



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

РТУ МИРЭА

Институт радиоэлектроники и информатики
Кафедра геоинформационных систем

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

Выполнил студент группы *ИКБО-30-23*

Павлов Н.С.

Принял ассистент кафедры ГИС

Корчемная А.И.

Практическая
работа выполнена

«__» _____ 2023 г.

«Зачтено»

«__» _____ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	4
2.1 Персональный вариант	4
2.2 Блок-схемы алгоритмов программы	4
2.3 Структурированный код программы с комментариями	8
2.4 Примеры тестирования, доказывающие работоспособность программы .	11
3 ВЫВОДЫ	12
4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	13

1 ПОСТАНОВКА ЗАДАЧИ

Требуется разработать блок-схему алгоритма и написать программу обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом (см. далее). 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, т.е. 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.

2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1 Персональный вариант

2.20. Создать квадратную матрицу размера 6 на 6. Матрица может заполняться как числами, вводимыми с клавиатуры, так и случайными числами (на выбор пользователя) из диапазона от 0 до 100. Над данной матрицей выполнить следующие преобразования. Поделить четные строки на нечетные, в результате чего получить матрицу размера 3 на 6. Далее поделить четные столбцы на нечетные, в результате чего получить матрицу 3 на 3. Столбцы полученной матрицы требуется упорядочить по возрастанию элементов. Предусмотреть корректную реакцию программы в случае возникновения деления на 0. Результаты преобразований вывести на экран.

2.2 Блок-схемы алгоритмов программы

На рисунках 2-4 представлены части блок-схемы основной программы. На рисунке 1 приведены блок-схемы подпрограмм.

