Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая 14, 1-5**»

Выполнил: Карамов Д.Э

Группа: ПР-24

Преподаватель: Мирошниченко Г.В

2025

**1**

**Задание:**

****

**Входные данные**

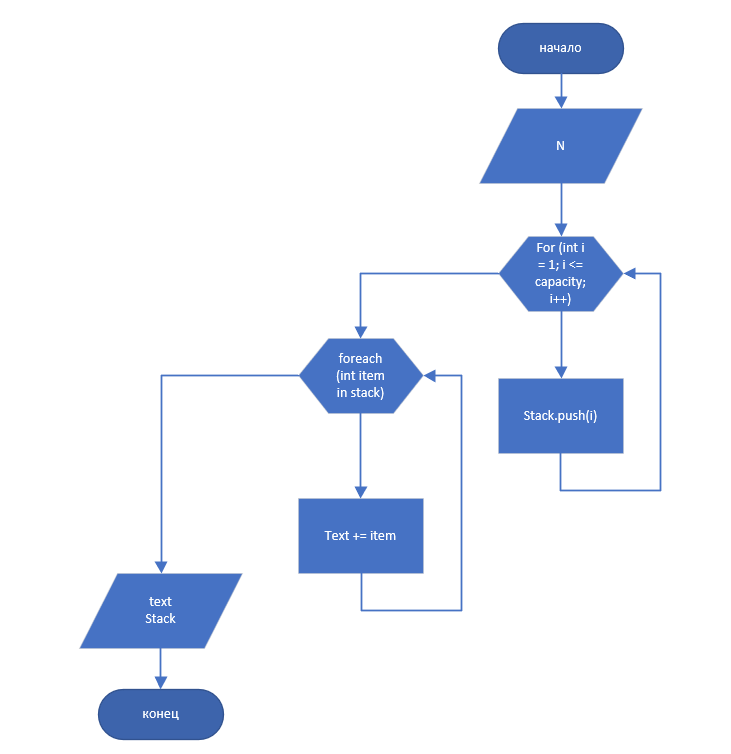
n - Int

**Выходные данные**

Stack 1 – n

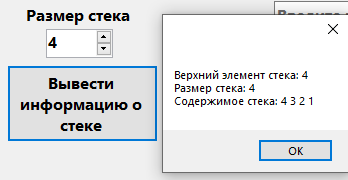
Text – string – содержимое стека

**Блок-схема**



**Тестовые ситуации**

**Результат**

****

**Листинг программы**

int capacity = Convert.ToInt32(numericUpDown1.Value);

Stack<int> stack = new Stack<int>();

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

{

stack.Push(i);

}

string text = "";

foreach (int item in stack)

{

text += $"{item} ";

}

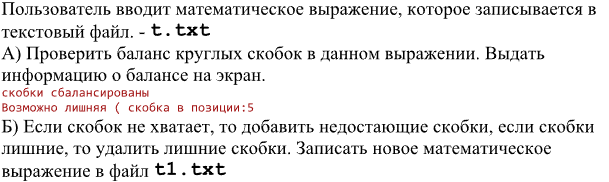
MessageBox.Show($"Верхний элемент стека: {stack.Peek()}\nРазмер стека: {stack.Count}\nСодержимое стека: {text}");

stack.Clear();

MessageBox.Show($"Новая размерность стека: {stack.Count}");

