Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Практическое занятие 14 »

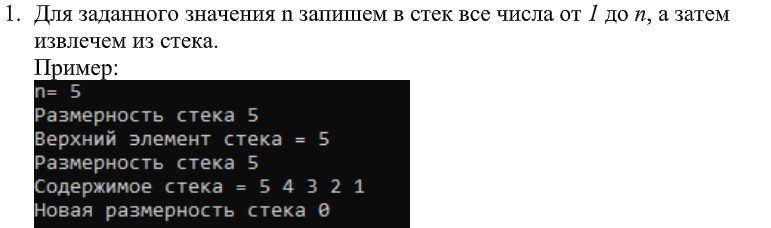
Выполнила: Зиновьева Кристина Дмитриевна

Группа: ПР-24

Преподаватель: Мирошниченко Г.В

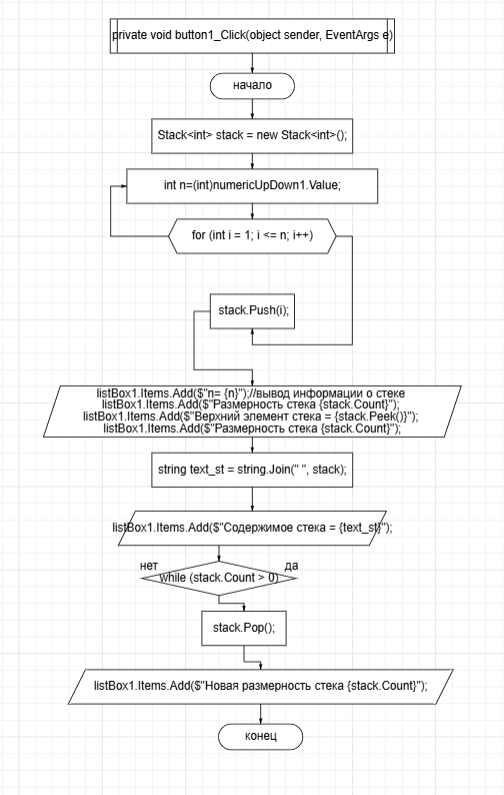
2025

**Задание:**

**Входные и выходные данные**

n-целое(int)

**Блоксхема**

****

**Код программы**

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

{

listBox1.Items.Clear();//очищаем

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

int n=(int)numericUpDown1.Value;//верхний элемент и размерность

for (int i = 1; i <= n; i++)//заполняем стек

{

stack.Push(i);

}

listBox1.Items.Add($"n= {n}");//вывод информации о стеке

listBox1.Items.Add($"Размерность стека {stack.Count}");

listBox1.Items.Add($"Верхний элемент стека = {stack.Peek()}");

listBox1.Items.Add($"Размерность стека {stack.Count}");

string text\_st = string.Join(" ", stack);//формируем строку с содержимым стека (в порядке извлечения)

listBox1.Items.Add($"Содержимое стека = {text\_st}");

while (stack.Count > 0)//извлекаем все элементы из стека

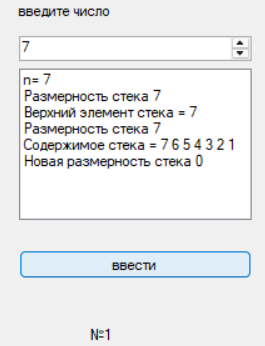
{

stack.Pop();//Возвращает элемент, расположенный в вершине стека, и извлекает его из стека

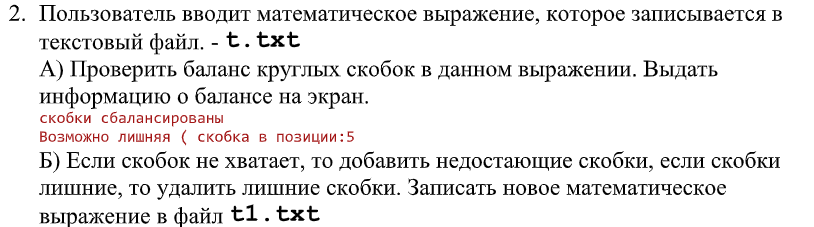
}

listBox1.Items.Add($"Новая размерность стека {stack.Count}");