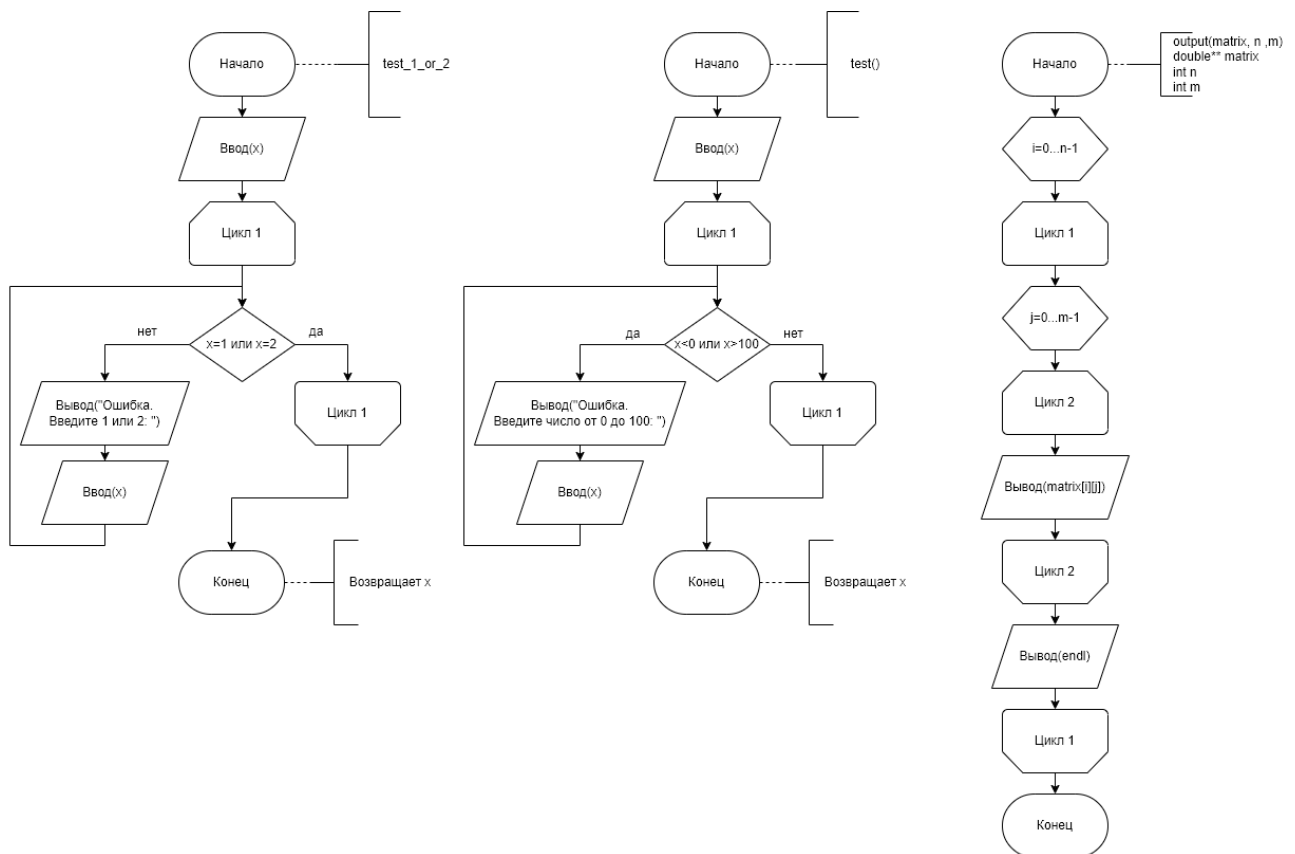


Рисунок 1 – Блок-схемы подпрограмм

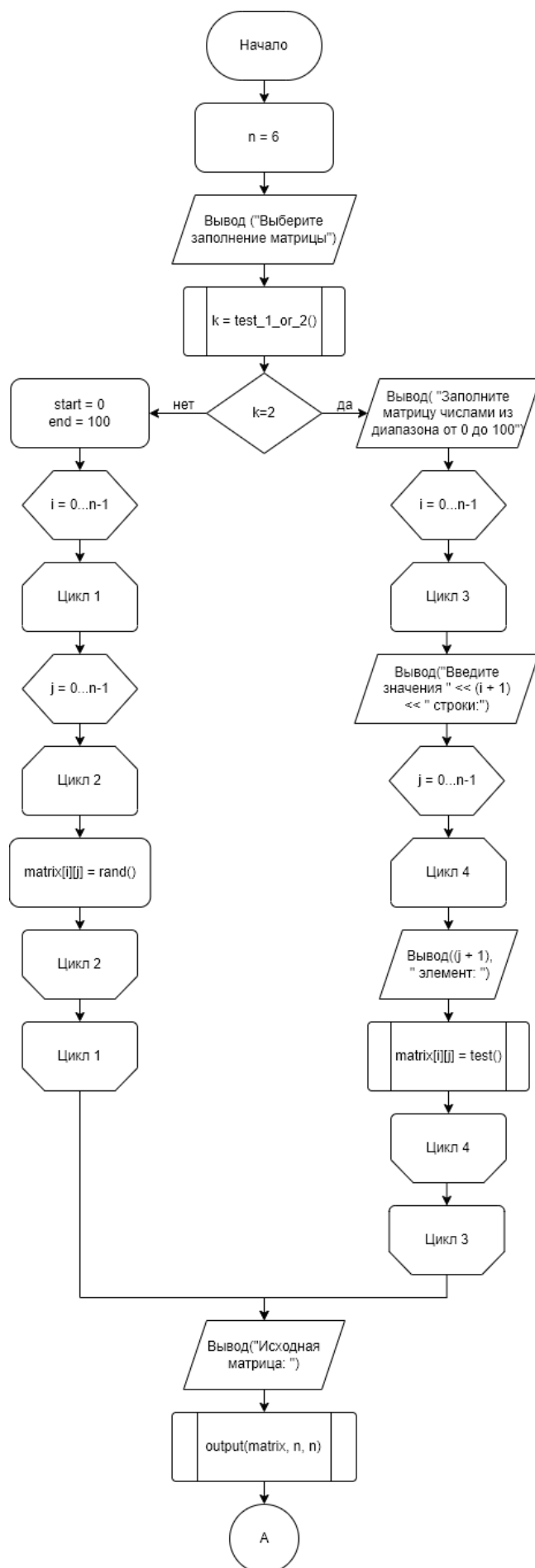


Рисунок 2 – Первая часть блок-схемы основной программы

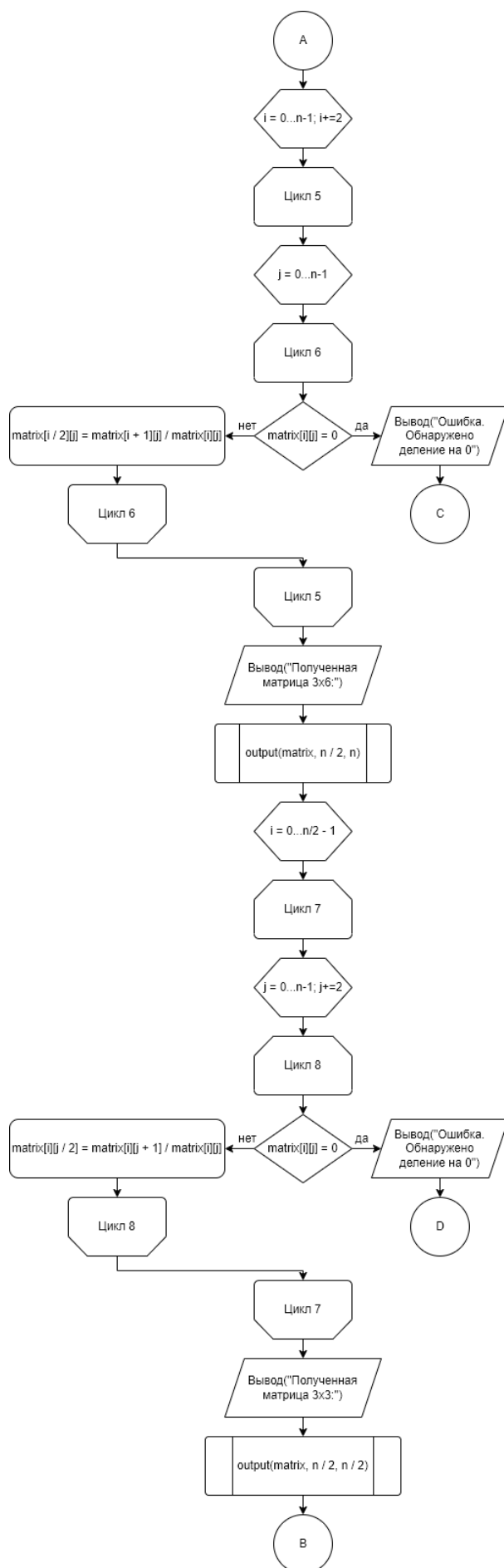


Рисунок 3 – Вторая часть блок-схемы основной программы

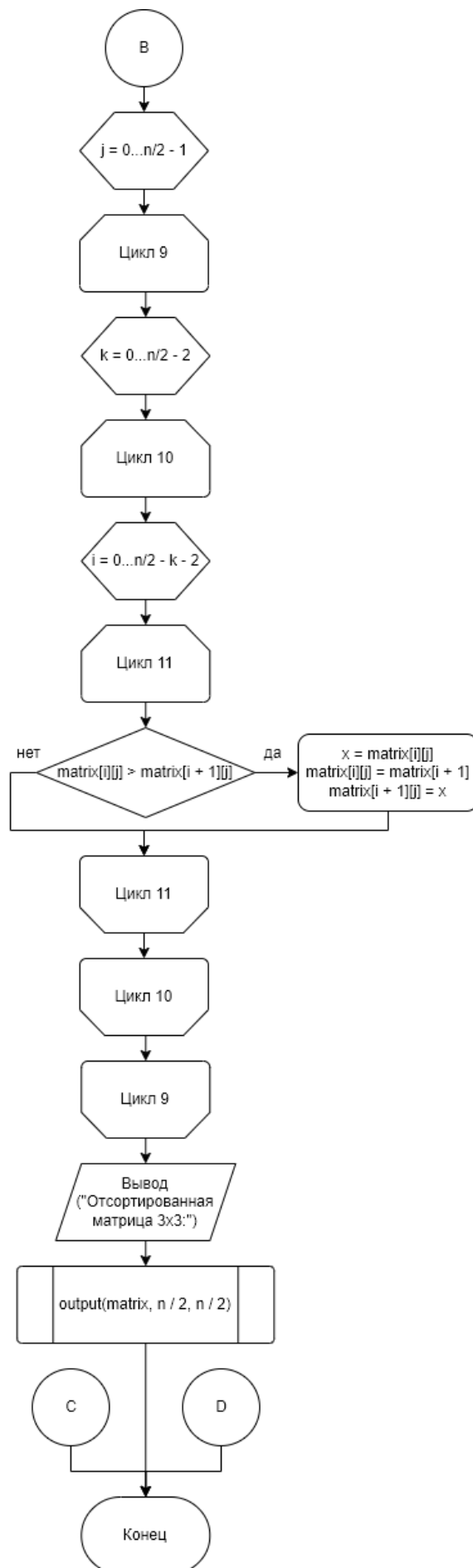


Рисунок 4 – Третья часть блок-схемы основной программы

2.3 Структурированный код программы с комментариями

На рисунках 5-9 представлена программа, реализованная на языке C++. Для удобства восприятия кода перед каждым блоком программы приведены комментарии, объясняющие что выполняется в данной части.

```
1  #include <iostream>
2  #include <math.h>
3  #include <ctime>
4  #include <string>
5  #include <stdio.h>
6
7  using namespace std;
8
9  //функция проверки правильности ввода (выбора способа заполнения)
10 double test_1_or_2()
11 {
12     double x;
13     cin >> x;
14     while (cin.fail() or cin.eof() or (round(x)!=x) or ((x != 1) and (x != 2))) {
15         cin.clear();
16         cin.ignore(numeric_limits<streamsize>::max(), '\n');
17         cout << "Ошибка. Введите 1 или 2: ";
18         cin >> x;
19     }
20     return x;
