Отчет по лабораторной работе №4

По курсу Логическое программирование.

Работу выполнил

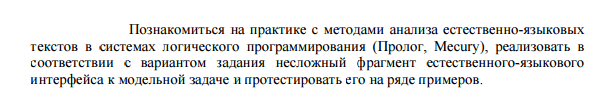
студент группы 8О-207Б

Сорокин Денис Михайлович

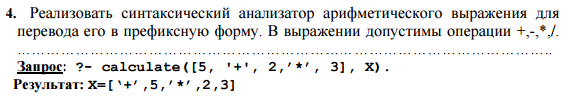
Сдано: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Тема:**

Обработка естественного языка

**Цель работы:**

**Задание:**



**Текст программы:**

%Программа написана и выполнена в web версии SWI-Prolog – SWISH http://swish.swi-prolog.org/

**calculate**(Infix, Prefix):-

**atomic\_list\_concat**(Infix, InfAtom), % встроееный предикат конкатенации элементов списка в один атом

**atom\_to\_term**(InfAtom, Term, **\_**), % конвертация атома в терм

**calc**(Term, Prefix).

**calc**(Term, Prefix):-

Term =.. [Prefix].

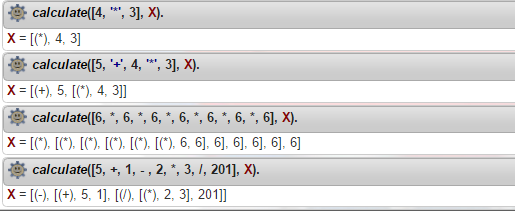
**calc**(Term, [Op, Left, Right]):-

Term =.. [Op, LTmp, RTmp],

**calc**(LTmp, Left),

**calc**(RTmp, Right).

**Протокол работы программы:**



**Выводы:**

В ходе выполнения данной лабораторной работы я научился обрабатывать естественный язык на языке Пролог. Мой задачей было перевести выражение из инфиксной (нормальной) в префиксную (обратная Польская нотация). На императивных языках я бы решал данную задачу с помощью алгоритма Дейкстры, но у нас есть Пролог! Да, на нем данная задача сводится к маленькому объему кода и весьма элегантному решению.

Для решения задачи я взял два встроенных предиката SWI-Prolog:

* **atomic\_list\_concat**
* **atom\_to\_term**

Первый для перевода списка токенов в один атом, т.е. *atomic\_list\_concat*([5, (+), 1], '5+1').

Второй же для перевода полученного атома в терм, т.е. *atom\_to\_term*('5+1', 5+1, []).

Далее полученный терм мы пропускаем через предикат calc. В нем терм «бьется» относительно операторов, а дальше два «осколка» тоже разбиваются рекурсивно, в итоге мы получаем префиксную форму.

В итоге, задача довольно быстро и красиво решена, да и кода немного. По-моему, это замечательно!