Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий Кафедра прикладной математики

Отчет защищен с оценкой \_ Преподаватель (подпись)

« » 2023 г.

Отчет

по лабораторной работе № 5

по дисциплине «Теория алгоритмических языков и трансляторов»

Студент гр. ПИ-02 Чередов Р.А.

Преподаватель, к.ф.-м.н., доцент,

Крючкова Е. Н.

Барнаул 2023

# Задание

# Отчет: СД, готовые для реализации синтаксического анализатора (без левой рекурсии, с минимумом примитивных СД и т.д.)

**124 Чередов Роман Алексеевич**

Программа: главная программа языка С++.. Допускается описание классов. Функции не имеют параметров и возвращают void.

Типы данных: double, int, short int.

Операции: все арифметические, сравнения.

Операторы: пустой, составной, присваивания и do {} while.

Операнды: простые переменные, поля классов, именованные константы , константы.

# Константы: целые в 10 c/c.

# Решение

Таблица приоритетов:

|  |  |  |
| --- | --- | --- |
| **Приоритет** | **Оператор** | **Описание** |
| **1**  **самый высокий** | ++ | Постинкремент (Suffix increment). |
| -- | Постдекремент (Suffix decrement). |
| () | Вызов функции. |
| . | Выбор элемента структуры по ссылке. |
| **2** | ++ | Прединкремент (Prefix increment). |
| -- | Преддекремент (Prefix decrement). |
| + | Унарный плюс. |
| - | Унарный минус. |
|  |  |
| **3** | \* | Умножение. |
| / | Деление. |
| % | Взятие по модулю числа (Modulo, получение остатка от деления). |
| **4** | + | Сложение. |
| - | Вычитание. |
| **5** | >> | Сдвиг вправо |
| << | Сдвиг влево |

|  |  |  |
| --- | --- | --- |
| **6** | >=  <=  > | Больше ил равно…  Меньше или равно…  Больше чем... |
| >= | Больше или равно... |
| **7** | == | Оператор определения равенства (Equal to). |
| != | Оператор определения неравенства (Not equal to) |
| **8** | = | Прямое присваивание. |
| += | Присваивание со сложением. |
| -= | Присваивание с вычитанием. |
| \*= | Присваивание с умножением. |
| /= | Присваивание с делением. |
| %= | Присваиванием с взятием остатка от деления. |
| **9**  **самый низкий** | , | Запятая (Comma) |

## КС-грамматика языка:

<программа> → <программа><описание> | ε

<описание> → <данные> | <функция> | <класс>

<данные> → <тип> <список>;| const <тип> <список>;

<тип> → double | int | short int |float| <идентификатор>

<класс>→class<идентификатор> {<тело класса>};

<тело класса>→тело класса> <данные>| <тело класса> <функция> | e

<список> → <список>, <переменная> | <переменная>

<переменная> → <идентификатор> | <идентификатор> = <выражение>

<функция> → void <идентификатор>(<параметры функции>) <составной оператор>| void main()<составной оператор>

<параметры функции> → <список параметров функции> | ε

<список параметров функции> → <тип> <идентификатор> |   
<список параметров функции>, <тип> <идентификатор>

<составной оператор> → {<операторы и описания>}

<операторы и описания> → <операторы и описания> <данные> | <операторы и описания> <оператор> | ε



<оператор> → <присваивание>; | <составной оператор> | <вызов функции> | <do {} while> |;

<do {} while> → do {<оператор>} while <выражение>

<присваивание> → <идентификатор> = <выражение> | <поле класса> = <выражение>

<выражение> → <выражение> == <выр. отношения> | <выражение>!= <выр. отношения> | <выр. отношения>

<выр. отношения> → <выр. отношения> > <слагаемое> | <выр. отношения> >= <слагаемое> | <выр. отношения> < <слагаемое> | <выр. Отношения> <= <cлагаемое> | <cлагаемое>

<слагаемое> → <слагаемое> + <множитель> | <слагаемое> - <множитель> | <множитель>

<множитель> → <множитель> ∗ <унар. операция> | <множитель> / <унар. операция> | <множитель> % <унар. операция> | <унар. операция>

<унар. операция> → + <эл.выр.> | - <эл.выр.> | ++<эл.выр.> | --эл.выр.> | <эл.выр.>

<эл. выр.> → <идентификатор> | <константа> | (<выражение>)| <поле класса>

<вызов функции> → <идентификатор>(<параметры>) <идентификатор> ()| <поле класса>

<параметры> → <параметры>, <выражение> | <выражение>

<идентификатор> → <буква> <окончание>

<окончание> → <окончание> <буква> | <окончание> <цифра> | ε

<константа> → <константа целая>

<константа целая> → <константа целая> <Цифра> | <Цифра>

<Цифра> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<Знак> → + | - | ε

<Буква> → A | a | B | b | C | c | D | d | E | e |

F | f | G | g | H | h | I | i | K | k |

L | l | M | m | N | n | O | o | P | p |

Q | q | R | r | S | s | T | t | V | v |

## X | x | Y | y | Z | z

## Синтаксические диаграммы:

## 

## 

## 

## 

## 

## 

## 

## 

## Пример правильного теста

## class MyOwnClass{

## int x;

## short int shortint = 2;

## double expperem = 1;

## };

## int main(){

## int i = 0;

## short int a1 = 2;

## int a2 = 4;

## return 0;

## do

## {

## y = 1;

## } while ( x > 0 );

## }

## 

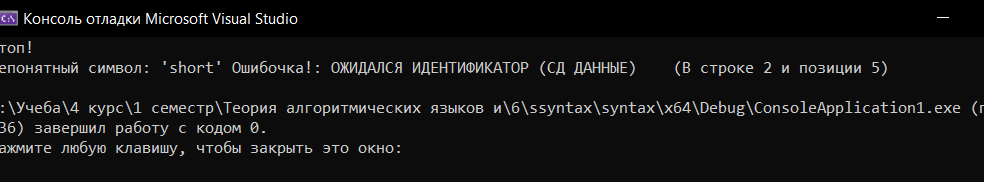
Примеры неправильного теста

class MyOwnClass{

int short const x;

short int shortint = 2;

double =3;};



class MyOwnClass{

double x =3(Пропущена точка с запятой)

int a =2;};

