Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

# Отчёт по программе «.«Коллекции c#. Собственные методы для работы

# с коллекциями»»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание**

Для заданного значения n запишем в стек все числа от 1 до n, а затем

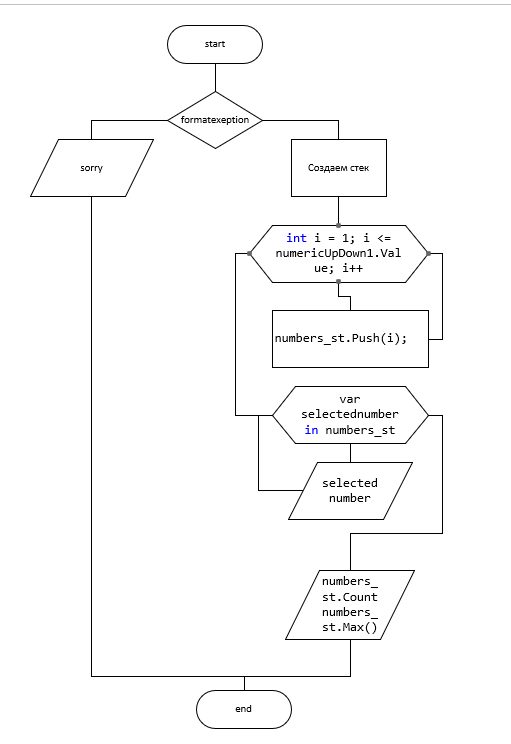
извлечем из стека.

**Входные и выходные данные**

Значение numericupdown1

Stack<int>numbers\_s -выводятся элементы стека

**Блок-схема**

****

**Листинг программы**

try

{

Stack<int> numbers\_st = new Stack<int>();

for (int i = 1; i <= numericUpDown1.Value; i++)

{

numbers\_st.Push(i);

}

listBox1.Items.Add("elements");

foreach (var selectednumber in numbers\_st)

{

listBox1.Items.Add(selectednumber);

}

listBox1.Items.Add($"Размер стека:{numbers\_st.Count}\n Max el:{numbers\_st.Max()}");

}

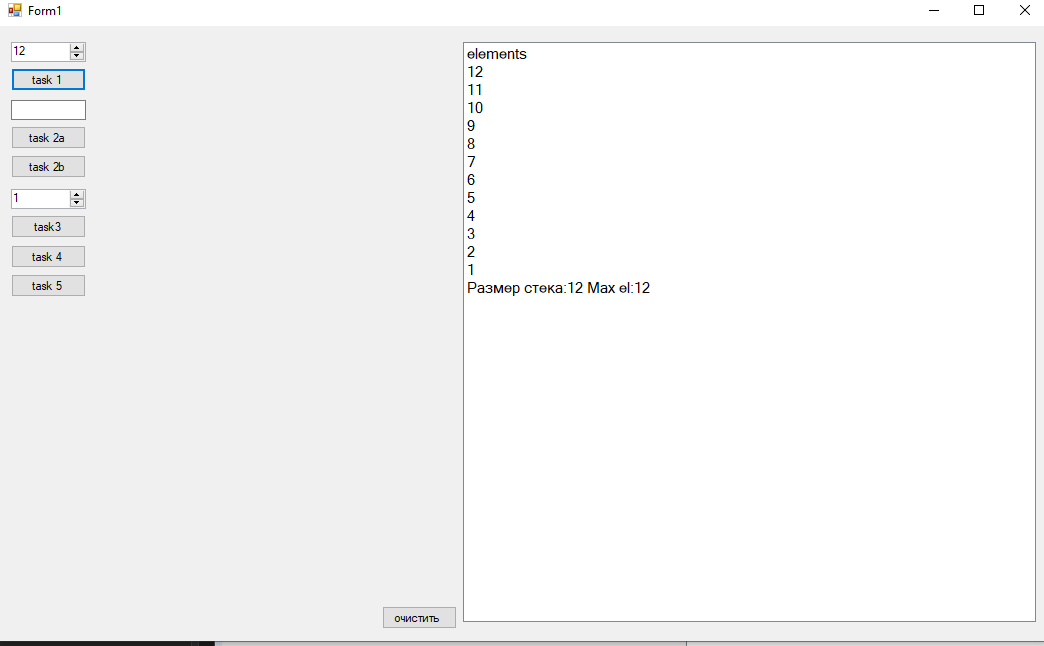
catch (FormatException)

{

MessageBox.Show("Проверьте корретность введеных вами данных");

}

**Тестовые ситуации**

****

**Задача**

Пользователь вводит математическое выражение, которое записывается в

текстовый файл. - t.txt

А) Проверить баланс круглых скобок в данном выражении. Выдать

информацию о балансе на экран.

