Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая №14**»

Выполнил: Рогов Андрей Михайлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2025

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

**Входные и выходные данные:**

n – крайнее значение, до которого должны записываться числа, целый.

count – размерность стека, целый;

upEl – верхний элемент стека, целый;

contArr – содержимое стека, строковый массив.

**На форме:**

z1 – “1” – label

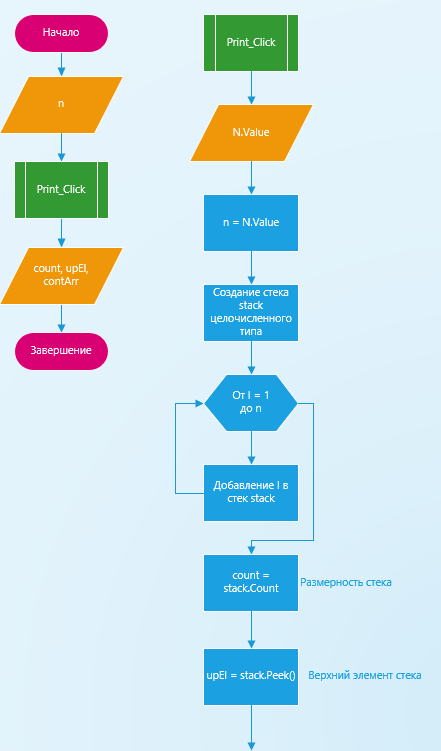
inputN – “Введите n” - label

N – numericUpDown (1, 100)

Print – “Вычислить стек” – button

Result - listBox

**Блок-схема:**



**Листинг программы:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Result.ClearSelected();

int n = (int)N.Value;

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

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

stack.Push(i);

int count = stack.Count;

int upEl = stack.Peek();

Result.Items.Add($"n = {n}");

Result.Items.Add($"Размерность стека = {count}");

Result.Items.Add($"Верхний элемент стека = {upEl}");

int[] arr = stack.ToArray();

string contArr = "";

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

contArr += arr[i] + " ";

Result.Items.Add($"Содержимое стека = {contArr}");

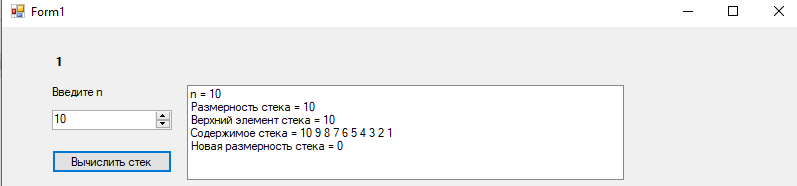
stack.Clear();

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

}

}

**Тестовые ситуации:**



**Задание 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.

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:**

primer – математическое выражение, строковый.

balance – математическое выражение со сбалансированными скобками, строковый.

**На форме:**

z2 – “2” – label

inputEx – “Введите математическое выражение” - label

Example – textBox

Calculate– “Проверить” – button

**Блок-схема:**

**Листинг программы:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

string primer = Example.Text;

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

{

sw.WriteLine(primer);

}

string balance;

using (StreamReader sr = File.OpenText("t.txt"))

{

string ex = sr.ReadLine();

balance = BalanceBrackets(ex);

}

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

{

sw.WriteLine(balance);

}

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

}

private string BalanceBrackets(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)

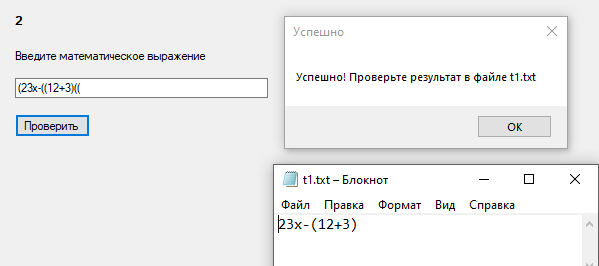
balanceEx.Remove(i, 1);

return balanceEx.ToString();

}

}

**Тестовые ситуации:**



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

**Входные и выходные данные:**

n – крайнее значение, до которого должны записываться числа, целый.

q – очередь из чисел от 1 до n, коллекция Queue.