**2**

**Задание:**

****

**Входные данные:**

Text – string – Выражение

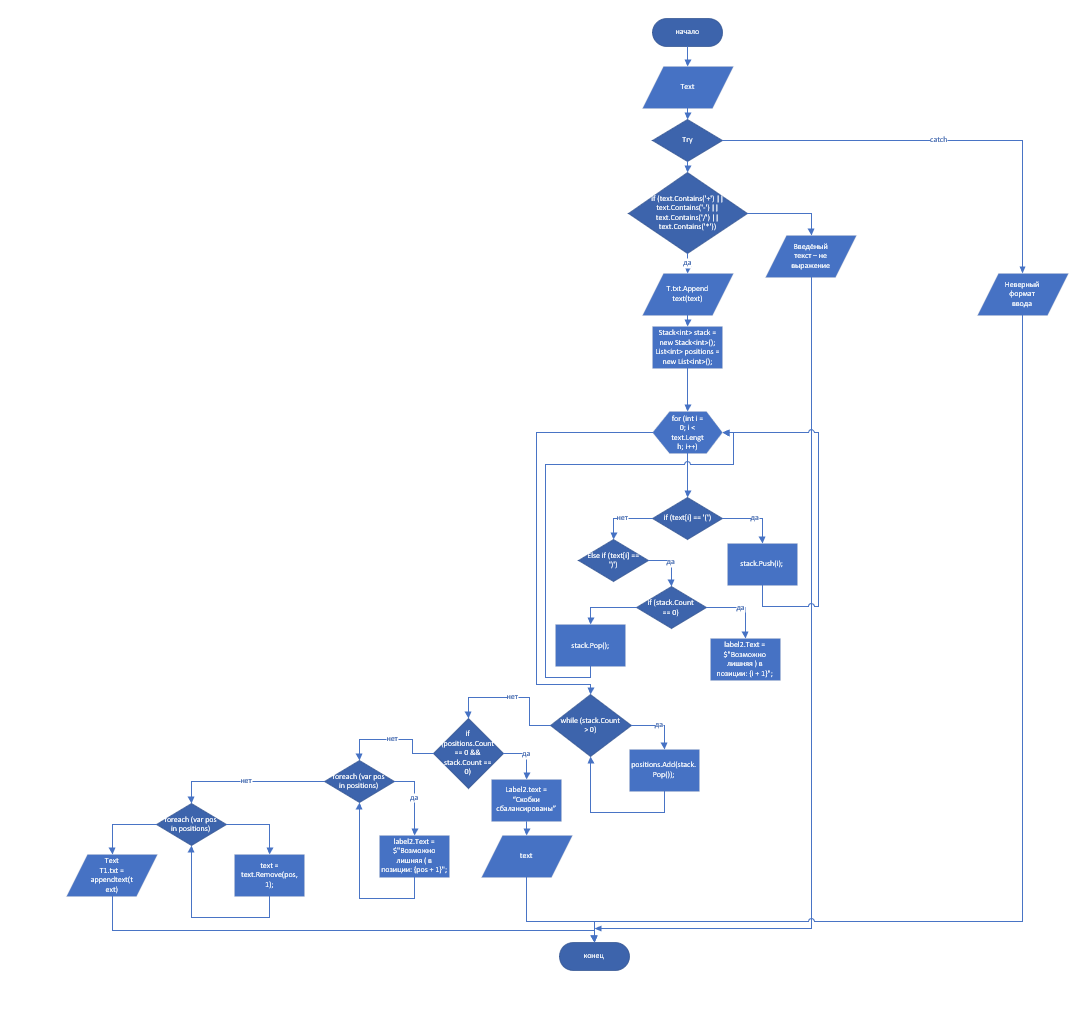
**Выходные данные:**

Text – string – Изменённое выражение

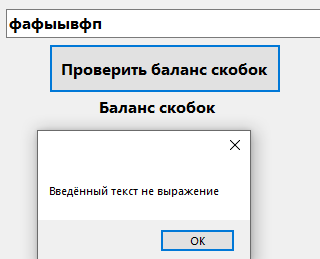
T.txt – Текстовый файл

T1.txt – текстовый файл

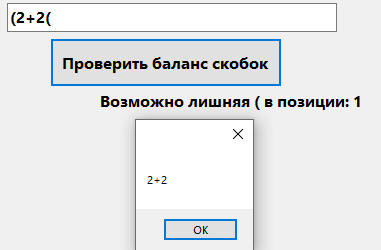
**Блок схема**

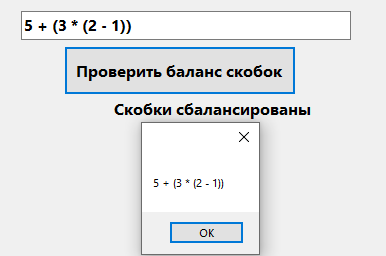
****

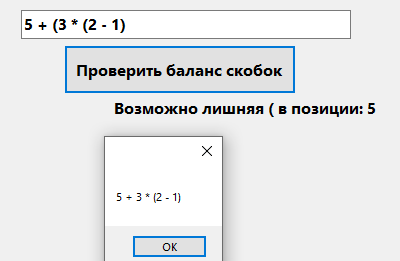
**Тестовые ситуации**

****

**Результат**

****

****

****

**Листинг программы**

try

{

string text = textBox1.Text;

if (text.Contains('+') || text.Contains('-') || text.Contains('/') || text.Contains('\*'))

{

using (StreamWriter sw = File.AppendText("t.txt"))

{

sw.WriteLine(text);

}

Stack<int> stack = new Stack<int>();

List<int> positions = new List<int>();

for (int i = 0; i < text.Length; i++)

{

if (text[i] == '(')

{

stack.Push(i);

}

else if (text[i] == ')')

{