скобки сбалансированы

Возможно лишняя ( скобка в позиции:5

Б) Если скобок не хватает, то добавить недостающие скобки, если скобки

лишние, то удалить лишние скобки. Записать новое математическое

выражение в файл t1.txt

**Входные и выходные данные**

А)

Строка texbox1

Б)

Строка texbox1

**Выходные данные**

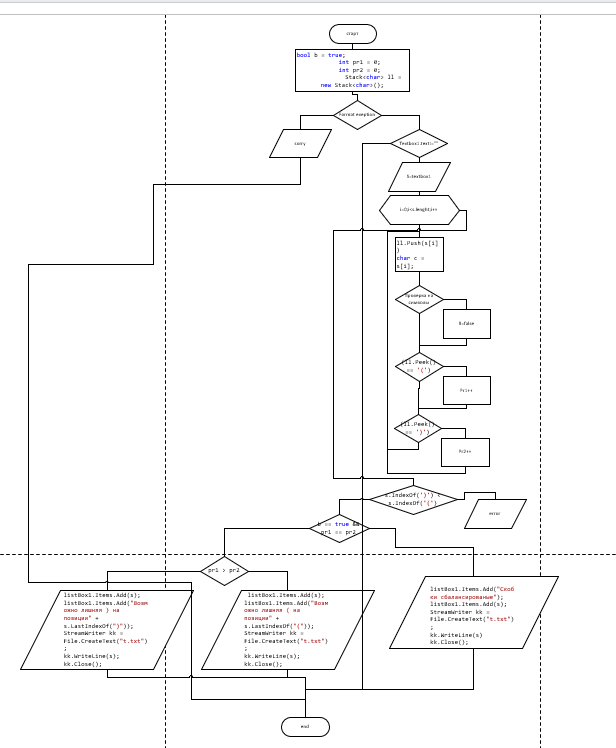
А)

Строка в listbox1, текстовый файл t.txt

Б)

Строка в listbox1, текстовый файл t1.txt

**Блок-схема**

****

**Листинг программы**

А)

bool b = true;

int pr1 = 0;

int pr2 = 0;

Stack<char> ll = new Stack<char>();

try

{

if (textBox1.Text != "")

{

string s = textBox1.Text;

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

{

ll.Push(s[i]);

char c = s[i];

if (ll.Peek() != '0' && ll.Peek() != '1' && ll.Peek() != '2' && ll.Peek() != '3' && ll.Peek() != '4' && ll.Peek() != '5' && ll.Peek() != '6' && ll.Peek() != '7' && ll.Peek() != '8' && ll.Peek() != '9' && ll.Peek() != '/' && ll.Peek() != '\*' && ll.Peek() != '-' && ll.Peek() != '+' && ll.Peek() != '.' && ll.Peek() != '(' && ll.Peek() != ')')

b = false;

if ((ll.Peek() == '('))

pr1++;

if (ll.Peek() == ')')

pr2++;

}

if (s.IndexOf(')') < s.IndexOf('('))

{

MessageBox.Show("Ваши скобки стоят не правильно");

}

else

{

if (b == true && pr1 == pr2)

{

listBox1.Items.Add("Скобки сбалансированые");

listBox1.Items.Add(s);

StreamWriter kk = File.CreateText("t.txt");

kk.WriteLine(s);

kk.Close();

}

else if (pr1 > pr2)

{

listBox1.Items.Add(s);

listBox1.Items.Add("Возможно лишняя ( на позиции" + s.LastIndexOf("("));

StreamWriter kk = File.CreateText("t.txt");

kk.WriteLine(s);

kk.Close();

}

else if (pr1 < pr2)

{

listBox1.Items.Add(s);

listBox1.Items.Add("Возможно лишняя ) на позиции" + s.LastIndexOf(")"));

StreamWriter kk = File.CreateText("t.txt");

kk.WriteLine(s);

kk.Close();

