**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 |  | Ковтун А.С. |
| 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.

|  |  |  |
| --- | --- | --- |
| **Функция** | **Описание** | **Оценка временной сложности** |
| **Class strl** | | |
| Strl() | Конструктор по умолчанию | O(1) |
| Strl(cons char [N]) | Конструктор с использованием массива символов | O(l) |
| Strl(double) | Конструктор из числа | O(n) |
| ~strl() | Деструктор | O(1) |
| Size\_t getSize() | Возвращение размера строки | O(1) |
| Void input(fstream&) | Ввод строки из файла | O(n) |
| Void output(fstream&) | Вывод строки в файл | O(n) |
| Bool isDigit() | Проверка является ли строка числом | O(n) |
| Bool isOperation() | Проверка является ли строка операцией | O(n) |
| Int getOperationWeight() | Возвращение приоритета операции | O(1) |
| Double perform(strl) | Выполнение действия с одним аргументом | O(1) |
| Double exe (strl, strl) | Выполнение действия с двумя аргументами | O(1) |
| Bool checkinfix(fstream&) | Проверка корректности инфиксной формы | O(n) |
| **Class element** | | |
| element(strl, elem\*) | Конструктор с использованием строки | O(1) |
| element(double, elem\*) | Конструктор с использованием числа | O(1) |
| bool isBracket() | Проверка является ли элемент скобкой | O(1) |
| int getpriority() | Возврат приоритета операции | O(1) |
| elem\* getNext() | Возврат следующего элемента стека | O(1) |
| strl getValue() | Возврат строки элемента | O(1) |
| **Class stack** | | |
| stack() | Конструктор по умолчанию | O(1) |
| ~stack() | Деструктор по умолчанию | O(1) |
| elem getTop() | Возвращение верхнего элемента стека | O(1) |
| void pop() | Удаление верхнего элемента стека | O(1) |
| void push(strl) | Добавление строки наверх стека | O(1) |
| void push(double) | Добавление числа наверх стека | O(n) |
| size\_t getSize() | Возвращение количества элементов в стеке | O(1) |
| bool isEmpty() | Проверка стека на пустоту | O(1) |
| void clear() | Очистка стека от элементов | O(n) |
| **Class expression** | | |
| expression(strl) | Конструктор постфиксной форму по инфиксной | O(1) |
| converttopf | Конвертация из инфиксной в постфиксную | O(n) |
| strl calculate() | Расчет результата в постфиксной форме | O(n^2) |
| strl getOutstr() | Возврат постфиксной формы | O(1) |

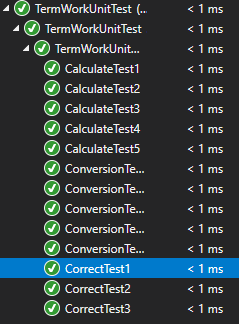
## Описание алгоритма решения.

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.

|  |  |
| --- | --- |
| Имя теста | Описание |
| СorrectTest1 | Закрывающая скобка в начале выражение - ошибка |
| СorrectTest2 | Нет пробелов между операндами и операциями |
| СorrectTest3 | Корректная строка |
| ConversionTest1 | Простое выражение |
| ConversionTest2 | Тригонометрия |
| ConversionTest3 | Деление |
| ConversionTest4 | Степень |
| ConversionTest5 | Много скобок |
| CalculateTest1 | Много скобок |
| CalculateTest2 | Тригонометрия |
| CalculateTest3 | Тригонометрия |
| CalculateTest4 | Натуральный логарифм |
| CalculateTest5 | Большие числа |

## Результат выполнения всех unit-тестов



## Примеры работы программы

https://media.discordapp.net/attachments/696700378997325874/792766597638520832/unknown.png

## Вывод:

Проделав курсовую работу, я научился писать калькулятор на языке С++, изучив структуры данных и алгоритмы.