

Лабораторная работа №2

Исследование способов анализа областей эквивалентности и построения тестовых последовательностей

1. Цель работы

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

2. Постановка задачи

Вариант — 22

Согласно варианту задания необходимо:

- Построить граф потоков управления.
- Вычислить цикломатическое число для построенного графа потоков управления.
- Определить независимые ветви программы.

3. Ход работы

3.1 Для построения графа потоков управления возьмём фрагмент программы из лабораторной работы №1, который обрабатывает строку:

```
void work_with_string() {
    getchar();
    std::string s;
    std::cout << "Введите строку:" << std::endl;
    std::cin >> s;
    //преобразование строки в массив символов для работы с символами
    char *str = const_cast<char*>(s.c_str());

    //обработка строки
    bool flag = false;
    for(int i = 0; i < s.length() + 1; i++) {
        if (!flag) {
            if(str[i] == '#') flag = true;
        }
        else {
            str[i] = '@';
        }
    }
}
```

```

    }
}
std::cout << "Обработанная строка:" << std::endl << str << std::endl;
system("pause");
}

```

Данный фрагмент кода представляет функцию `work_with_string()`, которая:

- Просит пользователя ввести строку.
- Преобразует введенную строку в массив символов с помощью `const_cast<char*>(s.c_str())`.
- Обработывает полученную строку: находит первый символ '#' и заменяет все символы после него на символ '@'.
- Выводит обработанную строку на экран.

По этому фрагменту кода был составлен граф потоков управления (рисунок 1).

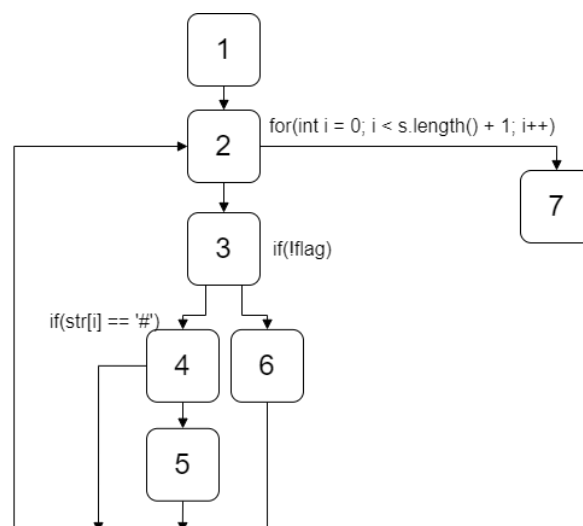


Рисунок 1 – Граф потоков управления подпрограммы обработки строки

3.2 Было рассчитано цикломатическое число: $C(G) = \text{количество дуг} - \text{количество узлов} + 2 = 8 - 7 + 2 = 3$.

3.3 Независимыми ветвями называются последовательности дуг вершин графа, ведущие из начальной вершину к конечной и отличающиеся по крайней мере одним узлом. Для полученного графа потоков управления получаем следующие независимые ветви:

- 1) 1, 2, 3, 4, 5, 6, 7
- 2) 1, 2, 3, 4, 5, 7
- 3) 1, 2, 3, 7
- 4) 1, 2, 3, 4, 7

5) 1, 2, 7

Выводы:

В ходе выполнения данной лабораторной работы были исследованы способы тестирования программного обеспечения, а именно тестирование ветвей. Были получены практические навыки нахождения цикломатического числа, позволяющего вычислить количество независимых ветвей, построения графа потоков управления, позволяющего найти эти независимые ветви. Были изучены основные подходы к структурному тестированию программного обеспечения. Были получены практические навыки построения графа потоков управления программы и определения независимых ветвей. Было установлено, что количество независимых ветвей в программе можно определить, вычислив цикломатическое число графа потоков управления программы.