**Листинг программы (если есть)**

****

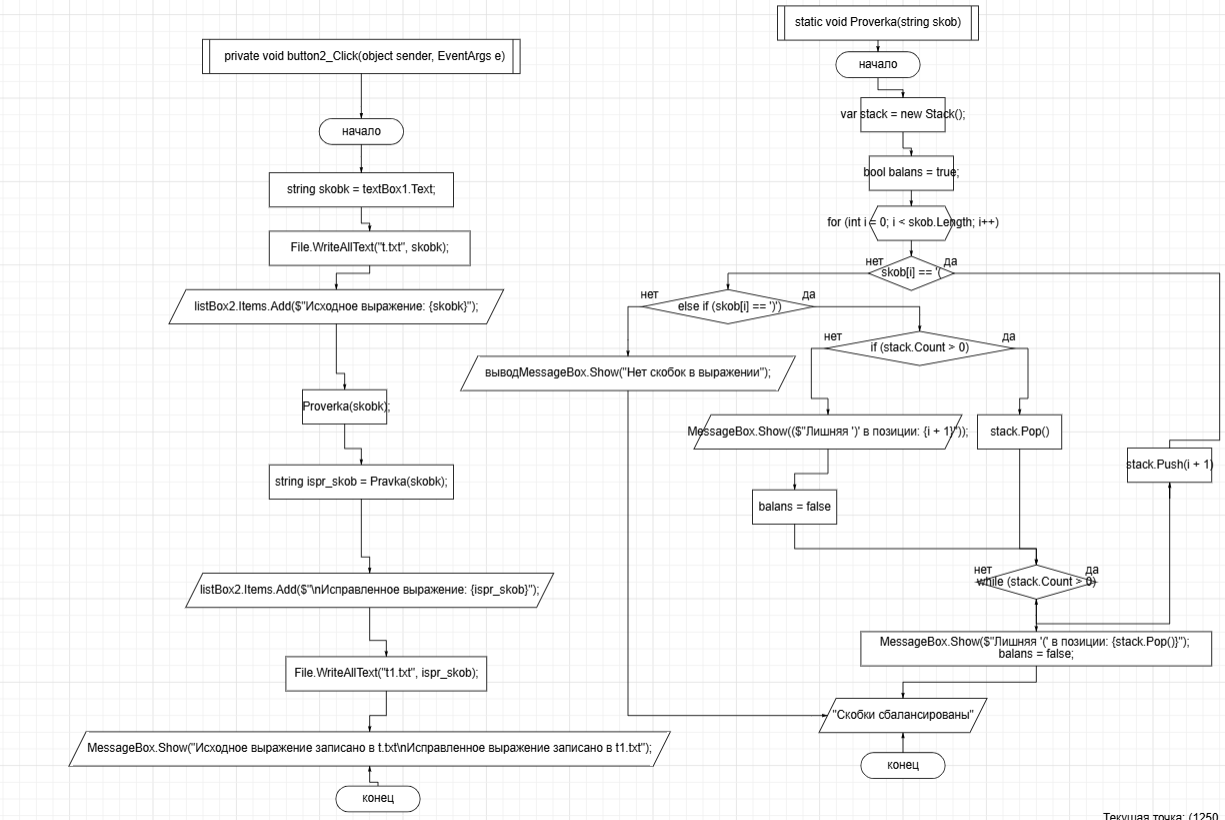
**Задание:**

**Входные и выходные данные**

string skobk-выражение вводит пользователь

ispr\_skob-исправленное выражение

**Блоксхема**

****

**Код программы**

private void button2\_Click(object sender, EventArgs e)

{

string skobk = textBox1.Text;

File.WriteAllText("t.txt", skobk);//записываем исходное выражение в файл t.txt

listBox2.Items.Add($"Исходное выражение: {skobk}");

Proverka(skobk); //проверка баланса скобок

string ispr\_skob = Pravka(skobk); // Исправление выражения

listBox2.Items.Add($"\nИсправленное выражение: {ispr\_skob}");

File.WriteAllText("t1.txt", ispr\_skob);

MessageBox.Show("Исходное выражение записано в t.txt\nИсправленное выражение записано в t1.txt");

}

static void Proverka(string skob)

{

var stack = new Stack();

bool balans = true;//правильность расстановки скобок

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

{

if (skob[i] == '(') stack.Push(i + 1);

