



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 .....      | 15 |
| 3 ВЫВОДЫ.....                       | 17 |
| 4 ИНФОРМАЦИОННЫЙ ИСТОЧНИК .....     | 18 |

## **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.

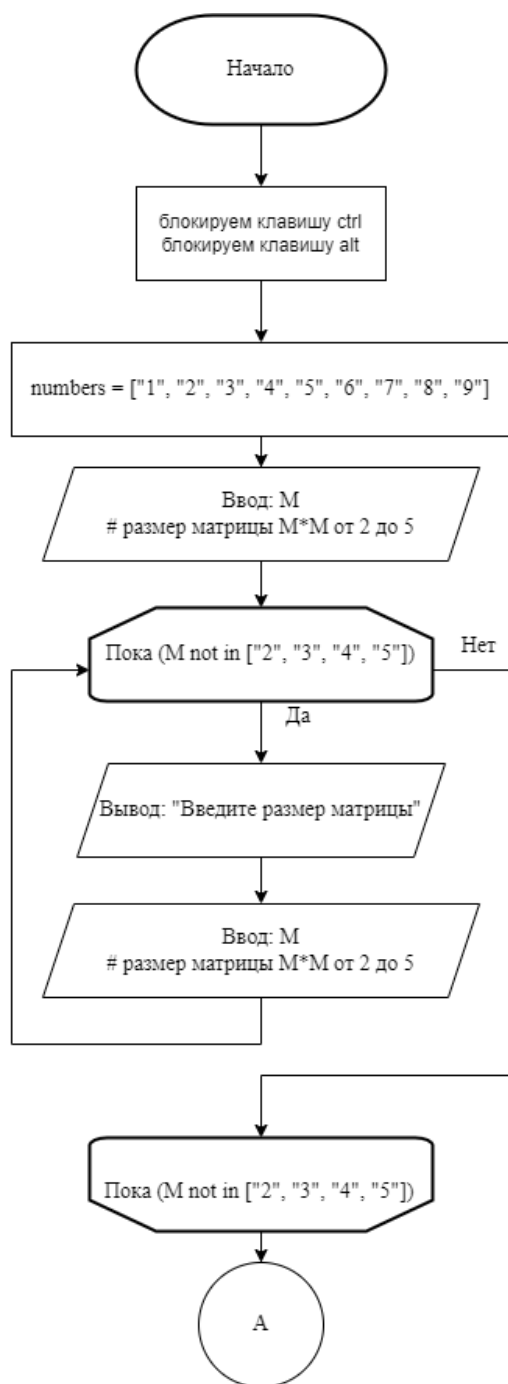


Рисунок 1 – Блок-схема программы, часть 1

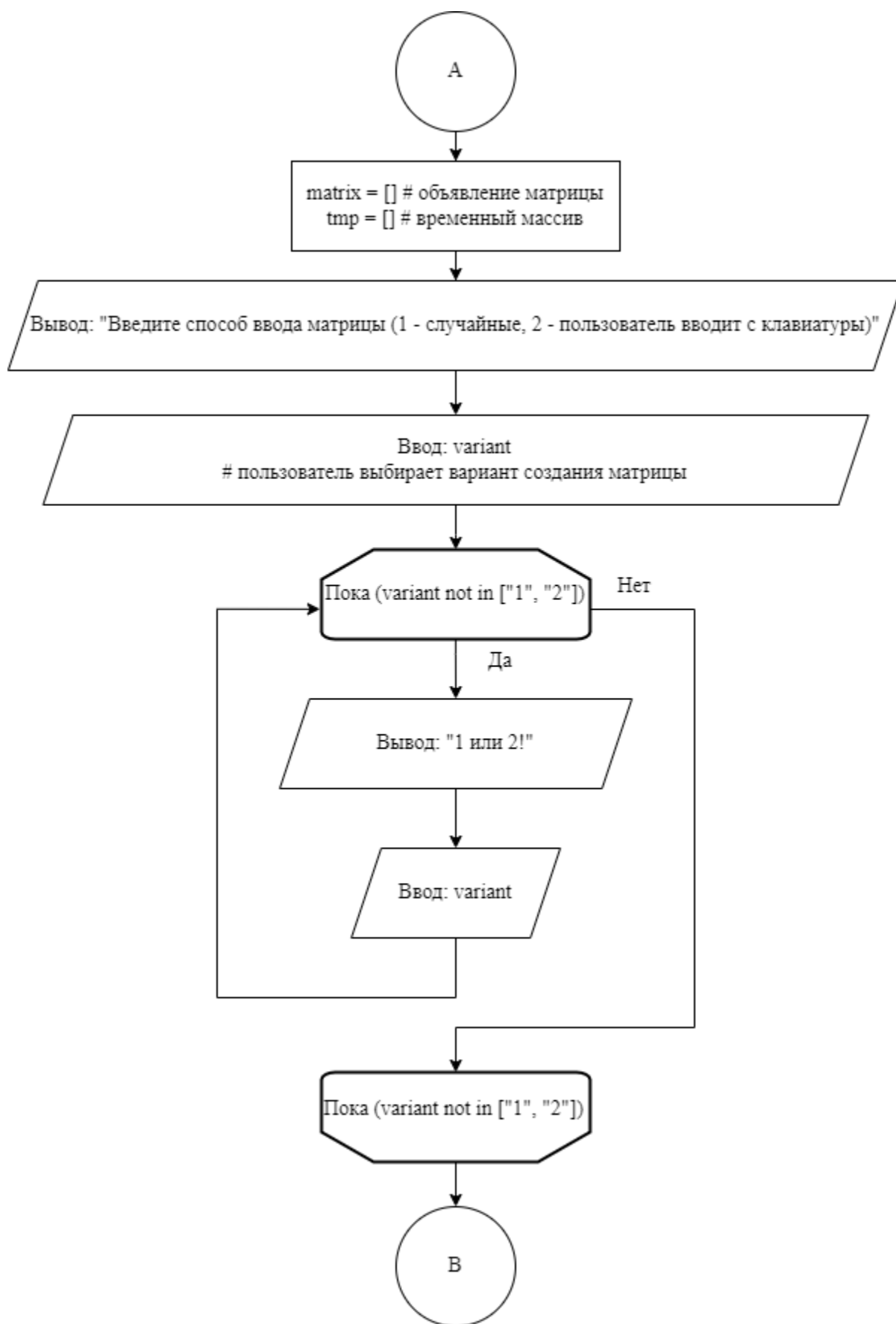


Рисунок 2 – Блок-схема программы, часть 2

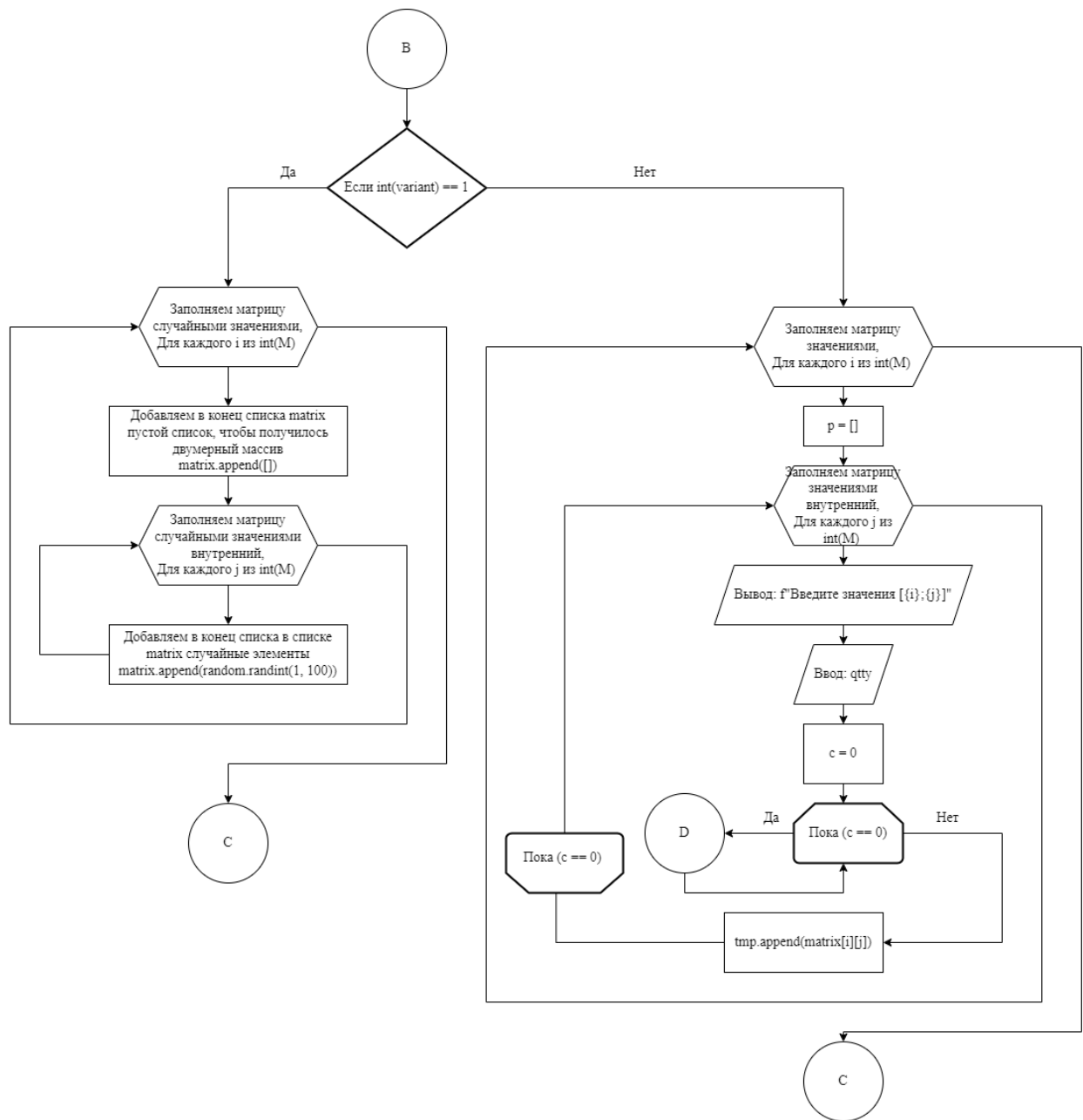


Рисунок 3 – Блок-схема программы, часть 3

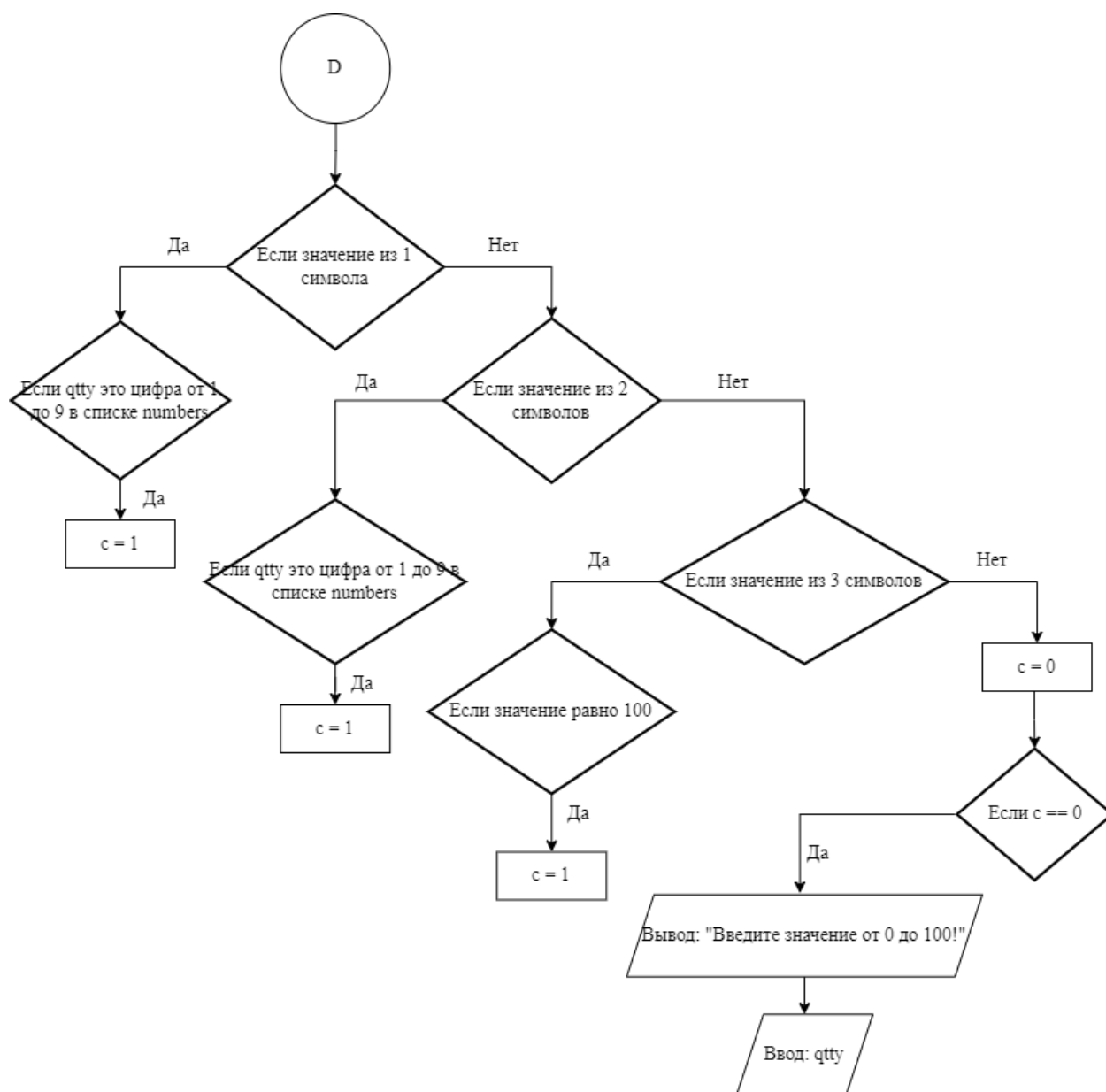


Рисунок 4 – Блок-схема программы, часть 4

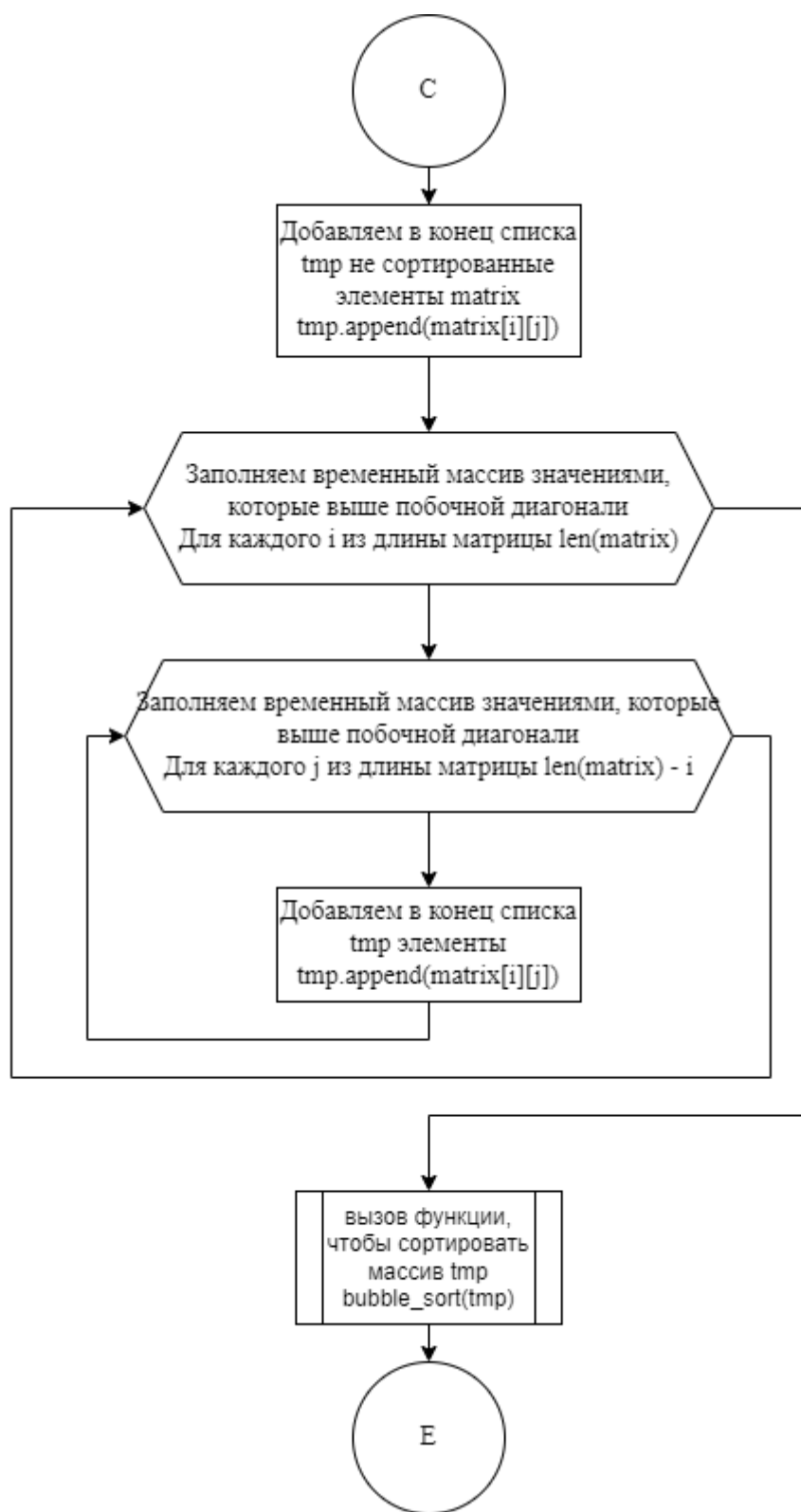


