

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

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Качество и метрология программного обеспечения»**  
**Тема: Анализ структурной сложности графовых моделей программ**

Студент гр. 7304

\_\_\_\_\_

Петруненко Д.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

**Цель работы:**

Изучение структурной сложности графовых моделей программ и метрик ее оценки.

**Задание:**

Выполнить оценивание структурной сложности двух программ с помощью критериев:

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

Варианты программ:

- Программа с заданной преподавателем 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:

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

## Ход работы

1. Был выбран вариант №13 для первой программы, у которой необходимо определить структурную сложность. Управляющий граф первой программы приведен на Рисунке 1:

### ВАРИАНТ

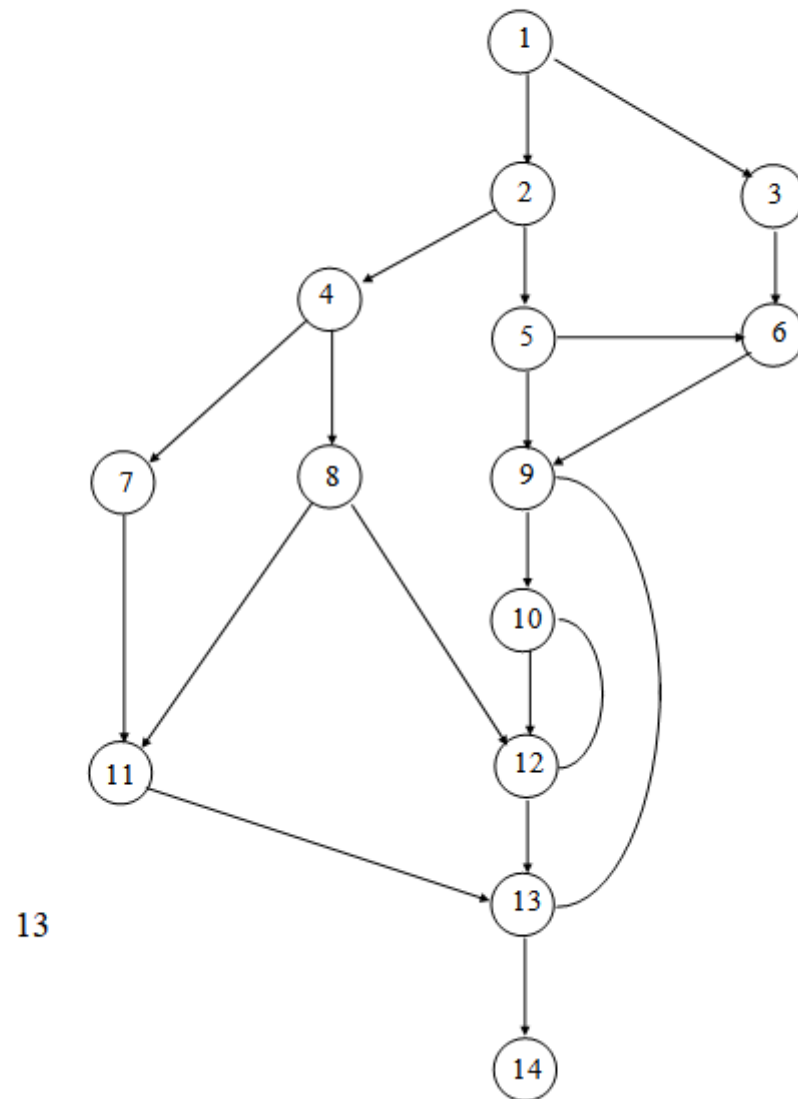


Рисунок 1: Управляющий граф первой программы для определения структурной сложности

- Ненаправленная дуга 10-12, 9-13 были удалены из графа, так как программа way.exe сообщает о некорректной структуре графа.

2. Была рассчитана структурная сложность первой программы вручную по первому критерию, а именно по минимальному покрытию вершин и дуг графа управления.

