

Compiladores
2016/01

Trabalho Prático 01:
Analisador Léxico e Tabela de Símbolos

Felipe Duarte, Gustavo Borba, Juan Lopes
Belo Horizonte, April 25, 2016

Contents

1	Introdução	1
1.1	A Linguagem Pas-c	1
2	Uso do compilador	2
2.1	Compilando o Compilador Pas-c	2
2.2	Compilando um programa em Pas-c	2
3	A Implementação do Compilador	3
3.1	A abordagem utilizada na implementação	3
3.2	A Estrutura do Projeto	3
3.3	Principais Classes da Aplicação	3
4	Resultados dos testes especificados	5
4.1	Teste 1	5
4.2	Teste 2	6
4.3	Teste 3	8
4.4	Teste 4	9
4.5	Teste 5	11
4.6	Teste 6	13
4.7	Teste 7	15
4.8	Teste 8	17

1 Introdução

O programa pas-c é um projeto desenvolvido na aula de Compiladores do Curso de Engenharia da Computação do CEFET-MG. Pas-c é a linguagem homônima, definida para esse compilador, cuja semântica dos comandos e expressões é a tradicional de linguagens como Pascal e C (daí seu nome).

1.1 A Linguagem Pas-c

Definimos a gramática da linguagem:

```

1  program ::= [ var decl-list ] begin stmt-list end
2  decl-list ::= decl ";" { decl ";" }
3  decl ::= ident-list is type
4  ident-list ::= identifier { "," identifier }
5  type ::= int | string
6  stmt-list ::= stmt ";" { stmt ";" }
7  stmt ::= assign-stmt | if-stmt | do-stmt | read-stmt | write-stmt
8  assign-stmt ::= identifier ":=" simple_expr
9  if-stmt ::= if condition then stmt-list end
10             | if condition then stmt-list else stmt-list end
11 condition ::= expression
12 do-stmt ::= do stmt-list stmt-suffix
13 stmt-suffix ::= while condition
14 read-stmt ::= in "(" identifier ")"
15 write-stmt ::= out "(" writable ")"
16 writable ::= simple_expr
17 expression ::= simple_expr | simple_expr relop simple_expr
18 simple_expr ::= term | simple_expr addop term
19 term ::= factor-a | term mulop factor-a
20 factor-a ::= factor | not factor | "-" factor
21 factor ::= identifier | constant | "(" expression ")"
22 relop ::= "=" | ">" | ">=" | "<" | "<=" | "<>"
23 addop ::= "+" | "-" | or
24 mulop ::= "*" | "/" | and
25 constant ::= integer_const | literal
26 integer_const ::= nozero { digit } | "0"
27 literal ::= " { caractere } "
28 identifier ::= ( letter ) { letter | digit }
29             | "_" ( letter | digit ) { letter | digit }
30 letter ::= [A-Za-z]
31 digit ::= [0-9]
32 nozero ::= [1-9]
33 caractere ::= um dos 256 caracteres do conjunto ASCII,
34             exceto "{", "}" e quebra de linha
35
```

2 Uso do compilador

2.1 Compilando o Compilador Pas-c

Para compilar o seu código (neste momento, apenas como analisador léxico), basta ter um compilador G++ versão 4.8 ou superior com cmake instalado em sua máquina.

Na pasta do projeto execute o comando:

```
1 | make
```

Esse comando deverá gerar a seguinte sequência de comandos:

```
1 | g++ -Iinclude -c -Wall -g src/Scanner.cpp -o src/Scanner.o
2 | g++ -Iinclude -c -Wall -g src/TestCase.cpp -o src/TestCase.o
3 | g++ -Iinclude -c -Wall -g src/Token.cpp -o src/Token.o
4 | g++ -Iinclude -c -Wall -g src/main.cpp -o src/main.o
5 | g++ src/Scanner.o src/TestCase.o src/Token.o src/main.o -o pasc
```

Isso deve gerar um arquivo executável **pasc** que compilará arquivos com extensão **.pasc**.

