

KeLang

Analizador Sintático

Implementação do Analisador Sintático

Sobre as classes implementadas

O analisador sintático foi feito em apenas uma classe, chamada *SyntaticAnalysis*. E nessa classe, foram descritos uma série de métodos de um parser descendente.

Em primeiro lugar, é importante explicitar o método *private eat(Token Type type)*. Ele é o método responsável por processar o token atual vindo do léxico. Se houve um *eat(token)*, então o sintático executa o procedimento necessário e requisita o próximo token pro léxico.

Assim, logo em seguida, todos os outros métodos são procedimentos associados a cada não-terminal da gramática. E tais procedimentos possuem cláusulas para cada produção do respectivo símbolo. São esses os métodos:

- public void program()
- public void identifier()
- public void body()
- public void declist()
- public void casePVG()
- public void decl()
- public void identlist()
- public void type()
- public void stmtlist()
- public void stmt()
- public void assignstmt()
- public void ifstmt()
- public void repeatstmt()
- public void stmstsuffix()
- public void whilestmt()
- public void stmtprefix()
- public void readstmt()
- public void writestmt()
- public void writetable()
- public void expresison()
- public void simpleexpr()
- public void term()
- public void factor_a()
- public void factor()
- public void relop()
- public void addop()
- public void mulop()
- public void integerconst()
- public void floatconst()
- public void literal()

Realização de testes

Todos os testes foram compilados e executados com o *javac* pelo terminal. Os resultados são apresentados pelos prints colocados nesse relatório, mas existe um arquivo .txt para cada teste, contendo os seus respectivos resultados.

Tanto nos prints, quanto nos *.txts, os resultados podem ser interpretados da seguinte maneira:

- Comi [token] -> o “token” foi reconhecido/processado
- method called [token] -> Com o “token” em processo, o método “method” foi chamado
- method out [token] -> Com o “token” em processo, o método “method” foi finalizado

```
execute called
program called [app]
Comi [APP]
identifier called [_teste1]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [a]
identlist called [a]
identifier called [a]
Comi [ID]
Comi [VG]
identifier called [b]
Comi [ID]
Comi [VG]
identifier called [c]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [_result]
identlist called [_result]
identifier called [_result]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [a]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]

stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [c]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
assignstmt called [b]
identifier called [b]
Comi [ID]
Comi [ATRIB]
simpleexpr called [10.23]
term called [10.23]
factor-a called [10.23]
factor called [10.23]
constant called [10.23]
integerconst called [10.23]
Comi [NUM_REAL]
addop called [+]
Comi [PLUS]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
addop called [+]
Comi [PLUS]
simpleexpr called [c]
term called [c]
factor-a called [c]
factor called [c]
identifier called [c]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [_result]
assignstmt called [_result]
identifier called [_result]
```

Teste 1

```
assignstmt called [_result]
identifier called [_result]
Comi [ID]
Comi [ATRIB]
simpleexpr called [()]
term called [()]
factor-a called [()]
factor called [()]
Comi [OPEN_PAR]
expression called [a]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
addop called [+]
Comi [PLUS]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
Comi [CLOSE_PAR]
addop called [-]
Comi [MINUS]
simpleexpr called [1]
term called [1]
factor-a called [1]
factor called [1]
constant called [1]
integerconst called [1]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [_result]
simpleexpr called [_result]
```

```
term called [1]
factor-a called [1]
factor called [1]
constant called [1]
integerconst called [1]
Comi [NUM_INT]
Sai do assignstmt
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 2

```
execute called
program called [app]
Comi [APP]
identifier called [_huebr]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [a]
identlist called [a]
identifier called [a]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [b]
identlist called [b]
identifier called [b]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [f]
identlist called [f]
identifier called [f]
Comi [ID]
Comi [VG]
identifier called [_f]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [a]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]

stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [f]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [f]
assignstmt called [f]
identifier called [f]
Comi [ID]
Comi [ATRIB]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [b]
assignstmt called [b]
identifier called [b]
Comi [ID]
Comi [ATRIB]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
addop called [+]
Comi [PLUS]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
mulop called [/]
Comi [DIV]
```

```
term called [2]
factor-a called [2]
factor called [2]
constant called [2]
integerconst called [2]
Comi [NUM_INT]
addop called [+]
Comi [PLUS]
simpleexpr called [(]
term called [(]
factor-a called [(]
factor called [(]
Comi [OPEN_PAR]
expression called [3]
simpleexpr called [3]
term called [3]
factor-a called [3]
factor called [3]
constant called [3]
integerconst called [3]
Comi [NUM_INT]
addop called [+]
Comi [PLUS]
simpleexpr called [5]
term called [5]
factor-a called [5]
factor called [5]
constant called [5]
integerconst called [5]
Comi [NUM_INT]
Comi [CLOSE_PAR]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [b]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
```

