

**Санкт-Петербургский национальный  
исследовательский университет  
информационных технологий, механики и  
ОПТИКИ**

Кафедра информатики и прикладной математики

**Основы разработки компиляторов**

Лабораторная работа 1

"Разработка лексического анализатора"

Вариант 7

**Выполнил:** Шкаруба Н.Е.

Группа: Р3318

2016 г

## Цель работы

Разработать лексический анализатор.

## БНФ

```
<Программа> ::= <Объявление переменных> <Описание  
вычислений>  
<Описание вычислений> ::= Begin <Список операторов> End.  
<Объявление переменных> ::= Var <Список переменных>  
<Список переменных> ::= <Идент>; | <Идент>, <Список  
переменных> |  
<Идент>; <Список переменных>  
<Список операторов> ::= <Оператор> | <Оператор>  
<Список операторов>  
<Оператор> ::= <Присваивание> | <Сложный  
оператор> | <Составной оператор>  
<Составной оператор> ::= Begin <Список операторов> End  
<Присваивание> ::= <Идент> = <Выражение>;  
<Выражение> ::= <Ун.оп.> <Подвыражение> |  
<Подвыражение>  
<Подвыражение> ::= (<Выражение>) | <Операнд> |  
<Подвыражение> <Бин.оп.> <Подвыражение>  
<Ун.оп.> ::= "-" | "not"  
<Бин.оп.> ::= "-" | "+" | "*" | "/" | "***" | ">" | "<" | "=="  
<Операнд> ::= <Идент> | <Const>  
| <Сложный оператор>:: IF "<Выражение>" <Оператор> |  
<Идент> ::= <Буква> <Идент> | <Буква>  
<Const> ::= <Цифра> <Const> | <Цифра>
```

## Список классов лексем реализуемого языка

```
Const(0)  
Id(1)  
Unary(2)  
Additive(3)  
Multiplicative(4)  
Logical(5)  
Assignment(6)  
Delimiter(7)  
Bracket(8)  
Error(9)  
EOF(10)  
KeyWord(1000)
```

## Описание классов программы

*Lexeme*: пользовательский тип.

*LexType*: определяет используемые типы лексем.

*Logger*: формирует данные для вывода результата в файл.

*Scanner*: блок сканирования (сканер) проходит по символно потоку программы и определяет

каждую лексему, устанавливая ее принадлежность  
тому или иному классу.

## Контрольные примеры

```
var
  a,m,n,sum,a;
begin
  m := a+1;
  n := m;
  sum := ((sum + m)+1)+2+3*5;
```

```
if (sum<m*1000) a:=m/1;
// Comment
sum:=sum*2;
n:=1000/sum;
end.
```

=IDs=	<0;7>	<0;6>
<a;0>	<2;1>	<1;1>
<m;1>	<0;7>	<0;4>
<n;2>	<3;1>	<0;0>
<sum;3>	<0;7>	<1;7>
=Consts=	<0;1>	<0;4>
<1;0>	<1;7>	<1;4>
<2;1>	<0;1000>	<6;1>
<3;2>	<1;1>	<7;1>
<0;3>	<0;6>	<8;1>
<3e8;4>	<0;1>	<9;1>
=Delims=	<1;3>	<!;10>
<;;0>	<0;0>	<1;4>
<;;1>	<1;7>	<3;1>
=OpenBrackets=	<2;1>	<0;6>
<;0>	<0;6>	<3;1>
=CloseBrackets=	<1;1>	<1;4>
<;0>	<1;7>	<1;7>
=KeyWords=	<3;1>	<2;1>
<begin;0>	<0;6>	<0;6>
<end;1>	<0;8>	<4;0>
<end.;2>	<0;8>	<0;4>
<if;3>	<3;1>	<3;1>
<var;4>	<1;3>	<1;7>
=Logical=	<1;1>	<2;1000>
<>;0>	<0;9>	<-2;11>
<<;1>	<1;3>	
<==;2>	<0;0>	
=Unary=	<0;9>	
<-;0>	<1;3>	
<not;1>	<1;0>	
=Additive=	<1;3>	
<-;0>	<2;0>	
<+;1>	<1;4>	
=Multiplicative=	<1;7>	
</;0>	<4;1>	
<;1>	<0;8>	
<;2>	<3;1>	
=Assignment=	<1;5>	
<:=;0>	<1;1>	
=Lexemes=	<1;4>	
<4;1000>	<3;0>	
<0;1>	<0;9>	
<0;7>	<5;1>	
<1;1>	<0;1>	