Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

# Отчёт по программе «Практическое занятие 14»

Выполнил: Полеев Андрей

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**

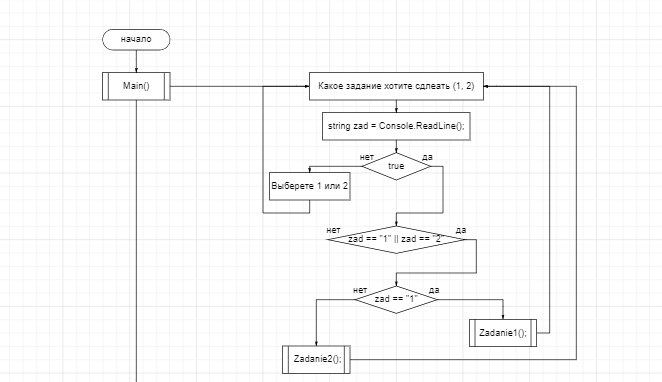
**Входные**

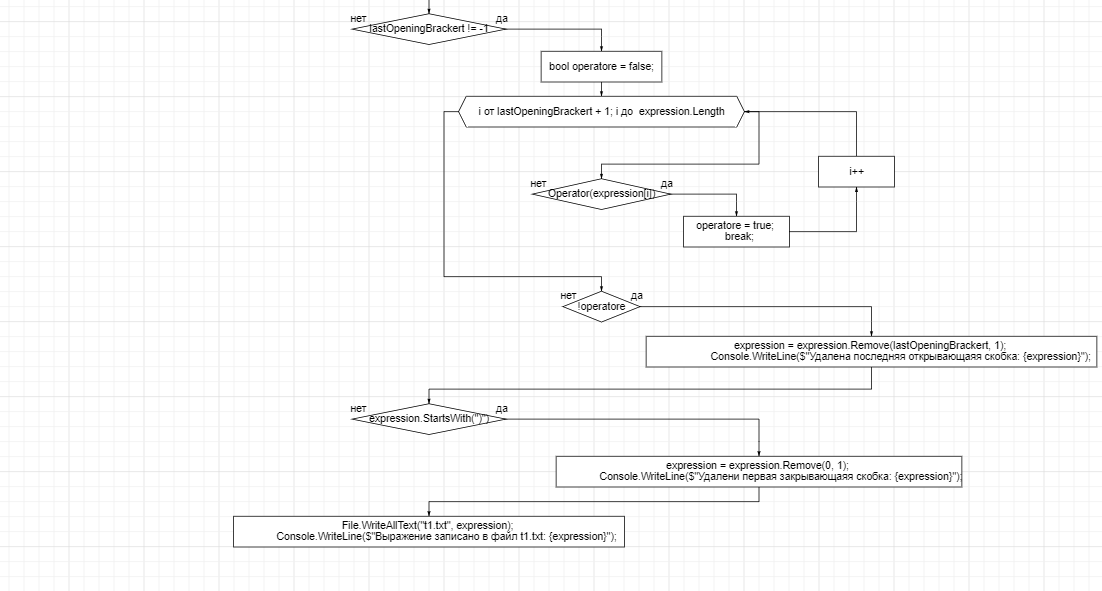
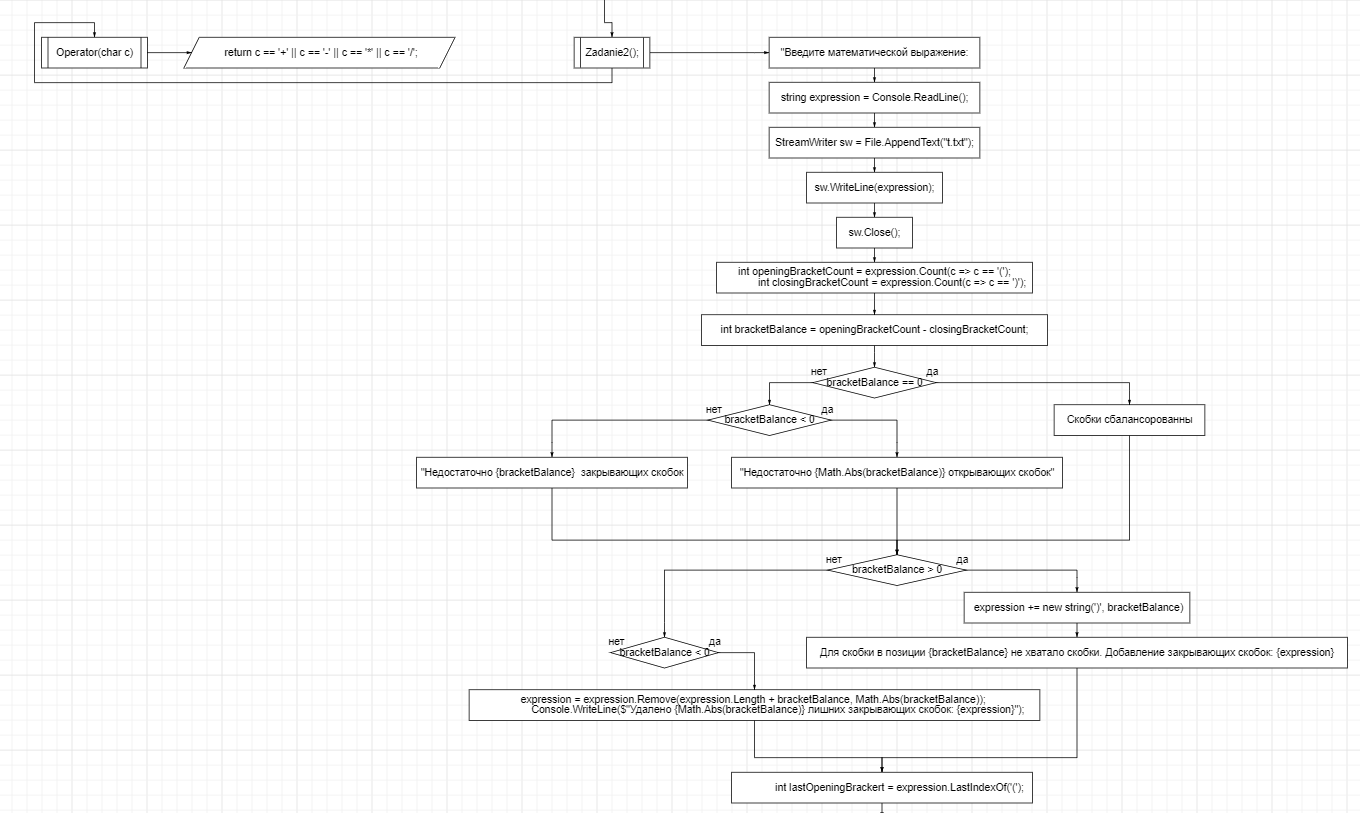
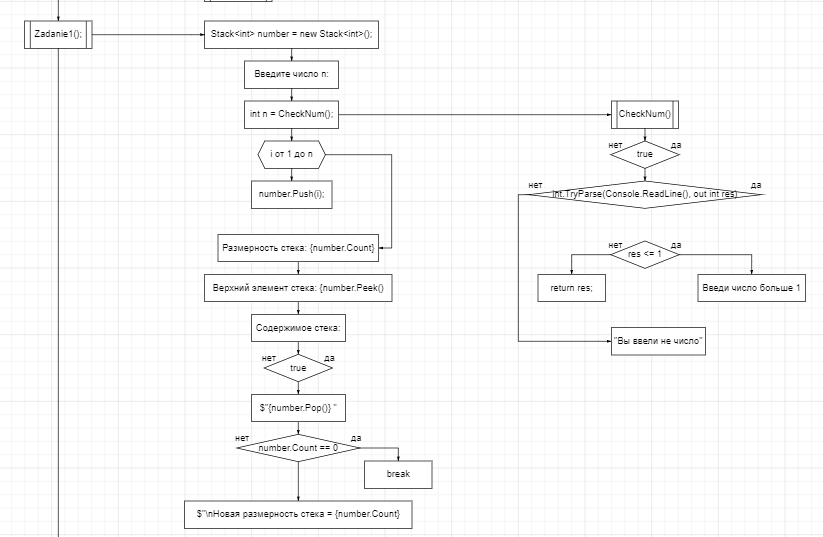
**number-**

