МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №2

по дисциплине “Тестирование Программного Обеспечения”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Строганов В. А.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №2**

**«Исследование способов структурного тестирования программного обеспечения»**

**Цель работы**

Исследовать основные подходы к структурному тестированию программного обеспечения. Приобрести практические навыки построения графа потоков управления и определения независимых ветвей программы.

**Вариант**

1) Написать программу, выполняющую заданные действия.

2) Построить граф потоков управления.

3) Вычислить цикломатическое число для построенного графа потоков управления.

4) Определить независимые ветви программы.

**Задачи**

Задача 1.

Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках,

которые не содержат отрицательных элементов.

Задача 2.

Дана строка. Преобразовать строку, заменив в ней каждую точку многоточием.

Задача 3.

Программа, которая считывает английский текст из файла и выводит на экран слова,

начинающиеся с гласных букв.

**Ход работы**

**Задача 1**

Для начала составим программу, выполняющую заданные действия. (приложение А).

Часть кода, выполняющего вычисление:

// Вычисление

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

ant[i] \*= M[i][j];

if (M[i][j] < 0) {

ant[i] = -1;

break;

}

Затем построим граф потоков управления. (рисунок 1).

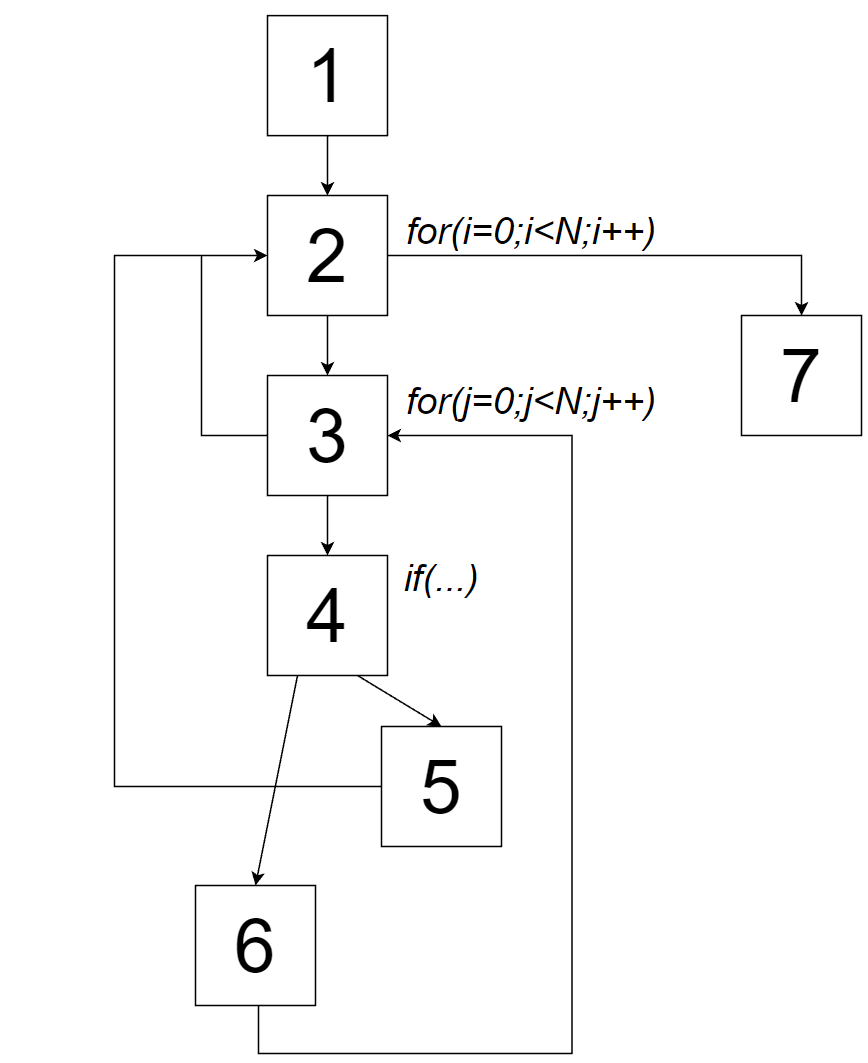


Рисунок 1 —­ Граф потоков управления.­

После этого вычислим для него цикломатическое число.

Следующим шагом определим независимые ветви графа потоков управления.

Независимые ветви:

1) 1, 2, 7.

