# 1. Мета роботи

Побудувати лексичний аналізатор для процедурної або функціональної мови програмування.

# 2. Завдання на роботу

1. По варианту задания определить, какие классы лексем будут в

вашем языке.

2. Составить контрольные примеры на реализуемом языке. Хотя бы один пример должен проверять поведение вашей программы при наличии недопустимых символов в транслируемом файле.

3. Запрограммировать и отладить модуль сканирования. Выполнить тестирование на контрольных примерах. Результатом работы должна быть таблица, содержащая лексемы и признаки их классов. Необходимо включить в результирующий файл информацию о номерах строк исходного текста

транслируемой программы.

# 3. Варіант індивідуального завдання

16

Подмножество языка PASCAL включает:

<Программа> ::= <Объявление переменных> <Описание вычислений> .

<Описание вычислений> ::= Begin <Список присваиваний> End

<Объявление переменных> ::= Var <Список переменных>

<Список переменных> ::= <Идент> | <Идент> , <Список переменных>

<Список присваиваний>::= <Присваивание> |

<Присваивание> <Список присваиваний>

<Присваивание> ::= <Идент> = <Выражение>

<Выражение> ::= <Ун.оп.> <Подвыражение> | <Подвыражение>

<Подвыражение> :: = ( <Выражение> ) | <Операнд> |

< Подвыражение > <Бин.оп.> <Подвыражение>

<Ун.оп.> ::= "-"

<Бин.оп.> ::= "-" | "+" | "\*" | "/"

<Операнд> ::= <Идент> | <Const>

<Идент> ::= <Буква> <Идент> | <Буква>

<Const> ::= <Цифра> <Const> | <Цифра>

На одной строке может быть только объявление переменных или один

оператор присваивания.

# 4. Виконання роботи

1) З варіанту завдання видно, що у мові є наступні класи лексем: ключові слова, ідентифікатори, оператори та константи.

2) Текст прикладів наведено нижче.

a)

**var** a**,**b**,**c**,**d

**begin**

a **=** 10

b **=** 20

d **=** 30

d **=** a**+**b**/**c**+**10

**end**

b)

**var** a**,**b**,**c**,**d

**begin**

a **=** 10

b **=** 20

d **=** 30

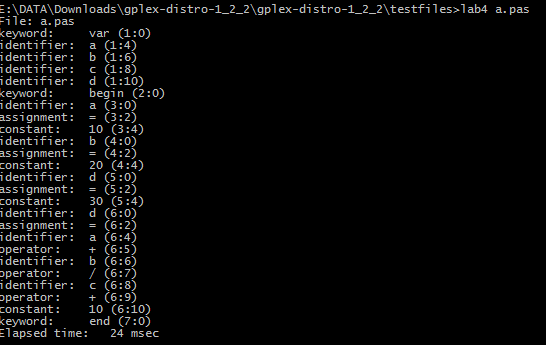
d **=** a**+**b**/**c**+**10

**(**\_**)**

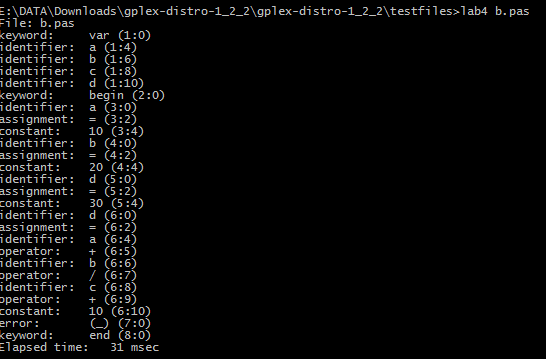
**end**

3)

a) Результат роботи на прикладі а:



b) Результат роботи на прикладі b:



Код програми:

%namespace lab4

%option noparser, caseInsensitive

alpha [a-zA-Z]

digits [0-9]+

error [^a-zA-Z0-9\+\-\\*\/=,\ \r\n]

operators [\+\-\\*\/]

%%

begin |

end |

var Console.WriteLine("keyword: " + yytext + " ("+ yyline + ":" + yycol +")");

{alpha}+ Console.WriteLine("identifier: " + yytext + " ("+ yyline + ":" + yycol +")");

= Console.WriteLine("assignment: " + yytext + " ("+ yyline + ":" + yycol +")");

{operators} Console.WriteLine("operator: " + yytext + " ("+ yyline + ":" + yycol +")");

{digits} Console.WriteLine("constant: " + yytext + " ("+ yyline + ":" + yycol +")");

{error}+ Console.WriteLine("error: " + yytext + " ("+ yyline + ":" + yycol +")");

%%

public static void Main(string[] argp) {

DateTime start = DateTime.Now;

int count = 0;

if (argp.Length == 0)

Console.WriteLine("Usage: WordCount filename(s), (wildcards ok)");

DirectoryInfo dirInfo = new DirectoryInfo(".");

for (int i = 0; i < argp.Length; i++) {

string name = argp[i];

FileInfo[] fInfo = dirInfo.GetFiles(name);

foreach (FileInfo info in fInfo)

{

try {

int tok;

FileStream file = new FileStream(info.Name, FileMode.Open);

Scanner scnr = new Scanner(file);

Console.WriteLine("File: " + info.Name);

do {

tok = scnr.yylex();

} while (tok > (int)Tokens.EOF);

count++;

} catch (IOException) {

Console.WriteLine("File " + name + " not found");

}

}

}

TimeSpan span = DateTime.Now - start;

Console.WriteLine("Elapsed time: {0,4:D} msec", (int)span.TotalMilliseconds);

}