Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление арифметических выражений»**

**Выполнил**:

студентка группы 381903-2

Петрова П.В.

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2020

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

Руководство программиста………………………………………………………………………... 5

[Руководство пользователя 6](#_Toc26962564)

[Описание программной реализации 7](#_Toc26962565)

[Подтверждение корректности 8](#_Toc26962566)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Разработать программу, выполняющую вычисление арифметического выражения с вещественными числами. Выражение в качестве операндов может содержать переменные и вещественные числа. Допустимые операции: +, -, /, \*, унарный -; наличие математических функций (sin, cos, ln, exp).

# Метод решения

В ходе данной лабораторной работы реализован шаблонный класс TStack, класс Lexem.

Класс Stack используется для реализации методов в классе Lexem.  
В классе Lexem есть конструкторы, определяющие что содержит выражение (число/ операцию), функции: getn – возвращает число, getop – возвращает операцию, есть две булевские функции is\_num is\_op, они проверяют символ: число это или нет, функция возврата приоритета.

Описание алгоритма перевода в ОПЗ:

На вход получаем массив лексем (размер массива N). Далее в цикле (от 1 до N) берем лексему, если это число, то кладем в новый массив лексем, если же это операция, то кладем ее в стек (предварительно проверив его на пустоту); Если пустой, то кладем в стек, если нет, то сравниваем приоритет операции с приоритетом той операции, которая лежит в стеке. Если приоритет операции уже лежавшей в стеке больше приоритета поступающей, то достаем ее и кладем в новый массив. После выполнения этого цикла добавляем все операции из стека в конец нового массива. На выходе получаем массив лексем в ОПЗ.  
Метод find\_des:  
На вход подается массив лексем в ОПЗ и размер массива. В цикле (по массиву) достаем лексему, если это операнд, то кладем в стек, если бинарная операция, то достаем из стека два числа, производим с ними это операцию и кладем обратно в стек, если унарная операция, то из стека достаем только одно число. После завершения цикла, результат будет лежать на вершине стека.

Метод work:

На вход получает строку, разбирает строку на лексемы, записывая из в массив, проводит проверку правильности математического выражения, переводит в ОПЗ и вычисляет значение.

# Руководство программиста.

Программа использует такие стандартные библиотеки, как: iostream, cmath, sring.

Реализован шаблоннный класс TStack. Private часть содержит массив данный типа T(шаблон) и две целочисленные переменные – top и size –вершина и размер стека соответственно. Public часть: два конструктора – дефолтный (выделяется память под массив данных, вершине приравнивается 0, size = 10) и с параметром, который задает размер стека. Функции «положить в стек», «взять из стека», «показать стек», функция булевского типа, которая определят заполненность стека, функция «очистки» стека (top=0). Класс TStack используется для перевода в ОПЗ и вычисления выражения.

Реализован класс Lexem. В private объявлены поля: переменная типа double(число), переменная типа std::string (строка) и переменная типа bool, определяющая какой символ подается. В public части реализованы методы, которые позволяют узнать число это или операция, узнать приоритет операции, а также возвращает значение числа или саму операцию. Этот класс используется для создания массива лексем и стека лексем.

# Руководство пользователя

1. Запустить программу
2. Ввести математическое выражение

Правила ввода математического выражения:  
- вводить без пробелов;

- доступны такие переменные, как: x0, x1, x2, x3, x4;

- выражение может содержать такие математические функции, как sin(), cos(), ln(), exp(), (их аргументы записывать в скобках);

- пример возможного выражения: sin(7.2)+77-exp(4)\*(777-ln(3.17))+8/0.25

# Описание программной реализации

Программа использует такие стандартные библиотеки, как: iostream, cmath, sring.

В файле stach.h реализован шаблонный класс TStack. Файл arithmetic.h – реализация основных методов и класса Lexem.

# Подтверждение корректности

Для подтверждения корректности в программе реализован метод cheking.

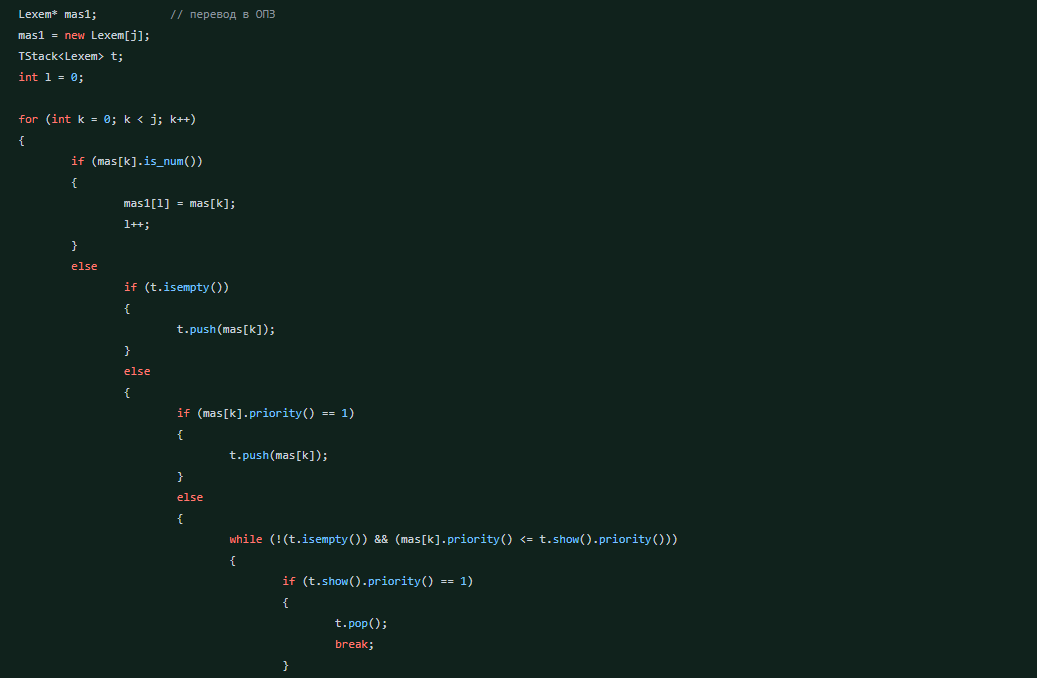
В нем на вход мы получаем массив лексем и его размер. В цикле проверяем, что следует после текущей лексемы, это может быть: бинарная операция,); унарная операция; (; число, унарная операция, (; бинарная операция,). В случае как-либо несовпадения выдается ошибка.