if (stack.Count == 0)

{

label2.Text = $"Возможно лишняя ) в позиции: {i + 1}";

}

else

{

stack.Pop();

}

}

}

while (stack.Count > 0)

{

positions.Add(stack.Pop());

}

if (positions.Count == 0 && stack.Count == 0)

{

label2.Text = "Скобки сбалансированы";

MessageBox.Show($"{text}");

}

else

{

foreach (var pos in positions)

{

label2.Text = $"Возможно лишняя ( в позиции: {pos + 1}";

}

foreach (var pos in positions)

{

text = text.Remove(pos, 1);

}

MessageBox.Show($"{text}");

}

using (StreamWriter sw = File.AppendText("t1.txt"))

{

sw.WriteLine(text);

}

}

else

{

MessageBox.Show("Введённый текст не выражение");

}

}

catch

{

MessageBox.Show("Ошибка");

}

**3**

**Задание**

****

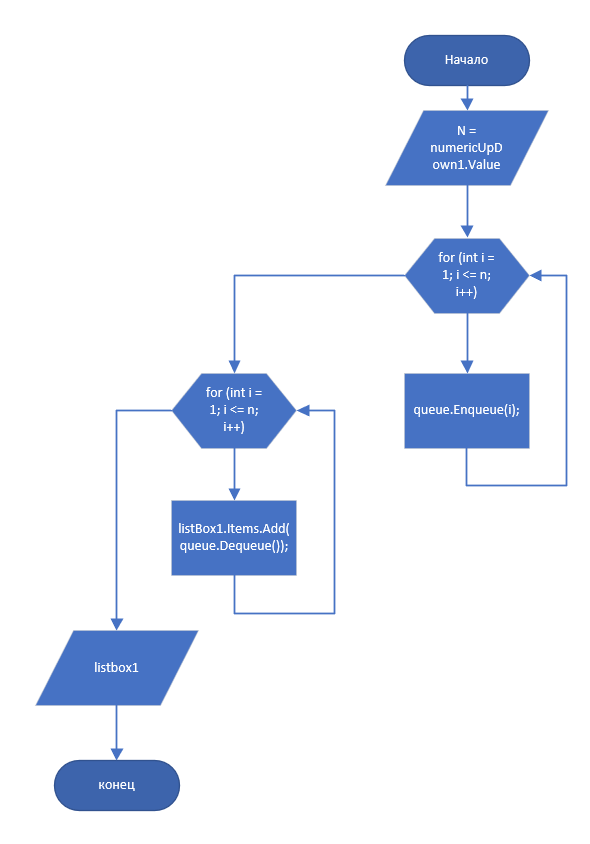
**Входные данные**

N – int – размер очереди

**Выходные данные**

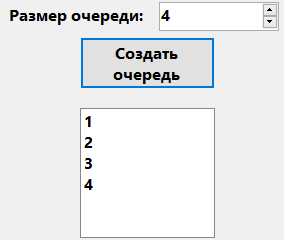
Queue – очередь из n чисел, выводится в listBox

