Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

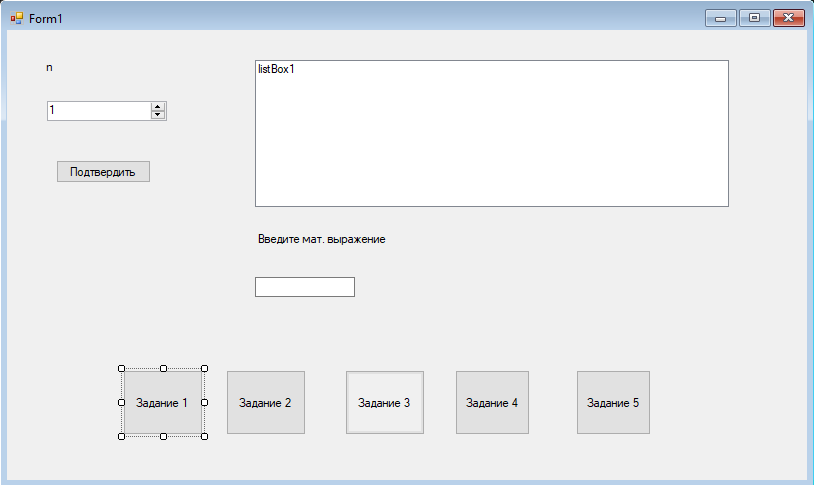
# Отчёт по программе «Практическое занятие 14 (3-5 задания)»

Выполнил: Азаренко Михаил Владимирович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Интерфейс формы:** ****

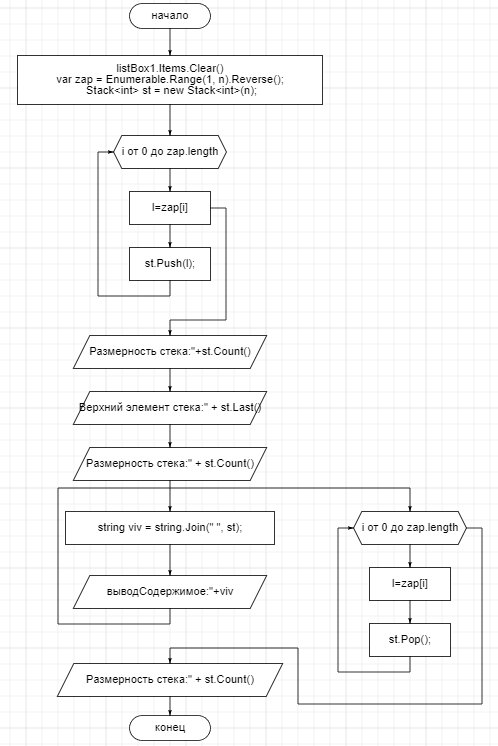
**1 задание:**

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

извлечем из стека.

**Входные данные**:

Int n-входное число

**Блок-схема**  ****

**Листинг программы (если есть)**

listBox1.Items.Clear();

var zap = Enumerable.Range(1, n).Reverse();

Stack<int> st = new Stack<int>(n);

foreach (var l in zap)

{

st.Push(l);

}

listBox1.Items.Add("Размерность стека:"+st.Count());

listBox1.Items.Add("Верхний элемент стека:" + st.Last());

listBox1.Items.Add("Размерность стека:" + st.Count());

string viv = string.Join(" ", st);

listBox1.Items.Add("Содержимое:"+viv);

foreach (int l in zap)

{

st.Pop();

}

listBox1.Items.Add("Размерность стека:" + st.Count());

