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**:

студент группы 382003-1

Ларин К.Д.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

Постановка задачи 3

Метод решения 4

Руководство пользователя 5

Описание программной реализации 6

Подтверждение корректности 7

Результаты экспериментов 8

Заключение 9

Приложение 10

**Постановка задачи**

Разработать программу, выполняющую вычисление арифметического выражения с вещественными числами. Выражение в качестве операндов может содержать переменные и вещественные числа. Допустимые операции известны: +, -, /, \*. Допускается наличие знака "-" в начале выражения или после открывающей скобки. Опционально - наличие математических функций (sin, соs, ln, exp, и т.д.) Программа должна выполнять предварительную проверку корректности выражения.

**Метод решения**

Предварительно выражение разбивается на последовавтельность токенов (тип хранящий один из возможных базовых элементов выражения: константа, переменная, оператор, функция). Последовательность токенов преобразуется в обратную польскую запись, чтобы не использовать приоритет операций при вычислении. Вычисление и преобразование выполняется с использованием структуры данных "стек".

**Руководство пользователя**

Пользователю предоставляется возможность указать число переменных, после ввести их имена и указать само выражение. В выражении допостимо использование перечисленных ранее переменных, вещественных констант, функций и операторов, указанных в functions\_parser\_table и скобок. Допускается пропуск умножения между константой и переменной, между скобкой и переменной. Допускается запись функций с одним аргумнтом без скобок как "sin x". При указании неверного числа аргументов или при совпадении имени переменной с именем функции поведение программы неопределено.

**Описание программной реализации**

Для хранения токена используется структура, содержащая union возможных типов и поле lexem\_type, которое обозначает, что в текущий момент хранится в объединении. Для хранения списка функций исмользуется functions\_parser\_table, формат: имя, объект приводимый к std::function, приоритет, количество операнодов. Для получения промежуточной информации о функции по ее имени используется функция find\_in\_table\_by\_name, возвращаестя значение в формате: промежуточное представление функции, приоритет функции, число аргументов. Для исполнения функции используется функция apply\_double\_function, которая принимает промежуточное представление функции, полученное из find\_in\_table\_by\_name и итератор на первый аргумент функции. Для добавления бинарных операторов с инфиксной формой записи, необходимо также добавить их имя в список operators, имя оператора должно иметь тип char. Функция parse\_lexem в классе Token пытается парсить очередную лексему и обрабатывает ошибки связанные с некорректной записью выражения.

**Подтверждение корректности**

Проверка корректности алгоритма осуществляется с помомщью gtests.

**Заключение**

Разработна программа, выполняющая вычисление арифметического выражения с вещественными числами. Выражение в качестве операндов может содержать переменные и вещественные числа.