Рисунок 5 – Блок-схема процедуры, часть 5



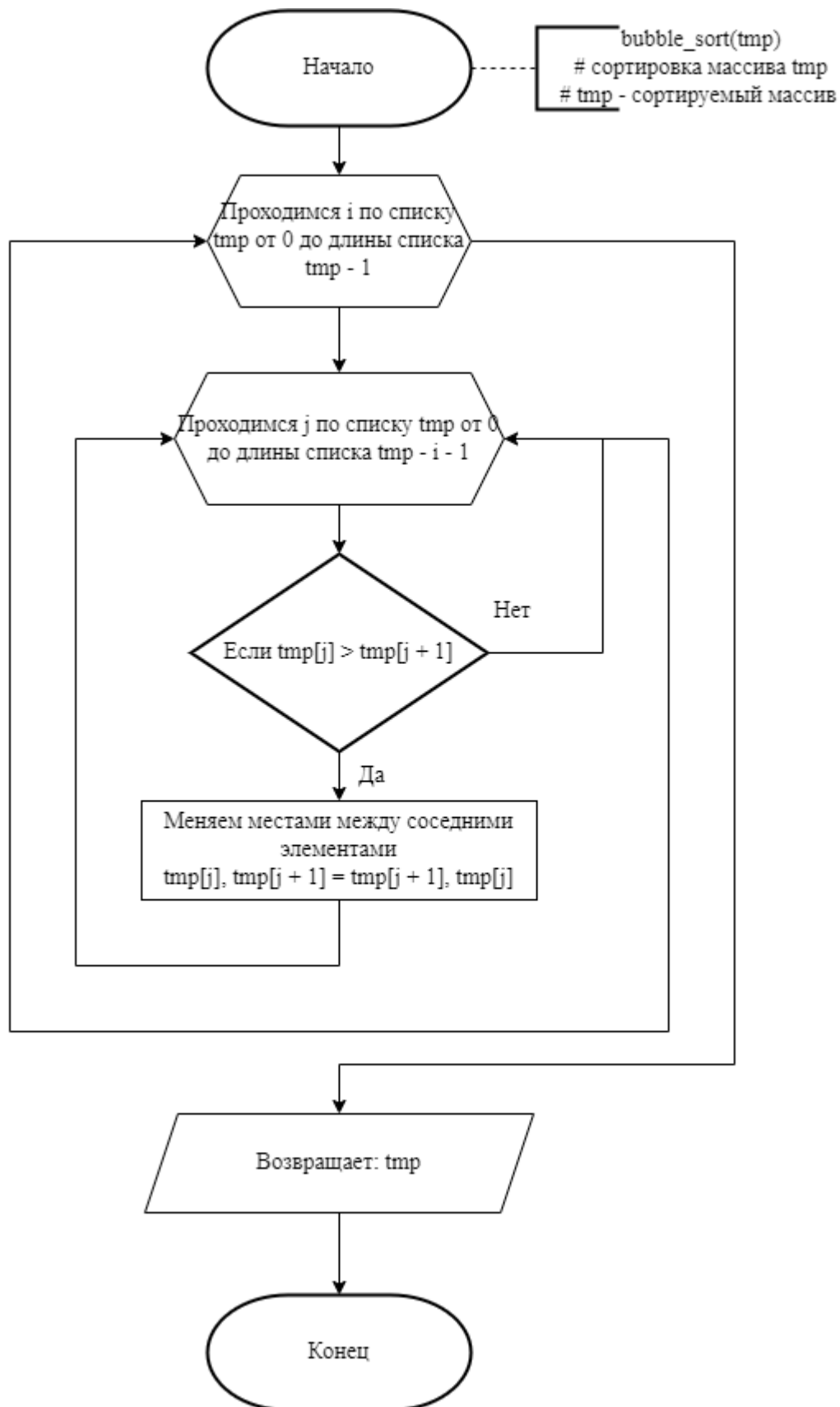


Рисунок 6 – Блок-схема процедуры bubble\_sort

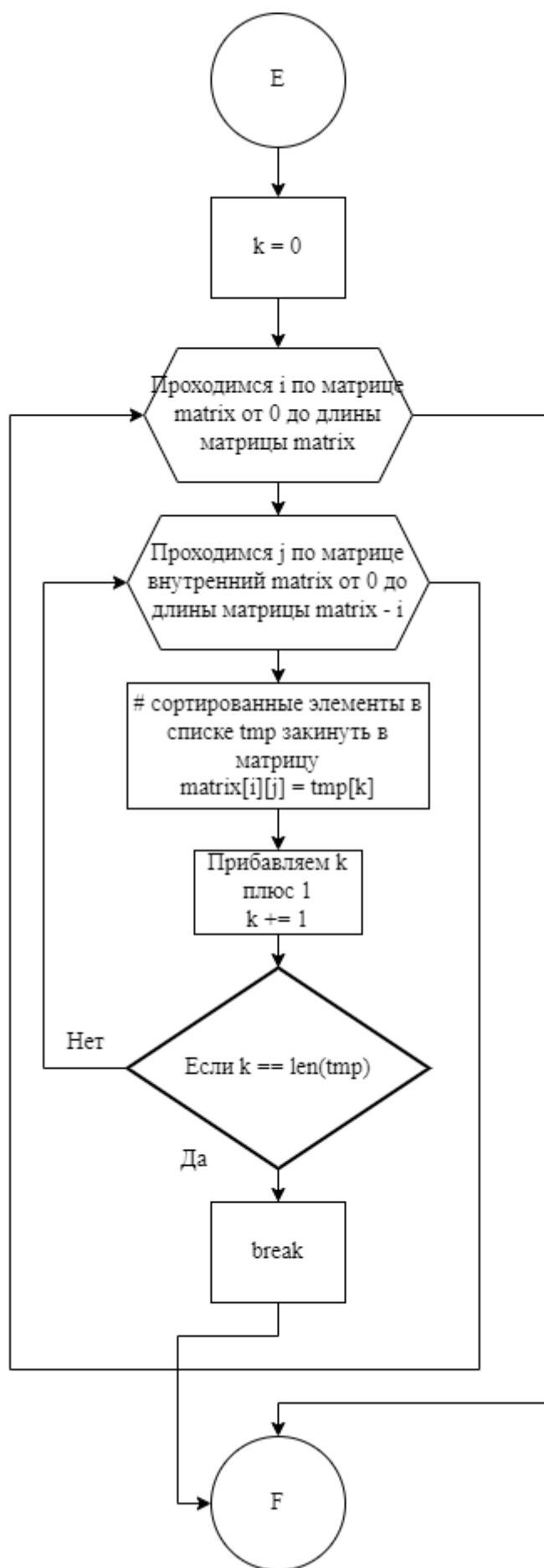


Рисунок 7 – Блок-схема программы, часть 7

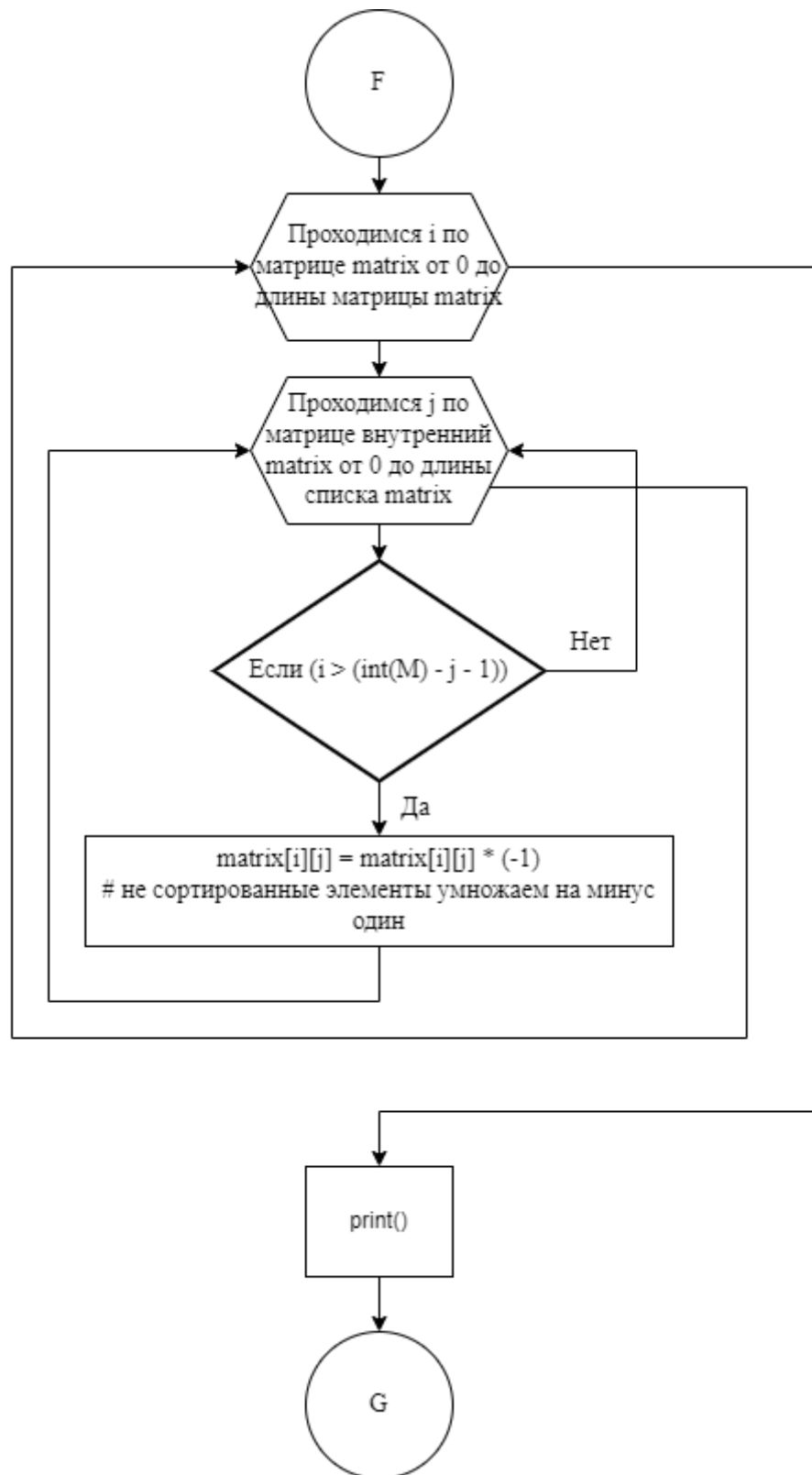


Рисунок 8 – Блок-схема программы, часть 8

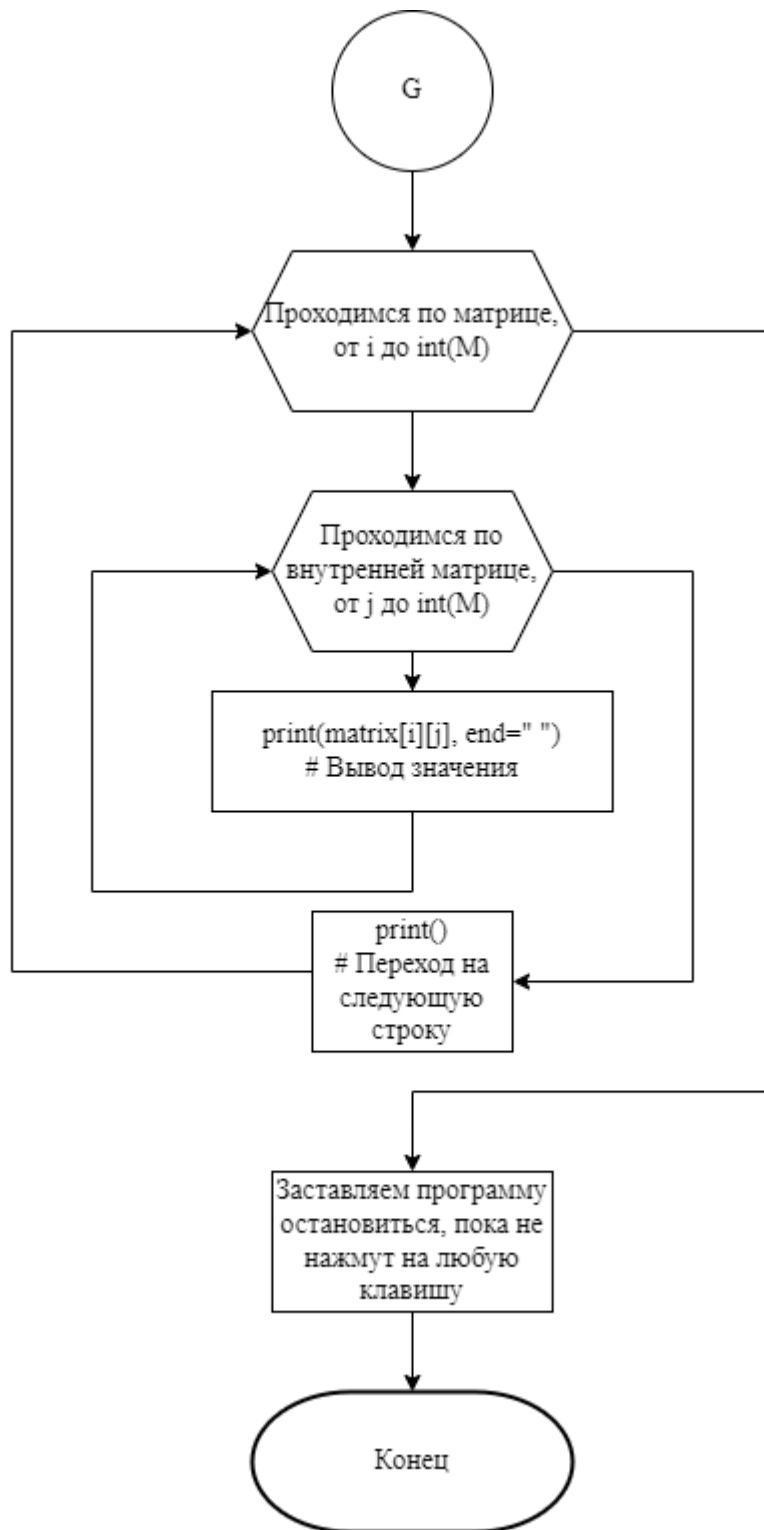


Рисунок 9 – Блок-схема программы, часть 9

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 import os # подключаем модуль для того, чтобы в конце программа могла ждать, пока не будет нажата любая клавиша
4
5
6 def bubble_sort(tmp):
7     for i in range(len(tmp) - 1):
