

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

Лабораторная работа №1
по дисциплине
«Основы разработки компиляторов»
Вариант 7

Выполнил:
Петкевич Константин Вячеславович
Группа Р3318

Преподаватель:
Лаздин Артур Вячеславович

Порядок выполнения работы:

- 1.1. По варианту задания определить, какие классы лексем будут в вашем языке.
 - 1.2. Составить контрольные примеры на реализуемом языке. Хотя бы один пример должен проверять поведение вашей программы при наличии недопустимых символов в транслируемом файле.
 - 1.3. Запрограммировать и отладить модуль сканирования. Выполнить тестирование на контрольных примерах. Результатом работы должна быть таблица, содержащая лексемы и признаки их классов, для числовых констант их внутреннее представление (шестнадцатеричное). Необходимо включить в результирующий файл информацию о номерах строк исходного текста транслируемой программы.
- Одинаковые идентификаторы и константы в таблицу повторно не записываются. Необходимо предусмотреть восстановление после ошибок.
- 1.4. Оформить отчет.

БНФ:

```
<Программа> ::= <Объявление переменных> <Описание вычислений>
<Описание вычислений> ::= <Список операторов>
<Объявление переменных> ::= Var <Список переменных> ;
<Список переменных> ::= <Идент> | <Идент> , <Список переменных>
<Список операторов> ::= <Оператор> | <Оператор> <Список операторов>
<Оператор> ::= <Присваивание> | <Сложный оператор>
<Присваивание> ::= <Идент> := <Выражение> ;
<Выражение> ::= <Ун.оп.> <Подвыражение> | <Подвыражение>
<Подвыражение> ::= ( <Выражение> ) | <Операнд> |
<Подвыражение> <Бин.оп.> <Подвыражение>
<Ун.оп.> ::= "-"
<Бин.оп.> ::= "-" | "+" | "*" | "/" | "<" | ">" | "=="
<Операнд> ::= <Идент> | <Const>
|<Сложный оператор> ::= IF <Выражение> THEN <Оператор> |
IF <Выражение> THEN <Оператор> ELSE <Оператор>
<Идент> ::= <Буква> <Идент> | <Буква>
<Const> ::= <Цифра> <Const> | <Цифра>
```

Используемые классы лексем:

Const (0)	// константы - числа
Id (1)	// идентификаторы – буквы (переменные, функции, ...)
Unary (2)	// '-' или 'not'
Add (3)	// '+' или '-'
Multiple (4)	// '*' или '/' или ' ' или '=='
Compare (5)	// '>' или '<'
Assign (6)	// ':='
Delimiter (7)	// ';' или ','
Bracket (8)	// '(' или ')'
Error (9)	
EOF (10)	
Keyword (100)	// 'var' 'IF' 'THEN' 'ELSE'

Пример входной программы:

Var a,b,sum;

a := 6;

b := 71;

sum := ((sum - a) + 1) / 1;

IF (sum < 200) THEN sum := 1;

Результат выполнения программы:

< 0 ; 9 >	< 2 ; 0 >	< sum ; 2 >
< 0 ; 1 >	< 1 ; 8 >	Add class with cIID == 3
< 0 ; 7 >	< 0 ; 4 >	< - ; 0 >
< 1 ; 1 >	< 2 ; 0 >	< + ; 1 >
< 0 ; 7 >	< 1 ; 7 >	Multiple class with cIID == 4
< 2 ; 1 >	< 1 ; 9 >	< / ; 0 >
< 1 ; 7 >	< 0 ; 8 >	Compare class with cIID == 5
< 0 ; 1 >	< 2 ; 1 >	< < ; 0 >
< 0 ; 6 >	< 0 ; 5 >	Assign class with cIID == 6
< 1 ; 6 >	< 3 ; 0 >	< : ; 0 >
< 0 ; 0 >	< 1 ; 8 >	< = ; 1 >
< 1 ; 7 >	< 2 ; 9 >	Delimiter class with cIID == 7
< 1 ; 1 >	< 2 ; 1 >	< , ; 0 >
< 0 ; 6 >	< 0 ; 6 >	< ; ; 1 >
< 1 ; 6 >	< 1 ; 6 >	Bracket class with cIID == 8
< 1 ; 0 >	< 2 ; 0 >	< (; 0 >
< 1 ; 7 >	< 1 ; 7 >	<) ; 1 >
< 2 ; 1 >		Keyword class with cIID == 9
< 0 ; 6 >	Const class with cIID == 0	< Var ; 0 >
< 1 ; 6 >	< 6 ; 0 >	< IF ; 1 >
< 0 ; 8 >	< 71 ; 1 >	< THEN ; 2 >
< 0 ; 8 >	< 1 ; 2 >	
< 2 ; 1 >	< 200 ; 3 >	
< 0 ; 3 >	Id class with cIID == 1	
< 0 ; 1 >	< a ; 0 >	
< 1 ; 8 >	< b ; 1 >	
< 1 ; 3 >		

Вывод: в результате выполнения лабораторной работы был разработан лексический анализатор, предназначенный для составления таблицы согласно принадлежности к классам соответствующих лексем.