2.2 Compilando um programa em Pas-c

Para executar o compilador, basta executar o comando:

```
pasc caminho_para_seu_codigo.pasc
```

3 A Implementação do Compilador

3.1 A abordagem utilizada na implementação

O analisador léxico da foi implementado na classe **Scannere** analisa caractere por caractere do arquivo fonte procurando casar os caracteres acumulados com o padrão de formação de um Token. Dentro da classe **Scanner** existe uma tabela que contem as palavras reservadas da linguagem.

3.2 A Estrutura do Projeto

O projeto segue a estrutura básica de todo projeto em C++, contendo uma pasta **include** e, dentro desta, separados por módulos, os headers (.h) com as declarações das classes. A pasta **src** contém a implementação das classes, bem como a função principal **main**:

```
include/ -- Pasta para os headers
include/frontend -- pasta contendo as definições para os analisadores.
include/backend -- ainda por fazer.
include/test -- Pasta com os headers referentes aos testes unitários
src/ -- Implementação das classes
tests/ -- Pasta com a implementação dos testes unitários
```

3.3 Principais Classes da Aplicação

- **Token:** A classe Token descreve a unidade lógica mais básica do compilador. Ela apresenta apenas um valor, em formato de cadeia de caracteres, e seu **Tipo**, a ser elucidado no próximo item.
- **TokenType:** TokenType trata-se apenas de um enum, que define, através de um bitmap, a lista de tokens reconhecidos pelo compilador, bem como um tipo UNKNOWN, para todo e qualquer token que não corresponder a nenhuma definição da linguagem.
- **Scanner:** O analisador léxico está concentrado basicamente na classe Scanner. Essa classe contém os métodos **getNumerical**, **getLiteral** e **getOperator**, que são responsáveis por captar os tokens, bem como o método **getString**. Este último é necessário por que uma vez que se abre um caracter delimitador de string, espaços em brancos passam a ser caracteres significativos. Também estão definidos nessa classe os contadores de linhas e colunas do código-fonte.

- **Scope:** Tabela de símbolos do compilador
- **Symbol:** Classe abstrata que representará símbolos da linguagem

4 Resultados dos testes especificados

4.1 Teste 1

Código fonte testado:

```
1  var
2    a, b, c is int;
3    result is int
4  begin
5    in (a);
6    in (c);
7    b := 10;
8    result := (a * c)/(b + 5 - 345);
9    out(result);
10 end
```

Saída encontrada:

```
1  Opening file tests/test1.psc ...
2  Token found: (TOK_VAR)
3  Token found: (TOK_ID, a)
4  Token found: (TOK_COMMA)
5  Token found: (TOK_ID, b)
6  Token found: (TOK_COMMA)
7  Token found: (TOK_ID, c)
8  Token found: (TOK_IS)
9  Token found: (TOK_INT)
10 Token found: (TOK_SEMICOLON)
11 Token found: (TOK_ID, result)
12 Token found: (TOK_IS)
13 Token found: (TOK_INT)
14 Token found: (TOK_BEGIN)
15 Token found: (TOK_IN)
16 Token found: (TOK_PAR_OPEN)
17 Token found: (TOK_ID, a)
18 Token found: (TOK_PAR_CLOSE)
19 Token found: (TOK_SEMICOLON)
20 Token found: (TOK_IN)
21 Token found: (TOK_PAR_OPEN)
22 Token found: (TOK_ID, c)
23 Token found: (TOK_PAR_CLOSE)
24 Token found: (TOK_SEMICOLON)
25 Token found: (TOK_ID, b)
26 Token found: (TOK_ASSIGN)
27 Token found: (TOK_CONST_INT, 10)
28 Token found: (TOK_SEMICOLON)
29 Token found: (TOK_ID, result)
30 Token found: (TOK_ASSIGN)
31 Token found: (TOK_PAR_OPEN)
```

```
32 | Token found: (TOK_ID, a)
33 | Token found: (TOK_MULT)
34 | Token found: (TOK_ID, c)
35 | Token found: (TOK_PAR_CLOSE)
36 | Token found: (TOK_DIV)
37 | Token found: (TOK_PAR_OPEN)
38 | Token found: (TOK_ID, b)
39 | Token found: (TOK_ADD)
40 | Token found: (TOK_CONST_INT, 5)
41 | Token found: (TOK_SUB)
42 | Token found: (TOK_CONST_INT, 345)
43 | Token found: (TOK_PAR_CLOSE)
44 | Token found: (TOK_SEMICOLON)
45 | Token found: (TOK_OUT)
46 | Token found: (TOK_PAR_OPEN)
47 | Token found: (TOK_ID, result)
48 | Token found: (TOK_PAR_CLOSE)
49 | Token found: (TOK_SEMICOLON)
50 | Token found: (TOK_END)
51 | Token found: (TOK_EOF)
```

