**Отче по теме:**

**“Вычисление арифметических выражений (стеки)”**

**Выполнил** студент 2 курса группы**: 381906-3**  
**ФИО:** Щеткин Артем Игоревич

Нижний Новгород 2020

Содержание:

Введение 3

Математическая постановка задачи 4

Описание методов 4

Программная реализация 5

Заключение 6

Список литературы 7

**Введение**

При вычислении произвольных арифметических выражений возникают две основные задачи: проверка корректности введённого выражения и выполнение операций в порядке, определяемом их приоритетами и расстановкой скобок. Существует алгоритм, позволяющий реализовать вычисление произвольного арифметического выражения за один просмотр без хранения промежуточных результатов. Для реализации данного алгоритма выражение должно быть представлено в постфиксной форме.

Ввиду того, что вычислять значение выражения, записанного в инфиксной форме, неудобно. Проще сначала перевести его в *постфиксную*, или *обратную польскую* запись, в которой знак операции записывается после своих операндов. Ниже, я продемонстрирую данный алгоритм.

Что же из себя представляет арифметическое выражение и стек?

**Арифметическое выражение** – это выражение, записанное с помощью констант, переменных, функций, знаков арифметических операций и круглых скобок. Результатом вычисления арифметического выражения является числовая константа.

**Стек** (англ. stack — стопка; читается стэк) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

**Математическая постановка задачи**

Алгоритм выполнения нашей программы:

1. Написать функцию, отвечающую за правильность введенных пользователем выражений.
2. Написать функцию перевода выражения из инфиксной формы в постфиксную или обратную польскую запись.
3. Написать функцию, вычисляющую выражение, которое записано в постфиксной форме и выводящую результат этого выражения.
4. Напоследок добавил функцию, объединяющую все функции выше в одну (для удобства пользователя).

**Описание методов**

Для работы со стеком реализовываются следующие операции:

* **Push** (Метод push() ставит значение на вершину стека)
* **Pop** (Метод pop() удаляет значение с вершины, то есть удаляет последний элемент)
* **Top** (Метод top() получает значение верхнего элемента, не удаляя его)
* **Size** (Метод size() возвращает количество элементов в стеке.)
* **Empty** (Булева функция empty() дает понять пуст стек или нет. Возвращает true если стек пуст)

Для описания структуры хранения используются след. переменные:

* **pMem** (указатель на память, выделенную для хранения стека)
* **MemSize** (размер выделенной памяти)
* **MaxMemSize** (размер памяти, выделяемый по умолчанию, если при создании стека явно не указано требуемое количество элементов памяти)
* **DataCount** (количество запомненных в стеке значений)
* **Hi** (индекс элемента массива, в котором хранится последнее добавленное значение стека)

**Программная реализация**

Весь код доступен по ссылке на github: <https://github.com/NickRimerQ/PolskayaZapis>

**Выбор инструментов**

Язык программирования С++, Microsoft Visual Studio 2019

**Руководство программиста**

* #include <iostream> - заголовочный файл, предоставляющий средства ввода-вывода, для стандартной консоли.
* #include <string> - заголовочный файл, содержащий методы со строками и переменные – строки.
* int main – глобальная функция.
* int check() – функция проверяет правильность написания выражений.
* String polskayaZapis() – функция преобразует выражение из инфиксной в постфиксную форму.
* int result() – функция считает выражение в постфиксной форме и возвращает результат.
* int together() – объединяет все три перечисленные функции в одну, и в случае ошибки при записи выражения сообщает пользователю, что выражение введено неверно.

**Заключение**

Я смог выполнить поставленные ранее задачи, т.е.реализована проверка правильности расстановки скобок, выполняется перевод записи выражения в постфиксную форму, правильно вычисляются корректно записанные арифметические выражения.

Также были проведены некоторые арифметические эксперименты:

1) 1+2

Ошибок: 0

Постфиксная форма: 1 2 +

Результат: 3

2) 1+2\*(3-2)-4

Ошибок: 0

Постфиксная форма: 1 2 3 2 -\*+4 –

Результат: -1

3) (((1+23)\*1-22)+5)\*2-(7

Ошибок: 1

Перевод и вычисление невозможно

4) 1+2/(3-3)

Ошибок: 0

Постфиксная форма: 1 2 3 3-/+

Результат: Деление на ноль невозможно

5) 1++1

Ошибок 0

Постфиксная форма 1 1 ++

Результат Недостаточно операндов

**Список литературы**

* Структуры данных и другие объекты в C++ - Майкл Мейн, Уолтер Савитч
* Эффективное программирование на C++ - Эндрю Кёниг, Барбара Э. Му
* Скиена С.С. Алгоритмы. Руководство по разработке
* Грокаем алгоритмы - Бхаргава Адитья