8         for j in range(len(tmp) - i - 1):
9             if tmp[j] > tmp[j + 1]:
10                 tmp[j], tmp[j + 1] = tmp[j + 1], tmp[j]
11     return tmp
12
13
14 keyboard.block_key("ctrl")
15 keyboard.block_key("alt")
16 numbers = ["1", "2", "3", "4", "5", "6", "7", "8", "9"]
17 print("Какой диапазон для размера матрицы?")
18 M = input() # размер матрицы M*M от 2 до 5
19 while (M not in ["2", "3", "4", "5"]):
20     print("Размер матрицы должен быть от 2 до 5!")
21     M = input()
22 matrix = [] # объявление матрицы
23 tmp = [] # временный массив
24 print("Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)")
25 variant = input() # пользователь выбирает вариант создания матрицы
26 while (variant not in ["1", "2"]):
27     print("1 или 2!")
28     variant = input()
29 if (int(variant) == 1):
30     for i in range(int(M)):
31         matrix.append([])
32         for j in range(int(M)):
33             matrix[i].append(random.randint(1, 100))
34
35     print()
36     print("Вывод изначальной матрицы")
37     for i in range(int(M)): # вывод изначальной матрицы
38         for j in range(int(M)):
39             print(matrix[i][j], end=" ")
40         print()
41
42     for i in range(len(matrix)): # закидываю элементы выше побочной диагонали