4.2 Teste 2

Código fonte testado:

```
1 | a, _ is int;
2 | b is int;
3 | nome is string;
4 | begin
5 |   in (a);
6 |   in (nome);
7 |   b := a * a;
8 |   b := b + a/2 * (3 + 5);
9 |   out (nome);
10 | out(b);
11 | end.
```

Saída encontrada:

```
1 | Opening file tests/test2.psc ...
2 | Token found: (TOK_ID, a)
3 | Token found: (TOK_COMMA)
4 | ERROR in line 1, column 6: Identifier  has no match with language structure.
5 | Token found: (TOK_IS)
6 | Token found: (TOK_INT)
7 | Token found: (TOK_SEMICOLON)
8 | Token found: (TOK_ID, b)
9 | Token found: (TOK_IS)
10 | Token found: (TOK_INT)
```



```
11 | Token found: (TOK_SEMICOLON)
12 | Token found: (TOK_ID, nome)
13 | Token found: (TOK_IS)
14 | Token found: (TOK_STRING)
15 | Token found: (TOK_SEMICOLON)
16 | Token found: (TOK_BEGIN)
17 | Token found: (TOK_IN)
18 | Token found: (TOK_PAR_OPEN)
19 | Token found: (TOK_ID, a)
20 | Token found: (TOK_PAR_CLOSE)
21 | Token found: (TOK_SEMICOLON)
22 | Token found: (TOK_IN)
23 | Token found: (TOK_PAR_OPEN)
24 | Token found: (TOK_ID, nome)
25 | Token found: (TOK_PAR_CLOSE)
26 | Token found: (TOK_SEMICOLON)
27 | Token found: (TOK_ID, b)
28 | Token found: (TOK_ASSIGN)
29 | Token found: (TOK_ID, a)
30 | Token found: (TOK_MULT)
31 | Token found: (TOK_ID, a)
32 | Token found: (TOK_SEMICOLON)
33 | Token found: (TOK_ID, b)
34 | Token found: (TOK_ASSIGN)
35 | Token found: (TOK_ID, b)
36 | Token found: (TOK_ADD)
37 | Token found: (TOK_ID, a)
38 | Token found: (TOK_DIV)
39 | Token found: (TOK_CONST_INT, 2)
40 | Token found: (TOK_MULT)
41 | Token found: (TOK_PAR_OPEN)
42 | Token found: (TOK_CONST_INT, 3)
43 | Token found: (TOK_ADD)
44 | Token found: (TOK_CONST_INT, 5)
45 | Token found: (TOK_PAR_CLOSE)
46 | Token found: (TOK_SEMICOLON)
47 | Token found: (TOK_OUT)
48 | Token found: (TOK_PAR_OPEN)
49 | Token found: (TOK_ID, nome)
50 | Token found: (TOK_PAR_CLOSE)
51 | Token found: (TOK_SEMICOLON)
52 | Token found: (TOK_OUT)
53 | Token found: (TOK_PAR_OPEN)
54 | Token found: (TOK_ID, b)
55 | Token found: (TOK_PAR_CLOSE)
56 | Token found: (TOK_SEMICOLON)
57 | Token found: (TOK_END)
58 | ERROR in line 11, column 5: Token . is not an operator
59 | Token found: (TOK_EOF)
```

4.3 Teste 3

Código fonte testado:

```
1  var
2  _cont is int;
3  media, altura, soma_ is int;
4  begin
5  _cont := 5;
6  soma = 0;
7  do
8
9  write({Altura: });
10 in (altura);
11 soma := soma altura;
12 _cont := _cont - 1;
13
14 while(_cont);
15 out({Media: });
16 out (soma / qtd);
17 end
```

