**МИНОБРНАУКИ РОССИИ**
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**
**Кафедра МОЭВМ**

# ОТЧЕТ

**по лабораторной работе №2**

**по дисциплине «Качество и метрология программного обеспечения»**

**Тема: Анализ структурной сложности графовых моделей программ**

Студент гр. 8304 _____ Рыжиков А.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2022

**Цель работы.**

Выполнить оценивание структурной сложности двух программ с помощью критериев:

- Минимального покрытия вершин и дуг графа управления;
- Выбора маршрутов на основе цикломатического числа графа.

Варианты программ:

- Программа с заданной структурой управляющего графа, выбираемой из файла zadan_struct.doc в соответствии с номером в списке группы;
- Программа из 1-ой лабораторной работы (управляющий граф составить самостоятельно).

Оцениваемые характеристики структурной сложности:

- Число учитываемых маршрутов проверки программы для заданного критерия;
- Цикломатическое число;
- Суммарное число ветвлений по всем маршрутам – оценка структурной сложности;

**Ход выполнения. Вариант 15**

1. Было проведено оценивание структурной сложности программы с заданной структурой управляющего графа — см. рис. 1.

Вычисление сложности программы по критерию минимального покрытия вершин и дуг графа управления:

m1: **1** – **3** – 14 – 15 S1=2

m2: **1** – **3** – 6 – 9 – **12** – 14 – 15 S2=3

m3: **1** – **2** – **5** – 7 – 9 – **12** – 9 – **12** – 14 – 15 S3=5

m4: **1** – **2** – **5** – **8** – **10** – **12** – 14 – 15 S4=6

m5: **1** – **2** – **5** – **8** – **10** – 13 – 14 – 15 S5=5

m6: **1** – **2** – **5** – **8** – 11 – 13 – 14 – 15 S6=4

m7: **1** – **2** – 4 – 6 – 9 – **12** – 14 – 15 S7=3

S=28

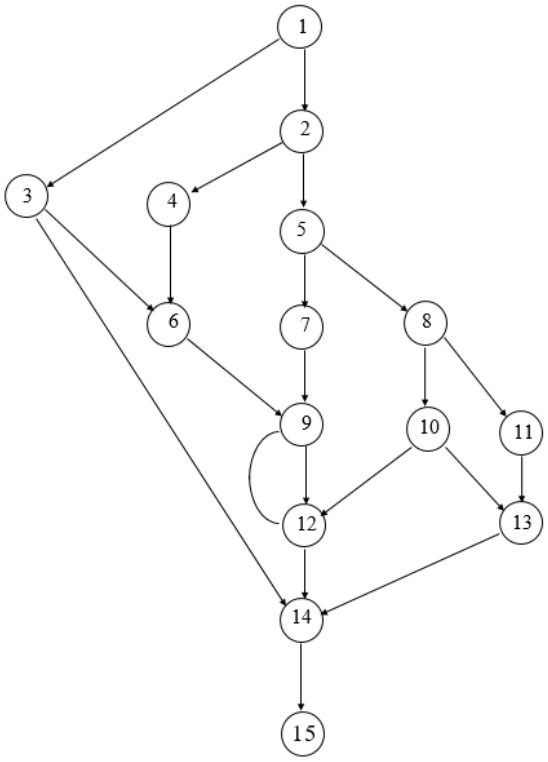


Рисунок 1 — Управляющий граф программы в соответствии с вариантом

Вычисление сложности программы с выбором маршрутов на основе цикломатического числа графа. Цикломатическое число Z исходного графа программы определяется формулой Z = Y - N +2*Ω, где Y – общее число дуг в графе; N – общее число вершин в графе; Ω – число связных компонент графа, число связных компонент графа Ω равно количеству дуг, необходимых для превращения исходного графа в максимально связный граф.:

