

Министерство образования и молодежной политики Свердловской
области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»



Отчёт по программе «Практическое занятие 14»

Выполнил: Сагутдинова Илона

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

Задание 1:

1. Для заданного значения n запишем в стек все числа от 1 до n , а затем извлечем из стека.

Пример:

```
n= 5
Размерность стека 5
Верхний элемент стека = 5
Размерность стека 5
Содержимое стека = 5 4 3 2 1
Новая размерность стека 0
```

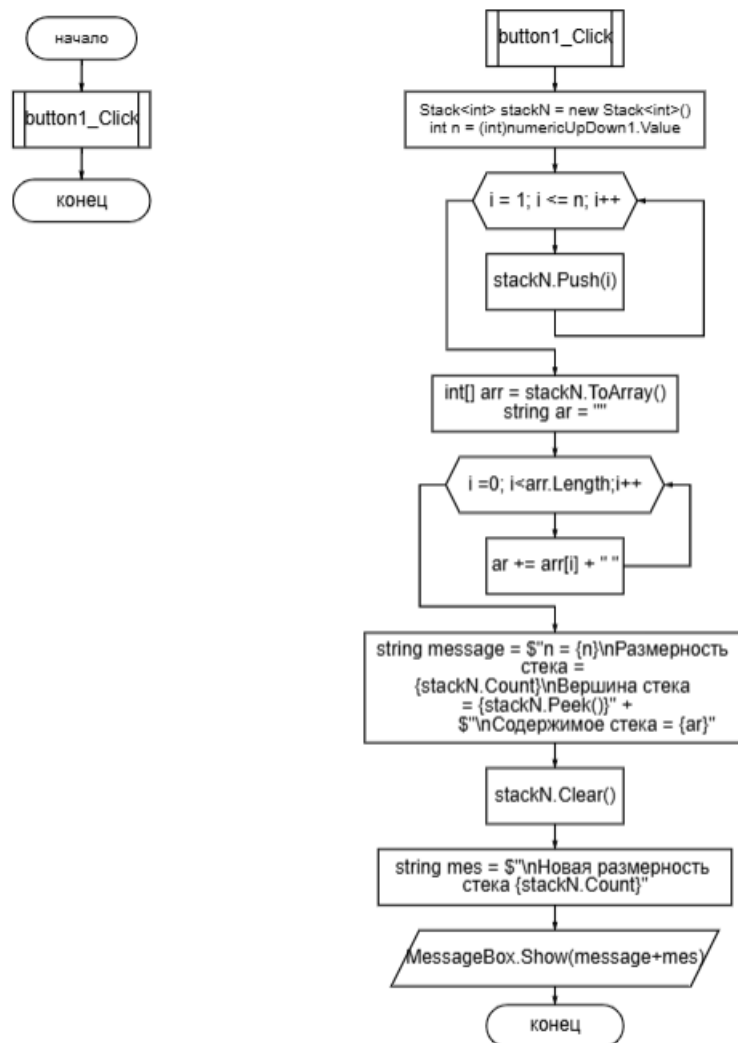
Входные данные:

N - целое число, int.

Выходные данные:

Вывод работы с числом n с помощью стека.

Блок-схема:



Листинг программы:

```
Stack<int> stackN = new Stack<int>();

private void button1_Click(object sender, EventArgs e)
{
    int n = (int)numericUpDown1.Value;

    for (int i = 1; i <= n; i++)
    {
        stackN.Push(i);
    }

    int[] arr = stackN.ToArray();

    string ar = "";

    for (int i = 0; i < arr.Length; i++)
    {
        ar += arr[i] + " ";
    }

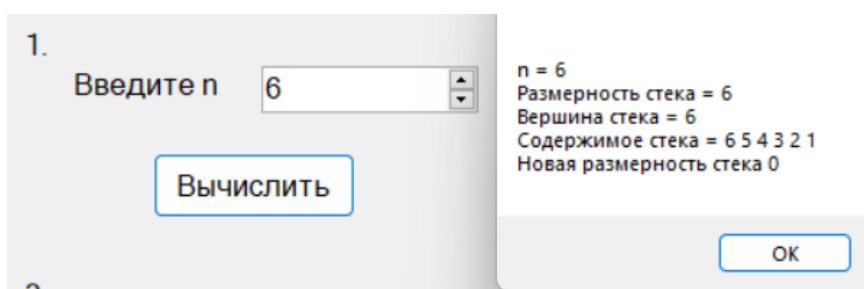
    string message = $"n = {n}\nРазмерность стека = {stackN.Count}\nВершина стека = {stackN.Peek()} " +
        $"Содержимое стека = {ar}";

    stackN.Clear();

    string mes = $"Новая размерность стека {stackN.Count}";

    MessageBox.Show(message+mes);
}
```

