Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра «Информационных технологий и систем»

**«Проектирование лексического анализатора»**

Лабораторная работа

по дисциплине:

«Теория языков программирования и методы трансляции»

Проверил:

\_\_\_\_\_\_\_\_ / Макаров В. А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 г.

Выполнил

студент группы 8091:

\_\_\_\_\_\_/ Лехновский А. Д.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

**Великий Новгород**

**2020**

**Постановка задачи**

**Формулировка**

Разработать программу, решающую задачу лексического разбора исходного текста программы. Распознаваемый язык выбирается самостоятельно. В качестве прототипов может быть выбран синтаксис следующих языков программирования: Паскаль, СИ, Бейсик.

**Представление входных данных**

Была выбрана своя грамматика, сделанная на основе **4** (Арифметические выражения в языке C) и **3** варианта (Циклы в языке C) с некоторыми изменениями (добавлениями и упрощениями).

Описание грамматики:

stats ::= stats stat

|

;

stat ::= iteration

| assignment\_exp ‘;’

| { stats }

;

iteration ::= 'while' '(' exp ')' stat ;

exp ::= assignment\_exp | exp ',' assignment\_exp ;

assignment\_exp ::= VAR ‘=’ additive\_exp ;

additive\_exp ::= mult\_exp

| additive\_exp '+' mult\_exp

| additive\_exp '-' mult\_exp

;

mult\_exp ::= cast\_exp

| mult\_exp '\*' cast\_exp

| mult\_exp '/' cast\_exp

| mult\_exp '%' cast\_exp

;

cast\_exp ::= unary\_exp

| '(' type\_name ')' cast\_exp ;

unary\_exp ::= VAR

| INT

| FLOAT

| '++' unary\_exp

| '--' unary\_exp

;

type\_name ::= ‘int’ | ‘double’ | ‘char’ | ‘long’ | ‘short’ ;

VAR ::= идентификатор

INT ::= целое число

FLOAT ::= вещественное число

CHAR ::= символ в одинарных кавычках

Алфавит состоит из: [[A-Z], [a-z], [0-9]], [int, double, char, long, short, while], ['**;**', '**{**', '**}**', '**(**', '**)**', '**,**'], [‘**-**’, ‘**+**‘, ‘**\***’, ‘**/**’, ‘**%**’, ‘**=**’], [‘\_’, '**.**']

**Классы лексем:**

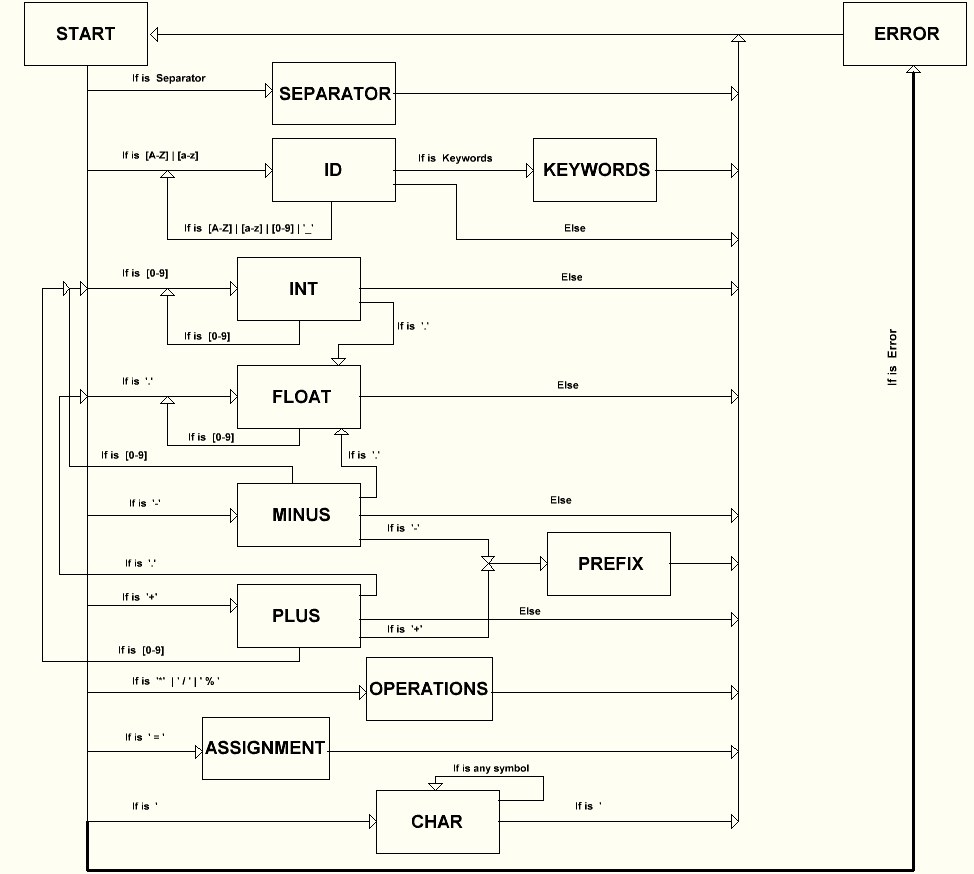
1. Разделители;
2. Идентификаторы;
3. Числовые константы;
4. Знаки операций;
5. Вещественные и целые числа.

**Описание выбранного метода решения задачи ЛА**

Для решения задачи лексического анализатора был выбран конечный автомат, так как он позволяет обрабатывать числа с плавающей точкой со знаком. Для хранения состояний были использованы enum, для хранения ключевых слов, разделителей и некоторых знаков операций задействованы списки.

**Описание реализации**

Общий принцип работы лексического анализатора был реализован в соответствии со следующей схеме:



**Обработка ошибок**

Лексема является «ошибочной», если следующий рассматриваемый символ будет недопустимым.

**Методика тестирования**

При тестировании используется файл *input.txt*, в котором представлены примеры, написанные для данной грамматики. Полученные данные выводятся в виде условной таблицы в файл *output.txt*

**Вывод**

В ходе лабораторной работы был разработан лексический анализатор на основе рассматриваемой грамматики, позволяющий распознать простейшие лексемы.