Міністерство освіти і науки України

Національний технічний університет України

"Київський політехнічний інститут"

Факультет прикладної математики

Кафедра спеціалізованих комп’ютерних систем

Лабораторна робота №1

Дисципліна:

"Основи проектування трансляторів"

Тема:

"Лексичний аналізатор"

Варіант 14

|  |  |
| --- | --- |
| Виконав  Студент групи КВ-92  Сміюн В.В. | Перевірив  Марченко О.І. |

Київ 2011

Розробка лексичного аналізатора (сканеру, scanner )

Постановка задачі

Розробити програму лексичного аналізатора (ЛА) для підмножини мови програмування SIGNAL.

Програма має забезпечувати наступне (якщо це передбачається граматикою варіанту):

* згортання ідентифікаторів;
* згортання ключових слів;
* згортання цілих десяткових констант;
* згортання дійсних десяткових констант;
* згортання строкових констант, якщо вони визначені в заданій мові;

Також у всіх варіантах необхідно забезпечити:

* видалення коментарів, заданих у вигляді (\*<текст коментарю>\*)

Для кодування лексем необхідно використовувати числові діапазони, вказані в Таблиці 1.

Діапазони кодування лексем

|  |  |
| --- | --- |
| Вид лексеми | Числовий діапазон |
| односимвольні роздільники та знаки операцій (: / ; + і т.д.) | 0 – 255,  Тобто коди ASCII |
| багатосимвольні роздільники ( := ,<= ,<=, і т.д.) | 301 - 400 |
| цілі десяткові константи | 401 - 500 |
| символьні константи | 501 - 600 |
| рядкові константи | 601 - 700 |
| ключові слова (BEGIN, END, FOR) та ідентифікатори | 701 – 1000 |

**Входом** ЛА має бути наступне:

* вихідна програма, написана підмножиною мови SIGNAL відповідно до варіанту;
* таблиця кодів ASCII з атрибутами для визначення токенів;
* таблиця багато символьних роздільників;
* таблиця ідентифікаторів, в яку попередньо занесені ключові слова з атрибутом ключового слова;

**Вихід** ЛА має бути наступним:

* закодований рядок лексем;
* таблиці ідентифікаторів, числових, символьних та рядкових констант, сформовані для конкретної програми;

Варіант 14

<signal-program> --> <program>

<program> --> PROGRAM <procedure-identifier> ; <block>.

<block> --> <declarations> BEGIN <statements-list> END

<declarations> --> <label-declarations>

<label-declarations> --> LABEL <unsigned-integer> <labels-list>; |  
<empty>

<labels-list> --> , <unsigned-integer> <labels-list> |  
<empty>

<statements-list> --> <statement> <statements-list> |  
<empty>

<statement> --> <unsigned-integer> : <statement> |  
GOTO <unsigned-integer> ; |  
<condition-statement> ENDIF ;|  
;

<condition-statement> --> <incomplete-condition-statement><alternative-part>

<incomplete-condition-statement> --> IF <conditional-expression> THEN <statements-list>

<conditional-expression> --> <variable-identifier> = <unsigned-integer>

<alternative-part> --> ELSE <statements-list> |  
<empty>

<variable-identifier> --> <identifier>

<procedure-identifier> --> <identifier>

<identifier> --> <letter><string>

16 23.<string> --> <letter><string> |  
<digit><string> |  
<empty>

<unsigned-integer> --> <digit><digits-string>

<digits-string> --> <digit><digits-string> |  
<empty>

<digit> --> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<letter> --> A | B | C | D | ... | Z

dg

dg

Інший символ

dg, lt

lt

Інший символ

Інший символ

Інший символ

\*D

(

)

\*D

\*D

eof

Інший символ

eof

Будь-який роздільний символ

dm ( крім “(” )

ws

er

Будь-який символ

eof

PROGRAM ONE;

LABEL 10;

BEGIN

56:

IF IDN = 22 THEN

GOTO 56;

ELSE

GOTO 20;

ENDIF;

20:

GOTO 56;