Тестовые ситуации:



Задание 2:

2. 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**

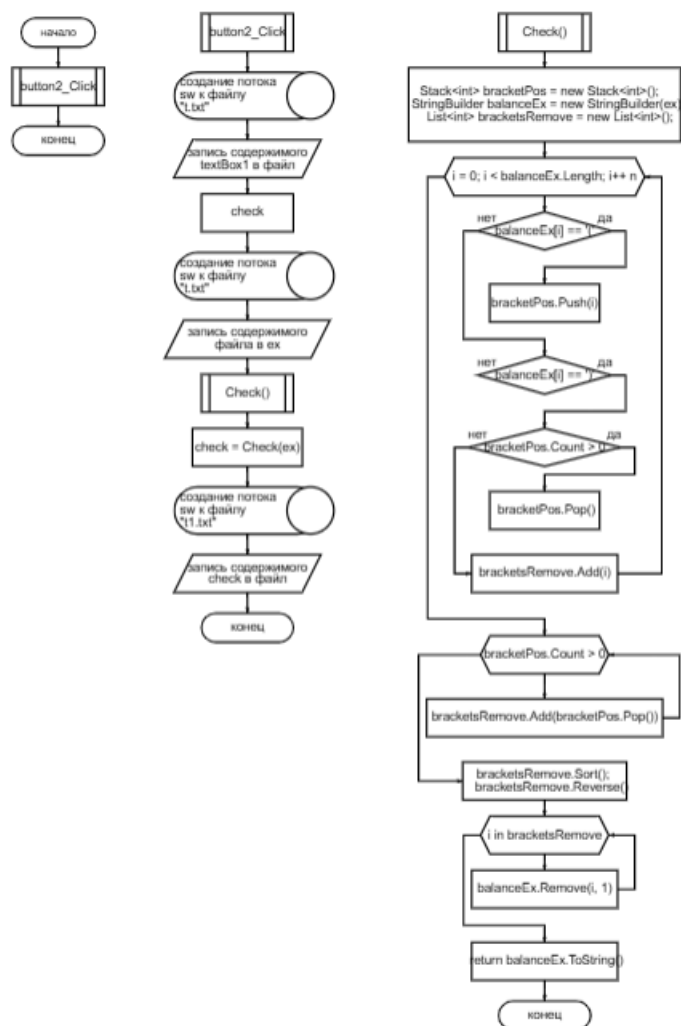
Входные данные:

TextBox1 - содержит математическое выражение, строка, string.

Выходные данные:

Запись в файл исправленного математического выражения.

Блок-схема:



Листинг программы:

```
private void button2_Click(object sender, EventArgs e)
{
    using (StreamWriter sw = File.CreateText("t.txt"))
    {
        sw.WriteLine(textBox1.Text);
    }

    string check;

    using (StreamReader sr = File.OpenText("t.txt"))
    {
        string ex = sr.ReadLine();

        check = Check(ex);
    }

    using (StreamWriter sw = File.CreateText("t1.txt"))
    {
        sw.WriteLine(check);
    }

    MessageBox.Show("Успешно! Проверьте результат в файле t1.txt", "Успешно", MessageBoxButtons.OK);
}

private string Check(string ex)
{
    Stack<int> bracketPos = new Stack<int>();

    StringBuilder balanceEx = new StringBuilder(ex);

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

    for (int i = 0; i < balanceEx.Length; i++)
    {
        if (balanceEx[i] == '(')
            bracketPos.Push(i);

        else if (balanceEx[i] == ')')
        {
            if (bracketPos.Count > 0)
                bracketPos.Pop();

            else
```

```

        bracketsRemove.Add(i);
    }
}
while (bracketPos.Count > 0)
    bracketsRemove.Add(bracketPos.Pop());
bracketsRemove.Sort();
bracketsRemove.Reverse();
foreach (int i in bracketsRemove)
    balanceEx.Remove(i, 1);
return balanceEx.ToString();
}

