**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

ОТЧЁТ

**по курсовой работе**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: «Преобразование алгебраический формул из инфиксной в постфиксную форму записи и вычисление значения выражения»**

**Вариант 1.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9302 |  | Горбанев А.В |
| Преподаватель |  | Тутуева А.В. |

Санкт-Петербург

2020

## Постановка задачи и описание реализуемого класса и методов.

Задание: Необходимо реализовать простейшую версию калькулятора. Пользователю должен быть доступен ввод математического выражения, состоящего из чисел и арифметических знаков. Программа должна выполнить проверку корректности введенного выражения. В случае некорректного ввода необходимо вывести сообщение об ошибке с указанием позиции некорректного ввода. В противном выводится постфиксная запись введенного выражения, а также отображается результат вычисления.

Для реализации мне потребовался класс stack содержащий в себе класс Elem.

class Stack

|  |  |  |
| --- | --- | --- |
| Поля | Описание | |
| Elem\*top | Указатель на начало списка | |
| size\_t size; | Переменная, хранящая размер списка | |
| Методы | Описание | Оценка временной сложности |
| void push(char data) | Добавление в конец списка | O(1) |
| void pop() | Удаление последнего элемента | O(1) |
| Void getDataNow() | Получение значения элемента из стека не вынимая его | O(1) |

class Elem

|  |  |
| --- | --- |
| Поля | Описание |
| Elem\*next | Указатель на следующий элемент стека |
| Char data | Данные элемента |

Также для реализации работы мне понадобились данные функции:

|  |  |
| --- | --- |
| Функции | Описание |
| string convertInt(double number) | Перевод числа в строку |
| int check\_string(string infix); | Проверка строки на корректно введенные данные |
| void string\_to\_double(double& a, double& b, string after, string before); | Перевод двух строк в числа |
| void one\_string\_to\_double(double& b, string after); | Перевод одной строки в число |
| string string\_power(string infix); | Реализация в строке возведения в степень |
| string string\_exp(string infix); | Замена в строке символа e на число |
| string string\_multiplication(string infix); | Реализация в строке умножения чисел |
| string string\_pi(string infix); | Замена в строке символа pi на число |
| string string\_division(string infix); | Реализация в строке деления чисел |
| string string\_addition(string infix); | Реализация в строке суммирования чисел |
| string string\_computation(string infix); | Реализация в строке вычитания чисел |
| string string\_cos\_procedure(string infix); | Реализация в строке вычисления косинусов |
| string string\_sin\_procedure(string infix); | Реализация в строке вычисления синусов |
| string string\_ctg\_procedure(string infix); | Реализация в строке вычисления котангенсов |
| string string\_tg\_procedure(string infix); | Реализация в строке вычисления тангенсов |
| string string\_ln\_procedure(string infix); | Реализация в строке вычисления натуральных логарифмов |
| string string\_log\_procedure(string infix); | Реализация в строке вычисления десятичных логарифмов |
| string minus\_plus\_procedure(string infix); | Замена в строке повторных символов суммирования или вычитания |
| string string\_brackets(string infix, int& brackets\_open); | Работа со скобками в строке |
| string implementation(string infix); | Выполнение набора действия для получения результата |

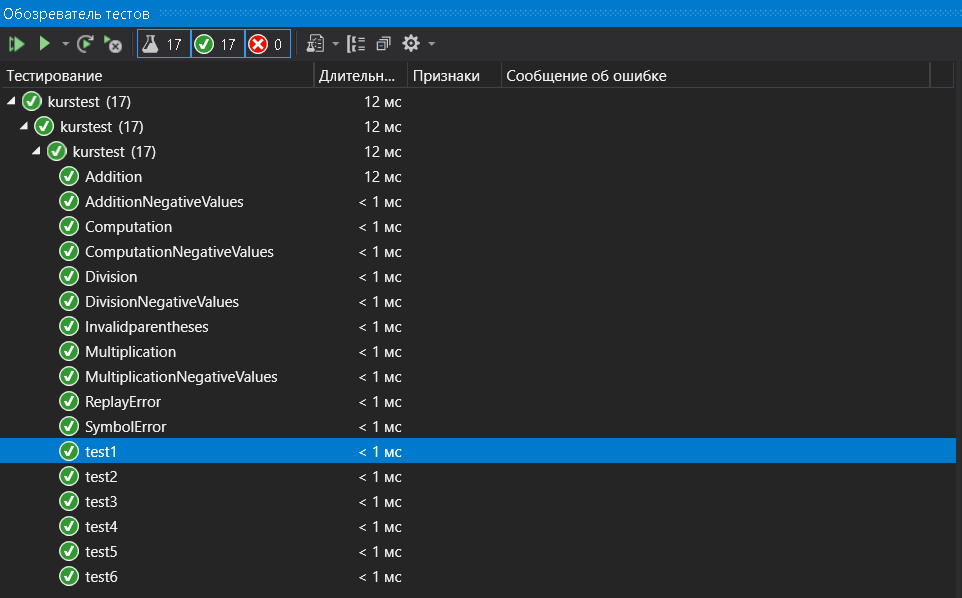
## Описание алгоритма решения.

Алгоритм:  
Пользователем вводится выражение в консоль. Далее программа считывает введенную строку. Далее идет работа отдельно с выражениями в каждых скобках, поочередно выполняя функции для вычисления возможных операций в данном выражении, это выполняется до тех пор, пока не будут раскрыты все скобки, в конечном счете получается итоговое значение.

Описание реализованных unit-тестов.

|  |  |
| --- | --- |
| Название теста | Что проверяет |
| Addition | Проверка суммирования значения |
| AdditionNegativeValues | Проверка суммирования отрицательных значений |
| Computation | Проверка вычитания значений |
| ComputationNegativeValues | Проверка вычитания отрицательных значений |
| Multiplication | Проверка умножения значений |
| MultiplicationNegativeValues | Проверка умножения отрицательных значений |
| Division | Проверка деления значений |
| DivisionNegativeValues | Проверка деления отрицательных значений |
| test1 | Проверка работы калькулятора в разных ситуациях и вычислениях |
| test2 |
| test3 |
| test4 |
| test5 |
| test6 |
| SymbolError | Проверка работы проверки на корректность введенных символов |
| ReplayError | Проверка работы проверки на нахождение повторно введенных символов |
| Invalidparentheses | Проверка работы проверки на правильно введенных скобки |

## Результат выполнения всех unit-тестов



## Примеры работы программы