}

}

}

}

catch (FormatException)

{

MessageBox.Show("Проверьте корретность введеных вами данных");

}

Б)

Stack<char> ll = new Stack<char>();

try

{

if (textBox1.Text != "")

{

string s = textBox1.Text;

foreach(char c in s)

{

if (c == '(')

{

ll.Push(c);

}

else if (c == ')')

{

if (ll.Count > 0 && ll.Peek() == '(') ll.Pop();

else ll.Push(c);

}

}

if (s.IndexOf(')') < s.IndexOf('('))

{

MessageBox.Show("Ваши скобки стоят не правильно");

}

else

{

string new\_s = s;

while (ll.Count > 0)

{

if (ll.Peek() == '(')

{

new\_s += ')';

ll.Pop();

}

else

{

new\_s = new\_s.Remove(new\_s.LastIndexOf(')'), 1);

ll.Pop();

}

}

File.WriteAllText("t1.txt", new\_s);

listBox1.Items.Add(new\_s);

}

}

}

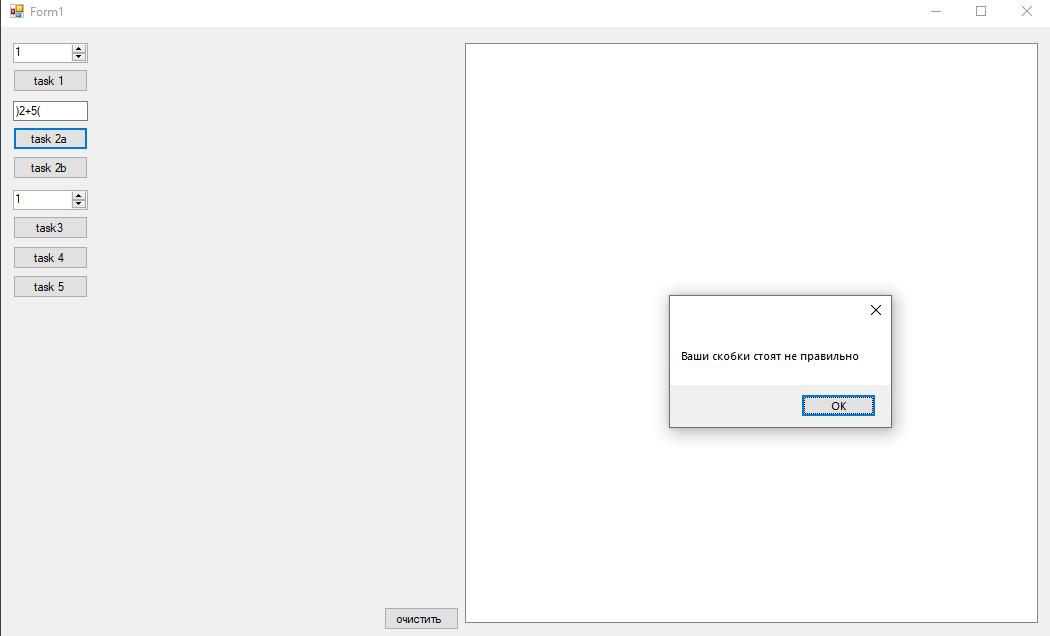
catch (FormatException)