Saída encontrada:

```
1  Opening file  tests/test3.psc ...
2  Token found: (TOK_VAR)
3  Token found: (TOK_ID, _cont)
4  Token found: (TOK_IS)
5  Token found: (TOK_INT)
6  Token found: (TOK_SEMICOLON)
7  Token found: (TOK_ID, media)
8  Token found: (TOK_COMMA)
9  Token found: (TOK_ID, altura)
10 Token found: (TOK_COMMA)
11 Token found: (TOK_ID, soma)
12 ERROR in line 3, column 18: Identifier  has no match with language structure.
13 Token found: (TOK_IS)
14 Token found: (TOK_INT)
15 Token found: (TOK_SEMICOLON)
16 Token found: (TOK_BEGIN)
17 Token found: (TOK_ID, _cont)
18 Token found: (TOK_ASSIGN)
19 Token found: (TOK_CONST_INT, 5)
20 Token found: (TOK_SEMICOLON)
21 Token found: (TOK_ID, soma)
22 Token found: (TOK_EQUALS)
23 Token found: (TOK_CONST_INT, 0)
24 Token found: (TOK_SEMICOLON)
25 Token found: (TOK_DO)
26 Token found: (TOK_ID, write)
```

```
27 | Token found: (TOK_PAR_OPEN)
28 | Token found: (TOK_CONST_STR, Altura: )
29 | Token found: (TOK_PAR_CLOSE)
30 | Token found: (TOK_SEMICOLON)
31 | Token found: (TOK_IN)
32 | Token found: (TOK_PAR_OPEN)
33 | Token found: (TOK_ID, altura)
34 | Token found: (TOK_PAR_CLOSE)
35 | Token found: (TOK_SEMICOLON)
36 | Token found: (TOK_ID, soma)
37 | Token found: (TOK_ASSIGN)
38 | Token found: (TOK_ID, soma)
39 | Token found: (TOK_ID, altura)
40 | Token found: (TOK_SEMICOLON)
41 | Token found: (TOK_ID, _cont)
42 | Token found: (TOK_ASSIGN)
43 | Token found: (TOK_ID, _cont)
44 | Token found: (TOK_SUB)
45 | Token found: (TOK_CONST_INT, 1)
46 | Token found: (TOK_SEMICOLON)
47 | Token found: (TOK_WHILE)
48 | Token found: (TOK_PAR_OPEN)
49 | Token found: (TOK_ID, _cont)
50 | Token found: (TOK_PAR_CLOSE)
51 | Token found: (TOK_SEMICOLON)
52 | Token found: (TOK_OUT)
53 | Token found: (TOK_PAR_OPEN)
54 | Token found: (TOK_CONST_STR, Media: )
55 | Token found: (TOK_PAR_CLOSE)
56 | Token found: (TOK_SEMICOLON)
57 | Token found: (TOK_OUT)
58 | Token found: (TOK_PAR_OPEN)
59 | Token found: (TOK_ID, soma)
60 | Token found: (TOK_DIV)
61 | Token found: (TOK_ID, qtd)
62 | Token found: (TOK_PAR_CLOSE)
63 | Token found: (TOK_SEMICOLON)
64 | Token found: (TOK_END)
65 | Token found: (TOK_EOF)
```

4.4 Teste 4

Código fonte testado:

```
1 | var
2 |
3 | i, j, k, @total, lsoma is int;
4 | begin
```

```
5 | i := 4 * (5-3 * 50 / 10;  
6 | j := i * 10;  
7 | k := i* j / k;  
8 | k := 4 + a $;  
9 | out(i);  
10 | out(j);  
11 | out(k);  
12 | end
```