END

(\*\*\*)

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Menus, StdCtrls, CheckLst;

type

Tabl= record

key: string;

number: integer;

end;

TSymbol = record

value: Char;

attr: Byte;

end;

TForm1 = class(TForm)

OpenDialog1: TOpenDialog;

MainMenu1: TMainMenu;

Ffle1: TMenuItem;

Open1: TMenuItem;

Exit1: TMenuItem;

StaticText1: TStaticText;

StaticText2: TStaticText;

StaticText3: TStaticText;

Button1: TButton;

Memo1: TMemo;

Memo2: TMemo;

Memo3: TMemo;

procedure Open1Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

function Gets: TSymbol;

procedure FillAttributes;

Procedure ConstTabSearch(bf1:string;Te:integer; var num:integer;var C:boolean);

Procedure IdnTabSearch(bf2:string; Te:integer; var num:integer; var IP:boolean);

Procedure KeyTabSearch(bf:string; var num:integer; var K:boolean);

Procedure ConstTabForm(bf3:string;num:integer);

Procedure IdnTabForm(bf4:string;num:integer);

procedure main();

procedure Exit1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

Attributes: array [1..256] of Byte;

symbol: TSymbol;

lexCode: Word;

buf, buf1: string;

fname: TFileName;

SuppressOutput,CT,KT,IT: Boolean;

FINP,FOUT, tablf: TextFile;

char,chIdN, chKT:integer;

ind:integer;

j,CTE,IDN,ch:byte;

KeyTabl:array[1..9] of Tabl;

ConstTabl:array[1..99] of Tabl;

IdnTabl:array[1..99] of Tabl;

implementation

{$R \*.dfm}

procedure TForm1.Open1Click(Sender: TObject);

begin

// створення об'єкту OpenDialog - назначення на нашу змінну OpenDialog

openDialog1 := TOpenDialog.Create(self);

// Установка початкового каталогу, щоб зробити його поточним

openDialog1.InitialDir := GetCurrentDir;

// Тільки дозволені існуючі файли можуть бути вибрані

openDialog1.Options := [ofFileMustExist];

// Дозволено вибирати тільки .txt і .sig файли

openDialog1.Filter :=

'text files|\*.txt|signal files|\*.sig';

// Вибір файлів мови Signal за замовчуванням

openDialog1.FilterIndex := 2;

// Показ діалогу відкриття файлу

if openDialog1.Execute

then begin

ShowMessage('File : '+openDialog1.FileName);

fname :=openDialog1.FileName;

end

else ShowMessage('Відкриття відмінено');

// Звільнення діалогу

openDialog1.Free;

end;

function TForm1.Gets: TSymbol;

begin

Read(FINP, Result.value);

ch:= Ord(Result.value);

Result.attr := Attributes[ch];

end;

//Опис таблиць

procedure TForm1.FillAttributes;

var i: integer;

begin

//error

for i:=1 to 256 do

Attributes[i]:=5;

//числова константа

for i:=48 to 57 do

Attributes[i]:=1;

for i:=65 to 90 do

//ідентифікатор

Attributes[i]:=2;

//коментар

Attributes[40]:=3;

//роздільник

Attributes[44]:=4;

Attributes[46]:=4;

Attributes[58]:=4;

Attributes[59]:=4;

Attributes[61]:=4;

//whitespace

Attributes[9]:=0;

Attributes[10]:=0;

Attributes[13]:=0;

Attributes[32]:=0;

KeyTabl[1].key:='PROGRAM';

KeyTabl[1].number:=701;

KeyTabl[2].key:='LABEL';

KeyTabl[2].number:=702;

KeyTabl[3].key:='BEGIN';

KeyTabl[3].number:=703;

KeyTabl[4].key:='GOTO';

KeyTabl[4].number:=704;

KeyTabl[5].key:='ENDIF';

KeyTabl[5].number:=705;

KeyTabl[6].key:='IF';

KeyTabl[6].number:=706;

KeyTabl[7].key:='THEN';

KeyTabl[7].number:=707;