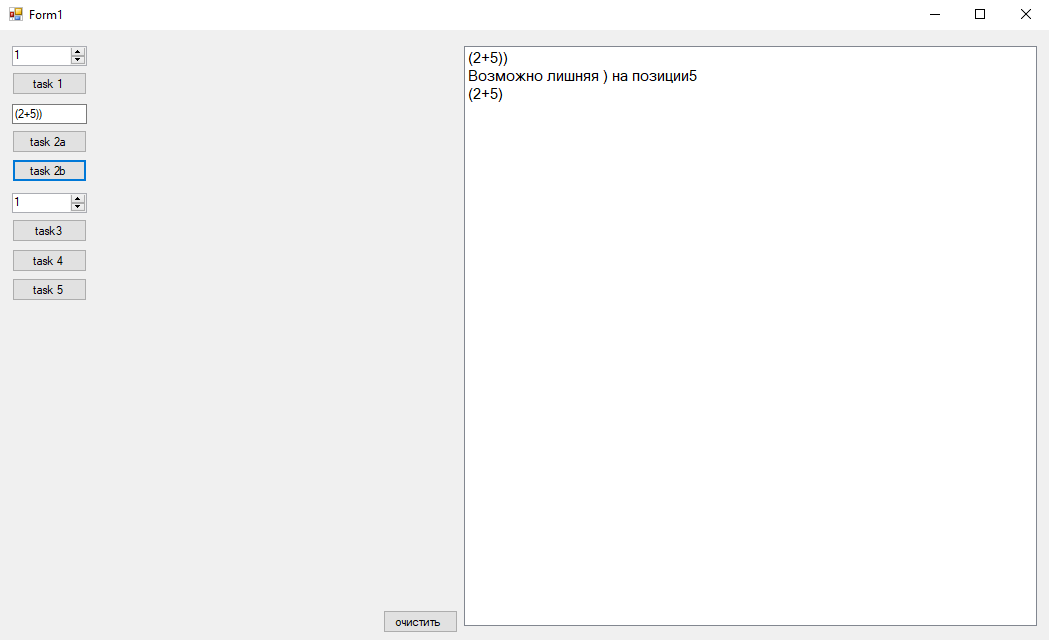
{

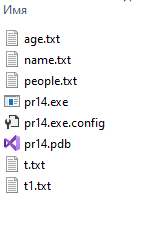
MessageBox.Show("Проверьте корретность введеных вами данных");

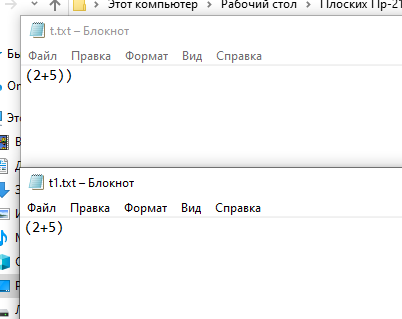
}

**Тестовые ситуации**

****

****

****

****

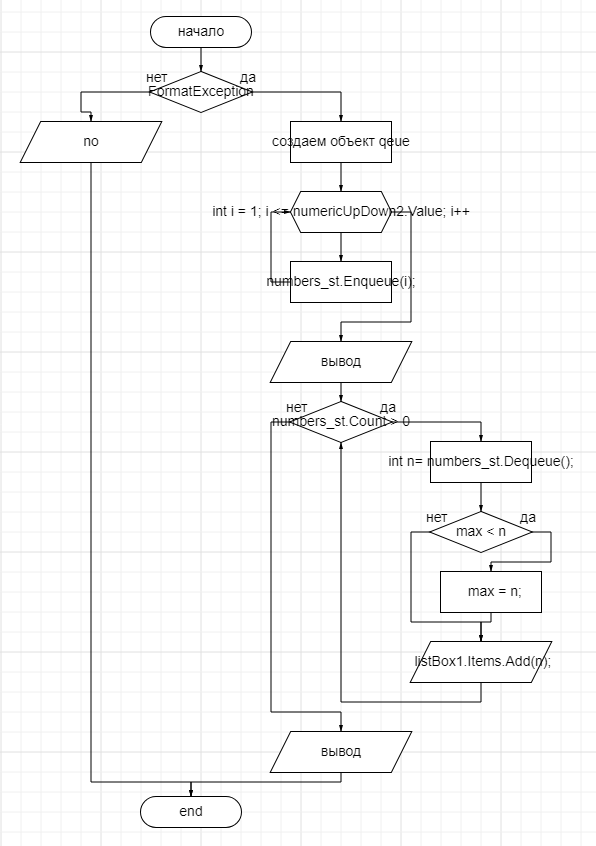
**Задача**

**Входные выходные данные**

Значение numericupdown1

Queue<int> numbers\_st-выводятся элементы Queue

**Блок-схема**

****

**Листинг программы**

try

{

Queue<int> numbers\_st = new Queue<int>();

for (int i = 1; i <= numericUpDown2.Value; i++)

{

numbers\_st.Enqueue(i);

}

listBox1.Items.Add("elements");

int max = 0;

listBox1.Items.Add($"Размер стека:{numbers\_st.Count}\n ");

while (numbers\_st.Count > 0)

{

int n= numbers\_st.Dequeue();

if (max < n)

{

max = n;

}

listBox1.Items.Add(n);