2) 1, 2, 3, 2, 7.

3) 1, 2, 3, 4, 5, 2, 7.

4) 1, 2, 3, 4, 6, 3, 2, 7.

**Задача 2**

Для начала составим программу, выполняющую заданные действия. (приложение Б).

Часть кода, решающая задачу:

// Решение

for (int i = 0; i < s.size(); i++)

{

if (s[i] == '.') {

s.insert(i++, ".");

s.insert(i++, ".");

}

}

Затем построим граф потоков управления. (рисунок 2).

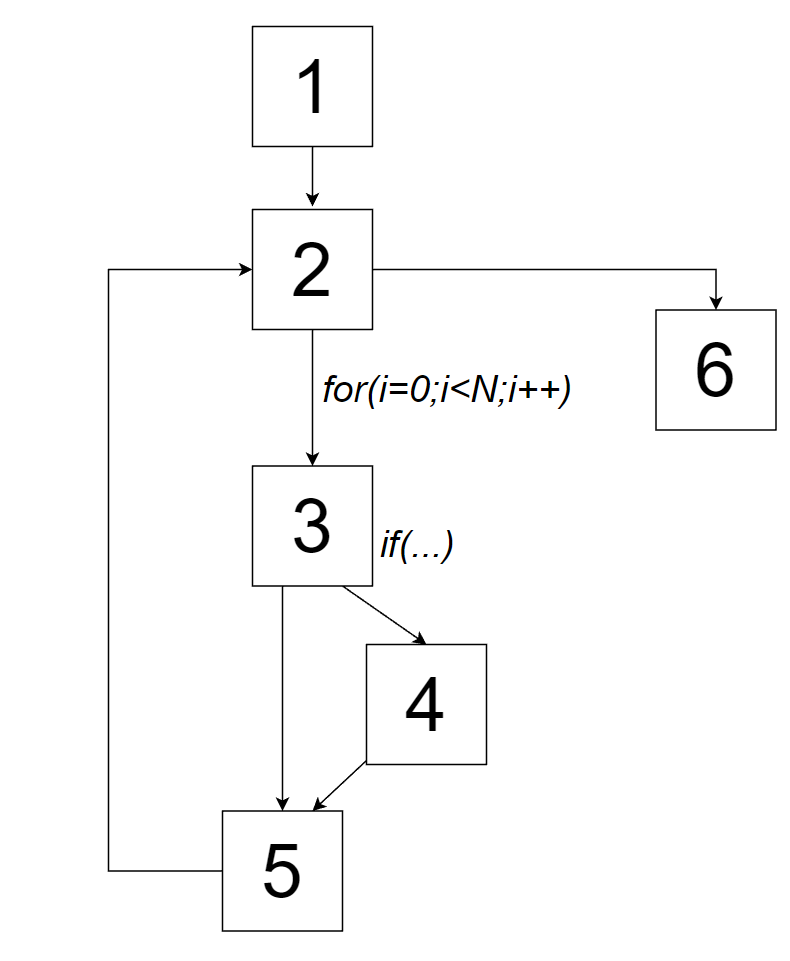


Рисунок 2 —­ Граф потоков управления.­

После этого вычислим для него цикломатическое число.

Следующим шагом определим независимые ветви графа потоков управления.

Независимые ветви:

1) 1, 2, 6.

2) 1, 2, 3, 5, 6.

3) 1, 2, 3, 4, 5, 6.

**Задача 3**

Для начала составим программу, выполняющую заданные действия. (приложение В).

Часть кода, решающая задачу:

// Решение

ifstream in("text.txt");

while (!in.eof()) {

in >> slovo;

if (slovo[0] == 'A'

|| …

}

Затем построим граф потоков управления. (рисунок 3).