Цикломатическое число **Z = 21 - 15 + 2×1 = 7**

m1: **1** – **3** – 14 – 15 S1=2

m2: **1** – **3** – 6 – 9 – **12** – 14 – 15 S2=3

m3: **1** – **2** – **5** – 7 – 9 – **12** – 9 – **12** – 14 – 15 S3=5

m4: **1** – **2** – **5** – **8** – **10** – **12** – 14 – 15 S4=6

m5: **1** – **2** – **5** – **8** – **10** – 13 – 14 – 15 S5=5

m6: **1** – **2** – **5** – **8** – 11 – 13 – 14 – 15 S6=4

m7: **1** – **2** – 4 – 6 – 9 – **12** – 14 – 15 S7=3

S=28

Также было проведено оценивание структурной сложности заданной программы с помощью программы ways (вход программы см. в приложении А) — см. рис 2 и 3.

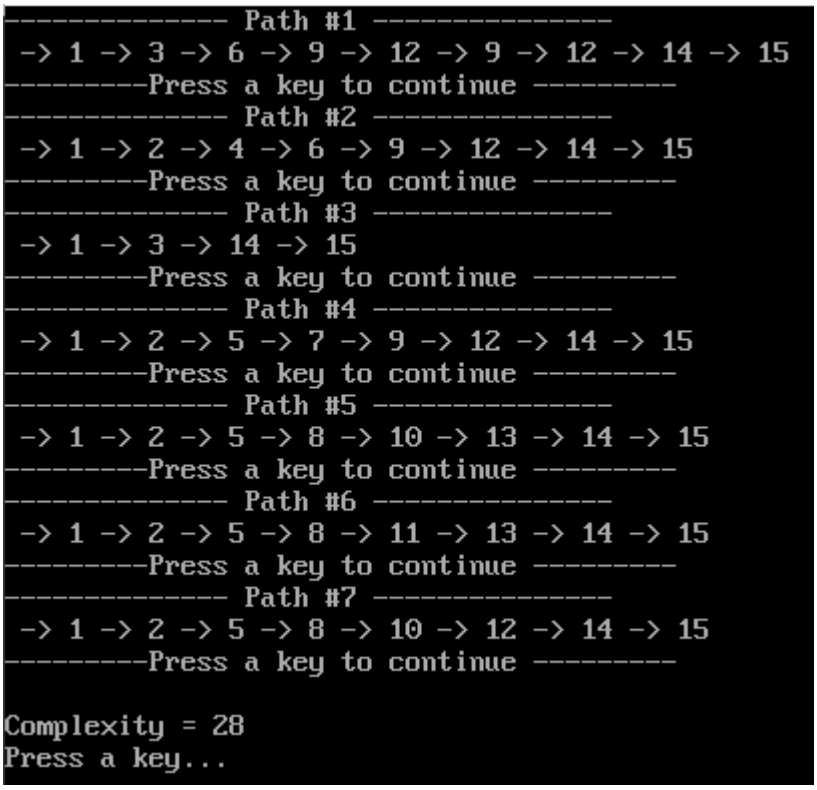


Рисунок 2 — Результат программы ways по критерию минимального покрытия

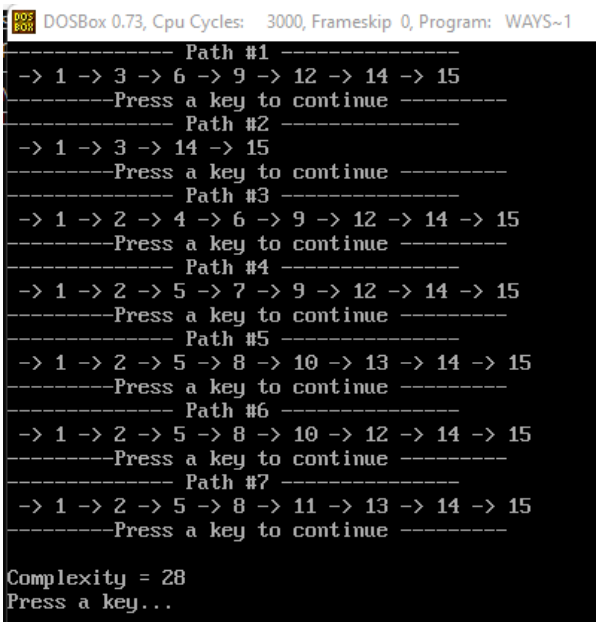


Рисунок 3 — Результат программы ways по путям по цикломатическому числу

Результат программы ways по критерию минимального покрытия (рис. 2) совпал с ручным расчетом. Результат программы по путям по цикломатическому числу (рис. 3) совпал с ручным расчетом.

2. Для программы из 1-ой лабораторной работы был составлен управляющий граф — см. рис. 4 (см. код программы в приложении В с комментариями, отражающими соответствие операторов и вершин).