**Блок схема**

****

**Тестовые ситуации**

**Результат**

****

**Листинг программы**

listBox1.Items.Clear();

Queue<int> queue = new Queue<int>();

int n = Convert.ToInt32(numericUpDown1.Value);

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

{

queue.Enqueue(i);

}

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

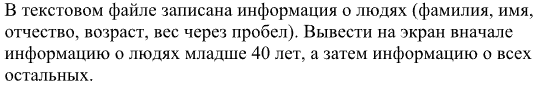
{

listBox1.Items.Add(queue.Dequeue());

}

**4**

**Задание**

****

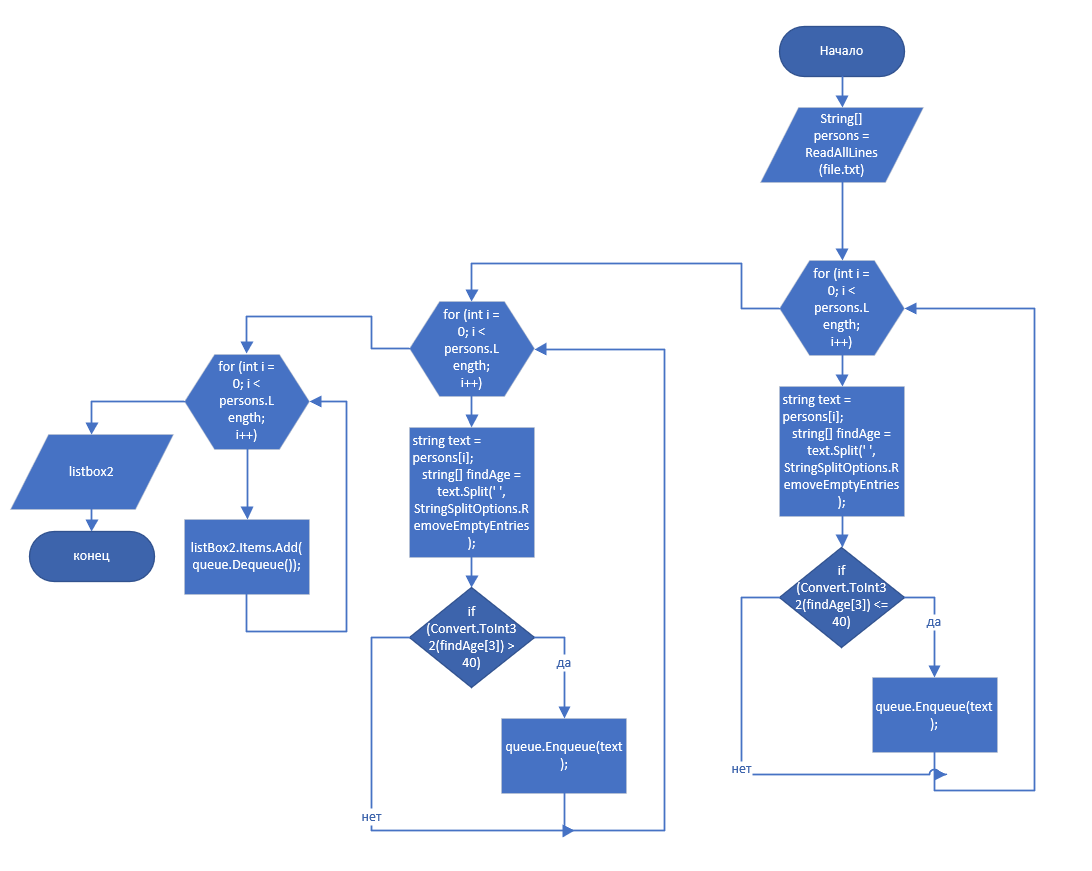
**Входные данные**

Persons – string[] – Считанный из файла массив строк

**Выходные данные**

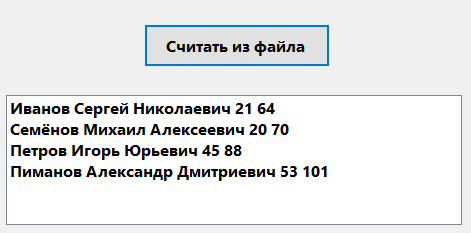
Queue – Очередь из людей выводится в listBox