**На форме:**

z3 – “3” – label

n – “Введите n” - label

inputN – numericUpDown (1, 100)

Perform – “Выполнить” – button

Result - listBox

**Блок-схема:**

**Листинг программы:**

public partial class pr14\_35 : Form

{

public pr14\_35()

{

InitializeComponent();

}

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

{

int n = (int)InputN.Value;

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

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

q.Enqueue(i);

Result.Items.Clear();

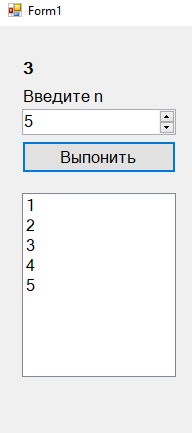
while (q.Count > 0)

Result.Items.Add(q.Dequeue().ToString());

}

}

**Тестовые ситуации:**



**Задание 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 и выходные данные:**

path – имя файла, строковый.

younger40 – очередь из людей младше 40, коллекция Queue;

older40 – очередь из людей старше 40, коллекция Queue;

**На форме:**

z4 – “4” – label

Path– “Введите путь к файлу” - label

InputPath – textBox

CheckFile – “Проверить” – button

FileContent - listBox

**Блок-схема:**

**Листинг программы:**

public class Person

{

public string surname { get; set; }

public string name { get; set; }

public string patronimic { get; set; }

public int age { get; set; }

public int weight { get; set; }

public string Info()

{

return $"{surname} {name} {patronimic} {age} {weight}";

}

}

public partial class pr14\_35 : Form

{

public pr14\_35()

{

InitializeComponent();

}

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

{

string path = InputPath.Text;

if (File.Exists(path))

{

Queue<Person> people = new Queue<Person>();

using (StreamReader sr = File.OpenText(path))

{

string line;

while (!(sr.EndOfStream))

{

line = sr.ReadLine();

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

if (parts.Length == 5)

{

try

{

Person p = new Person

{

surname = parts[0],

name = parts[1],

patronimic = parts[2],

age = int.Parse(parts[3]),

weight = int.Parse(parts[4])

};

people.Enqueue(p);

}

catch (FormatException)

{

MessageBox.Show("В файле ошибка в формате данных", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

MessageBox.Show("В файле неверная запись", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

Queue<Person> younger40 = new Queue<Person>();

Queue<Person> older40 = new Queue<Person>();

while (people.Count > 0)

{

Person p = people.Dequeue();

if (p.age < 40)

younger40.Enqueue(p);

else

older40.Enqueue(p);

}

FileContent.Items.Clear();

FileContent.Items.Add("Люди младше 40 лет:\r\n");

foreach (Person p in younger40)

FileContent.Items.Add(p.Info() + "\r\n");

FileContent.Items.Add("Остальные:\r\n");

foreach (Person p in older40)

FileContent.Items.Add(p.Info() + "\r\n");

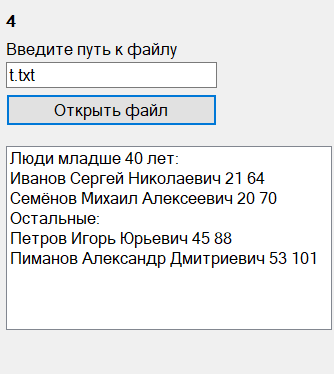
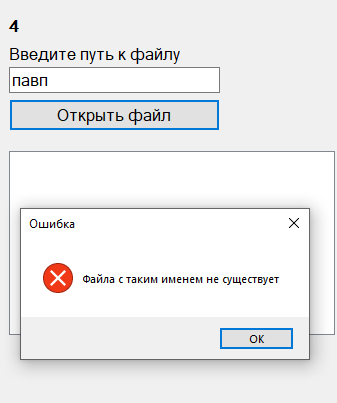
}

else

MessageBox.Show("Файла с таким именем не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**Тестовые ситуации:**



**Задание 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 и выходные данные:**

path\_fio – имя файла, содержащий фио, строковый.

path\_aw– имя файла, содержащий возраст и вес, строковый.

item – элементы группы, var.