}

listBox1.Items.Add($"Max el:{max}\n ");

}

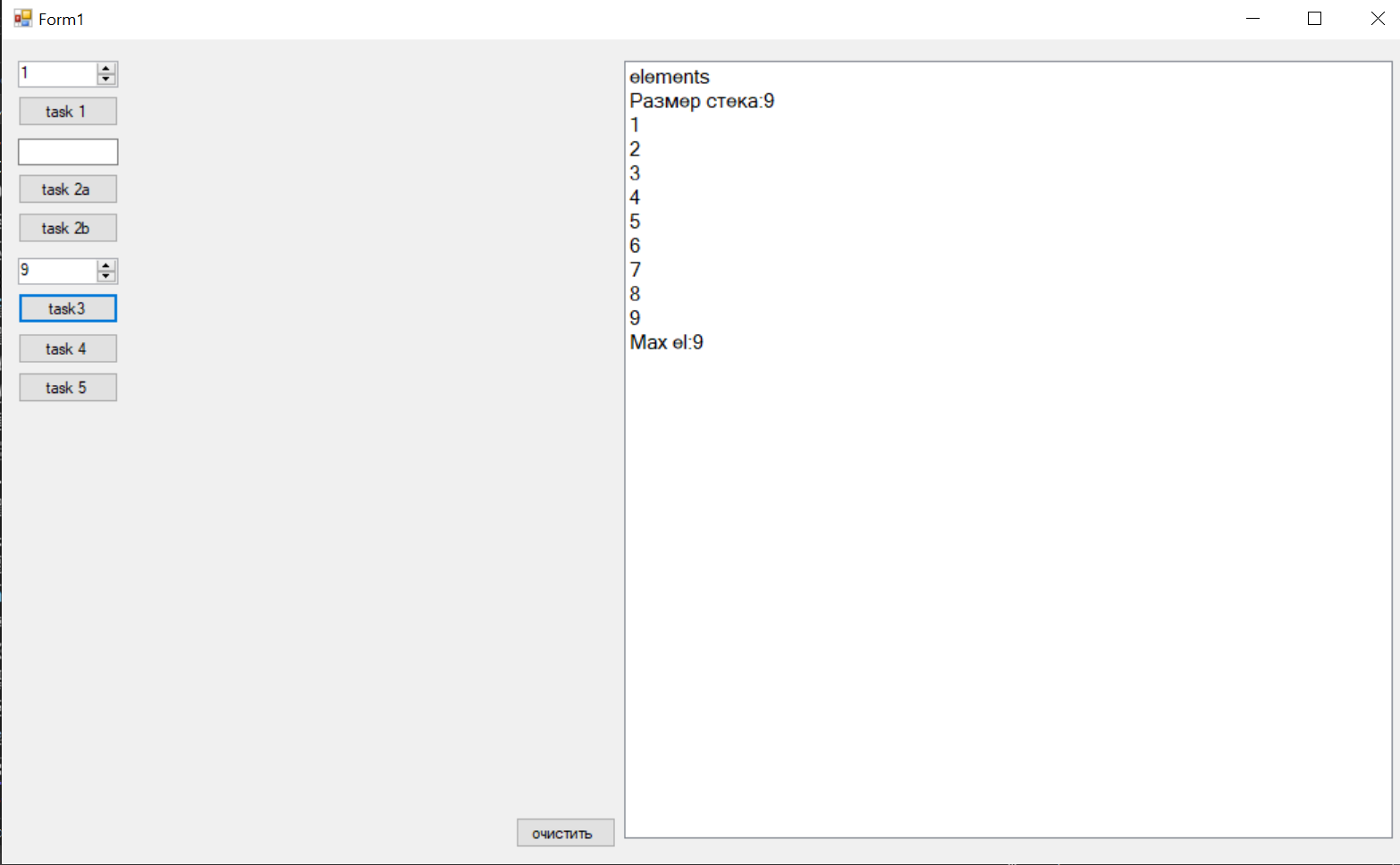
catch (FormatException)

{

MessageBox.Show("Проверьте корретность введеных вами данных");