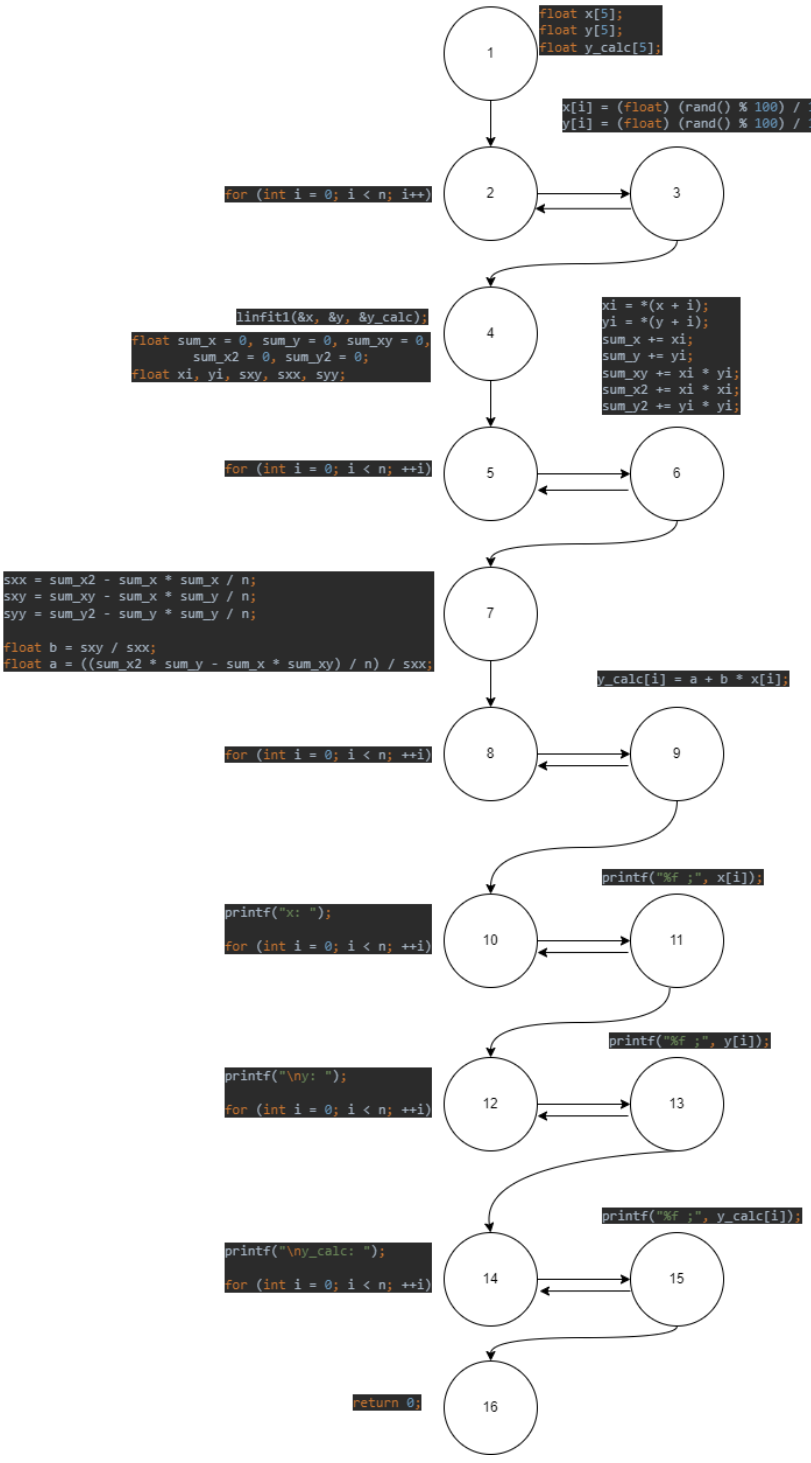


Рисунок 4 — Управляющий граф программы из лаб. работы 1

Вычисление сложности программы по критерию минимального покрытия вершин и дуг графа управления:

m1: 1 – 2 – **3** – 2 – **3** – 4 – 5 – **6** – 5 – **6** – 7 – 8 – **9** – 8 – **9** – 10 – **11** – 10 – **11** S1=6 – 12 – **13** – 12 – **13** – 14 – **15** – 14 – **15** – 16

S=12

Вычисление сложности программы с выбором маршрутов на основе цикломатического числа графа:

Цикломатическое число **Z = 21 - 16 + 2×1 = 7**

m1: 1 – 2 – **3** – 4 – 5 – **6** – 7 – 8 – **9** – 10 – **11** – 12 – **13** – 14 – **15** – 16 S1=6

m2: 2 – **3** – 2 S2=1

m3: 5 – **6** – 5 S3=1

m4: 8 – **9** – 8 S4=1

m5: 10 – **11** – 10 S5=1

m6: 12 – **13** – 12 S6=1

m7: 14 – **15** – 14 S7=1

S=12

Также было проведено оценивание структурной сложности заданной программы с помощью программы ways (вход программы см. в приложении С) — см. рис 5, 6 и 7.

