

Математическое обеспечение ЭВМ

Лабораторная работа № 2: “Построение лексического блока”

Разработать программу, реализующую лексический блок для заданного языка.

Максимальная длина идентификатора - не менее 10 литер.

Лексический блок реализуется как подпрограмма, вызываемая синтаксическим анализатором.

Лексический блок для каждой лексемы должен указывать ее класс и значение.

Лексический блок должен строить таблицу(ы) идентификаторов и констант.

Возможны следующие виды лексем:

- идентификаторы;
- ключевые слова;
- целые числа;
- разделители.

Программа разрабатывается как приложение с графическим интерфейсом на языке C# в среде Visual Studio.

Отчёт по лабораторной работе должен содержать:

- Титульный лист
- Задание на работу
- Описание работы программы
- Грамматику для лексем
- Граф лексического блока
- Алгоритмы выполнения основных операций на псевдокоде
- Тесты
- Распечатки экранов при работе программы
- Листинг программы

Оператор присваивания:
`<идентификатор>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)<оператор>[else <оператор>]`

Оператор цикла:
`while(<лог.выр.>)<оператор>`

Арифметическое выражение:

`<E> ::= <T><E-список>`
`<E-список> ::= +<T><E-список>`
`<E-список> ::= ε`
`<T> ::= <F><T-список>`
`<T-список> ::= *<F><T-список>`
`<T-список> ::= ε`
`<F> ::= <P>`
`<F> ::= <R>`
`<P> ::= <Id>`
`<P> ::= <Int>`
`<R> ::= <P>^<P>`

Логическое выражение:

`<лог.выр.> ::= <P><лог.опер.><P>`
`<лог.опер.> ::= >=`
`<лог.опер.> ::= <=`

Пример программы:

```
d=2*3+1;  
a=3+2*d^2;  
c=2*d+3*a;  
if(d>=c) a=d^2; else a=d^3;  
i=1;  
while(i<=10) i=i+1;
```

Оператор присваивания:
`<идентификатор>=<ар.выр.>;`

Условный оператор:
`ЕСЛИ(<лог.выр.>)`
`<совокупность операторов>`
`ИНАЧЕ`
`<совокупность операторов>`
`КОНЕЦ_ЕСЛИ`

Оператор цикла:
`ЦИКЛ_ПОКА(<лог.выр.>)`
`<совокупность операторов>`
`КОНЕЦ_ЦИКЛ`

Арифметическое выражение:
`<E> ::= <T> <E-список>`
`<E-список> ::= + <T> <E-список>`
`<E-список> ::= ε`
`<T> ::= <F> <T-список>`
`<T-список> ::= * <F> <T-список>`
`<T-список> ::= ε`
`<F> ::= <Id>`
`<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F> <лог.опер.> <F>`
`<лог.опер.> ::= >`
`<лог.опер.> ::= <`

Пример программы:

```
d=2*3+1;  
a=3+2*d;  
c=2*d+3*a;  
ЕСЛИ(d>c)  
    a=d*2;  
ИНАЧЕ  
    a=d;  
КОНЕЦ_ЕСЛИ  
i=1;  
ЦИКЛ_ПОКА(i<10)  
    i=i+1;  
КОНЕЦ_ЦИКЛ
```

Вариант 3 - Вагапов А.Р.

Оператор присваивания:
`<ид.>=<ар.выр.>;`

Условный оператор:
`ЕСЛИ(<лог.выр.>) ТО`
`<совокупность операторов>`
`ИНАЧЕ`
`<совокупность операторов>`
`КОНЕЦ_ЕСЛИ`

Оператор цикла:
`ЦИКЛ(<ид.>=<F>,<F>)`
`<совокупность операторов>`
`КОНЕЦ_ЦИКЛ`

Арифметическое выражение:
`<E> ::= <T> <E-список>`
`<E-список> ::= + <T> <E-список>`
`<E-список> ::= ε`
`<T> ::= <F> <T-список>`
`<T-список> ::= * <F> <T-список>`
`<T-список> ::= ε`
`<F> ::= <Id>`
`<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F> <лог.опер.> <F>`
`<лог.опер.> ::= >`
`<лог.опер.> ::= <`

Пример программы:

```
d=2*3+1;  
a=3+2*d;  
c=2*d+3*a;  
ЕСЛИ(d>c)ТО  
    a=d*2;  
ИНАЧЕ  
    a=d+3;  
КОНЕЦ_ЕСЛИ  
j=3;  
ЦИКЛ(i=1,d)  
    a=a+j;  
КОНЕЦ_ЦИКЛ
```

Вариант 4 - Гаджиханов В.

