

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г. И. Носова»
(ФГБОУ ВО «МГТУ им. Г.И. Носова»)

Кафедра вычислительной техники и программирования

Лабораторная работа №2

по дисциплине «Метрология и стандартизация программного обеспечения»
название лабораторной работы: «Оценка структурной сложности программы»

Исполнитель: Варламов М.Н. студент 3 курса, группа АВб-19-1

Руководитель: Сибилева Н.С., ст. преподаватель каф. ВТиП

Магнитогорск, 2022

СОДЕРЖАНИЕ

Введение.....	3
1 Постановка задания.....	4
2 Выполнения задания.....	4
Заключение.....	6

ВВЕДЕНИЕ

Необходимо для натурального числа, заданного с клавиатуры получить новое число, дописав к нему цифру s (также введенную с клавиатуры) в начале и в конце. Разработанный алгоритм решения задачи, оформить согласно ГОСТ 19.701-90. Построить граф потока управления (с выделенными ветвлениями). Предусмотреть возможные маршруты тестирования в соответствии с первым и вторым критериями. Рассчитать матрицы смежности и достижимости. Рассчитать значения цикломатического числа.

1 ПОСТАНОВКА ЗАДАНИЯ

Для натурального числа, заданного с клавиатуры получить новое число, дописав к нему цифру *s* (также введенную с клавиатуры) в начале и в конце.

2 ВЫПОЛНЕНИЯ ЗАДАНИЯ

Исходный код решения представлен на рисунке 1.

```
7  ▶ public static void main(String[] args) {
8      Scanner scanner = new Scanner(System.in);
9      String aLine;
10     boolean flag = false;
11     do {
12         System.err.print("Введите натуральное число: ");
13         aLine = scanner.next();
14         try {
15             Integer.valueOf(aLine);
16             flag = true;
17         } catch (NumberFormatException e) {
18             System.err.println("Введено не натуральное число");
19         }
20     } while (!flag);
21     String bLine = "";
22     flag = false;
23     do {
24         System.err.print("Введите цифру: ");
25         bLine = scanner.next();
26         if (bLine.length() == 1 && Character.isDigit(bLine.charAt(0)))
27             flag = true;
28         else
29             System.err.println("Введена не цифра");
30     } while (!flag);
31     System.err.println("Новое число = " + bLine + aLine + bLine);
32 }
```

Рисунок 1 – Исходный код решения

На рисунке 2 представлен разработанный алгоритм решения задачи, оформленный согласно ГОСТ 19.701-90.