KeyTabl[8].key:='ELSE';

KeyTabl[8].number:=708;

KeyTabl[9].key:='END';

KeyTabl[9].number:=709;

end;

Procedure TForm1.ConstTabSearch(bf1:string;Te:integer; var num:integer;var C:boolean);

var i:byte;

Begin

for i:=1 to Te do

begin

if bf1=ConstTabl[i].Key then

begin

num:=ConstTabl[i].number;

C:=true;

break;

end;

C:=false;

end;

end;

Procedure TForm1.IdnTabSearch(bf2:string; Te:integer; var num:integer; var IP:boolean);

var i:byte;

Begin

for i:=1 to Te do

begin

if bf2=IdnTabl[i].Key then

begin

num:=IdnTabl[i].number;

IP:=true;

break;

end;

IP:=false;

end;

end;

Procedure TForm1.KeyTabSearch(bf:string; var num:integer; var K:boolean);

var i:byte;

Begin

for i:=1 to 9 do

begin

if bf=KeyTabl[i].Key then

begin

num:=KeyTabl[i].number;

K:=true;

break;

end

else

K:=false;

end;

end;

Procedure TForm1.ConstTabForm(bf3:string;num:integer);

Begin

ConstTabl[num].key:=bf3;

ConstTabl[num].number:=num+400;

end;

Procedure TForm1.IdnTabForm(bf4:string;num:integer);

Begin

IdnTabl[num].key:=bf4;

IdnTabl[num].number:=num+709;

end;

procedure TForm1.main();

Begin

//відкриття файлу початкової програми

AssignFile(FINP,fname);

reset(FINP);

//вивід початкового файлу на екран

readln(FINP, buf1);

while (not EOF(FINP)) do

begin

memo3.Lines.Add(buf1);

readln(finp, buf1);

end;

reset(FINP);

AssignFile(tablf,'tabl.txt');

AssignFile(FOUT,'Output.txt');

rewrite(FOUT);

rewrite(tablf);

writeln(fout, 'lexem string:');

writeln(fout, ' ');

ind:=0;

CTE:=0;IDN:=0;

CT:=false; KT:=false; IT:=false;

//початкове встановлення таблиць ідентифікаторів і констант

FillAttributes;

if eof(FINP) then

begin

memo1.lines.add('Empty file');

Writeln(FOUT,'Empty file');

end;

symbol:=Gets;

repeat

buf := '';

lexCode := 0;

SuppressOutput := False;

case symbol.attr of

0: (\*whitespace\*)

begin

while not eof(FINP) do

begin

symbol := Gets;

if symbol.attr <> 0 then

Break;

end;

SuppressOutput := True;

end;

1: (\*константа\*)

begin

while not eof(FINP) and (symbol.attr = 1) do

begin

buf := buf + symbol.value;

symbol := Gets;

end;

ConstTabSearch(buf,CTE,char,CT);

if CT=true then

lexCode := char

else

begin

CTE:=CTE+1;

ConstTabForm(buf,CTE);

lexCode := CTE+400;

end;

end;

2: (\*ідентифікатор\*)

begin

while not eof(FINP) and ((symbol.attr = 2)

or (symbol.attr = 1)) do

begin

buf := buf + symbol.value;

symbol := Gets;

end;

KeyTabSearch(buf,chKT,KT);

if KT=true then

lexCode := chKT

else begin

IdnTabSearch(buf,IDN,chIDN,IT);

if IT=true then

lexCode :=chIDN

else

begin

IDN:=IDN+1;

IdnTabForm(buf,IDN);

lexCode := IDN+709;

end;

end;

end;

3: (\*можливий коментар, тобто зустрінута '(' \*)

begin

if eof(FINP) then

begin

memo1.Lines.add('Illegal symbol');

Write(fout,' Illegal symbol');

inc(ind);

SuppressOutput :=true;

end

else

begin

symbol := Gets;

if symbol.value = '\*' then

begin

if eof(FINP) then

begin

memo1.Lines.add('\*) expected but end of file found');