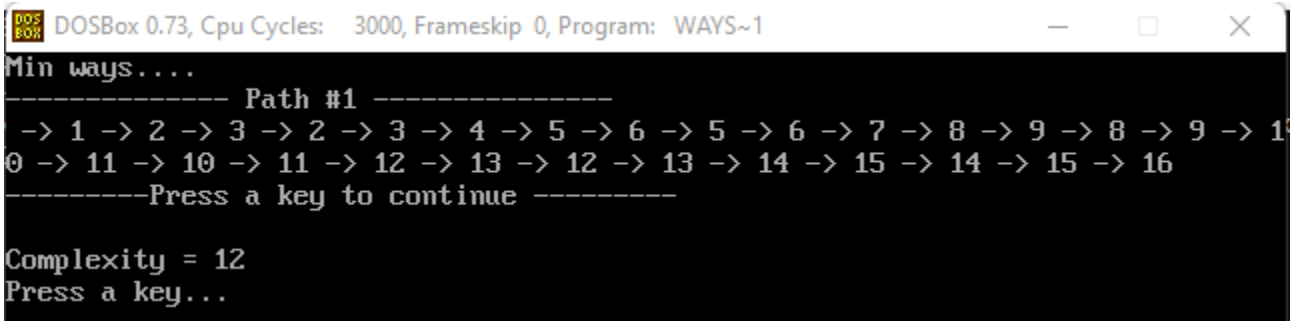


Рисунок 5 — Результат программы ways по критерию минимального покрытия

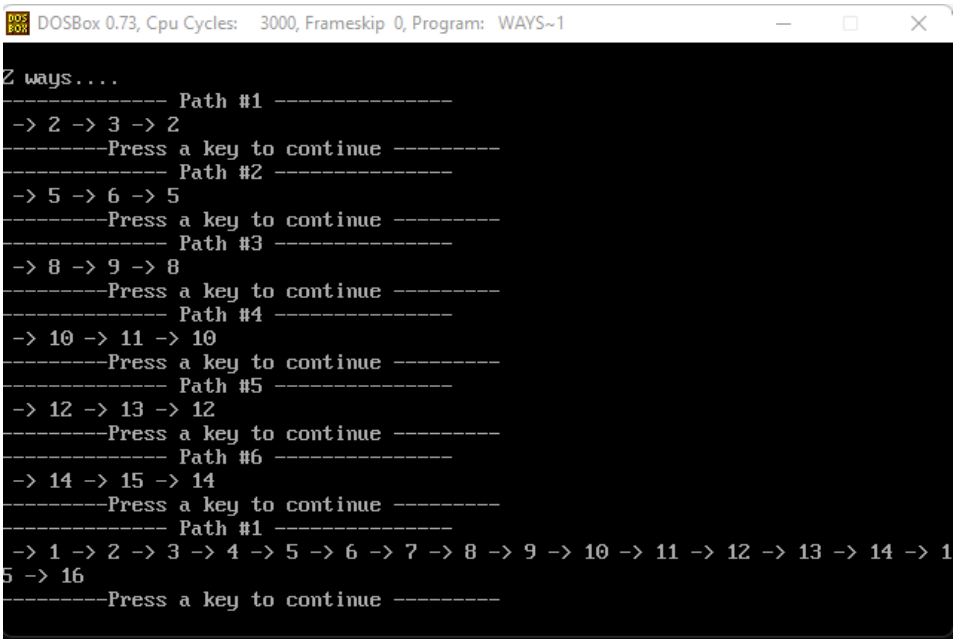


Рисунок 6 — Результат программы ways по путям по цикломатическому числу

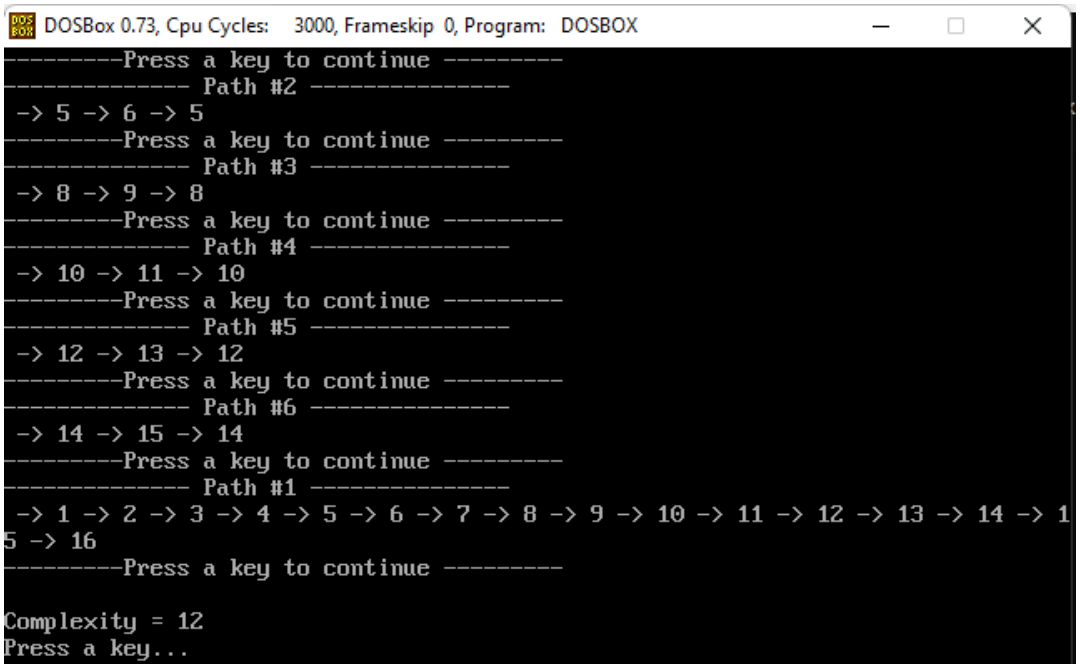


Рисунок 7 — Результат программы ways по путям по цикломатическому числу

**Выводы.**

В результате выполнения данной лабораторной работы была оценена структурная сложности двух программ с помощью критериев - минимального покрытия вершин и дуг графа управления; и выбора маршрутов на основе цикломатического числа графа.

# ПРИЛОЖЕНИЕ А. УПРАВЛЯЮЩИЙ ГРАФ ПРОГРАММЫ В СООТВЕТСТВИИ С ВАРИАНТОМ

Nodes{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}
Top{1}
Last{15}
Arcs{
arc(1,3);
arc(1,2);
arc(2,4);
arc(4,6);
arc(3,6);
arc(3,14);
arc(6,9);
arc(2,5);
arc(5,7);
arc(5,8);
arc(8,10);
arc(8,11);
arc(10,13);