**Блок схема**

****

**Тестовые ситуации**

**Результат**

****

**Листинг программы**

listBox2.Items.Clear();

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

string[] persons = File.ReadAllLines("file.txt");

for (int i = 0; i < persons.Length; i++)

{

string text = persons[i];

string[] findAge = text.Split(' ', StringSplitOptions.RemoveEmptyEntries);

if (Convert.ToInt32(findAge[3]) <= 40)

{

queue.Enqueue(text);

}

}

for (int i = 0; i < persons.Length; i++)

{

string text = persons[i];

string[] findAge = text.Split(' ');

if (Convert.ToInt32(findAge[3]) > 40)

{

queue.Enqueue(text);

}

}

for (int i = 0; i < persons.Length; i++)

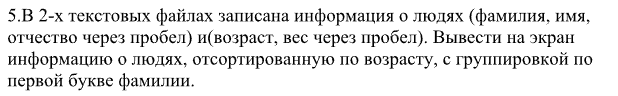
{

listBox2.Items.Add(queue.Dequeue());

}

**5**

**Задание**

****

**Входные данные**

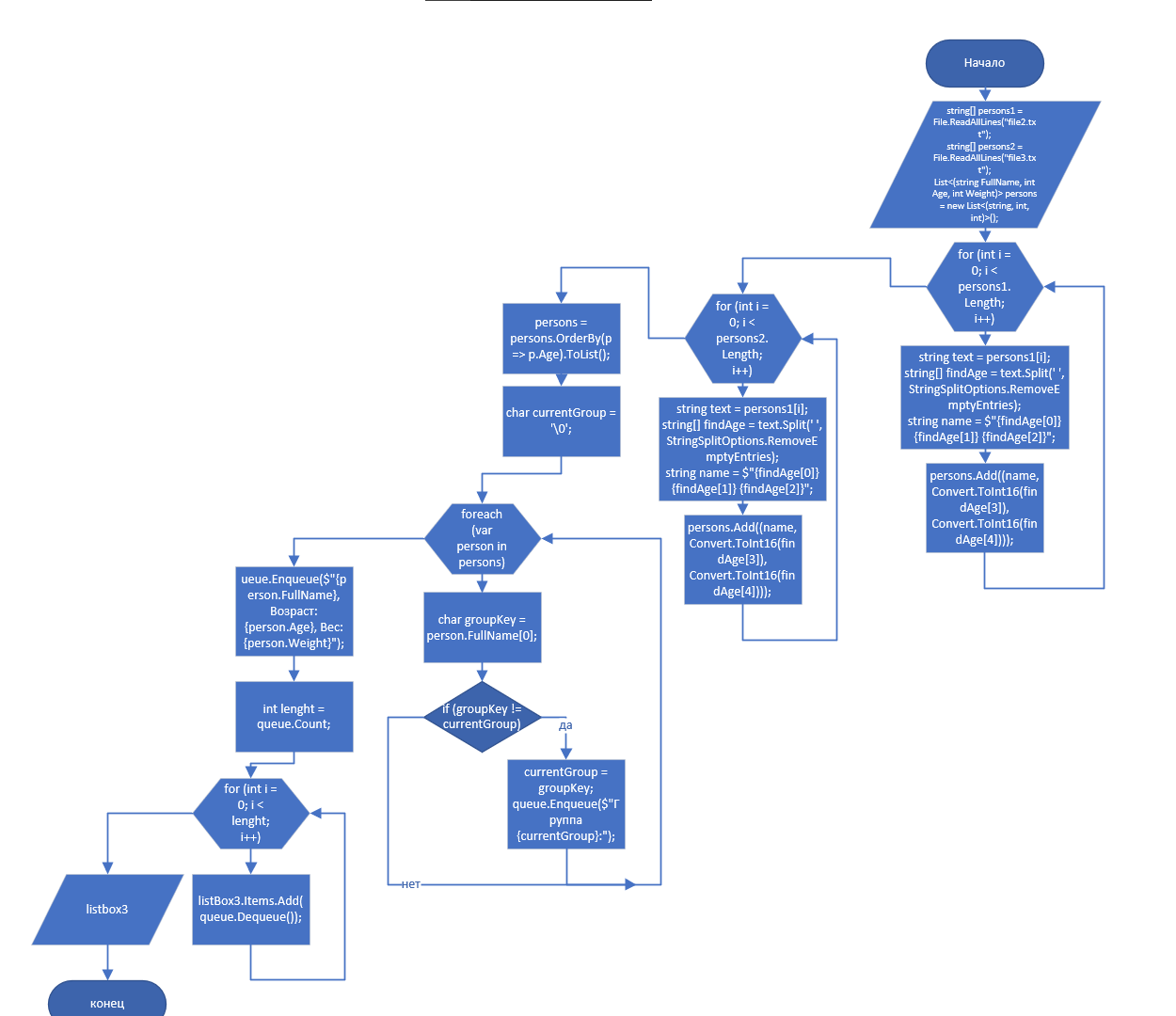
Persons1 – string[] – Считанный из файла массив строк