Ветвления в вершинах: 1, 2, 4, 5, 8

M1: **1-2-4**-7-11-13-14 (3 ветвлений)

M2: **1-2-4-8**-12-13-14 (4 ветвления)

M3: **1-2-4-8**-11-13-14 (4 ветвления)

M4: **1-3-6-9-10-12-13-14** (1 ветвления)

M5: **1-2-5-9-10-12-13-14** (3 ветвления)

M6: **1-2-5-6-9-10-12-13-14** (3 ветвления)

Количество маршрутов:  $M = 6$

Сложность:  $S = 3 + 4 + 4 + 1 + 3 + 3 = 18$

3. Было рассчитано вручную цикломатическое число графа первой программы для дальнейшего применения второго критерия. Для данной программы число вершин в графе равно 14, число дуг – 18, число связных компонент графа 1, тогда:  $Z = Y - N + 2 * P = 18 - 14 + 2 * 1 = 6$ , то есть цикломатическое число равно 6.

4. Для управляющего графа первой программы вручную были построены 8 линейно-независимых циклов и линейно-независимых маршрутов, после чего была подсчитана структурная сложность по второму критерию.

M1: **1-2-4**-7-11-13-14 (3 ветвлений)

M2: **1-2-4-8**-12-13-14 (4 ветвления)

M3: **1-2-4-8**-11-13-14 (4 ветвления)

M4: **1-3-6-9-10-12-13-14** (1 ветвления)

M5: **1-2-5**-9-10-12-13-14 (3 ветвления)

M6: **1-2-5**-6-9-10-12-13-14 (3 ветвления)

Сложность:  $S = 3 + 4 + 4 + 1 + 3 + 3 = 18$

5. Для первой программы была подсчитана структурная сложность по двух критериям с помощью программы ways.exe. Результаты расчета структурной сложности по первому критерию приведены на Рисунке 2, по второму критерию на Рисунке 3:

```
Min ways....
----- Path #1 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 3 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 5 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 4 -> 8 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 4 -> 8 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #6 -----
-> 1 -> 2 -> 5 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----

Complexity = 18
Press a key...
```

Рисунок 2: Программный расчет структурной сложности первой программы по первому критерию

```
Z ways....
----- Path #1 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 4 -> 8 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 4 -> 8 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 5 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 5 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #6 -----
-> 1 -> 3 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----

Complexity = 18
Press a key...
```

Рисунок 3: Программный расчет структурной сложности первой программы по второму критерию

6. Для программы из первой лабораторной работы был составлен управляющий граф. Данный управляющий граф представлен на Рисунке 4:

4:

