



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт искусственного интеллекта
Кафедра общей информатики

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 12
Элементы алгоритмизации и процедурного программирования
по дисциплине
«ИНФОРМАТИКА»

Выполнил студент группы ИМБО-01-22

Ким К.С.

Принял
Ассистент

Павлова Е.С.

Практическая работа выполнена

«_» декабря 2022 г.

Подпись студента

«Зачтено»

«_» декабря 2022 г.

Подпись преподавателя

Москва 2022

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
1.1 Персональный вариант.....	3
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	4
2.2 Код программы.....	13
2.3 Примеры тестирования	16
3 ВЫВОДЫ.....	18
4 ИНФОРМАЦИОННЫЙ ИСТОЧНИК	19

1 ПОСТАНОВКА ЗАДАЧИ

Требуется разработать блок-схему алгоритма и написать программу обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом. При этом требуется контролировать типы и диапазоны вводимых данных, а также предусмотреть обработку других исключительных ситуаций (если они есть), например, ситуацию деления на ноль. Блок-схема должна быть полной, то есть должна описывать и процесс диалога с пользователем, и контроль вводимых данных, и подпрограммы вычислений с обработкой возможных исключительных операций. Блок-схема должна изображаться по ГОСТу. При обнаружении ошибки ввода или ошибки вычислений программа должна информативно уведомлять пользователя о причине ошибки. Если ошибка произошла на этапе ввода данных, то программа должна просить пользователя повторить ввод.

1.1 Персональный вариант

2.8. Создать квадратную матрицу размера $M \times M$, где M является целым числом из диапазона $[2, 5]$. Конкретный размер матрицы задается пользователем. Матрица содержит только целые числа из диапазона $[1, 100]$, которые могут быть как случайными, так и вводиться пользователем. Отсортировать по возрастанию элементы, принадлежащие или лежащие выше побочной диагонали матрицы, остальные элементы умножить на минус один. Результаты обработки матрицы вывести на экран.

2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1 Блок-схемы алгоритмов программы

Блок-схема алгоритма, решающего поставленную задачу показана на рис. 1–9.