}

**Тестовые ситуации**

****

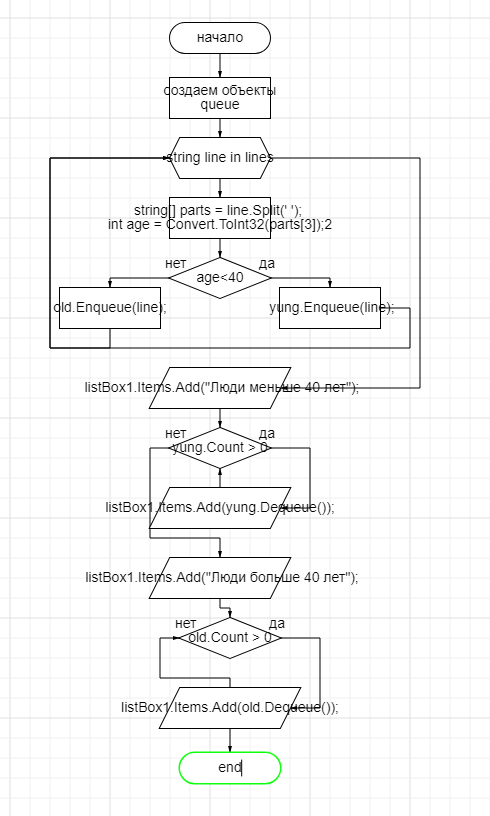
**Задача**

**Входные и выходные данные**

Фаил

Queue-выводятся элементы Queue

**Блок-схема**

****

**Листинг программы**

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

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

string[] lines = File.ReadAllLines("people.txt");

foreach(string line in lines)

{

string[] parts = line.Split(' ');

int age = Convert.ToInt32(parts[3]);

if (age < 40)

{

yung.Enqueue(line);

}

else

{

old.Enqueue(line);

}

}

listBox1.Items.Add("Люди меньше 40 лет");

while (yung.Count > 0)

{

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

}

listBox1.Items.Add("Люди больше 40 лет");

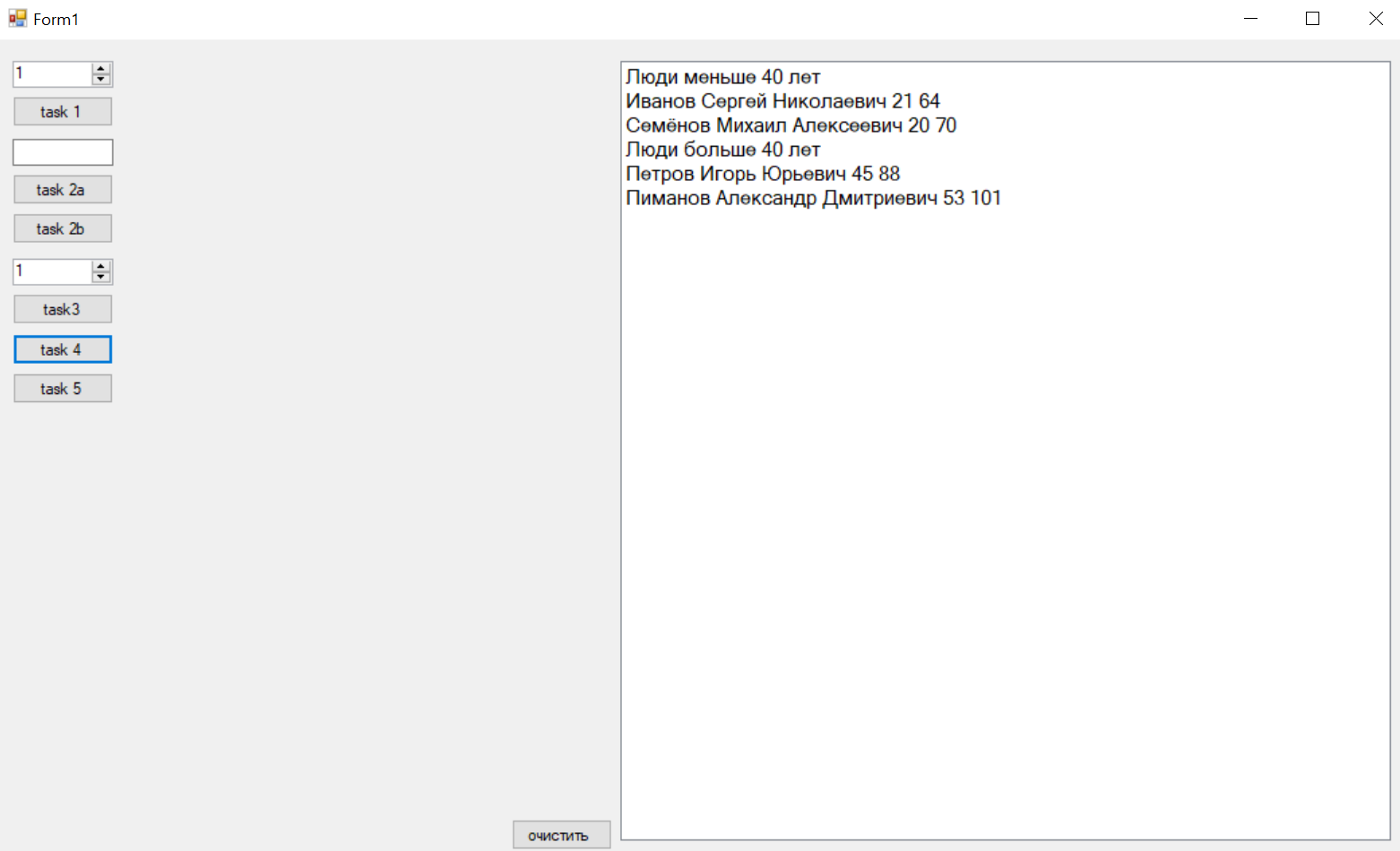
while (old.Count > 0)

