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 | Реализация алгоритмов "Определение правильной скобочной последовательности" и "Вычисление обратной польской записи" |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 3-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Дудикова Дмитрия Евгеньевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-232 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | | 31.05.2024 | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

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

Омск 2024

**Содержа**ние

[Введение 3](#_Toc22848)

[Теоретическая часть 4](#_Toc32058)

[Разработка кода 5](#_Toc16304)

[Результаты](#_Toc2637) 9

[Заключение 10](#_Toc23786)

[Список используемой литературы 1](#_Toc21612)1

# Введение

В наши дни алгоритмы обработки данных в мире информационных технологий играют важную роль во многих областях, начиная от программирования и заканчивая анализом больших объёмов данных. В этой расчётно-графической работе будут реализованы два важных алгоритма: "Определение правильной скобочной последовательности" и "Вычисление обратной польской записи". Эти алгоритмы являются фундаментальными в программировании и на практике применяются в различных сценариях, таких как обработка формул и проверка корректности вводимых данных.

В данной работе эти алгоритмы реализованы на языке программирования C#.

Целью расчётно-графической работы является изучение и практическое применение на платформе C# алгоритмов определения правильной последовательности скобок и вычисления обратной польской записи.

# Теоретическая часть

Для реализации обоих алгоритмов понадобятся знания о классах Stack и Array и об их методах, операторах выбора if, else-if и switch, а также об операторе итерации foreach.

Класс Stack представляет собой структуру данных стек, которая реализует принцип LIFO (Last In, First Out), то есть последний добавленный элемент будет извлечен первым. Этот класс обеспечивает удобный способ работы с данными в стиле стека, где новые элементы добавляются и извлекаются с одного конца структуры данных.

Класс Array представляет собой основной инструмент для работы с массивами, которые являются одной из фундаментальных структур данных. Этот класс предоставляет разнообразные методы и свойства для создания, изменения и обращения к элементам массива.

Оператор if выполняет инструкцию только в том случае, если заданное логическое выражение вычисляется true. Оператор if-else позволяет выбрать, какие из двух путей кода следует выполнять на основе логического выражения.

Оператор if может быть в любой из следующих двух форм:

1. оператор if с частью else выбирает для выполнения один из двух операторов на основе значения логического выражения;
2. оператор if без части else выполняется, только если логическое выражение имеет значение true.

В нашем случае мы будем использовать оператор if для проверки нескольких условий.

Оператор switch представляет собой управляющую конструкцию, которая позволяет выбирать между несколькими вариантами выполнения кода на основе значения одного выражения или переменной. Он позволяет сравнить выражение с различными вариантами значений (case) и выполнить соответствующий блок кода для первого совпадающего значения. Если совпадающее значение не найдено, выполнится код в блоке default.

Оператор итерации foreach предназначен для удобного перебора элементов коллекций или массивов без явного использования индексации. Этот оператор обеспечивает упрощённый итеративный доступ к элементам структур данных, что делает его очень удобным для использования в циклах.

# Разработка кода

Этот код представляет собой консольное приложение на языке C#. Программа отображает меню с четырьмя пунктами: проверка правильной скобочной последовательности, вычисление выражения в польской нотации, информация об авторе и выход. Пользователь выбирает пункт меню, после чего выполняется соответствующая функция.

using System;

using System.Collections.Generic;

class Program

{

static void Polish()

{

Console.WriteLine("Введите выражение в обратной польской записи: ");

string polishString = Console.ReadLine();

var elements = polishString.Split(' ');

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

foreach (var element in elements)

{

if (element == "+" || element == "-" || element == "\*" || element == "/")

{

double a = 0, b = 0;

try

{

a = stack.Pop();

b = stack.Pop();

}

catch

{

Console.WriteLine("Неверное выражение: недостаточно операндов для операции");

return;

}

try

{

switch (element)

{

case "+":

stack.Push(b + a); break;

case "-":

stack.Push(b - a); break;

case "\*":

stack.Push(b \* a); break;

case "/":

if (a == 0) throw new Exception("На ноль делить нельзя!");

stack.Push(b / a); break;

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

return;

}

}

else

{

try

{

stack.Push(double.Parse(element));

}

catch

{

Console.WriteLine("Запись введена неверно!");

return;

}

}

}

if (stack.Count == 1)

{

Console.WriteLine($"Результат: {stack.Peek()}");

}

else

{

Console.WriteLine("Неверное выражение: остаток в стеке");

}

}

static void Staples()

{

Console.WriteLine("Введите выражение, содержащее 3 вида скобок: () {} []: ");

string str = Console.ReadLine();

char[] staples = new char[] { '(', ')', '[', ']', '{', '}' };

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

foreach (char c in str)

{

int index = Array.IndexOf(staples, c);

if (c == '(' || c == '[' || c == '{') stack.Push(c);

else if (index != -1)

{

if (stack.Count > 0 && stack.Peek() == staples[index - 1]) stack.Pop();

else

{

Console.WriteLine("Неверно расставлены скобки");

return;

}

}

}

if (stack.Count > 0) Console.WriteLine("Неверно расставлены скобки");

else Console.WriteLine("Скобки расставлены верно");