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)then<оператор>[else <оператор>]`

Оператор цикла:
`for(<ид.>:=<F>,<F>[,<F>])<оператор>`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= -<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= %<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >
<лог.опер.> ::= <`

Пример программы:

```
d:=23-1;  
a:=3-254%d;  
c:=200%d-367%a;  
if(d>c) then a:=d%2; else a:=d%3;  
h:=10;  
for(i:=1,d,2) h:=h-1;
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)<оператор>[else <оператор>]`

Оператор цикла:
`while(<лог.выр.>) do <оператор>`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <P>
<F> ::= <R>
<P> ::= <Id>
<P> ::= <Int>
<R> ::= <P>^<P>`

Логическое выражение:
`<лог.выр.> ::= <P><лог.опер.><P>
<лог.опер.> ::= >=`
`<лог.опер.> ::= <=`

Пример программы:

```
d:=2*3+1;  
a:=3+2*d^2;  
c:=2*d+3*a;  
if(d>=c) a:=d^2; else a:=d^3;  
i:=1;  
while(i<=10) i:=i+1;
```

Оператор присваивания:
`<ид.>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)then<оператор>[else <оператор>]`

Оператор цикла:
`for <ид.>=<F> to <F> do <оператор>`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= -<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= /<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= =
<лог.опер.> ::= #`

Пример программы:

```
d=23-1;  
a=3-254/d;  
c=200/d-367/a;  
if(d#c) a=d-2; else a=d/3;  
h=10;  
for i=1 to d do h=h-1;
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)<оператор>[else <оператор>]`

Оператор цикла:
`for(<ид.>:=<F>,<F>)
 <совокупность операторов>
end_for`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= /<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >=
<лог.опер.> ::= ==`

Пример программы:

```
d:=13+1;
a:=3+254/d;
c:=200/d+367/a;
if(d>=c) a:=d/2; else a:=d/3;
h:=1;
for(i:=1,d)
    h:=h+1;
    a:=a+h;
end_for
```

Оператор присваивания:
`<идентификатор>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)then<оператор>[else <оператор>]`

Оператор цикла:
`while(<лог.выр.>)
 <совокупность операторов>
end_while`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <P>
<F> ::= <R>
<P> ::= <Id>
<P> ::= <Int>
<R> ::= <P>^<P>`

Логическое выражение:
`<лог.выр.> ::= <P><лог.опер.><P>
<лог.опер.> ::= =
<лог.опер.> ::= #`

Пример программы:

```
d=2*3+1;  
a=3+2*d^2;  
c=2*d+3*a;  
if(d=c) a=d*2; else a=d^3;  
i=1;  
while( i # 10)  
    i=i+1;  
end_while
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>) <оператор> [else <оператор>]`

Оператор цикла:
`do <совокупность операторов> until <лог.выр.>;`

Арифметическое выражение:
`<E> ::= <T> <E-список>
<E-список> ::= -<T> <E-список>
<E-список> ::= ε
<T> ::= <F> <T-список>
<T-список> ::= /<F> <T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F> <лог.опер.> <F>
<лог.опер.> ::= >
<лог.опер.> ::= <`

Пример программы:

```
d:=23-1;
a:=3-254/d;
c:=200/d-367/a;
if(d>c) a:=d/2; else a:=d/3;
h:=10;
do
    h:=h-1;
until h<0;
```

Оператор присваивания:
`<ид.>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)then<оператор>[else <оператор>]`

Оператор цикла:
`do <совокупность операторов> while <лог.выр.>;`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= =
<лог.опер.> ::= #`

Пример программы:

```
d=23*2+1;
a=3+2*d;
c=2*d+3*a;
if(d#c) a=d*2; else a=d+3;
h=10;
do
    h=h-1;
while h#0;
```

Оператор присваивания:
<ид.>=<ар.выр.>;

Условный оператор:
if(<лог.выр.>) <совокупность операторов>
[else <совокупность операторов>]
endif

Оператор цикла:
do <ид.>=<F>,<F> begin <совокупность операторов> end

Арифметическое выражение:
<E> ::= <T><E-список>
<E-список> ::= -<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= %<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>

Логическое выражение:
<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >
<лог.опер.> ::= <

Пример программы:

```
d=23-1;
a=3-254%d;
c=200%d-367%a;
if(d>c) a=d%2; else a=d%3; endif
h=10;
do i=1,d begin h=h-1; end
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>) then <совок. операторов> [else < совок. операторов>] endif`

Оператор цикла:
`for <ид.>:=<F> to <F> [step <F>] do <совок. операторов> end`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= =
<лог.опер.> ::= !=`

Пример программы:

```
a:=16*3+1;
b:=11+2*a;
c:=3*a+2;
if(b!=c) then a:=4*b; b:=12; else a:=2*b+3; endif
k:=0; s:=0;
for i:=1 to 10 step 2 do
    k:=k+1;
    s:=s+k;
end
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>) <оператор> [else <оператор>]`

Оператор цикла:
`repeat <совок. операторов> until <лог.выр.>;`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= =
<лог.опер.> ::= !=`

Пример программы:

```
a:=16*3+1;
b:=11+2*a;
c:=3*a+2;
if(b!=c) a:=4*b; else a:=2*b+3;
k:=0; s:=0;
repeat
    k:=k+1;
    s:=s+k;
until k=10;
```

Оператор присваивания:
`<ид.>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>) <совок. операторов> [else < совок. операторов>] endif`

Оператор цикла:
`for (<ид.>=<Е> ; <лог.выр.> ; <ид.>=<Е>) <совок. операторов> end`

Арифметическое выражение:
`<Е> ::= <Т><Е-список>
<Е-список> ::= +<Т><Е-список>
<Е-список> ::= ε
<Т> ::= <F><Т-список>
<Т-список> ::= *<F><Т-список>
<Т-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >
<лог.опер.> ::= <`

Пример программы:

```
a=16*3+1;
b=11+2*a;
c=3*a+2;
if(b>c) a=4*b; b=12; else a=2*b+3; endif
k=0; s=0;
for( i=1; i<10; i=i+1)
    k=k+1;
    s=s+k;
end
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>) then <оператор> [else <оператор>]`

Оператор цикла:
`for (<ид.>:=<F>, <F>) while (<лог.выр.>) do <оператор>`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >=
<лог.опер.> ::= <`

Пример программы:

```
a:=16*3+1;
b:=11+2*a;
c:=3*a+2;
if(b>=c) then a:=4*b; else a:=2*b+3;
s:=0;
for ( i:=1,10 ) while ( s<30 ) do
  s:=s+i;
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`ЕСЛИ(<лог.выр.>) ТО <совок. операторов> [ИНАЧЕ < совок. операторов>] КОНЕЦ`

Оператор цикла:
`ЦИКЛ <совок. операторов> ПОКА <лог.выр.>;`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= =
<лог.опер.> ::= !=`

Пример программы:

```
a:=16*3+1;
b:=11+2*a;
c:=3*a+2;
ЕСЛИ (b!=c) ТО a:=4*b; b:=12; ИНАЧЕ a:=2*b+3; КОНЕЦ
k:=0; s:=0;
ЦИКЛ
    k:=k+1;
    s:=s+k;
ПОКА k!=10;
```

Оператор присваивания:
`<ид.>:=<ар.выр.>;`

Условный оператор:
`ЕСЛИ <лог.выр.> ТО <совок. операторов> [ИНАЧЕ < совок. операторов>] КОНЕЦ`

Оператор цикла:
`ЦИКЛ (<ид.>:=<E>, <лог.выр.>, <ид.>:=<E>) <совок. операторов> КОНЕЦ`

Арифметическое выражение:
`<E> ::= <T><E-список>
<E-список> ::= +<T><E-список>
<E-список> ::= ε
<T> ::= <F><T-список>
<T-список> ::= *<F><T-список>
<T-список> ::= ε
<F> ::= <Id>
<F> ::= <Int>`

Логическое выражение:
`<лог.выр.> ::= <F><лог.опер.><F>
<лог.опер.> ::= >
<лог.опер.> ::= <`

Пример программы:

```
a:=16*3+1;
b:=11+2*a;
c:=3*a+2;
ЕСЛИ b>c ТО a:=4*b; b:=12; ИНАЧЕ a:=2*b+3; КОНЕЦ
k:=0; s:=0;
ЦИКЛ ( i:=1, i<10, i:=i+1 )
    k:=k+1;
    s:=s+k;
КОНЕЦ
```

Оператор присваивания:
`<идентификатор>=<ар.выр.>;`

Условный оператор:
`if(<лог.выр.>)<оператор>[else <оператор>]`

Оператор цикла:
`ЦИКЛ(<лог.выр.>)<оператор>`

Арифметическое выражение:

`<E> ::= <T><E-список>`
`<E-список> ::= +<T><E-список>`
`<E-список> ::= ε`
`<T> ::= <F><T-список>`
`<T-список> ::= *<F><T-список>`
`<T-список> ::= ε`
`<F> ::= <P>`
`<F> ::= <R>`
`<P> ::= <Id>`
`<P> ::= <Int>`
`<R> ::= <P>^<P>`

Логическое выражение:

`<лог.выр.> ::= <P><лог.опер.><P>`
`<лог.опер.> ::= >=`
`<лог.опер.> ::= <=`

Пример программы:

```
d=2*3+1;  
a=3+2*d^2;  
c=2*d+3*a;  
if(d>=c) a=d*2; else a=d^3;  
i=1;  
ЦИКЛ(i<=10) i=i+1;
```