**На форме:**

z5 – “5” – label

FioPath – “Введите путь к файлу с фио” - label

InputFioPath – textbox

AwPath – “Введите путь к файлу с возрастом и весом” - label

InputAw– textBox

OpenFiles - “Открыть файлы” – button

FilesContent - listBox

**Блок-схема:**

**Листинг программы:**

public class Person

{

public string surname { get; set; }

public string name { get; set; }

public string patronimic { get; set; }

public int age { get; set; }

public int weight { get; set; }

public string Info()

{

return $"{surname} {name} {patronimic} {age} {weight}";

}

}

public partial class pr14\_35 : Form

{

public pr14\_35()

{

InitializeComponent();

}

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

{

string path\_fio = InputFioPath.Text;

string path\_aw = InputAw.Text;

if (File.Exists(path\_aw) && File.Exists(path\_fio))

{

Queue<Person> people = new Queue<Person>();

using (StreamReader sr = File.OpenText(path\_fio))

{

using (StreamReader sr\_aw = File.OpenText(path\_aw))

{

string names;

string aws;

while (!(sr.EndOfStream))

{

names = sr.ReadLine();

string[] names\_parts = names.Split(' ');

aws = sr\_aw.ReadLine();

string[] aws\_parts = aws.Split(' ');

if (names\_parts.Length == 3 && aws\_parts.Length == 2)

{

try

{

Person p = new Person

{

surname = names\_parts[0],

name = names\_parts[1],

patronimic = names\_parts[2],

age = int.Parse(aws\_parts[0]),

weight = int.Parse(aws\_parts[1])

};

people.Enqueue(p);

}

catch (FormatException)

{

MessageBox.Show("В файле ошибка в формате данных", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

MessageBox.Show("В файле неверная запись", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

var groupped = people

.OrderBy(p => p.age)

.GroupBy(p => p.surname

.FirstOrDefault())

.SelectMany(group => new[] { $"Группа: {group.Key}" }

.Concat(group.Select(p => p.Info())));

FilesContent.Items.Clear();

foreach (var item in groupped)

FilesContent.Items.Add(item);

}

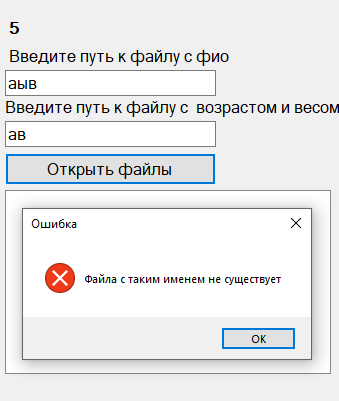
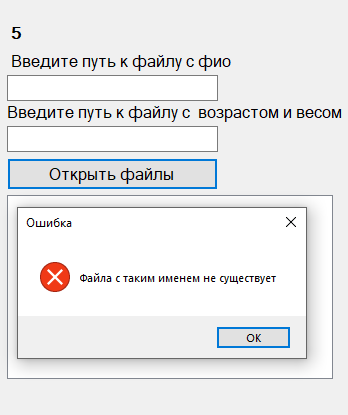
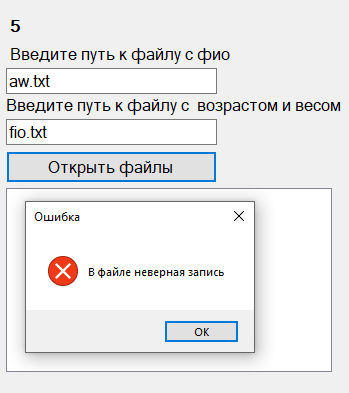
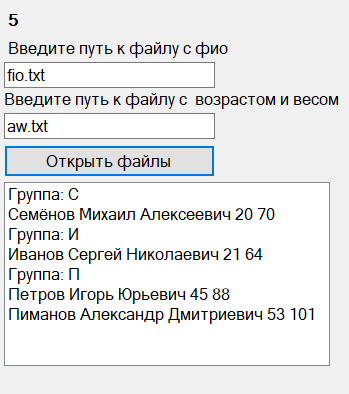
else

MessageBox.Show("Файла с таким именем не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

**Тестовые ситуации:**



**Вывод:** я научилась работать с коллекциями stack и queue, а также с методами библиотеки linq.