}

static void AboutAuthor()

{

Console.WriteLine("Выполнил студент первого курса ОмГТУ группы ФИТ-232 Дудиков Дмитрий Евгеньевич");

}

static void Main()

{

bool exit = false;

while (!exit)

{

Console.Clear();

Console.WriteLine("Меню программы:");

Console.WriteLine("1. Скобочная последовательность");

Console.WriteLine("2. Польская запись");

Console.WriteLine("3. Об авторе");

Console.WriteLine("4. Выход");

Console.Write("Выберите пункт меню: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

Console.Clear();

Staples();

break;

case "2":

Console.Clear();

Polish();

break;

case "3":

Console.Clear();

AboutAuthor();

break;

case "4":

exit = true;

break;

default:

Console.Clear();

Console.WriteLine("Неверный выбор. Попробуйте снова.");

break;

}

if (!exit)

{

Console.WriteLine("Нажмите любую клавишу для возврата в меню...");

Console.ReadKey();

}

}

}

}

# Результаты

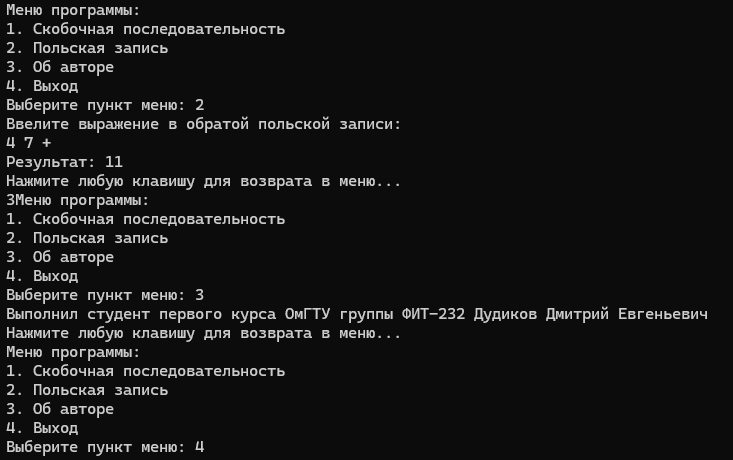


Рисунок 1 – Результат безошибочной работы программы

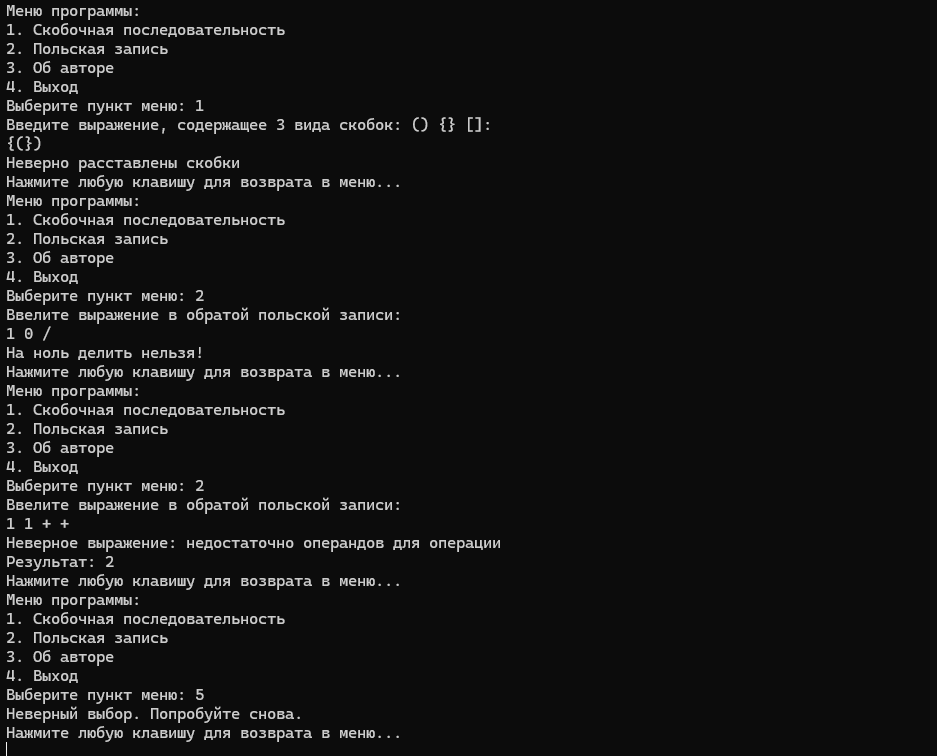


Рисунок 2 – Результат работы программы с ошибками

# Заключение

В ходе выполнения расчётно-графической работы был разработан код, представляющий собой консольную программу. Этот код демонстрирует основы работы с консольным вводом/выводом, обработкой стеков и базовыми арифметическими операциями, а также организацию простого пользовательского интерфейса на основе меню.

# Список используемой литературы

1. Stack Класс // Microsoft Learn URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.collections.stack?view=net-8.0 (дата обращения: 29.05.2024).
2. Локальные функции – Руководство по языку C# // Microsoft Learn URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/local-functions (дата обращения: 28.05.2024).