Saída encontrada:

```
1 | Opening file tests/test4.psc ...  
2 | Token found: (TOK_VAR)  
3 | Token found: (TOK_ID, i)  
4 | Token found: (TOK_COMMA)  
5 | Token found: (TOK_ID, j)  
6 | Token found: (TOK_COMMA)  
7 | Token found: (TOK_ID, k)  
8 | Token found: (TOK_COMMA)  
9 | ERROR in line 3, column 12: Token @ is not an operator  
10 | Token found: (TOK_ID, total)  
11 | Token found: (TOK_COMMA)  
12 | Token found: (TOK_CONST_INT, 1)  
13 | Token found: (TOK_ID, soma)  
14 | Token found: (TOK_IS)  
15 | Token found: (TOK_INT)  
16 | Token found: (TOK_SEMICOLON)  
17 | Token found: (TOK_BEGIN)  
18 | Token found: (TOK_ID, i)  
19 | Token found: (TOK_ASSIGN)  
20 | Token found: (TOK_CONST_INT, 4)  
21 | Token found: (TOK_MULT)  
22 | Token found: (TOK_PAR_OPEN)  
23 | Token found: (TOK_CONST_INT, 5)  
24 | Token found: (TOK_SUB)  
25 | Token found: (TOK_CONST_INT, 3)  
26 | Token found: (TOK_MULT)  
27 | Token found: (TOK_CONST_INT, 50)  
28 | Token found: (TOK_DIV)  
29 | Token found: (TOK_CONST_INT, 10)  
30 | Token found: (TOK_SEMICOLON)  
31 | Token found: (TOK_ID, j)  
32 | Token found: (TOK_ASSIGN)  
33 | Token found: (TOK_ID, i)  
34 | Token found: (TOK_MULT)  
35 | Token found: (TOK_CONST_INT, 10)  
36 | Token found: (TOK_SEMICOLON)  
37 | Token found: (TOK_ID, k)  
38 | Token found: (TOK_ASSIGN)  
39 | Token found: (TOK_ID, i)
```

```
40 | Token found: (TOK_MULT)
41 | Token found: (TOK_ID, j)
42 | Token found: (TOK_DIV)
43 | Token found: (TOK_ID, k)
44 | Token found: (TOK_SEMICOLON)
45 | Token found: (TOK_ID, k)
46 | Token found: (TOK_ASSIGN)
47 | Token found: (TOK_CONST_INT, 4)
48 | Token found: (TOK_ADD)
49 | Token found: (TOK_ID, a)
50 | ERROR in line 8, column 12: Token $ is not an operator
51 | Token found: (TOK_SEMICOLON)
52 | Token found: (TOK_OUT)
53 | Token found: (TOK_PAR_OPEN)
54 | Token found: (TOK_ID, i)
55 | Token found: (TOK_PAR_CLOSE)
56 | Token found: (TOK_SEMICOLON)
57 | Token found: (TOK_OUT)
58 | Token found: (TOK_PAR_OPEN)
59 | Token found: (TOK_ID, j)
60 | Token found: (TOK_PAR_CLOSE)
61 | Token found: (TOK_SEMICOLON)
62 | Token found: (TOK_OUT)
63 | Token found: (TOK_PAR_OPEN)
64 | Token found: (TOK_ID, k)
65 | Token found: (TOK_PAR_CLOSE)
66 | Token found: (TOK_SEMICOLON)
67 | Token found: (TOK_END)
68 | Token found: (TOK_EOF)
```

4.5 Teste 5

Código fonte testado:

```
1 | var
2 |
3 |   j, k is int;
4 |   a, j real;
5 |
6 | begin
7 |   read(j);
8 |   read(K);
9 |
10 |  if (k <> 0)
11 |    result := j/k
12 |  else
13 |    result := 0
14 |  end;
```

```
15 |  
16 | out(result);  
17 | end
```