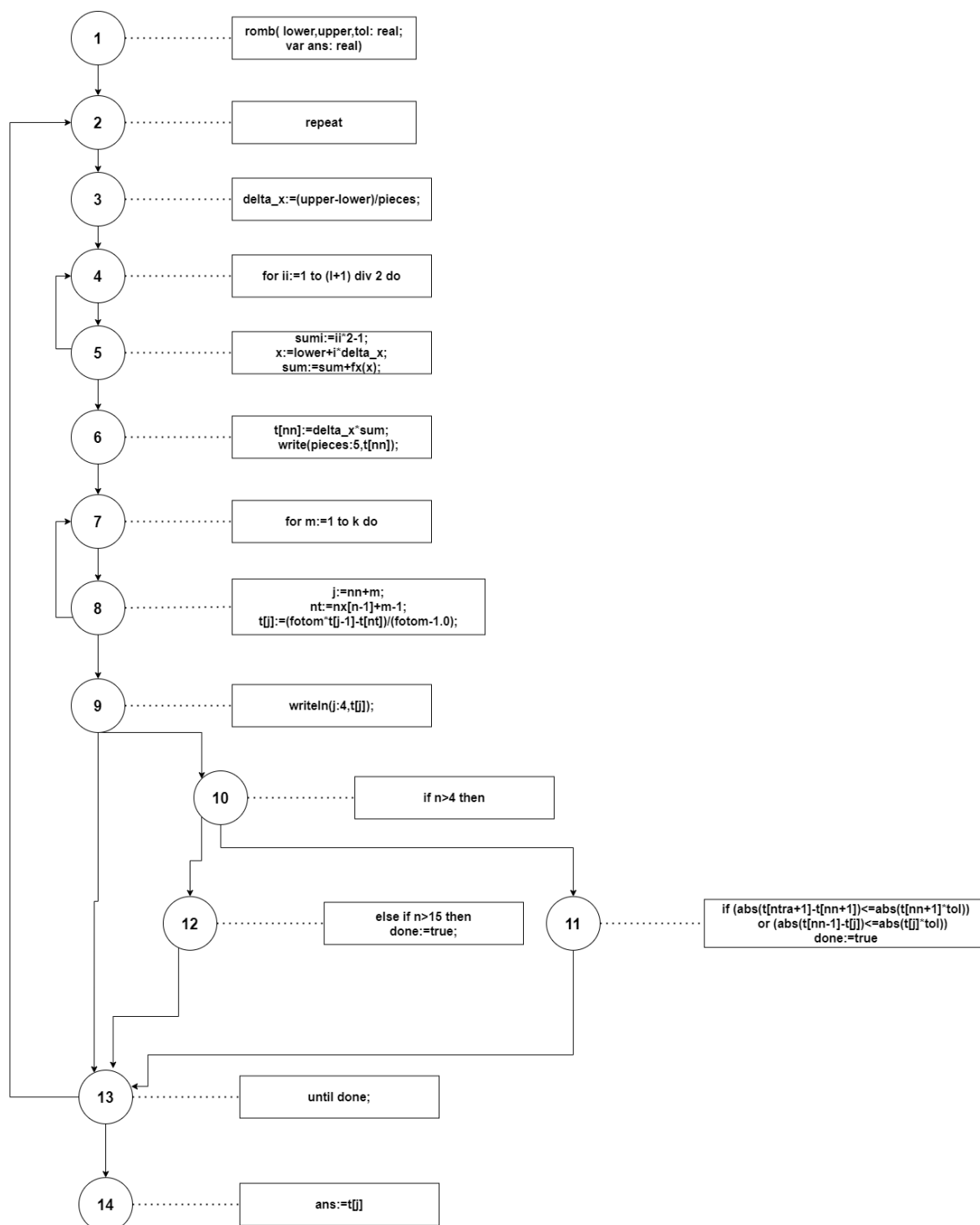


Рисунок 4: Управляющий граф программы из первой лабораторной работы для определения структурной сложности

7. Была рассчитана структурная сложность программы из первой лабораторной работы вручную по первому критерию, а именно по минимальному покрытию вершин и дуг графа управления.

Ветвления в вершинах: 5, 8, 9, 10, 13

M1: 1-2-3-4-5-4-5-6-7-8-7-8-9-10-11-13-2-3-4-5-6-7-8-9-13-2-3-4-5-6-7-8-9-10-12-13-14 (16 ветвлений)

Количество маршрутов:  $M = 1$

Сложность:  $S = 16$

8. Было рассчитано вручную цикломатическое число графа программы из первой лабораторной работы для дальнейшего применения второго критерия. Для данной программы число вершин в графе равно 14, число дуг – 18, число связных компонент графа 1, тогда:  $Z = Y - N + 2 * P = 18 - 14 + 2 * 1 = 6$ , то есть цикломатическое число равно 6.

9. Для управляющего графа программы из первой лабораторной работы вручную были построены 6 линейно-независимых цикла и линейно – независимых маршрута, после чего была подсчитана структурная сложность по второму критерию.

M1: 1-2-3-4-5-6-7-8-9-13-14 (4 ветвления)

M2: 4-5-4 (1 ветвление)

M3: 7-8-7 (1 ветвления)

M4: 2-3-4-5-6-7-8-9-13-2 (4 ветвления)

M5: 1-2-3-4-5-6-7-8-9-10-11-13-14 (5 ветвления)

M6: 1-2-3-4-5-6-7-8-9-10-12-13-14 (5 ветвления)

Сложность:  $S = 4 + 1 + 1 + 4 + 5 + 5 = 20$

10. Для программы из первой лабораторной работы была подсчитана структурная сложность по двум критериям с помощью программы ways.exe. Результаты расчета структурной сложности по первому критерию приведены на Рисунке 5, по второму критерию на Рисунке 6.

```

Min ways....
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 4 -> 5 -> 6 -> 7 -> 8 -> 7 -> 8 -> 9 -> 10 -> 11 ->
13 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 13 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7
-> 8 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----

Complexity = 16
Press a key...