```

Тестовые ситуации:

2.
Введите выражение

((2+4))

Записать

Успешно! Проверьте результат в файле t1.txt

OK

(2+4)

Задание 3:

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

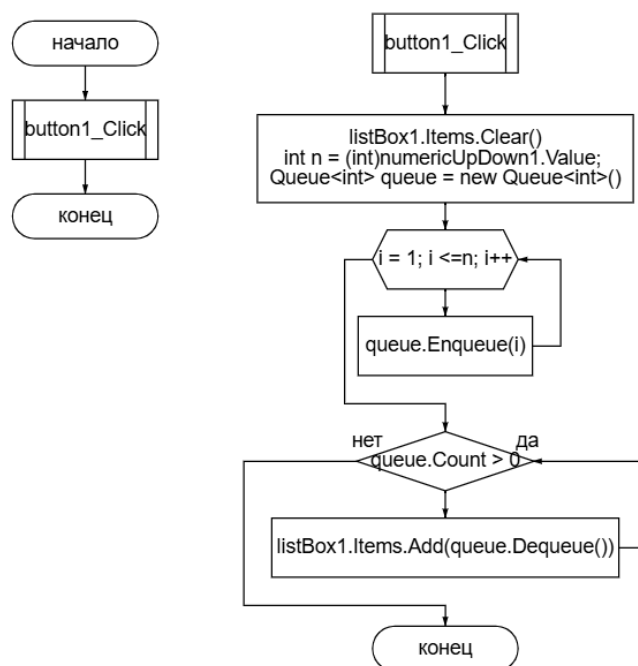
Входные данные:

N - целое число, int.

Выходные данные:

Вывод работы с числом n с помощью queue.

Блок-схема:



Листинг программы:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int n = (int)numericUpDown1.Value;

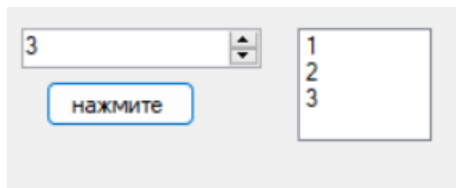
    Queue<int> queue = new Queue<int>();

    for (int i = 1; i <= n; i++)
    {
        queue.Enqueue(i);
    }

    while (queue.Count > 0)
```

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

Тестовые ситуации:



The screenshot shows a Windows application window with a light gray background. On the left, there is a text box containing the number '3'. Below the text box is a button with the text 'нажмите' (click) in Russian. To the right of the text box and button is a list box containing the numbers '1', '2', and '3' stacked vertically. The list box has a vertical scrollbar on its right side.

Задание 4:

4. 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.

Иванов Сергей Николаевич 21 64

Петров Игорь Юрьевич 45 88

Семёнов Михаил Алексеевич 20 70

Пиманов Александр Дмитриевич 53 101

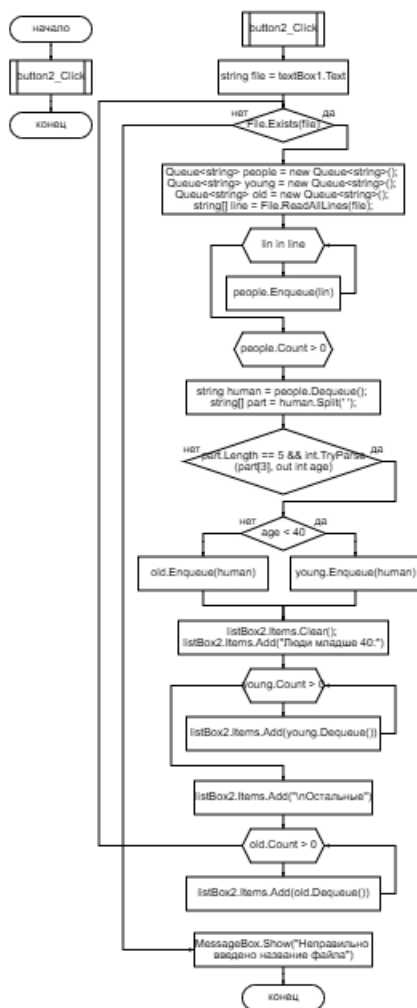
Входные данные:

f.txt - файл в котором записана информация о людях и их возрасте.

Выходные данные:

Вывод на listBox2 сначала людей младше 40 и остальных.

Блок-схема:



Листинг программы:

```

private void button2_Click(object sender, EventArgs e)
{
    string file = textBox1.Text;

    if (File.Exists(file))
    {
        Queue<string> people = new Queue<string>();
        Queue<string> young = new Queue<string>();
        Queue<string> old = new Queue<string>();

        string[] line = File.ReadAllLines(file);

        foreach (string lin in line)
        {
            people.Enqueue(lin);
        }

        while (people.Count > 0)
        {
            string human = people.Dequeue();
            string[] part = human.Split(" ");

            if (part.Length == 5 && int.TryParse(part[3], out int age))
            {
                if (age < 40)
                {
                    young.Enqueue(human);
                }
                else
                {
                    old.Enqueue(human);
                }
            }
        }

        ListBox2.Items.Clear();
        ListBox2.Items.Add("Люди младше 40.");

        while (young.Count > 0)
        {
            ListBox2.Items.Add(young.Dequeue());
        }

        ListBox2.Items.Add("нОстальные");

        while (old.Count > 0)
        {
            ListBox2.Items.Add(old.Dequeue());
        }

        MessageBox.Show("Неправильно  
введено название файла");
    }
}

```

```

string human = people.Dequeue();

string[] part = human.Split(' ');

if (part.Length == 5 && int.TryParse(part[3], out int age))
{
    if (age < 40)
        young.Enqueue(human);
    else
        old.Enqueue(human);
}
}

listBox2.Items.Clear();

listBox2.Items.Add("Люди младше 40:");

while (young.Count > 0)
    listBox2.Items.Add(young.Dequeue());

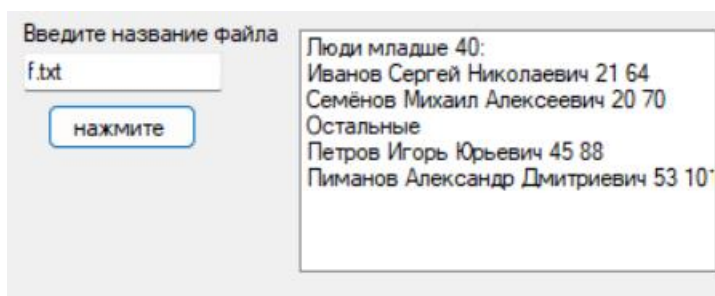
listBox2.Items.Add("\nОстальные");

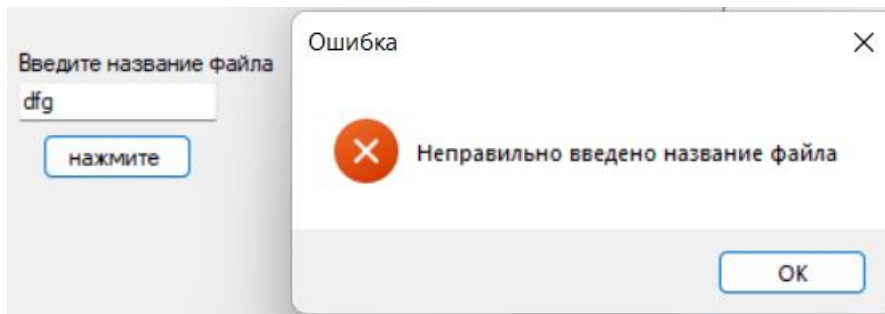
while (old.Count > 0)
    listBox2.Items.Add(old.Dequeue());
}

else
    MessageBox.Show("Неправильно введено название файла", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

Тестовые ситуации:





Задание 5:

5. 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.

Входные данные:

F2.txt - файл в котором записано ФИО людей.

F3.txt - файл в котором записан возраст и вес людей.

Выходные данные:

Вывод на listBox3 отсортированные фамилии людей по первой букве и по возрасту.