Saída encontrada:

```
1 | Opening file tests/test5.psc ...  
2 | Token found: (TOK_VAR)  
3 | Token found: (TOK_ID, j)  
4 | Token found: (TOK_COMMA)  
5 | Token found: (TOK_ID, k)  
6 | Token found: (TOK_IS)  
7 | Token found: (TOK_INT)  
8 | Token found: (TOK_SEMICOLON)  
9 | Token found: (TOK_ID, a)  
10 | Token found: (TOK_COMMA)  
11 | Token found: (TOK_ID, j)  
12 | Token found: (TOK_ID, real)  
13 | Token found: (TOK_SEMICOLON)  
14 | Token found: (TOK_BEGIN)  
15 | Token found: (TOK_ID, read)  
16 | Token found: (TOK_PAR_OPEN)  
17 | Token found: (TOK_ID, j)  
18 | Token found: (TOK_PAR_CLOSE)  
19 | Token found: (TOK_SEMICOLON)  
20 | Token found: (TOK_ID, read)  
21 | Token found: (TOK_PAR_OPEN)  
22 | Token found: (TOK_ID, K)  
23 | Token found: (TOK_PAR_CLOSE)  
24 | Token found: (TOK_SEMICOLON)  
25 | Token found: (TOK_IF)  
26 | Token found: (TOK_PAR_OPEN)  
27 | Token found: (TOK_ID, k)  
28 | Token found: (TOK_DIFF)  
29 | Token found: (TOK_CONST_INT, 0)  
30 | Token found: (TOK_PAR_CLOSE)  
31 | Token found: (TOK_ID, result)  
32 | Token found: (TOK_ASSIGN)  
33 | Token found: (TOK_ID, j)  
34 | Token found: (TOK_DIV)  
35 | Token found: (TOK_ID, k)  
36 | Token found: (TOK_ELSE)  
37 | Token found: (TOK_ID, result)  
38 | Token found: (TOK_ASSIGN)  
39 | Token found: (TOK_CONST_INT, 0)  
40 | Token found: (TOK_END)  
41 | Token found: (TOK_SEMICOLON)  
42 | Token found: (TOK_OUT)  
43 | Token found: (TOK_PAR_OPEN)  
44 | Token found: (TOK_ID, result)
```

```
45 | Token found: (TOK_PAR_CLOSE)
46 | Token found: (TOK_SEMICOLON)
47 | Token found: (TOK_END)
48 | Token found: (TOK_EOF)
```

4.6 Teste 6

Código fonte testado:

```
1 | var
2 | a, b, c, maior is int;
3 | nomecompletodoalunodeposgraduacao is string;
4 | start
5 | read(a);
6 | read(b);
7 | read(c;
8 |
9 | maior := 0;
10 | if ( a>b and a>c ) then
11 | maior := a;
12 | else
13 | if (b>c) then
14 | maior := b;
15 | else
16 | maior := c;
17 | end
18 | end
19 | Out({Maior idade: });
20 | out(maior);
21 | end
```

Saída encontrada:

```
1 | Opening file tests/test6.psc ...
2 | Token found: (TOK_VAR)
3 | Token found: (TOK_ID, a)
4 | Token found: (TOK_COMMA)
5 | Token found: (TOK_ID, b)
6 | Token found: (TOK_COMMA)
7 | Token found: (TOK_ID, c)
8 | Token found: (TOK_COMMA)
9 | Token found: (TOK_ID, maior)
10 | Token found: (TOK_IS)
11 | Token found: (TOK_INT)
12 | Token found: (TOK_SEMICOLON)
13 | ERROR in line 3, column 35: Identifier nomecompletodoalunodeposgraduacao
    exceed the maximum size of 15 characters
14 | Token found: (TOK_IS)
15 | Token found: (TOK_STRING)
```

```
16 | Token found: (TOK_SEMICOLON)
17 | Token found: (TOK_ID, start)
18 | Token found: (TOK_ID, read)
19 | Token found: (TOK_PAR_OPEN)
20 | Token found: (TOK_ID, a)
21 | Token found: (TOK_PAR_CLOSE)
22 | Token found: (TOK_SEMICOLON)
23 | Token found: (TOK_ID, read)
24 | Token found: (TOK_PAR_OPEN)
25 | Token found: (TOK_ID, b)
26 | Token found: (TOK_PAR_CLOSE)
27 | Token found: (TOK_SEMICOLON)
28 | Token found: (TOK_ID, read)
29 | Token found: (TOK_PAR_OPEN)
30 | Token found: (TOK_ID, c)
31 | Token found: (TOK_SEMICOLON)
32 | Token found: (TOK_ID, maior)
33 | Token found: (TOK_ASSIGN)
34 | Token found: (TOK_CONST_INT, 0)
35 | Token found: (TOK_SEMICOLON)
36 | Token found: (TOK_IF)
37 | Token found: (TOK_PAR_OPEN)
38 | Token found: (TOK_ID, a)
39 | Token found: (TOK_GT)
40 | Token found: (TOK_ID, b)
41 | Token found: (TOK_AND)
42 | Token found: (TOK_ID, a)
43 | Token found: (TOK_GT)
44 | Token found: (TOK_ID, c)
45 | Token found: (TOK_PAR_CLOSE)
46 | Token found: (TOK_THEN)
47 | Token found: (TOK_ID, maior)
48 | Token found: (TOK_ASSIGN)
49 | Token found: (TOK_ID, a)
50 | Token found: (TOK_SEMICOLON)
51 | Token found: (TOK_ELSE)
52 | Token found: (TOK_IF)
53 | Token found: (TOK_PAR_OPEN)
54 | Token found: (TOK_ID, b)
55 | Token found: (TOK_GT)
56 | Token found: (TOK_ID, c)
57 | Token found: (TOK_PAR_CLOSE)
58 | Token found: (TOK_THEN)
59 | Token found: (TOK_ID, maior)
60 | Token found: (TOK_ASSIGN)
61 | Token found: (TOK_ID, b)
62 | Token found: (TOK_SEMICOLON)
63 | Token found: (TOK_ELSE)
64 | Token found: (TOK_ID, maior)
```



```
65 | Token found: (TOK_ASSIGN)
66 | Token found: (TOK_ID, c)
67 | Token found: (TOK_SEMICOLON)
68 | Token found: (TOK_END)
69 | Token found: (TOK_END)
70 | Token found: (TOK_ID, Out)
71 | Token found: (TOK_PAR_OPEN)
72 | Token found: (TOK_CONST_STR, Maior idade: )
73 | Token found: (TOK_PAR_CLOSE)
74 | Token found: (TOK_SEMICOLON)
75 | Token found: (TOK_OUT)
76 | Token found: (TOK_PAR_OPEN)
77 | Token found: (TOK_ID, maior)
78 | Token found: (TOK_PAR_CLOSE)
79 | Token found: (TOK_SEMICOLON)
80 | Token found: (TOK_END)
81 | Token found: (TOK_EOF)
```