arc(11,13);
arc(10,12);
arc(7,9);
arc(9,12);
arc(12,9);
arc(12,14);
arc(13,14);
arc(14,15);
}

# ПРИЛОЖЕНИЕ В. ПРОГРАММА НА ЯЗЫКЕ СИ ИЗ ЛАБ. РАБОТЫ 1

```c
#include <stdio.h>
#include <stdlib.h>

int n = 5;

void linfit2(float x[n], float y[n], float *y_calc) {

    float sum_x = 0, sum_y = 0, sum_xy = 0,
            sum_x2 = 0, sum_y2 = 0;
    float xi, yi, sxy, sxx, syy;
    for (int i = 0; i < n; ++i) {

        xi = x[i];
        yi = y[i];
        sum_x += xi;
        sum_y += yi;
        sum_xy += xi * yi;
        sum_x2 += xi * xi;
        sum_y2 += yi * yi;
    }

    sxx = sum_x2 - sum_x * sum_x / n;
    sxy = sum_xy - sum_x * sum_y / n;
    syy = sum_y2 - sum_y * sum_y / n;

    float b = sxy / sxx;
    float a = ((sum_x2 * sum_y - sum_x * sum_xy) / n) / sxx;
    for (int i = 0; i < n; ++i) {
        *(y_calc + i) = a + b * x[i];
    }
}

int main() {