{

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

}

**Тестовые ситуации**

****

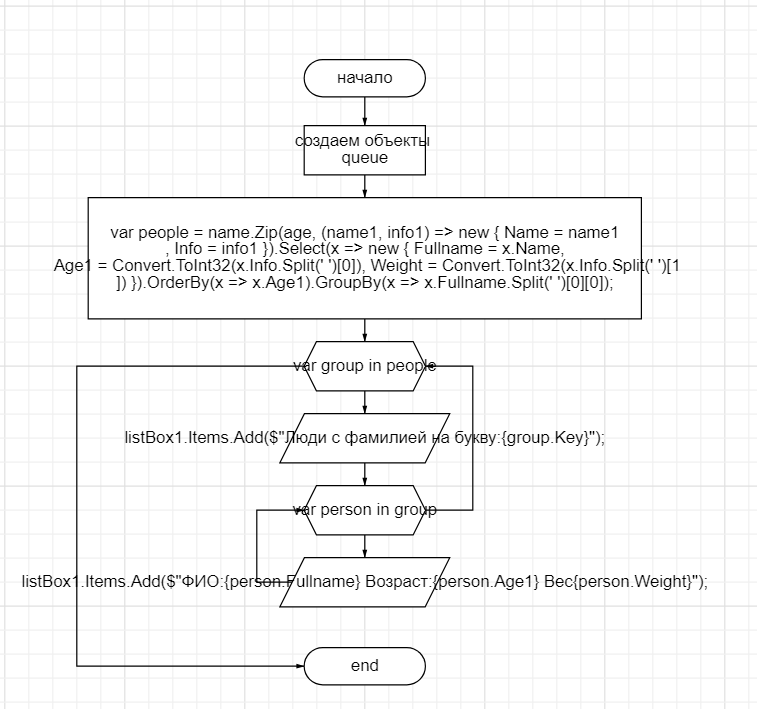
**Задача**

**Входные и выходные данные**

2 файла

Объект Linq

**Блок-схема**

****

**Листинг программы**

Queue<string> name = new Queue<string>(File.ReadAllLines("name.txt"));

Queue<string> age = new Queue<string>(File.ReadAllLines("age.txt"));

var people = name.Zip(age, (name1, info1) => new { Name = name1, Info = info1 }).Select(x => new { Fullname = x.Name, Age1 = Convert.ToInt32(x.Info.Split(' ')[0]), Weight = Convert.ToInt32(x.Info.Split(' ')[1]) }).OrderBy(x => x.Age1).GroupBy(x => x.Fullname.Split(' ')[0][0]);

foreach(var group in people)

{

listBox1.Items.Add($"Люди с фамилией на букву:{group.Key}");

foreach (var person in group)

{

listBox1.Items.Add($"ФИО:{person.Fullname} Возраст:{person.Age1} Вес{person.Weight}");

}

}

**Тестовые ситуации**

****