4.7 Teste 7

Código fonte testado:

```
1 | var
2 |   s, p is int;
3 |   var is int;
4 |
5 | begin
6 |   s := 0;
7 |
8 |   do
9 |     out({Valor a ser somado: });
10 |    in(p);
11 |    s := s+p;
12 |   while p!=0;
13 |
14 |   out({Total a pagar: R$});
15 |   out(s);
16 | end.
```

Saída encontrada:

```
1 | Opening file tests/test7.psc ...
2 | Token found: (TOK_VAR)
3 | Token found: (TOK_ID, s)
4 | Token found: (TOK_COMMA)
5 | Token found: (TOK_ID, p)
6 | Token found: (TOK_IS)
7 | Token found: (TOK_INT)
8 | Token found: (TOK_SEMICOLON)
```

```
9 | Token found: (TOK_VAR)
10 | Token found: (TOK_IS)
11 | Token found: (TOK_INT)
12 | Token found: (TOK_SEMICOLON)
13 | Token found: (TOK_BEGIN)
14 | Token found: (TOK_ID, s)
15 | Token found: (TOK_ASSIGN)
16 | Token found: (TOK_CONST_INT, 0)
17 | Token found: (TOK_SEMICOLON)
18 | Token found: (TOK_DO)
19 | Token found: (TOK_OUT)
20 | Token found: (TOK_PAR_OPEN)
21 | Token found: (TOK_CONST_STR, Valor a ser somado: )
22 | Token found: (TOK_PAR_CLOSE)
23 | Token found: (TOK_SEMICOLON)
24 | Token found: (TOK_IN)
25 | Token found: (TOK_PAR_OPEN)
26 | Token found: (TOK_ID, p)
27 | Token found: (TOK_PAR_CLOSE)
28 | Token found: (TOK_SEMICOLON)
29 | Token found: (TOK_ID, s)
30 | Token found: (TOK_ASSIGN)
31 | Token found: (TOK_ID, s)
32 | Token found: (TOK_ADD)
33 | Token found: (TOK_ID, p)
34 | Token found: (TOK_SEMICOLON)
35 | Token found: (TOK_WHILE)
36 | Token found: (TOK_ID, p)
37 | ERROR in line 12, column 12: Token ! is not an operator
38 | Token found: (TOK_EQUALS)
39 | Token found: (TOK_CONST_INT, 0)
40 | Token found: (TOK_SEMICOLON)
41 | Token found: (TOK_OUT)
42 | Token found: (TOK_PAR_OPEN)
43 | Token found: (TOK_CONST_STR, Total a pagar: R$)
44 | Token found: (TOK_PAR_CLOSE)
45 | Token found: (TOK_SEMICOLON)
46 | Token found: (TOK_OUT)
47 | Token found: (TOK_PAR_OPEN)
48 | Token found: (TOK_ID, s)
49 | Token found: (TOK_PAR_CLOSE)
50 | Token found: (TOK_SEMICOLON)
51 | Token found: (TOK_END)
52 | ERROR in line 16, column 5: Token . is not an operator
53 | Token found: (TOK_EOF)
```