    float x[5];
    float y[5];
    float y_calc[5];

    for (int i = 0; i < n; i++) {
        x[i] = (float) (rand() % 100) / 100;
        y[i] = (float) (rand() % 100) / 100;
    }

    linfit2(x, y, &y_calc);

    printf("x: ");

    for (int i = 0; i < n; ++i) {
        printf("%f ;", x[i]);
    }

    printf("\ny: ");

    for (int i = 0; i < n; ++i) {
        printf("%f ;", y[i]);
    }

    printf("\ny_calc: ");

    for (int i = 0; i < n; ++i) {
        printf("%f ;", y_calc[i]);
    }
}
```

# ПРИЛОЖЕНИЕ С. УПРАВЛЯЮЩИЙ ГРАФ ПРОГРАММЫ ИЗ ЛАБ. РАБОТЫ 1

Nodes{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}
Top{1}
Last{16}
Arcs{
arc(1,2);
arc(2,3);
arc(3,2);
arc(3,4);
arc(4,5);
arc(5,6);
arc(6,5);
arc(6,7);
arc(7,8);
arc(8,9);
arc(9,8);
arc(9,10);
arc(10,11);
arc(11,10);
arc(11,12);
arc(12,13);
arc(13,12);
arc(13,14);
arc(14,15);
arc(15,14);
arc(15,16);
}