Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое задание №14**»

Выполнил: З иятдинова Алина Ленаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:** **3.** Для заданного значения n запишем в очередь все числа от 1 до n, а затем извлечем их из очереди.

**Входные и выходные данные**

str – данные из текстового поля, строковый тип

n-число n, целочисленный тип.

**Листинг программы (если есть)**

private void button1\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

string str = textBox1.Text;

if (str != string.Empty)

{

if (CheckTextBox(str) == "")

{

int n = Convert.ToInt32(str);

Queue que = new Queue();

for (int i = 1; i <= n; i++)

{

que.Enqueue(i);

}

listBox1.Items.Add("Размерность очереди " + que.Count);

listBox1.Items.Add("Верхний элемент очереди = " + que.Peek());

string s = "";

while (que.Count != 0)

{

s += que.Dequeue() + " ";

}

listBox1.Items.Add("Вызываемая очередь = " + s);

}

else

{

MessageBox.Show($"{CheckTextBox(str)}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = string.Empty;

}

}

else

{

MessageBox.Show("Пустая строка", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

static string CheckTextBox(string st)

{

string mes = "";

foreach (char i in st)

{

if (!Char.IsNumber(i))

{

mes = "Некорректные данные. Повторите попытку.";

break;

}

}

if (mes == "")

{

int num = Convert.ToInt32(st);

if (num < 1)

{

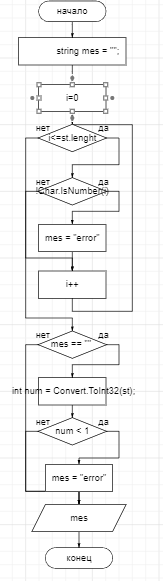
mes = "Число должно быть больше нуля.";

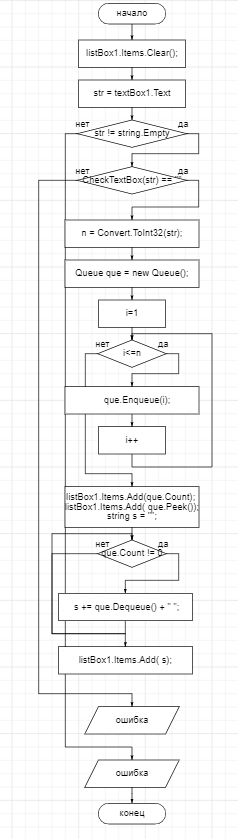
}

}

return mes;

}

**Блок схема**



**Задание:**

В текстовом файле записана информация о людях (фамилия, имя,

отчество, возраст, вес через пробел). Вывести на экран вначале

информацию о людях младше 40 лет, а затем информацию о всех

остальных.

**Входные и выходные данные**

filename – название файла, строковый тип.

str – строка файла, строковый тип.

**Листинг программы (если есть)**

listBox2.Items.Clear();

listBox3.Items.Clear();

if (textBox2.Text != string.Empty && textBox3.Text != string.Empty)

{

if (CheckAge(textBox3.Text) == "")

{

string filename = textBox2.Text + ".txt";

if (File.Exists(filename))

{

string[] stroka = File.ReadAllLines(filename);

if (stroka.Length == 0)

{

MessageBox.Show("Файл пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

Queue<string> set1 = new Queue<string>();

Queue<string> set2 = new Queue<string>();

StreamReader sr = new StreamReader(s);

while (!sr.EndOfStream)

{

string str = sr.ReadLine();

if (str != String.Empty)

{

set1.Enqueue(str);

}

}

sr.Close();

foreach (object i in set1)

{

listBox2.Items.Add(i);

}

for (int i = 0; i < listBox2.Items.Count; i++)

{

string list\_str = listBox2.Items[i].ToString();

string[] split = list\_str.Split(' ');

if (Convert.ToDouble(split[3]) <= Convert.ToDouble(textBox3.Text))

{

listBox3.Items.Add(list\_str);

}

else

{

set2.Enqueue(list\_str);

}

}

foreach (object i in set2)

{

listBox3.Items.Add(i);

}

}

}

else

{

MessageBox.Show("Файла с таким названием нет", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show($"{CheckAge(textBox3.Text)}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