43         for j in range(len(matrix) - i):
44             tmp.append(matrix[i][j])
45
46     bubble_sort(tmp) # сортируем элементы
47
48     k = 0
49     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
50         for j in range(len(matrix) - i):
51             matrix[i][j] = tmp[k]
52             k += 1
53         if (k == len(tmp)):
54             break
55
```

```

55
56 else:
57     for i in range(int(M)): # наполнение матрицы пользовательскими числами
58         p = []
59         for j in range(int(M)):
60             print(f"Введите значения [{i};{j}]")
61             qtty = input()
62             c = 0
63             while (c == 0):
64                 if (len(qtty) == 1):
65                     if (qtty[0] in numbers):
66                         c = 1
67                 elif (len(qtty) == 2):
68                     if (qtty[0] in numbers and qtty[1] in numbers):
69                         c = 1
70                 elif (len(qtty) == 3):
71                     if (int(qtty) == 100):
72                         c = 1
73                 else:
74                     c = 0
75             if (c == 0):
76                 print("Введите значение от 0 до 100!")
77                 qtty = input()
78             p.append(int(qtty))
79         matrix.append(p)
80
81     print()
82     print("Вывод изначальной матрицы")
83     for i in range(int(M)): # вывод изначальной матрицы
84         for j in range(int(M)):
85             print(matrix[i][j], end=" ")
86         print()
87
88     for i in range(len(matrix)): # закидываю элементы выше побочной диагонали
89         for j in range(len(matrix) - i):
90             tmp.append(matrix[i][j])
91
92     bubble_sort(tmp) # сортируем элементы
93
94     k = 0
95     for i in range(len(matrix)): # сортированные элементы закидываю в матрицу
96         for j in range(len(matrix) - i):
97             matrix[i][j] = tmp[k]
98             k += 1
99             if (k == len(tmp)):
100                 break
101
102     for i in range(int(M)): # оставшиеся не сортированные элементы умножаю на минус один
103         for j in range(int(M)):
104             if (i > (int(M) - j - 1)):
105                 matrix[i][j] = matrix[i][j] * (-1)
106
107     print()
108     print("Вывод конечной матрицы")
109     for i in range(int(M)): # вывод конечной матрицы
110         for j in range(int(M)):
111             print(matrix[i][j], end=" ")
112         print()
113     os.system("pause")

```

## 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.