
35 }
```

```

36
37 void modifyMatrix()
38 {
39     for (int i = 0; i < MATRIX_SIZE; i++)
40     {
41         for (int j = 0; j < MATRIX_SIZE; j++)
42         {
43             matrix[i][j] += calculateDigitSum(matrix[i][MATRIX_SIZE - 1 -
i]);
44         }
45     }
46 }
47
48 bool isNegative(int number)
49 {
50     return number < 0;
51 }
52
53 void checkRowForAlternatingSigns()
54 {
55     int numberOfRow{};
56     std::cin >> numberOfRow;
57     for (int j = 1; j < MATRIX_SIZE; j++)
58     {
59         if (isNegative(matrix[numberOfRow][j]) ==
isNegative(matrix[numberOfRow][j - 1]))
60         {
61             std::cout << "Строка содержит знакопередающиеся элементы" <<
std::endl;
62             break;
63         }
64     }
65 }
66
67 void findFirstZeroElement()
68 {
69     for (int i = MATRIX_SIZE / 3; i < MATRIX_SIZE; i++)
70     {
71         for (int j = 0; j < MATRIX_SIZE; j++)
72         {
73             if (matrix[i][j] == 0)
74             {
75                 std::cout << i << " " << j << std::endl;
76                 return;
77             }
78         }
79     }
80 }
81
82 int main()
83 {
84     generateRandomMatrix();
85     printMatrix();
86     modifyMatrix();

```

```

87     checkRowForAlternatingSigns();
88     findFirstZeroElement();
89     printMatrix();
90     return 0;
91 }

```

Характеристики программы:

Номер	Оператор	Число вхождений	Номер	Операнд MATRIX_	Число вхождений
1	=	16	1	SIZE	9
2	==	2	2	matrix	7
3	<	6	3	i	15
4	++	4	4	j	18
5	()	16	5	number	7
6	+=	1	6	17	1
7	;	32	7	25	1
8	&	3	8	-12	1
9	<=	1	9	10	2
10	[]	20	10	3	1
11	%	2	11		
12	/	1	12		
13	-	4	13		
14	<<	9	14		
15	>>	1	15		
16	rand	1	16		
17	cout	4	62		
18	for	11			
19	if	2			
20	main	1			
21	generateRandomMatrix	2			
22	printMatrix	3			
23	modifyMatrix	2			
	checkRowForAlternating				
24	Signs	2			
25	findFirstZeroElement	2			
26	calculateDigitSum	2			
27	isNegative	3			
28	endl	3			
156					

Число уникальных операторов (n1)	156
Число уникальных операндов (n2)	62
Общее число операторов(N1)	147
Общее число операндов(N2)	98
Словарь программы (n)	41
Экспериментальная длина программы(Nэ)	245
Теоретическая длина программы(N)	180,10
Объем программы(V)	1312,60
Потенциальный объем(V*)	43,02
Граничный объем(Vгр)	126,26
Уровень программы(L)	0,033
Сложность программы(S)	30,512
Оценка уровня программы(L^)	0,013
Интеллект программы(I)	17,144
Работа по программированию(E)	40049,68
Время программирования(T)	2224,98
Ожидание времени кодирования(T^)	32595,849
Уровень языка программирования(λ)	0,0462
Ожидаемое число ошибок(B)	0,438

Вывод: в ходе данной лабораторной работы были изучены основные методы и подходы оценки качества программного продукта.