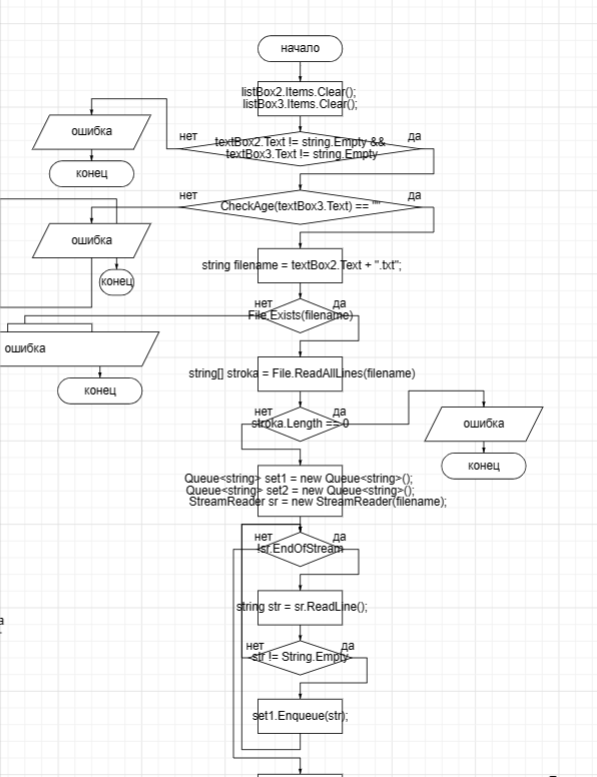
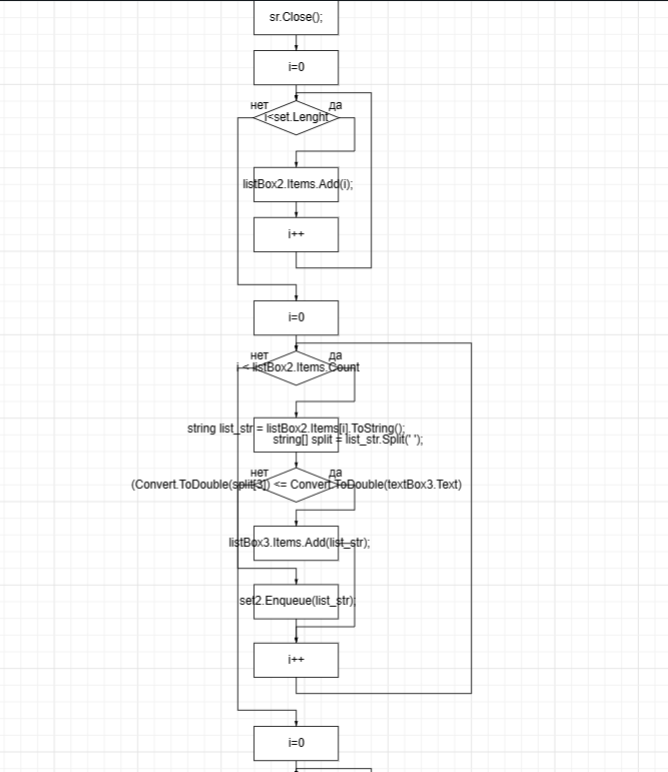
else

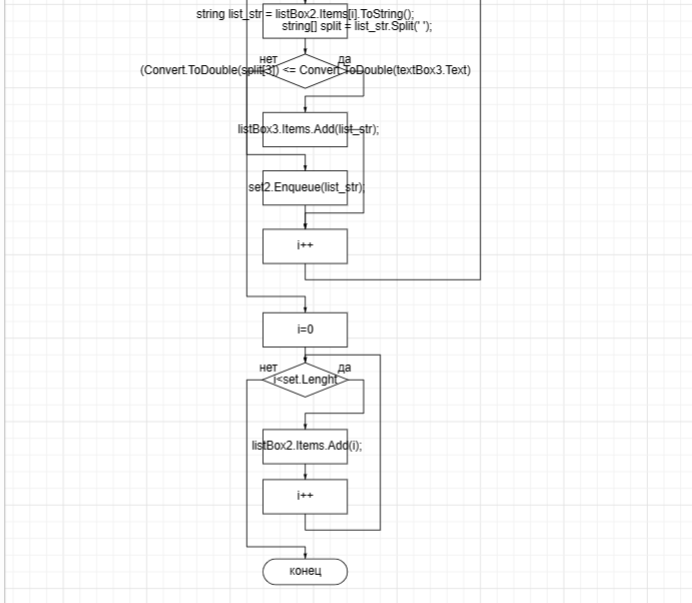
{

MessageBox.Show("Не все поля заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**Блок схема**

**Задание:**

В текстовом файле записана информация о людях (фамилия, имя, отчество,

возраст, вес через пробел). Вывести на экран информацию о людях,

отсортированную по возрасту.

**Входные и выходные данные**

filename – название файла, строковый тип.

str – строка файла, строковый тип.

**Листинг программы (если есть)**

listBox4.Items.Clear();

listBox5.Items.Clear();

if (textBox4.Text != string.Empty)

{

string s = textBox4.Text + ".txt";

if (File.Exists(s))

{

string[] stroka = File.ReadAllLines(s);

if (stroka.Length == 0)

{

MessageBox.Show("Файл пуст", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

Queue<string> set1 = new Queue<string>();

Queue<string> set2 = new Queue<string>();

List<string> list = new List<string>();

StreamReader sr = new StreamReader(s);

while (!sr.EndOfStream)

{

string str = sr.ReadLine();

if (str != String.Empty)

{

set1.Enqueue(str);

}

}

sr.Close();

foreach (object i in set1)

{

listBox4.Items.Add(i);

}

for (int i = 0; i < listBox4.Items.Count; i++)

{

string list\_str = listBox4.Items[i].ToString();

string[] split = list\_str.Split(' ');

set2.Enqueue(split[3]);

list.Add(split[3]);

}

list.Sort();

for (int j = 0; j < list.Count; j++)

{

for (int i = 0; i < listBox4.Items.Count; i++)