Блок-схема:

Листинг программы:

```

private void button3_Click(object sender, EventArgs e)
{
    string file = textBox2.Text;
    string file2 = textBox3.Text;
    if (File.Exists(file) && File.Exists(file2))
    {
        Queue<People> people = new Queue<People>();
        using (StreamReader sw = File.OpenText(file))
        {
            using (StreamReader sr = File.OpenText(file2))
            {
                string name;
                string age;
                while (!sr.EndOfStream && !sw.EndOfStream)
                {
                    name = sw.ReadLine();
                    age = sr.ReadLine();
                    if (string.IsNullOrEmpty(name) || string.IsNullOrEmpty(age))
                    {
                        MessageBox.Show("Один из файлов содержит пустые строки.");
                        break;
                    }
                    string[] part_name = name.Split(' ');
                    string[] part_age = age.Split(' ');
                    if (part_name.Length == 3 && part_age.Length == 2)
                    {
                        try
                        {
                            People peop = new People();
                            peop.SetSurname(part_name[0]);
                            peop.SetName(part_name[1]);
                            peop.SetSurname2(part_name[2]);
                            peop.SetAge(int.Parse(part_age[0]));
                            peop.SetVes(int.Parse(part_age[1]));
                            people.Enqueue(peop);
                        }
                        catch (FormatException)
                        {
                            MessageBox.Show("Неверный формат данных в файле");
                            break;
                        }
                    }
                    else
                    {
                        MessageBox.Show("Неверный формат строк в файле.");
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
}

var group = people
    .Where(peop => peop != null)
    .OrderBy(peop => peop.GetAge())
    .GroupBy(peop => peop.GetName()?.[0].ToString())
    .SelectMany(grou => new[] { $"Группа:{grou.Key}" })
    .Concat(grou.Select(peop => peop.Info()));

listBox3.Items.Clear();
foreach (var item in group)
{
    if (!string.IsNullOrEmpty(item))
    {
        listBox3.Items.Add(item);
    }
}
}
else
    MessageBox.Show("Неправильно введено название файла", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

internal class People

{

private string surname;

private string name;

private string surname2;

private int age;

private int ves;

public string GetSurname ()

```
{  
    return surname;  
}  
  
public void SetSurname(string n)  
{  
    surname = n;  
}  
  
public string GetName()  
{  
    return surname;  
}  
  
public void SetName(string n)  
{  
    name = n;  
}  
  
public string GetSurname2()  
{  
    return surname;  
}  
  
public void SetSurname2(string n)  
{  
    surname2 = n;  
}  
  
public int GetAge()  
{  
    return age;  
}  
  
public void SetAge(int n)  
{  
    age = n;  
}  
  
public int GetVes()  
{  
    return ves;  
}
```

```

    }

    public void SetVes(int n)

    {
        ves = n;
    }

    public string Info()

    {
        return $"{GetSurname()} {GetName()} {GetSurname2()}, Возраст: {GetAge()}, Вес: {GetVes()}";
    }
}

```

Тестовые ситуации:

Введите название файлов

f2.txt

f3.txt

нажмите

Группа:И
Иванов Иванов Иванов, Возраст: 25, Вес: 70
Группа:П
Петров Петров Петров, Возраст: 30, Вес: 80
Группа:С
Сидоров Сидоров Сидоров, Возраст: 35, Вес: 90

Введите название файлов

sdf

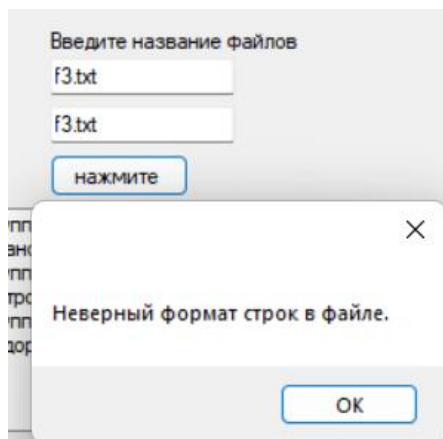
f3.txt

нажмите

Ошибка

Неправильно введено название файла

ОК



Вывод: я изучила коллекции stack и queue.