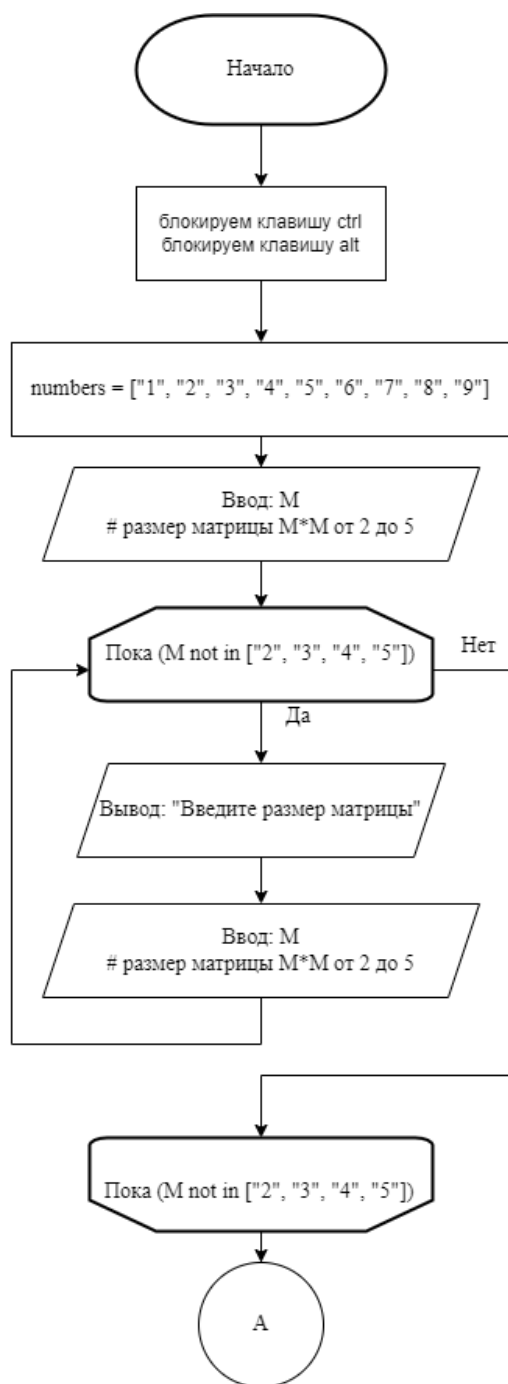


Рисунок 1 – Блок-схема программы, часть 1

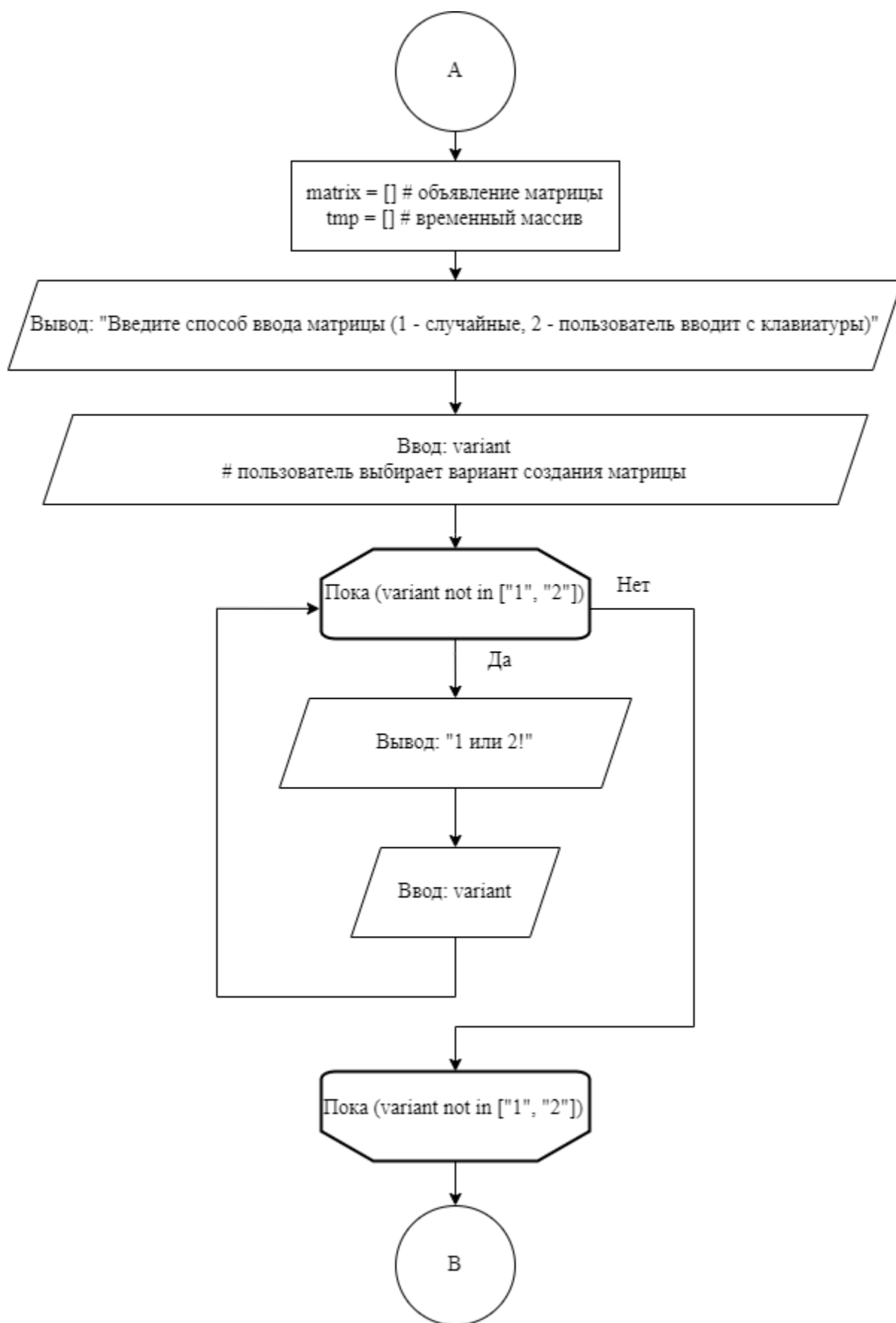


Рисунок 2 – Блок-схема программы, часть 2

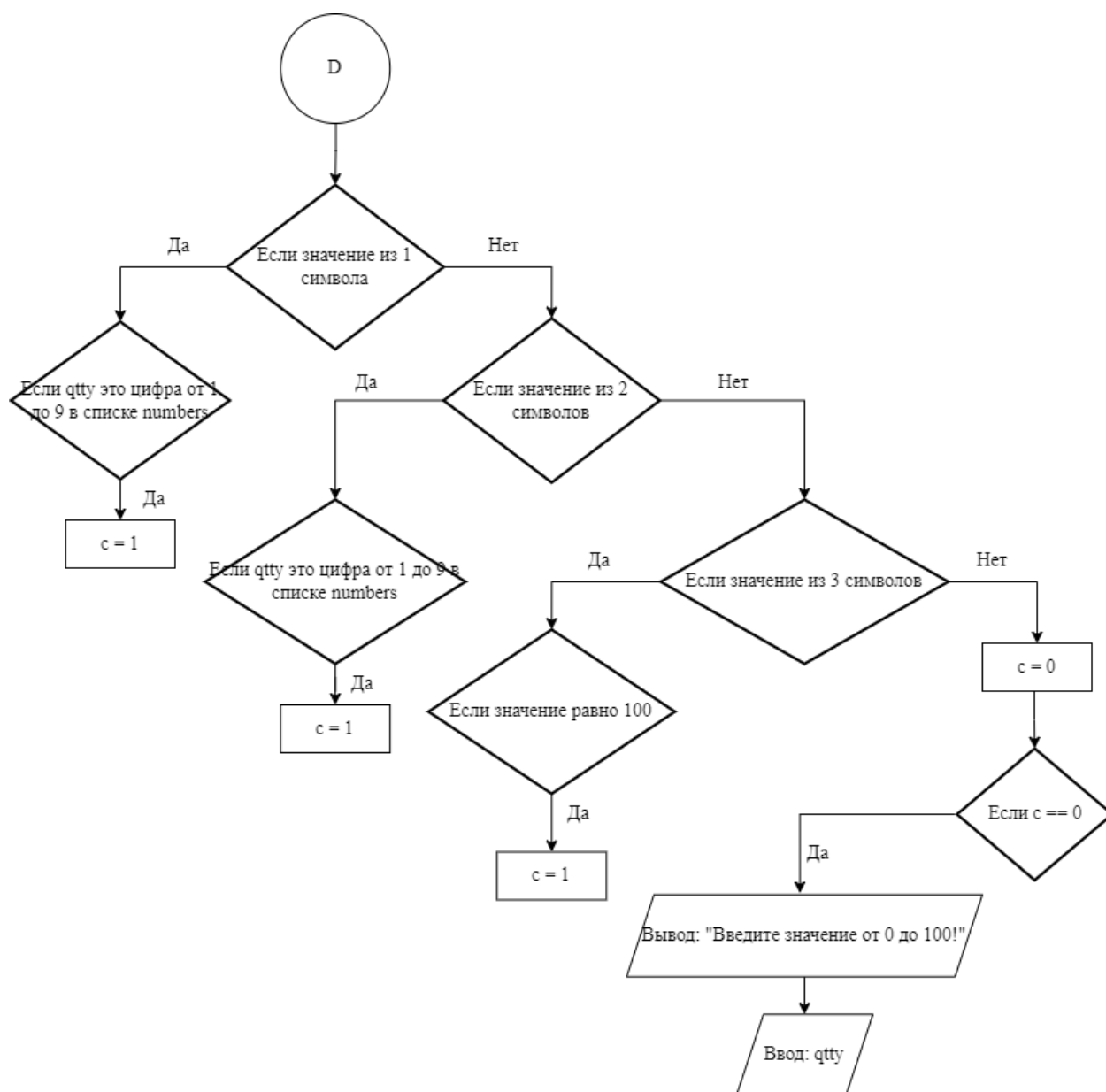


Рисунок 4 – Блок-схема программы, часть 4

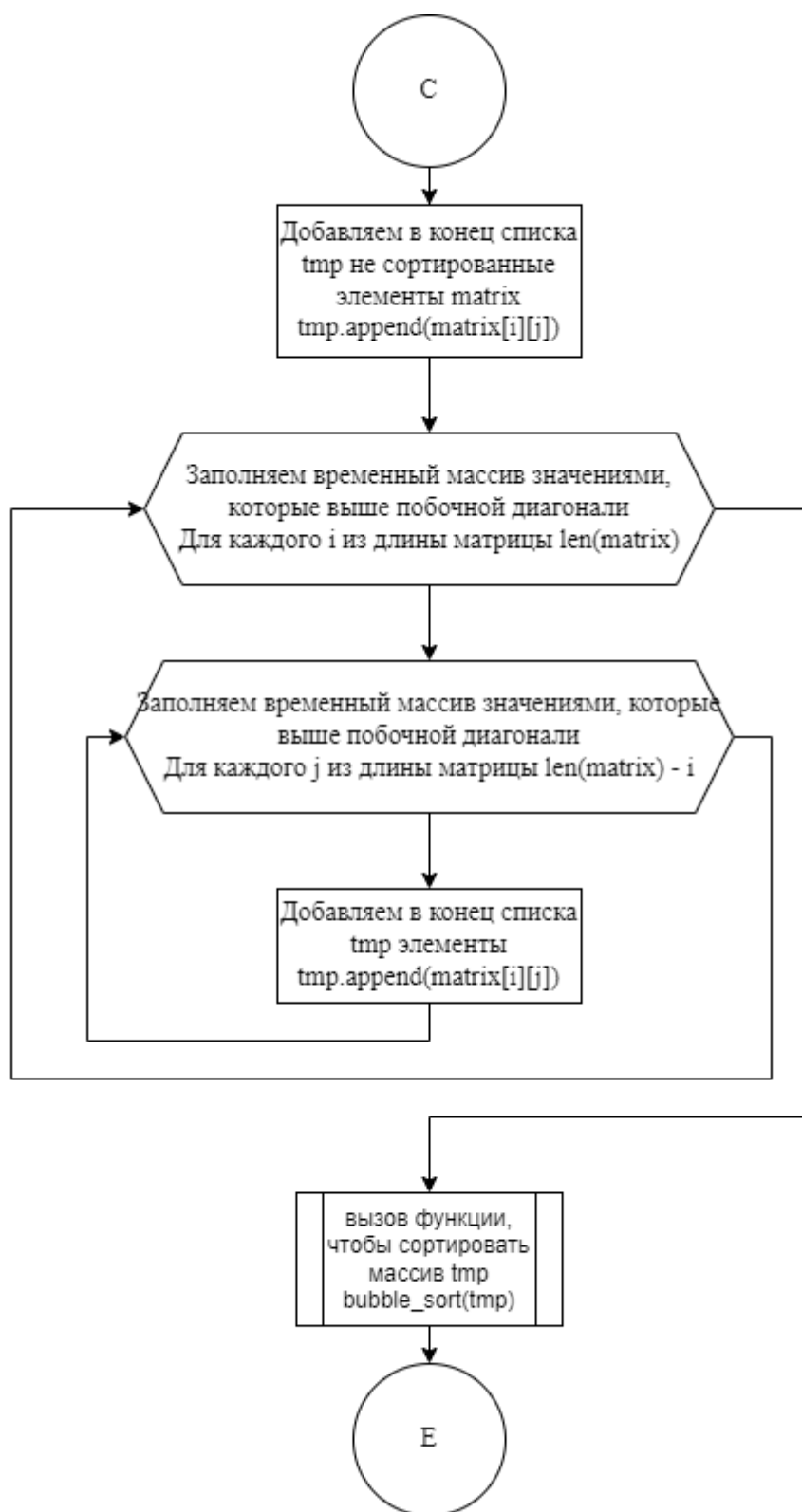


Рисунок 5 – Блок-схема процедуры, часть 5

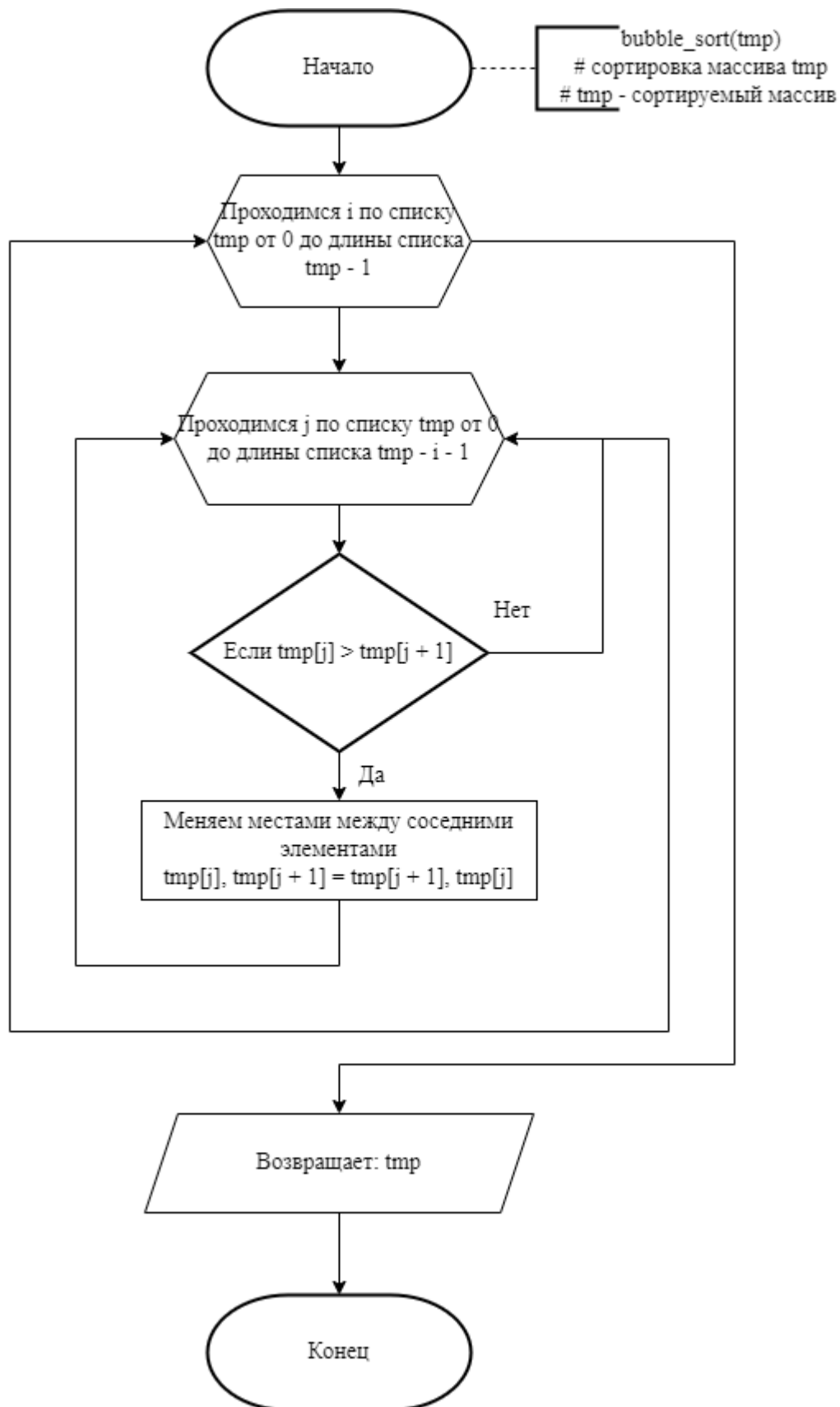


Рисунок 6 – Блок-схема процедуры bubble_sort

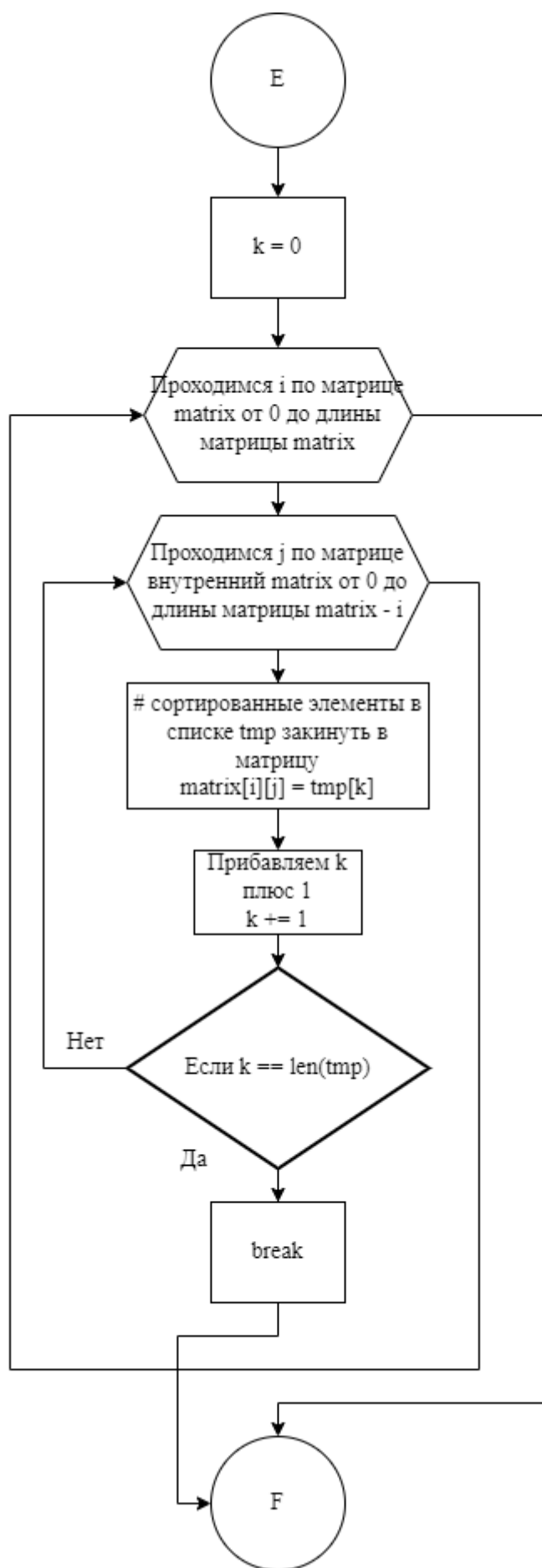


Рисунок 7 – Блок-схема программы, часть 7

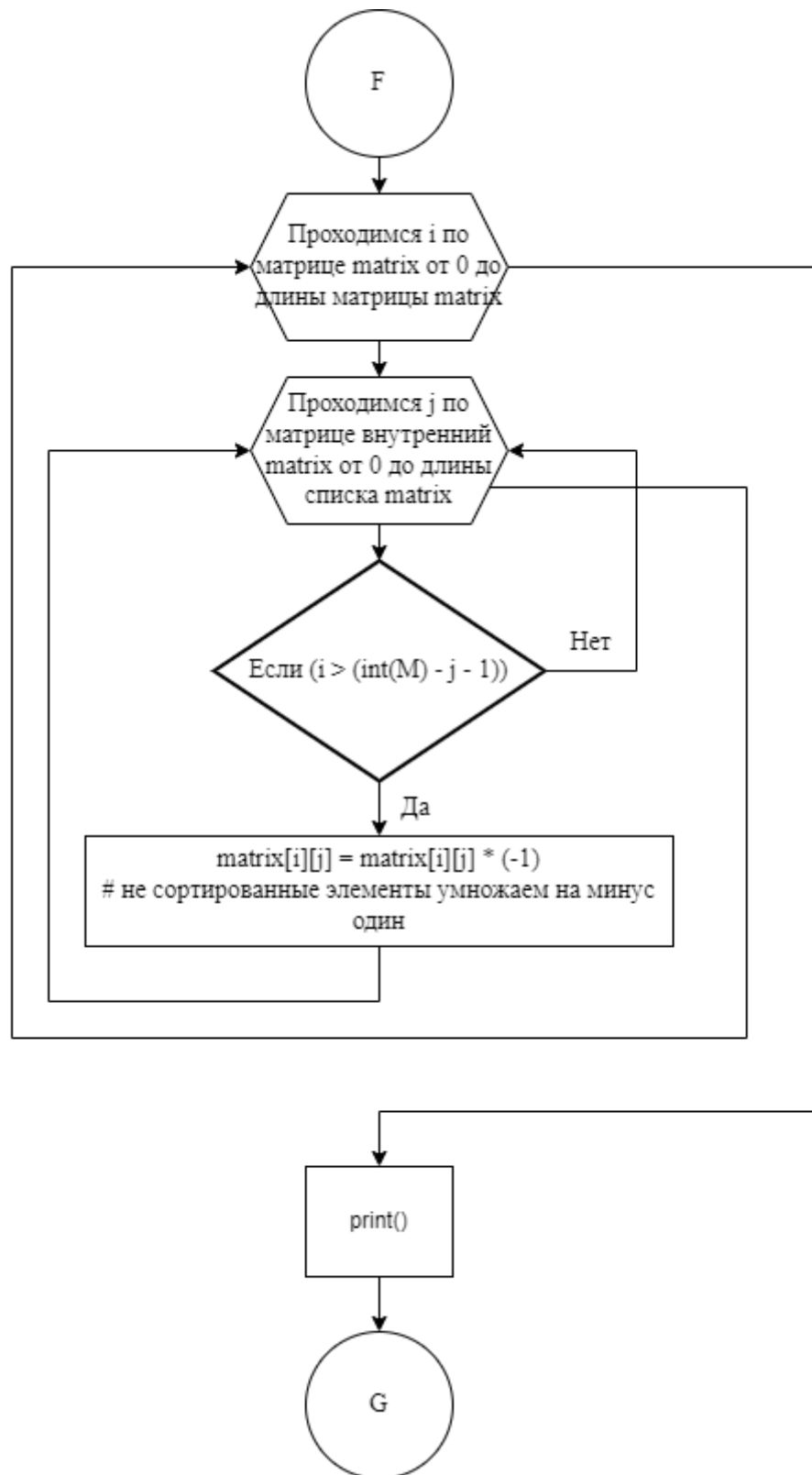


Рисунок 8 – Блок-схема программы, часть 8

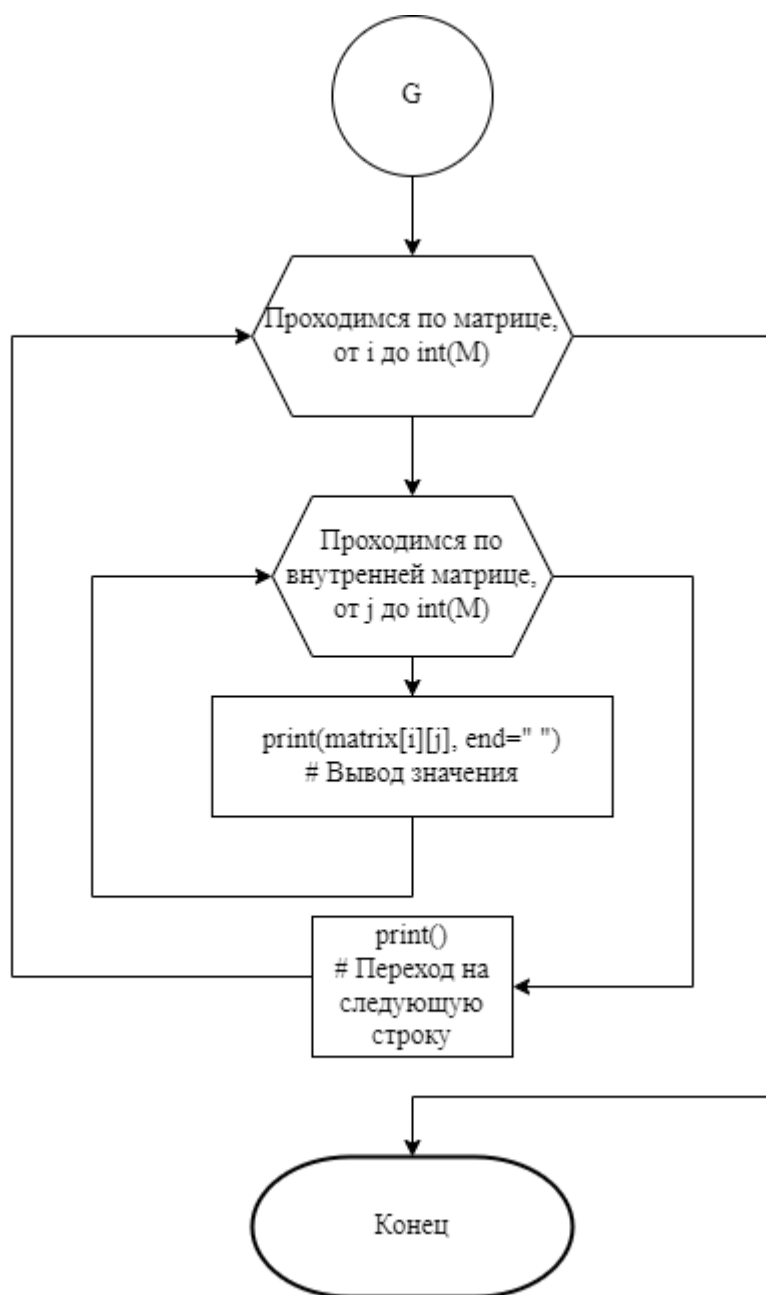


Рисунок 9 – Блок-схема программы, часть 9

На основе блок-схемы был реализован код на языке Python 3 в среде разработки PyCharm.