{

string list\_str = listBox4.Items[i].ToString();

string[] split = list\_str.Split(' ');

if (split[3] == list[j])

{

listBox5.Items.Add(list\_str);

break;

}

}

}

}

}

else

{

MessageBox.Show("Файла с таким названием нет", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

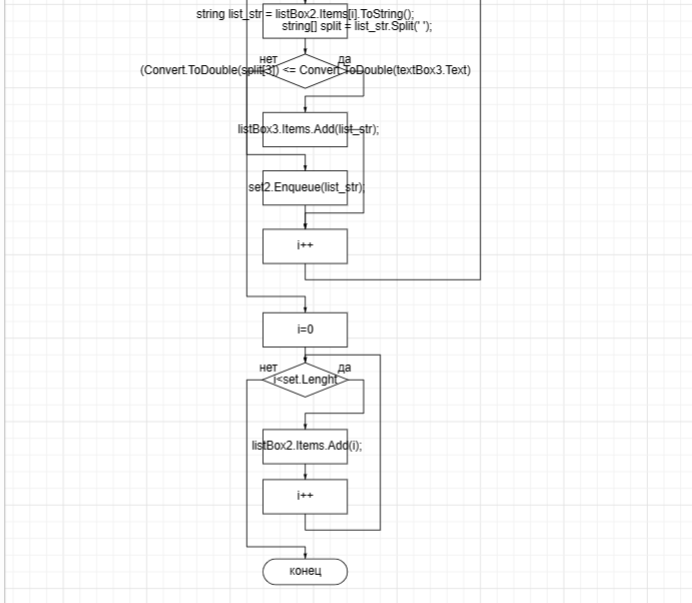
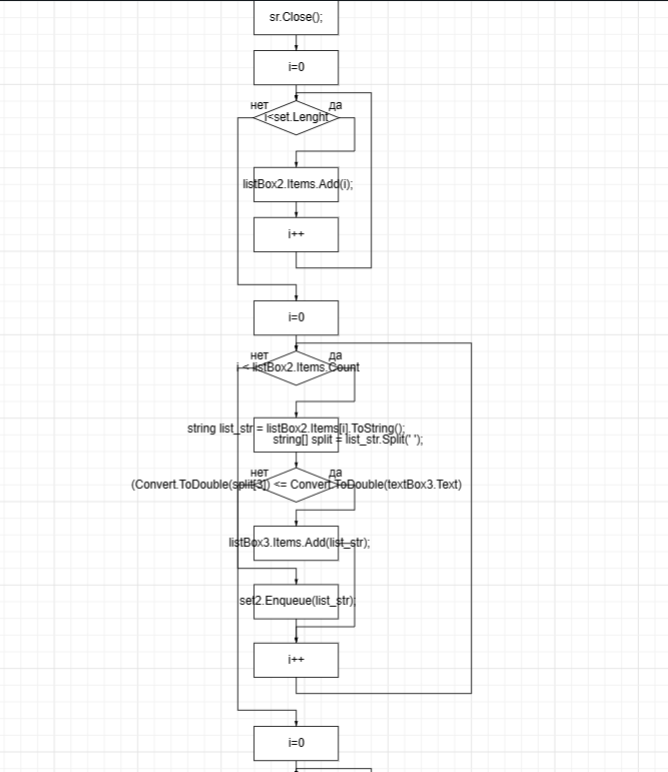
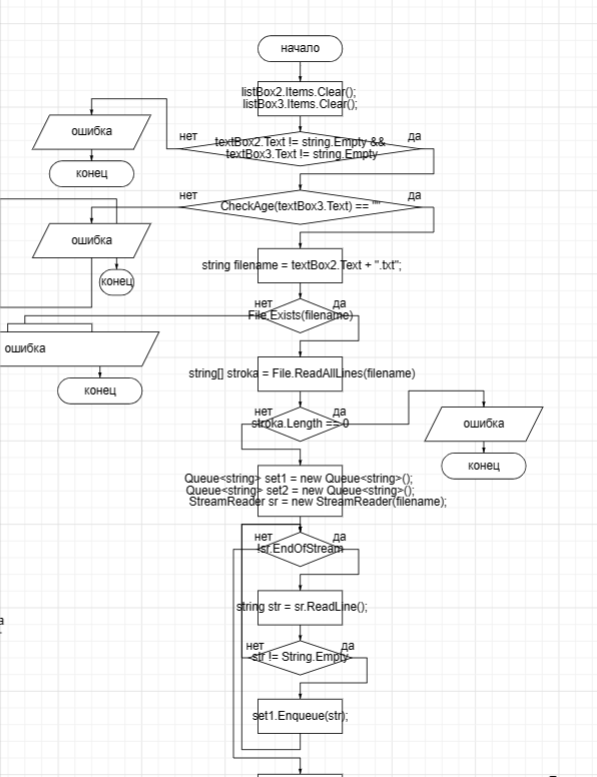
else

{

MessageBox.Show("Не все поля заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**Блок схема**



**Тестовые ситуации**

Выполнены проверки на входные данные. Проверка на корректность данных, на наличие файла и граничные значения некоторых входных данных.