число n – int

Выходные

**Блок-схема**





**Листинг программы (если есть)**

static int CheckNum()

{

while (true)

{

if (int.TryParse(Console.ReadLine(), out int res))

{

if (res <= 1)

Console.WriteLine("Введи число больше 1");

else

return res;

}

else

Console.WriteLine("Вы ввели не число");

}

}

static void Zadanie1()

{

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

Console.Write("Введите число n: ");

int n = CheckNum();

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

{

number.Push(i);

}

Console.WriteLine($"Размерность стека: {number.Count}");

Console.WriteLine($"Верхний элемент стека: {number.Peek()}");

Console.Write("Содержимое стека: ");

while (true)

{

Console.Write($"{number.Pop()} ");

if (number.Count == 0)

break;

}

Console.WriteLine($"\nНовая размерность стека = {number.Count}");

}

static void Zadanie2()

{

Console.Write("Введите математической выражение: ");

string expression = Console.ReadLine();

StreamWriter sw = File.AppendText("t.txt");

sw.WriteLine(expression);

sw.Close();

int openingBracketCount = expression.Count(c => c == '(');

int closingBracketCount = expression.Count(c => c == ')');

int bracketBalance = openingBracketCount - closingBracketCount;

if (bracketBalance == 0)

{

Console.WriteLine("Скобки сбалансорованны");

}

else if (bracketBalance < 0)

{

Console.WriteLine($"Недостаточно {Math.Abs(bracketBalance)} открывающих скобок");

}

else

{

Console.WriteLine($"Недостаточно {bracketBalance} закрывающих скобок");

}

if (bracketBalance > 0)

{

expression += new string(')', bracketBalance);

Console.WriteLine($"Для скобки в позиции {bracketBalance} не хватало скобки. Добавление закрывающих скобок: {expression}");

}

else if (bracketBalance < 0)

{

expression = expression.Remove(expression.Length + bracketBalance, Math.Abs(bracketBalance));

Console.WriteLine($"Удалено {Math.Abs(bracketBalance)} лишних закрывающих скобок: {expression}");

}

int lastOpeningBrackert = expression.LastIndexOf('(');

if (lastOpeningBrackert != -1)

{

bool operatore = false;

for (int i = lastOpeningBrackert + 1; i < expression.Length; i++)

{

if (Operator(expression[i]))

{

operatore = true;

break;

}

}

if (!operatore)

{

expression = expression.Remove(lastOpeningBrackert, 1);

Console.WriteLine($"Удалена последняя открывающаяя скобка: {expression}");

}

}

if (expression.StartsWith(")"))

{

expression = expression.Remove(0, 1);

Console.WriteLine($"Удалени первая закрывающаяя скобка: {expression}");

}

File.WriteAllText("t1.txt", expression);

Console.WriteLine($"Выражение записано в файл t1.txt: {expression}");

}

static bool Operator(char c)

{

return c == '+' || c == '-' || c == '\*' || c == '/';

}

static void Main(string[] args)

{

Console.WriteLine("Какое задание хотите сдлеать (1, 2)");

string zad = Console.ReadLine();

while (true)

{

if (zad == "1" || zad == "2")

{

if (zad == "1")

{

Zadanie1();

Console.WriteLine("Какое задание хотите сдлеать (1, 2)");

zad = Console.ReadLine();

}

else

{

Zadanie2();

Console.WriteLine("Какое задание хотите сдлеать (1, 2)");

zad = Console.ReadLine();

}

}

else

{

Console.WriteLine("Выберете 1 или 2");

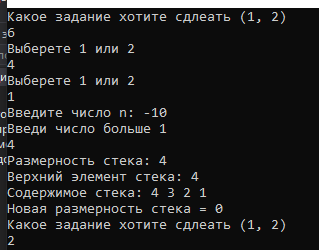
zad = Console.ReadLine();

}

}

}

**Тестовые ситуации**



**Вывод**