```

Рисунок 5: Программный расчет структурной сложности программы из первой лабораторной работы по первому критерию

```

Z ways....
----- Path #1 -----
-> 4 -> 5 -> 4
-----Press a key to continue -----
----- Path #2 -----
-> 7 -> 8 -> 7
-----Press a key to continue -----
----- Path #3 -----
-> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 13 -> 2
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 13 -> 14
-----Press a key to continue -----

Complexity = 20
Press a key...

```

Рисунок 6: Программный расчет структурной сложности программы из первой лабораторной работы по второму критерию



### **Выводы.**

В ходе выполнения лабораторной работы были изучены методы оценки структурной сложности программы. После чего проведена оценка структурной сложности двух программ: программы, соответствующей заданному варианту и программы, разработанной в первой лабораторной работе по двум критериям: минимальное покрытие дуг графа и выбор маршрутов на основе цикломатического числа графа. Расчеты были проведены как ручным, так и программным способом.

## ПРИЛОЖЕНИЕ А.

### КОД ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ.

```
PROGRAM ROMB1;

CONST TOL          = 1.0E-4;
VAR  DONE          : BOOLEAN;
      SUM,UPPER,LOWER : REAL;

FUNCTION FX(X: REAL): REAL;

BEGIN
  FX:=1.0/X
END;

PROCEDURE ROMB (
  LOWER,UPPER,TOL: REAL;
  VAR ANS: REAL);
VAR
  NX          : ARRAY[1..16] OF INTEGER;
  T           : ARRAY[1..136] OF REAL;
  DONE,ERROR  : BOOLEAN;
  PIECES,NT,I,II,N,NN,
  L,NTRA,K,M,J : INTEGER ;
  DELTA_X,C,SUM,FOTOM,X : REAL ;
BEGIN
  DONE:=FALSE;
  ERROR:=FALSE;
  PIECES:=1;
  NX[1]:=1;
  DELTA_X:=(UPPER-LOWER)/PIECES;
  C:=(FX(LOWER)+FX(UPPER))*0.5;
  T[1]:=DELTA_X*C;
  N:=1;
  NN:=2;
  SUM:=C;
  REPEAT
    N:=N+1;
    FOTOM:=4.0;
    NX[N]:=NN;
    PIECES:=PIECES*2;
    L:=PIECES-1;
    DELTA_X:=(UPPER-LOWER)/PIECES;
    FOR II:=1 TO (L+1) DIV 2 DO
      BEGIN
        I:=II*2-1;
        X:=LOWER+I*DELTA_X;
        SUM:=SUM+FX(X);
      END;
    T[NN]:=DELTA_X*SUM;
    WRITE(PIECES:5,T[NN]);
    NTRA:=NX[N-1];
    K:=N-1;
    FOR M:=1 TO K DO
      BEGIN
        J:=NN+M;
        NT:=NX[N-1]+M-1;
        T[J]:=(FOTOM*T[J-1]-T[NT])/(FOTOM-1.0);
      END;
    WRITELN(J:4,T[J]);
    IF N>4 THEN
```

```

BEGIN
  IF T[NN+1]<>0.0 THEN
    IF (ABS(T[NTRA+1]-T[NN+1])<=ABS(T[NN+1]*TOL))
      OR (ABS(T[NN-1]-T[J])<=ABS(T[J]*TOL)) THEN
      DONE:=TRUE
    ELSE IF N>15 THEN
      BEGIN
        DONE:=TRUE;
        ERROR:=TRUE
      END
    END;
    NN:=J+1;
    UNTIL DONE;
    ANS:=T[J]
  END;

BEGIN
  LOWER:=1.0;
  UPPER:=9.0;
  WRITELN;
  ROMB(LOWER,UPPER,TOL,SUM);
  WRITELN;
  WRITELN(CHR(7),'AREA= ',SUM)
END.

```