21 }
22
23 //функция проверки правильности ввода (значение ячеек матрицы)
24 double test()
25 {
26     double x;
27     cin >> x;
28     while (cin.fail() or cin.eof() or (x < 0) or (x > 100)) {
29         cin.clear();
30         cin.ignore(numeric_limits<streamsize>::max(), '\n');
31         cout << "Ошибка. Введите число из диапазона от 0 до 100: ";
32         cin >> x;
33     }
34     return x;
35 }
```

Рисунок 5 – Код программы, часть 1


```

37 //функция вывода матрицы размера n*m
38 void output(double** matrix, int n, int m)
39 {
40     for (int i = 0; i < n; i++) {
41         for (int j = 0; j < m; j++) {
42             printf("%10g", matrix[i][j]);
43         }
44         cout << endl;
45     }
46     cout << endl;
47 }
48
49 int main()
50 {
51     setlocale(LC_ALL, "rus");
52     const int n = 6;
53     //выделение памяти для двумерного массива типа double
54     double** matrix = new double* [n];
55     for (int j = 0; j < n; j++) {
56         matrix[j] = new double[n];
57     }
58     //выбор способа заполнения матрицы
59     cout << "Выберите заполнение матрицы (1 - случайное; 2 - с клавиатуры): ";
60     int k;
61     k = test_1_or_2();

```

Рисунок 6 – Код программы, часть 2

```

62     switch (k)
63     {
64         //заполнение матрицы сгенерированными числами
65         case 1:
66         {
67             srand(time(0));
68             int end = 100, start = 0;
69             for (int i = 0; i < n; i++) {
70                 for (int j = 0; j < n; j++) {
71                     matrix[i][j] = (rand() % (end - start + 1)) + start;
72                 }
73             }
74         }
75         break;
76         //заполнение матрицы с клавиатуры
77         case 2:
78         {
79             cout << "Заполните матрицу числами из диапазона от 0 до 100" << endl;
80             for (int i = 0; i < n; i++) {
81                 cout << "Введите значения " << (i + 1) << " строки:" << endl;
82                 for (int j = 0; j < n; j++) {
83                     cout << (j + 1) << " элемент: ";
84                     matrix[i][j] = test();
85                 }
86             }
87             break;
88         }
89         //вывод на экран введенной (или сгенерированной) матрицы
90         cout << "Исходная матрица: " << endl << endl;
91         output(matrix, n, n);