Persons2 – string[] – Считанный из файла массив строк

**Выходные данные**

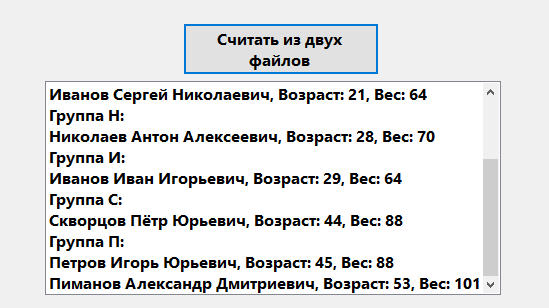
Queue – Очередь из людей выводится в listBox

**Блок схема**

****

**Тестовые ситуации**

**Результат**

****

**Листинг программы**

listBox3.Items.Clear();

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

string[] persons1 = File.ReadAllLines("file2.txt");

string[] persons2 = File.ReadAllLines("file3.txt");

List<(string FullName, int Age, int Weight)> persons = new List<(string, int, int)>();

for (int i = 0; i < persons1.Length; i++)

{

string text = persons1[i];

string[] findAge = text.Split(' ', StringSplitOptions.RemoveEmptyEntries);

string name = $"{findAge[0]} {findAge[1]} {findAge[2]}";

persons.Add((name, Convert.ToInt16(findAge[3]), Convert.ToInt16(findAge[4])));

}

for (int i = 0; i < persons2.Length; i++)

{

string text = persons2[i];

string[] findAge = text.Split(' ', StringSplitOptions.RemoveEmptyEntries);

string name = $"{findAge[0]} {findAge[1]} {findAge[2]}";

persons.Add((name, Convert.ToInt16(findAge[3]), Convert.ToInt16(findAge[4])));

}

persons = persons.OrderBy(p => p.Age).ToList();

char currentGroup = '\0';

foreach (var person in persons)

{

char groupKey = person.FullName[0];

if (groupKey != currentGroup)

{

currentGroup = groupKey;

queue.Enqueue($"Группа {currentGroup}:");

}

queue.Enqueue($"{person.FullName}, Возраст: {person.Age}, Вес: {person.Weight}");

}

int lenght = queue.Count;

for (int i = 0; i < lenght; i++)

{

listBox3.Items.Add(queue.Dequeue());

}

**Вывод**

**Я изучил коллекции Stack и Queue**