2.2 Код программы

Код программы на языке программирования Python представлен листингом 1.

Листинг 1 – Код программы

```
1 import random # подключение библиотеки для реализации случайных чисел
2 import keyboard # подключение модуля для блокировки клавиш
3
4
5 def bubble_sort(tmp):
6     for i in range(len(tmp) - 1):
7         for j in range(len(tmp) - i - 1):
8             if tmp[j] > tmp[j + 1]:
9                 tmp[j], tmp[j + 1] = tmp[j + 1], tmp[j]
10    return tmp
11
12
13 keyboard.block_key("ctrl")
14 keyboard.block_key("alt")
15 numbers = ["1", "2", "3", "4", "5", "6", "7", "8", "9"]
16 print("Какой диапазон для размера матрицы?")
17 M = input() # размер матрицы M*N от 2 до 5
18 while (M not in ["2", "3", "4", "5"]):
19     print("Размер матрицы должен быть от 2 до 5!")
20     M = input()
21 matrix = [] # объявление матрицы
22 tmp = [] # временный массив
23 print("Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)")
24 variant = input() # пользователь выбирает вариант создания матрицы
25 while (variant not in ["1", "2"]):
26     print("1 или 2!")
27     variant = input()
28 if (int(variant) == 1):
29     for i in range(int(M)):
30         matrix.append([])
31         for j in range(int(M)):
32             matrix[i].append(random.randint(1, 100))
33
34     print()
35     print("Вывод изначальной матрицы")
36     for i in range(int(M)): # вывод изначальной матрицы
37         for j in range(int(M)):
38             print(matrix[i][j], end=" ")
39         print()
40
41     for i in range(len(matrix)): # закидываю элементы выше побочной диагонали