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

{

if (stack.Count > 0) stack.Pop();

else

{

MessageBox.Show(($"Лишняя ')' в позиции: {i + 1}"));

balans = false;

}

}

}

while (stack.Count > 0)

{

MessageBox.Show($"Лишняя '(' в позиции: {stack.Pop()}");

balans = false;

}

if (balans) MessageBox.Show("Скобки сбалансированы");

}

static string Pravka(string ind)

{

Stack one\_skob = new Stack();

ArrayList two\_skob = new ArrayList(); //для лишних ')'

for (int i = 0; i < ind.Length; i++)//проходимсяпо всему выражению

{

if (ind[i] == '(')//лишня открывающая

{

one\_skob.Push(i);

}

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

{

if (one\_skob.Count > 0)//лишняя закрывающая

{

one\_skob.Pop();

}

else

{

two\_skob.Add(i);

}

}

}

string result = " ";

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

{

char zn = ind[i];

bool dop\_skob = true;

if (zn == '(' && one\_skob.Contains(i))//содержится ли i

{

dop\_skob = false;

}

if (zn == ')' && two\_skob.Contains(i)) // прверяем на лишнюю закрывающую скобку

{

dop\_skob = false;

}

if (dop\_skob)

{

result += zn;

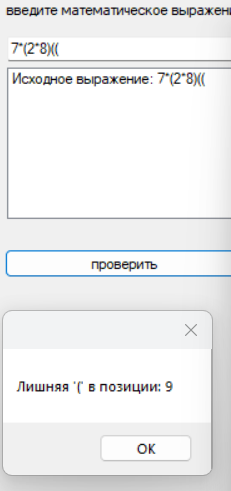
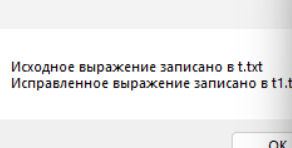
}

}

return result;

}

**Листинг программы (если есть)**

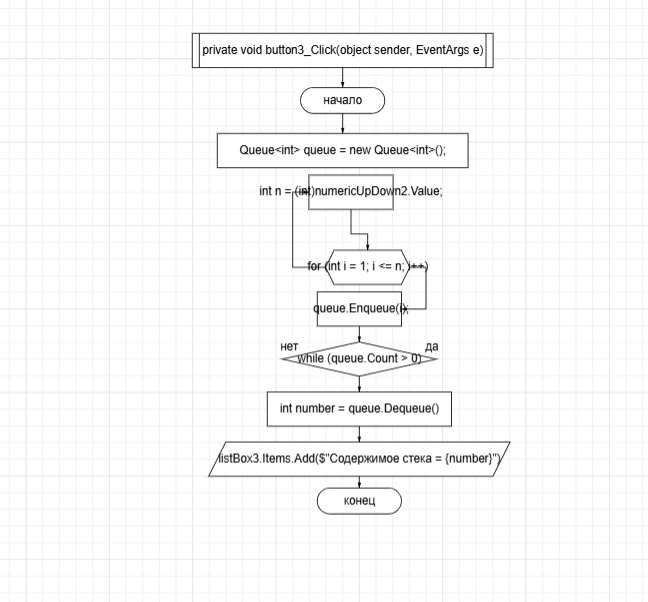
**** ****

**Задание:**

**Входные и выходные данные**

n-целое(int)

**Блоксхема**

****

**Код программы**

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

int n = (int)numericUpDown2.Value;

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

{

queue.Enqueue(i);//Добавляет объект v в конец очереди

}

while (queue.Count > 0)

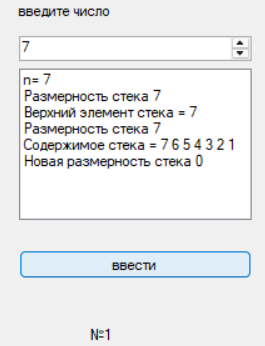
{

int number = queue.Dequeue();//Возвращает объект из начала вызывающей очереди, удаляя его из очереди

listBox3.Items.Add($"Содержимое стека = {number}");