4.8 Teste 8

Código fonte testado:

```
1  var
2      n, c, r is int;
3
4  begin
5      out({Digite um numero: });
6      in(n);
7
8      c:=1;
9      do
10         out(n);
11         out({ x });
12         out(c);
13         out({ = });
14         out(n*c);
15         out({\n});
16     while c !=10;
17
18 end.
```

Saída encontrada:

```
1  Opening file  tests/test8.psc ...
2  Token found: (TOK_VAR)
3  Token found: (TOK_ID, n)
4  Token found: (TOK_COMMA)
5  Token found: (TOK_ID, c)
6  Token found: (TOK_COMMA)
7  Token found: (TOK_ID, r)
8  Token found: (TOK_IS)
9  Token found: (TOK_INT)
10 Token found: (TOK_SEMICOLON)
11 Token found: (TOK_BEGIN)
12 Token found: (TOK_OUT)
13 Token found: (TOK_PAR_OPEN)
14 Token found: (TOK_CONST_STR, Digite um numero: )
15 Token found: (TOK_PAR_CLOSE)
16 Token found: (TOK_SEMICOLON)
17 Token found: (TOK_IN)
18 Token found: (TOK_PAR_OPEN)
19 Token found: (TOK_ID, n)
20 Token found: (TOK_PAR_CLOSE)
21 Token found: (TOK_SEMICOLON)
22 Token found: (TOK_ID, c)
23 Token found: (TOK_ASSIGN)
24 Token found: (TOK_CONST_INT, 1)
25 Token found: (TOK_SEMICOLON)
```

```
26 | Token found: (TOK_DO)
27 | Token found: (TOK_OUT)
28 | Token found: (TOK_PAR_OPEN)
29 | Token found: (TOK_ID, n)
30 | Token found: (TOK_PAR_CLOSE)
31 | Token found: (TOK_SEMICOLON)
32 | Token found: (TOK_OUT)
33 | Token found: (TOK_PAR_OPEN)
34 | Token found: (TOK_CONST_STR, x )
35 | Token found: (TOK_PAR_CLOSE)
36 | Token found: (TOK_SEMICOLON)
37 | Token found: (TOK_OUT)
38 | Token found: (TOK_PAR_OPEN)
39 | Token found: (TOK_ID, c)
40 | Token found: (TOK_PAR_CLOSE)
41 | Token found: (TOK_SEMICOLON)
42 | Token found: (TOK_OUT)
43 | Token found: (TOK_PAR_OPEN)
44 | Token found: (TOK_CONST_STR, = )
45 | Token found: (TOK_PAR_CLOSE)
46 | Token found: (TOK_SEMICOLON)
47 | Token found: (TOK_OUT)
48 | Token found: (TOK_PAR_OPEN)
49 | Token found: (TOK_ID, n)
50 | Token found: (TOK_MULT)
51 | Token found: (TOK_ID, c)
52 | Token found: (TOK_PAR_CLOSE)
53 | Token found: (TOK_SEMICOLON)
54 | Token found: (TOK_OUT)
55 | Token found: (TOK_PAR_OPEN)
56 | Token found: (TOK_CONST_STR, \n)
57 | Token found: (TOK_PAR_CLOSE)
58 | Token found: (TOK_SEMICOLON)
59 | Token found: (TOK_WHILE)
60 | Token found: (TOK_ID, c)
61 | ERROR in line 16, column 13: Token ! is not an operator
62 | Token found: (TOK_EQUALS)
63 | Token found: (TOK_CONST_INT, 10)
64 | Token found: (TOK_SEMICOLON)
65 | Token found: (TOK_END)
66 | ERROR in line 18, column 5: Token . is not an operator
67 | Token found: (TOK_EOF)
```