Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Разработка программы |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 26-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Фельде Сергея Дмитриевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-**231** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2024

**Содержание**

[Введение 3](#_Toc167657882)

[Постановка задачи 4](#_Toc167657883)

[Разработка кода 5](#_Toc167657884)

[Результаты 10](#_Toc167657885)

[Заключение 12](#_Toc167657886)

[Список использованной литературы 13](#_Toc167657887)

Введение

На сегодняшний день информационные технологии играют большую роль в жизни общества. Прогресс не стоит на месте, и теперь эти технологии применяются практически во всех сферах жизни человечества. Поэтому в наше время полезно владеть навыками алгоритмизации и программирования.

Постановка задачи

Необходимо создать программу, выполняющую 2 алгоритма: проверка скобочной последовательности на правильность, вычисления результата, принимая на вход обратный польский вид записи выражения.

Обратная польская запись – форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Автоматизация вычисления выражений в обратной польской нотации основана на использовании [стека](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA). Стек – это список элементов, организованных по принципу «последним пришёл – первым вышел».

Разработка кода

**Структура**

Код выполнен в среде Visual Studio 2022 на языке C#.

Программа состоит из двух методов: showMenu – метод, вызывающий меню, Main – метод, в котором происходит выполнение пунктов, выбранных в меню, включая алгоритмы проверки скобочной последовательности и вычисление выражения по обратной польской записи.

**Код программы**

using System;

namespace RGR\_Algor

{

internal class Program

{

public class Menu

{

public void showMenu()

{

Console.WriteLine("--------\nМЕНЮ:\n1) Начать скобочную последовательность\n2) Начать польскую запись\n3) Условия задачи\n4) Об авторе\n5) Выход\n--------");

}

}

static void Main()

{

Menu menu = new Menu();

menu.showMenu();

string p\_menu;

while ((p\_menu = Console.ReadLine()) != "5")

{

switch (p\_menu)

{

case "1":

var d\_s = new Dictionary<char, char>()

{

{ ')', '(' },

{ ']', '[' },

{ '}', '{' }

};

Console.WriteLine("Введите скобочное выражение: ");

string str = Console.ReadLine();

bool ok = true;

var stack = new Stack<char>();

foreach (char str\_s in str)

{

if (str\_s == '(' || str\_s == '[' || str\_s == '{')

stack.Push(str\_s);

else

{

if (stack.Count != 0)

{

if (d\_s[str\_s] == stack.Peek())

stack.Pop();

else

{

ok = false;

break;

}

}

else

{

ok = false;

break;

}

}

}

if (stack.Count != 0) ok = false;

if (ok)

Console.WriteLine("Верно");

else

Console.WriteLine("Неверно");

break;

case "2":

string arg;

string[] ar = new string[2];

Stack<double> st = new Stack<double>();

int num\_c = 0;

bool num\_minus = false;

bool isfirst = true;

Console.WriteLine("Введите число");

while ((arg = Console.ReadLine()) != "exit")

{

double num;

bool isNum = double.TryParse(arg, out num);

if (num\_minus) { if (isfirst) { num\_c -= 2; num\_minus = false; isfirst = false; } else { num\_c -= 1; num\_minus = false; } }

if (isNum)

{

st.Push(num);

num\_c += 1;

}

else

{

if (arg == "calc")

{

num\_c = 0;

isfirst = true;

Console.WriteLine("Результат: " + st.Pop() + "\nЕсли хотите завершить выполнение алгоритма, введите exit");

}

else if ((num\_c == 2 && isfirst) || ((isfirst == false) && num\_c == 1))

{

double op2;

switch (arg)

{

case "+":

st.Push(st.Pop() + st.Pop());

num\_minus = true;

break;

case "\*":

st.Push(st.Pop() \* st.Pop());

num\_minus = true;

break;

case "-":

op2 = st.Pop();

st.Push(st.Pop() - op2);

num\_minus = true;

break;

case "/":

op2 = st.Pop();

if (op2 != 0.0)

st.Push(st.Pop() / op2);

else

throw new Exception("Ошибка. Деление на ноль");

num\_minus = true;

break;

default:

Console.WriteLine("Ошибка. Неизвестная команда");

break;

}

}

else if (num\_c <= 1) throw new Exception("Недостаточно элементов для выполнения действия!");

else throw new Exception("Недостаточно операций для выполнения действия!");

}

Console.WriteLine("Введите число или знак операции или 'calc' для вывода ответа");

}

break;

case "3":

Console.WriteLine("1) Необходимо проверить скобочную последовательность на правильность\n2) Необходимо получить результат вычислений, принимая на вход обратный польский вид записи выражения");

break;

case "4":

Console.WriteLine("Фельде Сергей Дмитриевич\nСтудент группы ФИТ-231\nОМГТУ 2024");

break;

default:

Console.WriteLine("Ошибка. Неизвестная команда");

break;

}

menu.showMenu();

}

}

}

}

Результаты

На основе данных, вводимых пользователем, получены результаты, которые представлены на рисунках 1, 2, 3, 4.

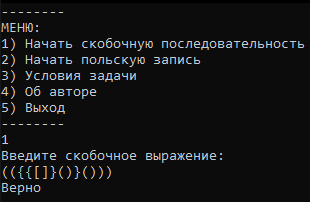


Рисунок 1 – Результаты работы алгоритма проверки скобочной последовательности

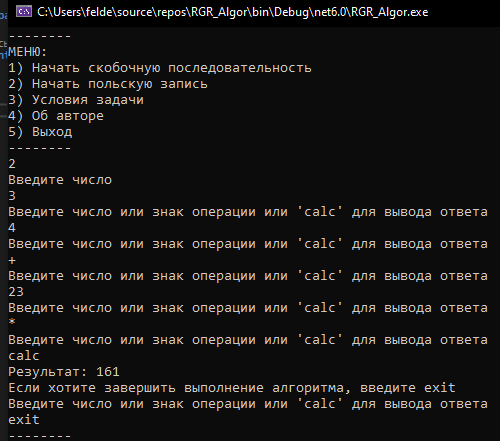


Рисунок 2 – Результаты работы алгоритма вычисления по обратной польской записи

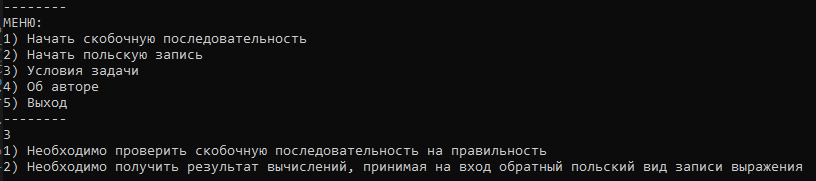


Рисунок 3 – Условия задачи

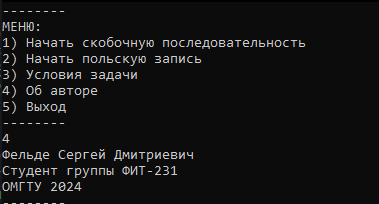


Рисунок 4 – Информация об авторе

Заключение

В ходе выполнения расчётно-графической работы был разработаны и реализованы алгоритмы проверки скобочной последовательности на правильность и вычисления результата обратной польской записи выражения.

Список использованной литературы

1. Microsoft Learn. https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: 20.05.24)