Writeln(fout,'\*) expected but end of file found');

inc(ind);

SuppressOutput :=true;

end

else

begin

symbol := Gets;

repeat

while not eof(FINP) and (symbol.value <> '\*') do

symbol := Gets;

if eof(FINP) then //якщо кінець файла

begin

memo1.Lines.add('\*) expected! Error!');

inc(ind);

Writeln(fout, ' \*) expected! Error!');

symbol.value := '+'; // все що завгодно, але не ')'

SuppressOutput :=true;

Break;

end

else //була '\*' і немає кінця файла

symbol := Gets;

until symbol.value = ')';

if symbol.value = ')' then

SuppressOutput := True;

if not eof(FINP) then

symbol := Gets;

end;

end

else

begin

memo1.Lines.add('Illegal symbol');

inc(ind);

Write(fout,' Illegal symbol');

SuppressOutput :=true;

end;

end;

end;

4: (\*роздільник\*)

begin

lexCode := Ord(symbol.value);

symbol := Gets;

end;

5: (\*помилка\*)

begin

memo1.lines.add('Illegal symbol');

inc(ind);

Write(fout,' Illegal symbol');

symbol := Gets;

SuppressOutput:= true;

end;

end; (\*case\*)

if not SuppressOutput then

begin memo1.Lines.add('Output: '+IntToStr(lexCode));

write(FOUT,' ',lexCode);

inc(ind);

end;

until eof(FINP)

//Вивід таблиць

Writeln(tablf,'');

ind:=0;

memo2.Lines.Strings[ind]:= 'Keywords Table:';

Writeln (tablf,'Keywords Table:');

Writeln(tablf,'');

for j:=1 to 9 do

begin

memo2.Lines.add( inttostr(KeyTabl[j].number)+' '+ KeyTabl[j].key);

Writeln(tablf, KeyTabl[j].number,' ', KeyTabl[j].key);

end;

if CTE<>0 then

begin

Writeln(tablf,'');

memo2.Lines.Add('Constant Table:');

Writeln (tablf,'Constant Table:');

Writeln(tablf,'');

for j:=1 to CTE do

begin

memo2.Lines.Add(inttostr(ConstTabl[j].number)+' '+ ConstTabl[j].key);

Writeln(tablf, ConstTabl[j].number,' ', ConstTabl[j].key);

end;

end;

if IDN<>0 then

begin

Writeln(FOUT,'');

memo2.Lines.Add('Identifier Table:');

Writeln (tablf,'Identifier Table:');

Writeln(tablf,'');

for j:=1 to IDN do

begin

memo2.Lines.Add(inttostr(IdnTabl[j].number)+' '+ IdnTabl[j].key);

Writeln(tablf, IdnTabl[j].number,' ', IdnTabl[j].key);

end;

end;

Writeln(tablf,'');

memo2.Lines.Add('Separators Table:');

Writeln (tablf,'Separators Table:');

Writeln(tablf,'');

memo2.Lines.Add('44 ,');

Writeln(tablf,'44',' ',',' );

memo2.Lines.Add('46 .');

Writeln(tablf,'46',' ','.');

memo2.Lines.Add('58 :');

Writeln(tablf,'58',' ',':');

memo2.Lines.Add('59 ;');

Writeln(tablf,'59',' ',';');

memo2.Lines.Add('61 =');

Writeln(tablf,'61',' ','=');

CloseFile(fout);

CloseFile(finp);

CloseFile(tablf);

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

main();

end;

procedure TForm1.Exit1Click(Sender: TObject);

begin

Close;

end;

end.

lexem string:

701 710 59 702 401 44 402 44 403 59 703 706 404 61 404 707 704 402 708 705 59 Illegal symbol 405 44 709

Keywords Table:

701 PROGRAM

702 LABEL

703 BEGIN

704 GOTO

705 ENDIF

706 IF

707 THEN

708 ELSE

709 END

Constant Table:

401 20

402 56

403 43

404 22

405 88

Identifier Table:

710 PROG

Separators Table:

44 ,

46 .

58 :

59 ;

61 =