42         for j in range(len(matrix) - i):
43             tmp.append(matrix[i][j])
44
45     bubble_sort(tmp) # сортируем элементы
46
47     k = 0
48     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
49         for j in range(len(matrix) - i):
50             matrix[i][j] = tmp[k]
51             k += 1
52         if (k == len(tmp)):
53             break
```

```

47     k = 0
48     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
49         for j in range(len(matrix) - i):
50             matrix[i][j] = tmp[k]
51             k += 1
52             if (k == len(tmp)):
53                 break
54
55     else:
56         for i in range(int(M)): # наполнение матрицы пользовательскими числами
57             p = []
58             for j in range(int(M)):
59                 print(f"Введите значения [{i};{j}]")
60                 qty = input()
61                 c = 0
62                 while (c == 0):
63                     if (len(qty) == 1):
64                         if (qty[0] in numbers):
65                             c = 1
66                     elif (len(qty) == 2):
67                         if (qty[0] in numbers and qty[1] in numbers):
68                             c = 1
69                     elif (len(qty) == 3):
70                         if (int(qty) == 100):
71                             c = 1
72                     else:
73                         c = 0
74                     if (c == 0):
75                         print("Введите значение от 0 до 100!")
76                         qty = input()
77                 p.append(int(qty))
78             matrix.append(p)
79
80     print()
81     print("Вывод изначальной матрицы")
82     for i in range(int(M)): # вывод изначальной матрицы
83         for j in range(int(M)):
84             print(matrix[i][j], end=" ")
85         print()
86
87     for i in range(len(matrix)): # закидываю элементы выше побочной диагонали
88         for j in range(len(matrix) - i):
89             tmp.append(matrix[i][j])
90
91     bubble_sort(tmp) # сортируем элементы
92
93     k = 0
94     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
95         for j in range(len(matrix) - i):
96             matrix[i][j] = tmp[k]
97             k += 1
98             if (k == len(tmp)):
99                 break