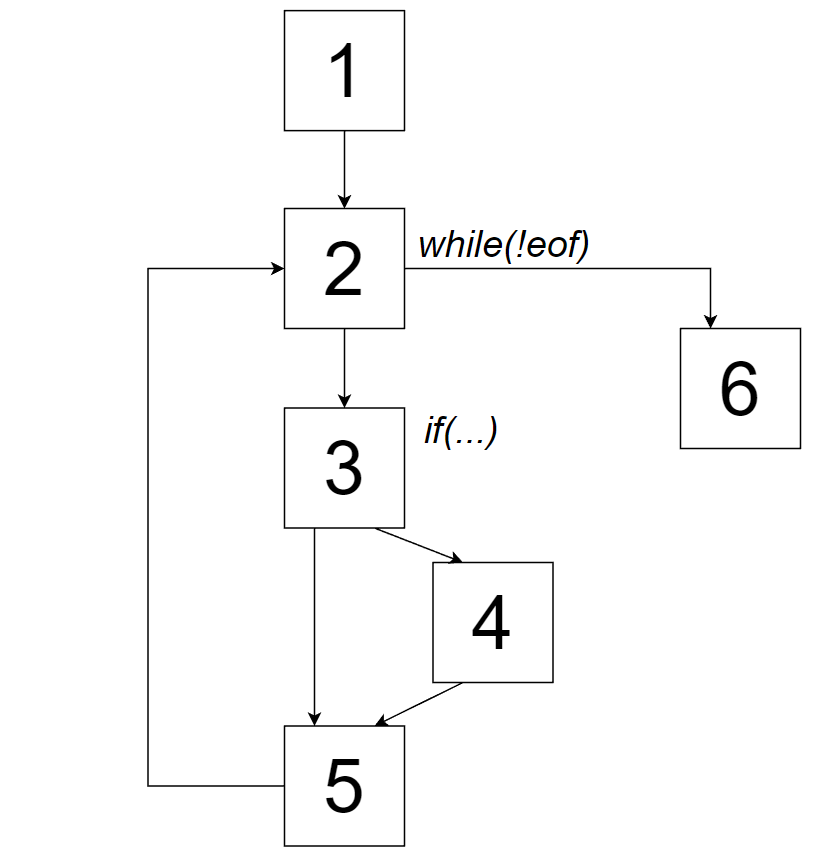


Рисунок 3 —­ Граф потоков управления.­

После этого вычислим для него цикломатическое число.

Следующим шагом определим независимые ветви графа потоков управления.

Независимые ветви:

1) 1, 2, 6.

2) 1, 2, 3, 4, 5, 6.

3) 1, 2, 3, 5, 6.

**Выводы**

В ходе лабораторной работы были исследованы основные подходы к структурному тестированию программного обеспечения, в ходе которого были получены знания о существовании метода "белого ящика", о его применимости для небольших программ и удобства наглядности. Также были приобретены практические навыки построения графа потоков управления и определения независимых ветвей программы.

**ПРИЛОЖЕНИЕ А**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Введите размерность квадратной матрицы N\*N" << endl;

int N;

cin >> N;

int\* ant = new int[N];

for (int i = 0; i < N; i++)

ant[i] = 1;

int\*\* M = new int\* [N];

for (int i = 0; i < N; i++)

M[i] = new int[N];

//Ввод значений в матрицу

cout << "Введите значения " << N \* N << " элементов матрицы :\n";

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++) {

cin >> M[i][j];

}

// Вычисление

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

ant[i] \*= M[i][j];

if (M[i][j] < 0) {

ant[i] = -1;

break;

}

//Вывод результата на экран

for (int i = 0; i < N; i++) {

if (ant[i] >= 0)

cout << "Произведение элементов в " << i << " строкe = " << ant[i] << endl;

else cout << "Строка " << i << " содержит отрицательные элементы\n";

}

}

**ПРИЛОЖЕНИЕ Б**

#include<iostream>

#include<Windows.h>

#include<string>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

string s;

cout << "Введите исходное предложение" << endl;

getline(cin, s);

for (int i = 0; i < s.size(); i++)

{

if (s[i] == '.') {

s.insert(i++, ".");

s.insert(i++, ".");

}

}

cout << "Полученное предложение\n";

cout << s << endl;

return 0;

}

**ПРИЛОЖЕНИЕ В**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

char slovo[20];

ifstream in("text.txt");

cout << "Слова, начинающиеся с гласных английских букв: \n";

while (!in.eof()) {

in >> slovo;

if (slovo[0] == 'A'

|| slovo[0] == 'a'

|| slovo[0] == 'E'

|| slovo[0] == 'e'

|| slovo[0] == 'Y'

|| slovo[0] == 'y'

|| slovo[0] == 'U'

|| slovo[0] == 'u'

|| slovo[0] == 'I'

|| slovo[0] == 'i'

|| slovo[0] == 'O'

|| slovo[0] == 'o')

cout << slovo << endl;

}

system("pause");

return 0;

}