# Заключение

В ходе данной лабораторной работы была написана программа, которая вычисляет значение математического выражения. Реализовано: шаблонный класс стек, класс лексем, алгоритм разбиения строки на лексемы, метод проверки, алгоритм перевода выражения в ОПЗ, алгоритм вычисления математического выражения.

# Приложение

Алгоритм перевода в ОПЗ: (рис.1, рис.2)

рис.1

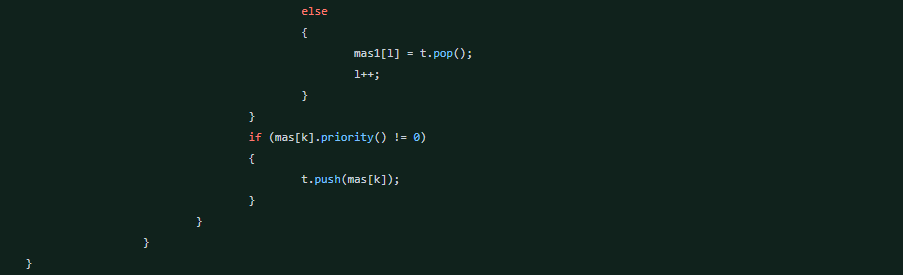


рис.2

Алгоритм вычисления: (рис.3, рис.4)

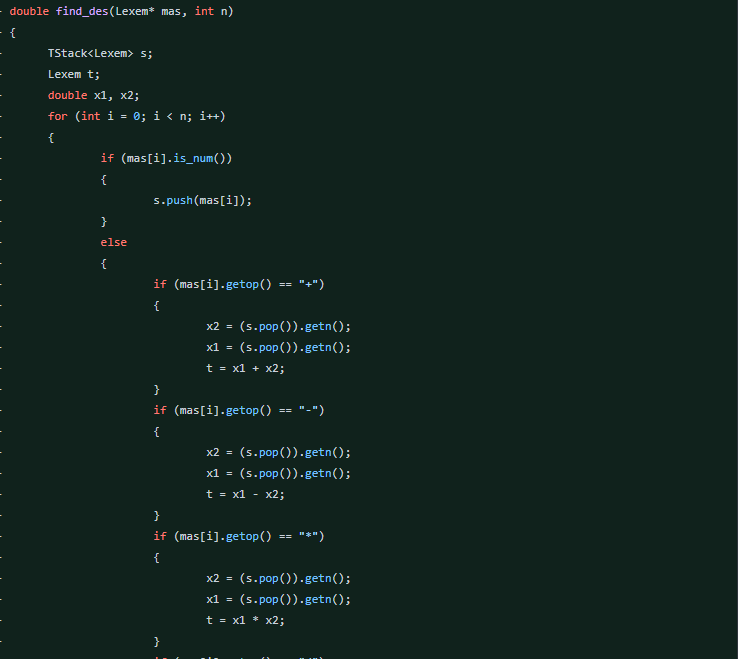
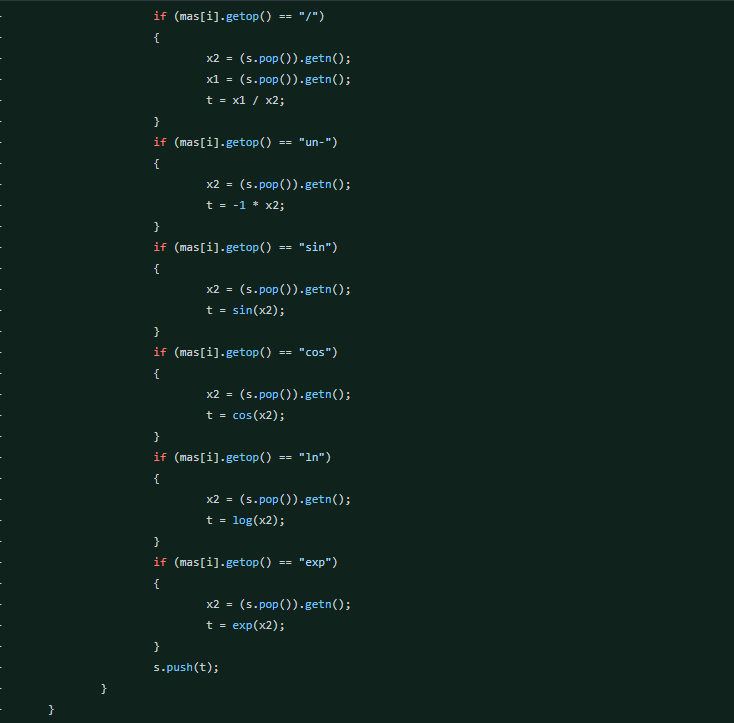


Рис.3

рис.4