```

Рисунок 7 – Код программы, часть 3

```

91      //деление четных строк на нечетные и перезапись матрицы
92      for (int i = 0; i < n; i += 2) {
93          for (int j = 0; j < n; j++) {
94              if (matrix[i][j] == 0) {
95                  cout << "Ошибка. Обнаружено деление на 0" << endl;
96                  exit(0);
97              }
98              else {
99                  matrix[i / 2][j] = matrix[i + 1][j] / matrix[i][j];
100              }
101          }
102      }
103      //вывод матрицы 3x6
104      cout << "Полученная матрица 3x6:" << endl << endl;
105      output(matrix, n / 2, n);
106      //деление четных столбцов на нечетные и перезапись матрицы
107      for (int i = 0; i < n / 2; i++) {
108          for (int j = 0; j < n; j += 2) {
109              if (matrix[i][j] == 0) {
110                  cout << "Ошибка. Обнаружено деление на 0" << endl;
111                  exit(0);
112              }
113              else {
114                  matrix[i][j / 2] = matrix[i][j + 1] / matrix[i][j];
115              }
116          }
117      }
118      //вывод матрицы 3x3
119      cout << "Полученная матрица 3x3:" << endl << endl;
120      output(matrix, n / 2, n / 2);

```

Рисунок 8 – Код программы, часть 4

```

121      //сортировка столбцов матрицы
122      for (int j = 0; j < n / 2; j++) {
123          for (int k = 0; k < n / 2 - 1; k++) {
124              for (int i = 0; i < n / 2 - k - 1; i++) {
125                  if (matrix[i][j] > matrix[i + 1][j]) {
126                      double x = matrix[i][j];
127                      matrix[i][j] = matrix[i + 1][j];
128                      matrix[i + 1][j] = x;
129                  }
130              }
131          }
132      }
133      //вывод отсортированной матрицы 3x3
134      cout << "Отсортированная матрица 3x3:" << endl << endl;
135      output(matrix, n / 2, n / 2);
136      //очистка памяти
137      for (int j = 0; j < n; j++) {
138          delete[] matrix[j];
139      }
140      delete[] matrix;
141  }

```

Рисунок 9 – Код программы, часть 5

2.4 Примеры тестирования, доказывающие работоспособность программы

На рисунке 10 приведен пример корректной работы программы. На рисунке 11 представлен пример с делением на 0.

```
Выберите заполнение матрицы (1 - случайное; 2 - с клавиатуры): 1
Исходная матрица:
      34      39      93      11      87      80
      72      32      69      57      13      47
      85      12      85      34      77      53
       4      31      35      74      62      36
      92      76       9      40      77      13
      51      47       3      33      46      35

Полученная матрица 3x6:
      2,11765  0,820513  0,741935  5,18182  0,149425  0,5875
      0,0470588  2,58333  0,411765  2,17647  0,805195  0,679245
      0,554348  0,618421  0,333333  0,825  0,597403  2,69231

Полученная матрица 3x3:
      0,387464  6,98419  3,93173
      54,8958  5,28571  0,843579
      1,11558  2,475  4,50669

Отсортированная матрица 3x3:
      0,387464  2,475  0,843579
      1,11558  5,28571  3,93173
      54,8958  6,98419  4,50669
```

Рисунок 10 – Корректный результат тестирования программы

```
Выберите заполнение матрицы (1 - случайное; 2 - с клавиатуры): 1
Исходная матрица:
      56      48      32      0      93      3
     100      80       4      59      7      97
      19      34      25      82      14      63
      39      41      29      37      71      82
      25       0      57      33      23      65
      73      24      65      52      60      33

Ошибка. Обнаружено деление на 0
```

Рисунок 11 – Реакция программы на деление на 0

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 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов, - М., МИРЭА – Российский технологический университет, 2020. – 102
2. ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем». Текст: электронный. URL: <https://docs.yandex.ru/docs/gost.html> (дата обращения: 26.11.2023)