```

```

65         c = 1
66     elif (len(qtty) == 2):
67         if (qty[0] in numbers and qty[1] in numbers):
68             c = 1
69     elif (len(qtty) == 3):
70         if (int(qtty) == 100):
71             c = 1
72     else:
73         c = 0
74     if (c == 0):
75         print("Введите значение от 0 до 100!")
76         qty = input()
77     p.append(int(qtty))
78     matrix.append(p)
79
80     print()
81     print("Вывод изначальной матрицы")
82     for i in range(int(M)): # вывод изначальной матрицы
83         for j in range(int(M)):
84             print(matrix[i][j], end=" ")
85         print()
86
87     for i in range(len(matrix)): # закидываю элементы выше побочной диагонали
88         for j in range(len(matrix) - i):
89             tmp.append(matrix[i][j])
90
91     bubble_sort(tmp) # сортируем элементы
92
93     k = 0
94     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
95         for j in range(len(matrix) - i):
96             matrix[i][j] = tmp[k]
97             k += 1
98             if (k == len(tmp)):
99                 break
100
101     for i in range(int(M)): # оставшиеся не сортированные элементы умножаю на минус один
102         for j in range(int(M)):
103             if (i > (int(M) - j - 1)):
104                 matrix[i][j] = matrix[i][j] * (-1)
105
106     print()
107     print("Вывод конечной матрицы")
108     for i in range(int(M)): # вывод конечной матрицы
109         for j in range(int(M)):
110             print(matrix[i][j], end=" ")
111         print()
112

```

2.3 Примеры тестирования

Было проведено тестирование программы, которое показало, что работает корректно. Результаты тестирования представлены листингами 2–4.

Листинг 2 – Тестирование со случайными числами

```
Какой диапазон для размера матрицы?  
4  
Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)  
1  
  
Вывод изначальной матрицы  
11 25 54 40  
95 19 19 8  
62 89 3 57  
35 51 42 60  
  
Вывод конечной матрицы  
11 19 19 25  
35 40 54 -8  
62 89 -3 -57  
95 -51 -42 -60  
  
Process finished with exit code 0
```


Листинг 3 – Тестирование с числами от клавиатуры

```
Какой диапазон для размера матрицы?
4
Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)
2
Введите значения [0;0]
2
Введите значения [0;1]
123
Введите значение от 0 до 100!
21
Введите значения [0;2]
74
Введите значения [0;3]
65
Введите значения [1;0]
53
Введите значения [1;1]
2
Введите значения [1;2]
43
Введите значения [1;3]
6
Введите значения [2;0]
34
Введите значения [2;1]
35
Введите значения [2;2]
23
Введите значения [2;3]
6
Введите значения [3;0]
88
Введите значения [3;1]
9
Введите значения [3;2]
99
Введите значения [3;3]
75

Вывод изначальной матрицы
2 21 74 65
53 2 43 6
34 35 23 6
88 9 99 75

Вывод конечной матрицы
2 2 21 34
35 43 53 -6
65 74 -23 -6
88 -9 -99 -75
```

3 ВЫВОДЫ

Разработана блок-схема алгоритма и написана программа обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом. При этом требуется проконтролированы типы и диапазоны вводимых данных, а также предусмотрена обработка других исключительных ситуаций (если они есть), например, ситуацию выхода за границу диапазона. Блок-схема изображена по ГОСТу. При обнаружении ошибки ввода или ошибки вычислений программа уведомляет пользователя о причине ошибки. Если ошибка произошла на этапе ввода данных, то программа просит пользователя повторить ввод.

4 ИНФОРМАЦИОННЫЙ ИСТОЧНИК

Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов — М., МИРЭА — Российский технологический университет, 2020. — 102 с.