Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Отчёт по лабораторной работе №2

по дисциплине «Лингвистические средства вычислительных систем»

Вариант 18

Выполнил студент гр.

Проверила:.

Тамбов 20

**Задание:***Разработать программу синтаксического анализа методом рекурсивного спуска. В качестве исходных данных используется последовательность лексем, полученных в результате работы лексического анализатора (Л.Р №1).*

**Отрывок программы:**

READ (H, B, M);

PI : = 3.14;

V : = PI \* H \* ( B \* B + M \* M + B \* M ) / 3

**Кодировочная таблица:**

|  |  |
| --- | --- |
| Символы | |
| Read | 1 |
| ( | 2 |
| ) | 3 |
| , | 4 |
| ; | 5 |
| : = | 6 |
| \* | 7 |
| + | 8 |
| / | 9 |
| Идентификатор | |
| H | 10 1 |
| B | 10 2 |
| M | 10 3 |
| Pi | 10 4 |
| V | 10 5 |
| Константа | |
| 3.14 | 11 1 |
| 3 | 11 2 |

**Результат работы лексического анализатора:**

1 2 10 1 4 10 2 4 10 3 3 5

10 4 6 11 1 5

10 5 6 10 4 7 10 1 7 2 10 2 7 10 2 8 10 3 7 10 3 8 10 2 7 10 3 9 11 2

**Грамматика:**

1 <Список операторов> → <оператор>{;<оператор>}

2 <оператор> → <ввод> / <присваивание>

3 <Ввод> → Read (<список переменных>)

4 <список переменных> → <идентификатор> {,<идентификатор>}

5 <идентификатор>→ H / B / M / V / PI

6<Присваивание> → <идентификатор> : = <арифметическое выражение>

7<арифметическое выражение> → <слагаемое>{+<слагаемое>}

8 <слагаемое> → <множитель> {\*/:<множитель>}

9 <множитель> → (<арифметическое выражение>) / <идентификатор> / <константа>

10 <константа> → 3.14 / 3

**Описание алгоритма:**

Начало программы начинается с подключения библиотек.

Затем объявляются переменные, объявляем массив, куда заносится код лексического анализатора и переменные i=0, k=0, z=0, j=0, h=0. Выводим заголовок таблицы последовательности четвёрок синтаксического анализатора.

Задаёмся циклом от 0 до 48.

Задаёмсяусловиемдля**« ; ».** Следующие условие определяет символ READ. Делаю, это так, условие проверяет, не равен ли код “1т.е. READ” кодам идентификатора и константы «10 и 11», если нет, проверяем, находятся после него, с права, скобки, если да, то выводим READ на дисплейи следующим действием ищем параметры “H, B, M”.

Нахождение параметров “H, B, M”. Зададимся циклом от i до 10. Условиями проверяем, является ли параметр истинным, это будет тогда когда он сравнивая после этого параметра стоящие символы совпадут, то он выводится на экран, в противном случае вывод ошибки либо «не найден какой-то из трёх параметров, либо все параметры не верны».

Таким же способом проверяем остальной код программы.

**Текст программы:**

#include<stdio.h>

#include<conio.h>

int main()

{

char leks[48]={1,2,10,1,4,10,2,4,10,3,3,5,10,4,6,11,1,5,10,5,6,10,4,7,10,1,7,2,10,2,7,10,2,8,10,3,7,10,3,8,10,2,7,10,3,9,11,2};

inti=0,k=0,z=0,j=0,h=0,m=0;

printf("+---------------------------------------+ \n");

printf("¦Operacija ¦ Op1 ¦ Op2 ¦ Rezultat ¦ \n");

printf("+----------+--------+--------+----------¦");

for(i=0;i<48;i++)

{

if(leks[11]==5 &&leks[17]==5)

{

if(leks[i]==1 &&leks[i-1]!=10 &&leks[i-1]!=11)

{

if(leks[i+1]==2 &&leks[i+10]==3)

{

printf("\n¦READ ¦ ¦ ¦ ¦\n");

for(k=i;k<i+10;k++)

{

if(leks[k]==10 &&leks[k+1]==1 &&leks[k+2]==4) z=1;

if(leks[k]==10 &&leks[k+1]==2 &&leks[k+2]==4) j=1;

if(leks[k]==10 &&leks[k+1]==3) h=1;

}

if(z==1 && j==1 && h==1) {printf("¦PARAM1 ¦ H ¦ ¦ ¦\n");

printf("¦PARAM2 ¦ B ¦ ¦ ¦\n");

printf("¦PARAM3 ¦ M ¦ ¦ ¦\n");}

}

}

if(leks[i]==6)

{

if(leks[i-1]==4 &&leks[i-2]==10 &&leks[i+1]==11 &&leks[i+2]==1) printf("¦:= ¦ #3.14 ¦ ¦ PI ¦\n");

}

k=i+10;

if(leks[k]==7 &&leks[k-1]==2 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==2)

printf("¦\* ¦ B ¦ B ¦ i1 ¦\n");

k=i+16;

if(leks[k]==7 &&leks[k-1]==3 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==3)

printf("¦\* ¦ M ¦ M ¦ i2 ¦\n");

k=i+22;

if(leks[k]==7 &&leks[k-1]==2 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==3)

printf("¦\* ¦ B ¦ M ¦ i3 ¦\n");

k=i+13;

if(leks[k]==8 &&leks[k-1]==2 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==3)

printf("¦+ ¦ i1 ¦ i2 ¦ i4 ¦\n");

k=i+19;

if(leks[k]==8 &&leks[k-1]==3 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==2)

printf("¦+ ¦ i3 ¦ i4 ¦ i5 ¦\n");

k=i+6;

if(leks[k]==7 &&leks[k-1]==1 &&leks[k-2]==10 &&leks[k+1]==2 )

printf("¦\* ¦ H ¦ i5 ¦ i6 ¦\n");

k=i+3;

if(leks[k]==7 &&leks[k-1]==4 &&leks[k-2]==10 &&leks[k+1]==10 &&leks[k+2]==1)

printf("¦\* ¦ i6 ¦ PI ¦ i7 ¦\n");

k=i+25;

if(leks[k]==9 &&leks[k-1]==3 &&leks[k+1]==11 &&leks[k+2]==2) printf("¦/ ¦ i7 ¦ #3 ¦ i8 ¦\n");

k=i;

if(leks[k]==6 &&leks[k-1]==5 &&leks[k-2]==10) printf("¦:= ¦ i8 ¦ ¦ V ¦\n");

}

}

printf("+---------------------------------------+");

{

}

getch();

return 0;

}

**Результат работы программы:**

