Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Отчёт по лабораторной работе №3

по дисциплине «Лингвистические средства вычислительных систем»

Вариант 18

Выполнил студент гр.

Проверила:.

Тамбов 20

**Задание:** *Разработать программу синтаксического анализа методом операторного предшествования. В качестве исходных данных использовать 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

**Результат работы лексического анализатора:**

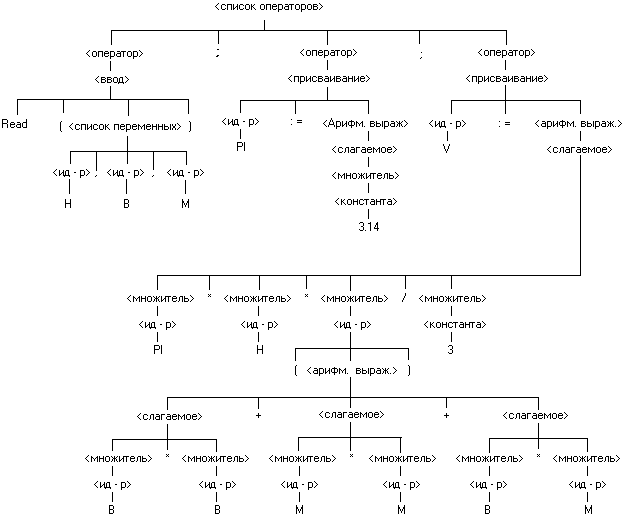
1 2 10 1 4 10 2 4 10 3 3 5

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

**Таблица предшествования**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лексема | Read | ( | ) | , | ; | : = | \* | + | / | ид | const |
| Read |  | = |  |  |  |  |  |  |  |  |  |
| ( |  | <∙ | = | <∙ |  |  |  | <∙ | <∙ | <∙ |  |
| ) |  |  | ∙> | = | ∙> |  |  |  | ∙> | <∙ |  |
| , |  |  |  | = |  |  |  |  |  | <∙ |  |
| ; |  |  |  |  |  |  |  |  |  | ∙> |  |
| : = |  |  |  |  |  |  |  |  |  | <∙ | <∙ |
| \* |  | <∙ | ∙> |  |  |  |  | ∙> | <∙ | <∙ |  |
| + |  |  | ∙> |  |  |  | <∙ | ∙> |  | <∙ |  |
| / |  |  |  |  |  |  |  |  |  |  | <∙ |
| ид |  |  | ∙> | ∙> |  | = | ∙> | ∙> |  |  |  |
| const |  |  |  |  | ∙> |  |  |  |  |  |  |



**Исходный код программы:**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

char\* Int(int one, int two)

{

if (one==1 && two==0) return "\nREAD";

if (one==2 && two==0) return "(";

if (one==3 && two==0) return ")";

if (one==5 && two==0) return ";\n";

if (one==6 && two==0) return ":=";

if (one==7 && two==0) return "\*";

if (one==8 && two==0) return "+";

if (one==9 && two==0) return "/";

if (one==10 && two==1) return "H ";

if (one==10 && two==2) return " B ";

if (one==10 && two==3) return " M ";

if (one==10 && two==4) return "PI ";

if (one==10 && two==5) return " V";

if (one==11 && two==1) return "3.14 ";

if (one==11 && two==2) return " 3\n";

return "";

}

int main()

{

int i=0,j=0,m=0,mass[3][31]=

{

1, 2, 10, 1, 4, 10, 2, 4, 10, 3, 3, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

10, 4, 6, 11, 1, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

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, 3, 9, 11, 2

};

int map[12][12]={0,0,0,0,0,0,0,0,0,0,0,0,

0,0,1,0,0,0,0,0,0,0,0,0,

0,0,2,1,2,0,0,0,2,2,2,0,

0,0,0,3,1,3,0,0,0,3,2,0,

0,0,0,0,1,0,0,0,0,0,2,0,

0,0,0,0,0,0,0,0,0,0,3,0,

0,0,0,0,0,0,0,0,0,0,2,2,

0,0,2,3,0,0,0,0,3,2,2,0,

0,0,0,3,0,0,0,2,3,0,2,0,

0,0,0,0,0,0,0,0,0,0,0,2,

0,0,0,3,3,0,1,3,3,0,0,0,

0,0,0,0,0,3,0,0,0,0,0,0

};

char str[80];

for(i=0;i<3;i++)

{ // Проверка на отношение <>

for(j=0;j<28;j++)

{

if(map[mass[i][j]][mass[i][j+1]]==2 && map[mass[i][j+1]][mass[i][j+3]]==3)

{// Если встретилось такое отношение, то выводим лексему

printf("%s",Int(mass[i][j+1],mass[i][j+2]));

mass[i][j+1]=0; // Проверка на нахождение повторяющихся символов и обнуление их.

mass[i][j+2]=0;

}

}

for(m=0;m<31;m++)

{ // Обработка идентификаторов

if(mass[i][m]!=0)

{

if(mass[i][m]==10 || mass[i][m]==11) {printf("%s",Int(mass[i][m],mass[i][m+1]));m++;}

else printf("%s",Int(mass[i][m],0));

}

}

for(m=0;m<10;m++)

{ // Далее выводим остальные лексемы в строке в соответствии с матрицей предшествий

if(map[mass[i][m]][mass[i][m+3]]==2 || map[mass[i][m]][mass[i][m+3]]==3)

{

printf("%s",Int(mass[i][m+3],0));

mass[i][m+3]=0;

}

}

}

return 0;

}

**Результат обработки:**