**Тестовые ситуации: **

**Задание:**

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

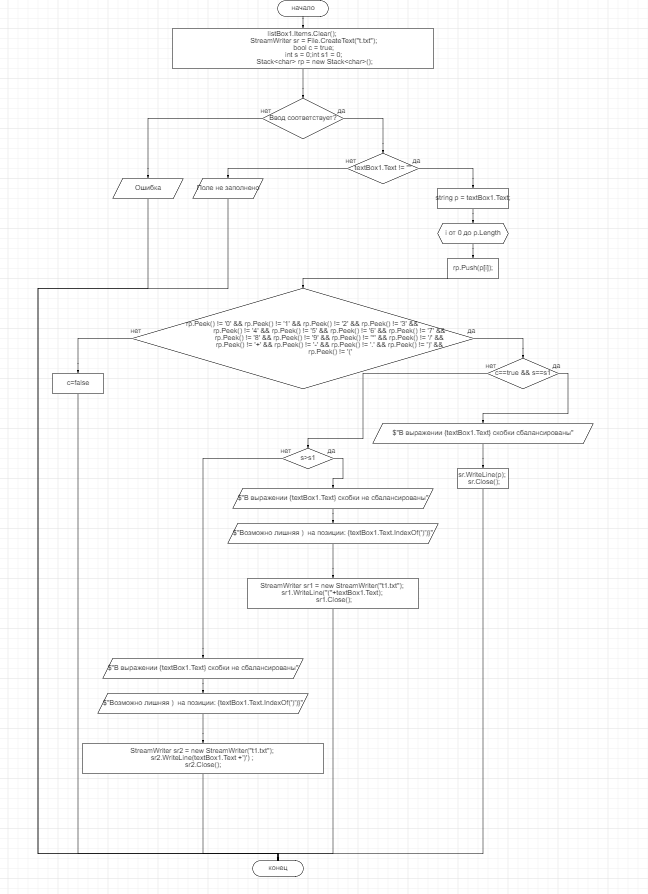
лишние, то удалить лишние скобки. Записать новое математическое выражение в файл t1.txt

**Входные и выходные данные**

**Вход**

String p-входное выражение

**Блок-схема**

****

**Листинг программы (если есть)**

listBox1.Items.Clear();

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

bool c = true;

int s = 0;int s1 = 0;

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

try

{

if (textBox1.Text != "")

{

string p = textBox1.Text;

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

{

rp.Push(p[i]);

if (rp.Peek() != '0' && rp.Peek() != '1' && rp.Peek() != '2' && rp.Peek() != '3' &&

rp.Peek() != '4' && rp.Peek() != '5' && rp.Peek() != '6' && rp.Peek() != '7' &&

rp.Peek() != '8' && rp.Peek() != '9' && rp.Peek() != '\*' && rp.Peek() != '/' &&

rp.Peek() != '+' && rp.Peek() != '-' && rp.Peek() != '.' && rp.Peek() != ')' &&

rp.Peek() != '(')

c = false;

if (rp.Peek() == ')') s++;

if (rp.Peek() == '(') s1++;

}

if(c==true && s==s1)

{

listBox1.Items.Add($"В выражении {textBox1.Text} скобки сбалансированы");

sr.WriteLine(p);

sr.Close();

}

else if(s>s1)

{

listBox1.Items.Add($"В выражении {textBox1.Text} скобки не сбалансированы");

listBox1.Items.Add($"Возможно лишняя ) на позиции: {textBox1.Text.IndexOf(')')}");

StreamWriter sr1 = new StreamWriter("t1.txt");

sr1.WriteLine("("+textBox1.Text);

sr1.Close();

}

else if (s < s1)

{

listBox1.Items.Add($"В выражении {textBox1.Text} скобки не сбалансированы");

listBox1.Items.Add($"Возможно лишняя ( на позиции: {textBox1.Text.IndexOf('(')}");

StreamWriter sr2 = new StreamWriter("t1.txt");

sr2.WriteLine(textBox1.Text +')') ;

sr2.Close();

}

}else

MessageBox.Show("Поле не заполнено");

}

catch(FormatException) { MessageBox.Show("Ошибка"); }

**Тестовые ситуации: **

**Задание:**

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

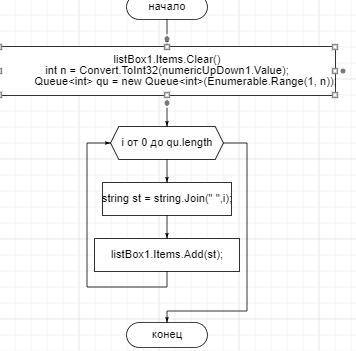
извлечем их из очереди.