}

**Листинг программы (если есть)**

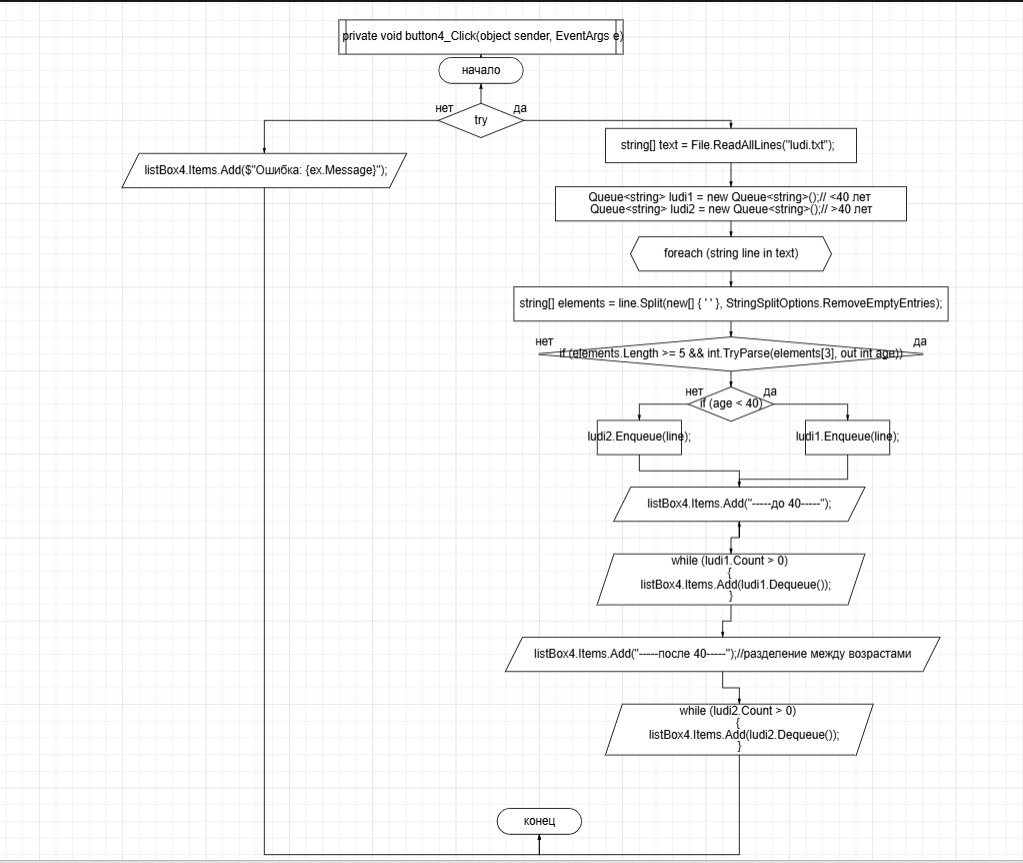
****

**Задание:**

**Входные и выходные данные**

Данные записанные в файле

**Блоксхема**

****

**Код программы**

private void button4\_Click(object sender, EventArgs e)

{

try

{

string[] text = File.ReadAllLines("ludi.txt");

Queue<string> ludi1 = new Queue<string>();// <40 лет

Queue<string> ludi2 = new Queue<string>();// >40 лет

foreach (string line in text)

{

string[] elements = line.Split(new[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

if (elements.Length >= 5 && int.TryParse(elements[3], out int age))//проверяем что строка состоит из 5 частей

{

if (age < 40)

ludi1.Enqueue(line);

else

ludi2.Enqueue(line);

}

}

listBox4.Items.Add("-----до 40-----");

while (ludi1.Count > 0)

{

listBox4.Items.Add(ludi1.Dequeue());

}

listBox4.Items.Add("-----после 40-----");//разделение между возрастами

while (ludi2.Count > 0)

{

listBox4.Items.Add(ludi2.Dequeue());

}

}

catch (Exception ex)

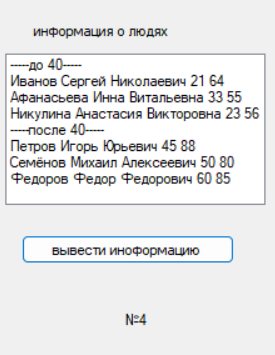
{

listBox4.Items.Add($"Ошибка: {ex.Message}");

}

}

**Листинг программы (если есть)**

****

**Вывод:** **Используя коллекции и методы работы с коллекциями, выполнили задания.**