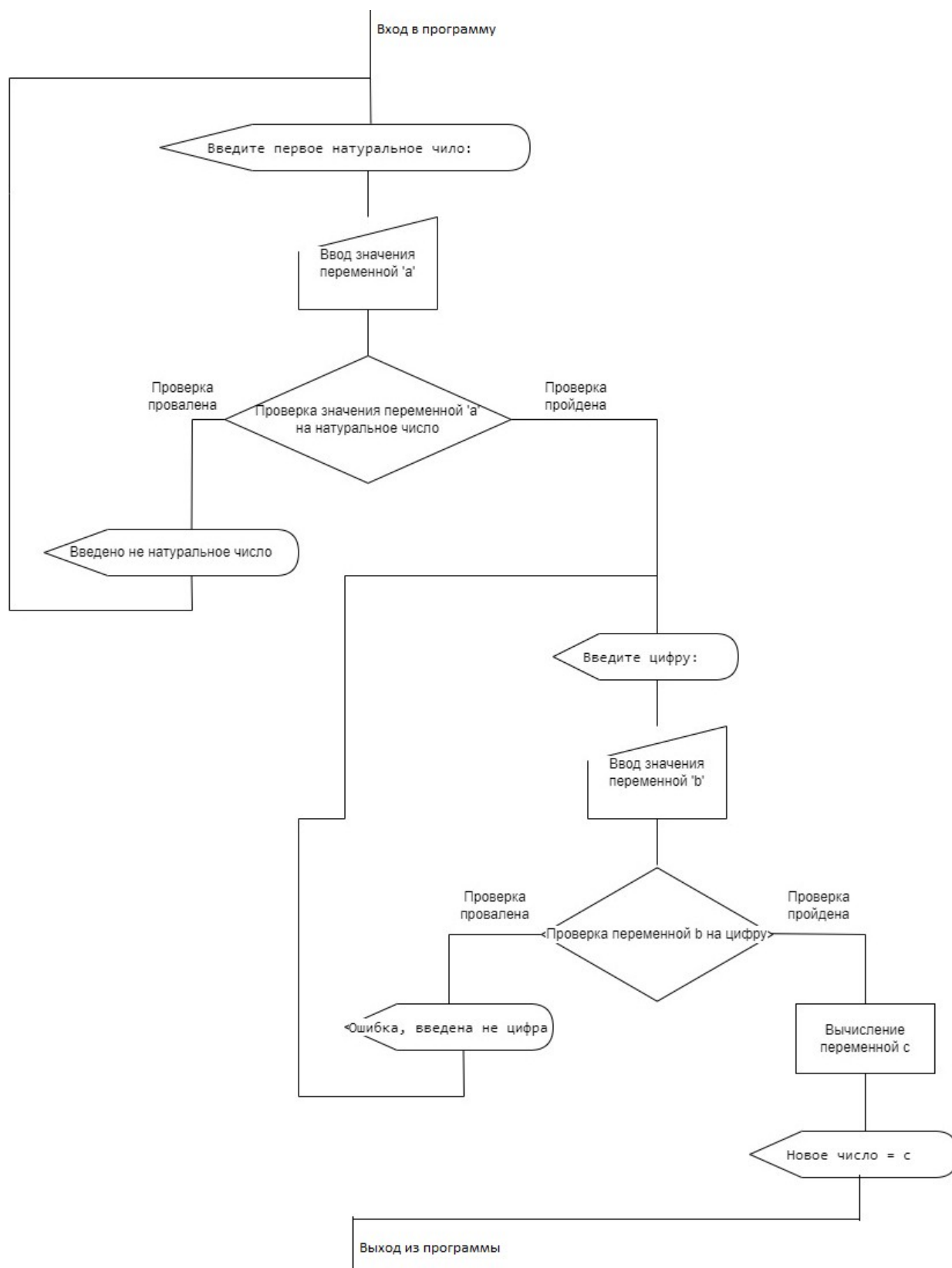


Рисунок 2 – Алгоритм решения

Поделим код на потоки управления (рисунок 3).

```
7 public static void main(String[] args) {
8     Scanner scanner = new Scanner(System.in);
9     String aLine;
10    A     boolean flag = false;
11    do {
12        B     System.err.print("Введите натуральное число: ");
13        aLine = scanner.next();
14        try {
15            Integer.valueOf(aLine);
16            flag = true;
17        } catch (NumberFormatException e) {
18            C     System.err.println("Введено не натуральное число");
19        }
20    D     } while (!flag);
21    String bLine = "";
22    E     flag = false;
23    do {
24        F     System.err.print("Введите цифру: ");
25        bLine = scanner.next();
26        G     if (bLine.length() == 1 && Character.isDigit(bLine.charAt(0)))
27        H         flag = true;
28        I     else
29            System.err.println("Введена не цифра");
30    J     } while (!flag);
31    K     System.err.println("Новое число = " + bLine + aLine + bLine);
32 }
```

Рисунок 3 – Потоки управления программы

Опишем потоки управления:

- A. Инициализация переменных
- B. Ввод числа, проверка введенного значения
- C. Обработка ошибки проверки
- D. Цикл, управляющий проверкой
- E. Инициализация переменных
- F. Ввод цифры
- G. Проверка введенного значения
- H. Установка значения для выхода из цикла
- I. Обработка не пройденной проверки
- J. Цикл, управляющий проверкой
- K. Расчёт и вывод результата

На рисунке 4 представлен граф потока управления.

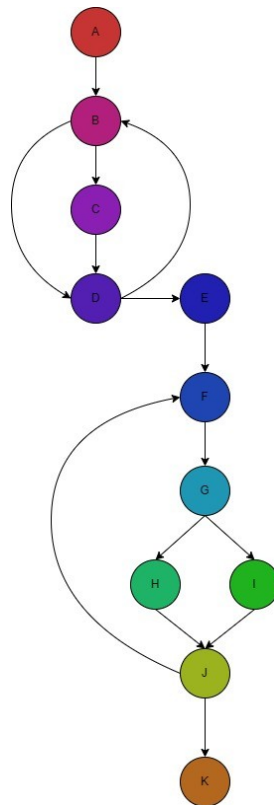


Рисунок 4 – Граф потока исполнения

С помощью графа можно рассчитать возможные маршруты тестирования. Так, например, при вводе значений “22” и “1”, маршрут будет A,B,D,E,F,G,H,J,K. Данный путь означает полностью корректные введенные данные пользователем. Если пользователь будет ошибаться при вводе первого числа, маршрут будет пополнен вершиной C в последовательности: A,B,C,D,E,F,G,H,J,K. Если пользователь будет ошибаться при вводе второй цифры, маршрут будет пополнен вершиной I в последовательности: A,B,D,E,F,G,I,J,K.

Из полученного графа построим матрицу смежности и достижимости (рисунок 5).

Матрица смежности												Матрица достижимости											
	A	B	C	D	E	F	G	H	I	J	K		A	B	C	D	E	F	G	H	I	J	K
A	0	1	0	0	0	0	0	0	0	0	0	1	0	12	9	16	12	19	15	9	9	12	10
B	0	0	1	1	0	0	0	0	0	0	0	2	0	16	12	21	16	24	19	15	15	18	12
C	0	0	0	1	0	0	0	0	0	0	0	3	0	9	7	12	9	15	9	10	10	10	8
D	0	1	0	0	1	0	0	0	0	0	0	4	0	12	9	16	12	19	15	9	9	20	10
E	0	0	0	0	0	1	0	0	0	0	0	5	0	0	0	0	0	8	0	0	0	0	8
F	0	0	0	0	0	0	1	0	0	0	0	6	0	0	0	0	0	0	8	0	0	0	0
G	0	0	0	0	0	0	0	1	1	0	0	7	0	0	0	0	0	0	0	8	8	0	0
H	0	0	0	0	0	0	0	0	0	1	0	8	0	0	0	0	0	0	0	0	0	8	0
I	0	0	0	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	8	0
J	0	0	0	0	0	1	0	0	0	0	1	10	0	0	0	0	0	8	0	0	0	0	8
K	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0

Рисунок 5 – Матрицы смежности и достижимости

По полученным данным рассчитаем значение цикломатического числа

$$Z = n - m + 1 - p = 14 - 11 + 2 - 1 = 4, \text{ где}$$

n – Количество рёбер графа,

m - Количество вершин графа,

p – Число связных компонентов графа

ЗАКЛЮЧЕНИЕ

В результате был разработан алгоритм решения задачи, и оформлен согласно ГОСТ 19.701-90, построен граф потока управления, который отображает множество всех возможных путей исполнения программы, представленное в виде графа. Рассчитаны возможные маршруты тестирования, описывающие все возможные пути исполнения программы в виде последовательности вершин, а также матрицы смежности, достижимости и цикломатическое число, благодаря которым мы можем определить структурную сложность программы .