**Входные и выходные данные**

**Вход**

Int n-входное число

**Блок-схема**

****

**Листинг программы (если есть)**

listBox1.Items.Clear();

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

Queue<int> qu = new Queue<int>(Enumerable.Range(1, n));

foreach(int i in qu)

{

string st = string.Join(" ",i);

listBox1.Items.Add(st);

}

**Тестовые ситуации: **

**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**

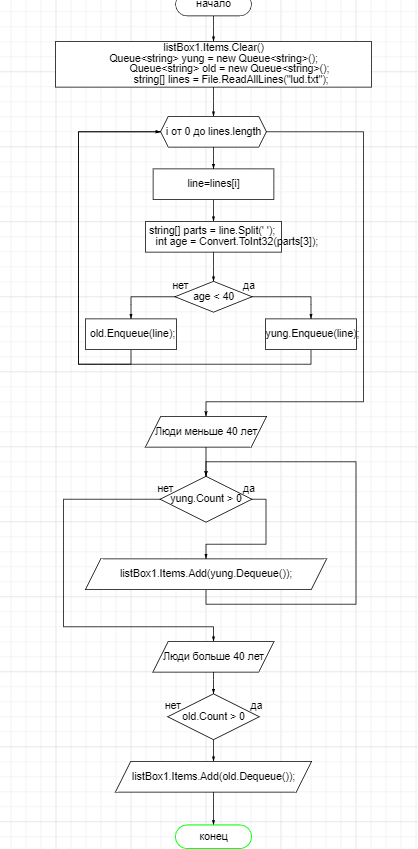
**Вход**

lud.txt-файл с входными данными

**Выход**

Price(int, цена товара), kol(int, количество товара), money(double, количество денег), st(double, цена покупки), edizm(string, валюта), vid(string, вид товара), name(string, название товара)

**Блок-схема**

****

**Листинг программы (если есть)**

listBox1.Items.Clear();

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

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

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

}

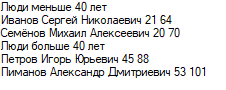
listBox1.Items.Add("Люди больше 40 лет");

while (old.Count > 0)

{

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

}

**Тестовые ситуации:** ****

**Задание:**

В 2-х текстовых файлах записана информация о людях (фамилия, имя,

отчество через пробел) и(возраст, вес через пробел). Вывести на экран

информацию о людях, отсортированную по возрасту, с группировкой по

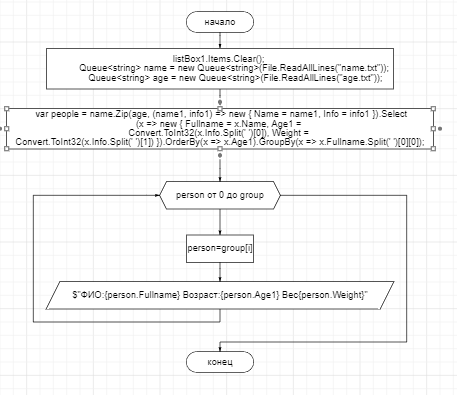
первой букве фамилии.

**Вход**

name.txt-файл с входными данными

age.txt-файл с входными данными

**Блок-схема**

****

**Листинг программы (если есть)**

listBox1.Items.Clear();

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}");

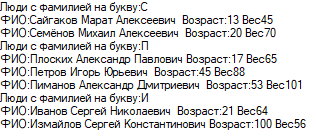
foreach (var person in group)

{

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

}

}

**Тестовые ситуации:** ****