```
Comi [WRITE]
Comi [OPEN_PAR]
writable called [f]
simpleexpr called [f]
term called [f]
factor-a called [f]
factor called [f]
identifier called [f]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [stop]
Not a valid statement, returning false
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 3

```
execute called
program called [app]
Comi [APP]
identifier called [_teste4]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [i]
identlist called [i]
identifier called [i]
Comi [ID]
Comi [VG]
identifier called [j]
Comi [ID]
Comi [VG]
identifier called [k]
Comi [ID]
Comi [VG]
identifier called [_total]
Comi [ID]
Comi [VG]
identifier called [_1soma]
Comi [ID]
Comi [VG]
identifier called [_teste4]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [i]
identlist called [i]
identifier called [i]
Comi [ID]
Comi [VG]
identifier called [a]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]

stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [i]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [j]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [k]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [i]
assignstmt called [i]
identifier called [i]
Comi [ID]
Comi [ATRIB]
simpleexpr called [4]
term called [4]
factor-a called [4]
factor called [4]
constant called [4]
integerconst called [4]
Comi [NUM_INT]
mulop called [*]
Comi [MUL]
term called [(]
factor-a called [(]
factor called [(]
Comi [OPEN_PAR]
```


expression called [5]
simpleexpr called [5]
term called [5]
factor-a called [5]
factor called [5]
constant called [5]
integerconst called [5]
Comi [NUM_INT]
addop called [-]
Comi [MINUS]
simpleexpr called [3]
term called [3]
factor-a called [3]
factor called [3]
constant called [3]
integerconst called [3]
Comi [NUM_INT]
mulop called [*]
Comi [MUL]
term called [50]
factor-a called [50]
factor called [50]
constant called [50]
integerconst called [50]
Comi [NUM_INT]
Comi [CLOSE_PAR]
mulop called [/]
Comi [DIV]
term called [10]
factor-a called [10]
factor called [10]
constant called [10]
integerconst called [10]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [j]
assignstmt called [j]
identifier called [j]
Comi [ID]
Comi [ATRIB]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]

identifier called [i]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [10]
factor-a called [10]
factor called [10]
constant called [10]
integerconst called [10]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [k]
assignstmt called [k]
identifier called [k]
Comi [ID]
Comi [ATRIB]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]
identifier called [i]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [j]
factor-a called [j]
factor called [j]
identifier called [j]
Comi [ID]
mulop called [/]
Comi [DIV]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [k]
assignstmt called [k]
identifier called [k]
Comi [ID]
Comi [ATRIB]

```
simpleexpr called [4]
term called [4]
factor-a called [4]
factor called [4]
constant called [4]
integerconst called [4]
Comi [NUM_INT]
addop called [+]
Comi [PLUS]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [i]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]
identifier called [i]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [j]
simpleexpr called [j]
term called [j]
factor-a called [j]
factor called [j]
identifier called [j]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
```

```
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [k]
simpleexpr called [k]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 4

```
execute called
program called [app]
Comi [APP]
identifier called [_teste4]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [i]
identlist called [i]
identifier called [i]
Comi [ID]
Comi [VG]
identifier called [j]
Comi [ID]
Comi [VG]
identifier called [k]
Comi [ID]
Comi [VG]
identifier called [_total]
Comi [ID]
Comi [VG]
identifier called [_1soma]
Comi [ID]
Comi [VG]
identifier called [_teste4]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [i]
identlist called [i]
identifier called [i]
Comi [ID]
Comi [VG]
identifier called [a]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [i]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [j]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [k]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [i]
assignstmt called [i]
identifier called [i]
Comi [ID]
Comi [ATRIB]
simpleexpr called [4]
term called [4]
factor-a called [4]
factor called [4]
constant called [4]
integerconst called [4]
Comi [NUM_INT]
mulop called [*]
Comi [MUL]
term called [(]
factor-a called [(]
factor called [(]
Comi [OPEN_PAR]
expression called [5]
simpleexpr called [5]
term called [5]
factor-a called [5]
factor called [5]
constant called [5]
integerconst called [5]
Comi [NUM_INT]
```

```

addop called [-]
Comi [MINUS]
simpleexpr called [3]
term called [3]
factor-a called [3]
factor called [3]
constant called [3]
integerconst called [3]
Comi [NUM_INT]
mulop called [*]
Comi [MUL]
term called [50]
factor-a called [50]
factor called [50]
constant called [50]
integerconst called [50]
Comi [NUM_INT]
Comi [CLOSE_PAR]
mulop called [/]
Comi [DIV]
term called [10]
factor-a called [10]
factor called [10]
constant called [10]
integerconst called [10]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [j]
assignstmt called [j]
identifier called [j]
Comi [ID]
Comi [ATRIB]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]
identifier called [i]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [10]
factor-a called [10]
factor called [10]
constant called [10]
integerconst called [10]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]

```

```

Comi [PVG]
stmt called [k]
assignstmt called [k]
identifier called [k]
Comi [ID]
Comi [ATRIB]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]
identifier called [i]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [j]
factor-a called [j]
factor called [j]
identifier called [j]
Comi [ID]
mulop called [/]
Comi [DIV]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [k]
assignstmt called [k]
identifier called [k]
Comi [ID]
Comi [ATRIB]
simpleexpr called [4]
term called [4]
factor-a called [4]
factor called [4]
constant called [4]
integerconst called [4]
Comi [NUM_INT]
addop called [+]
Comi [PLUS]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Sai do assignstmt

```

```
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [i]
simpleexpr called [i]
term called [i]
factor-a called [i]
factor called [i]
identifier called [i]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [j]
simpleexpr called [j]
term called [j]
factor-a called [j]
factor called [j]
identifier called [j]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [k]
simpleexpr called [k]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 5

```
execute called
program called [app]
Comi [APP]
identifier called [_teste5]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [j]
identlist called [j]
identifier called [j]
Comi [ID]
Comi [VG]
identifier called [k]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [a]
identlist called [a]
identifier called [a]
Comi [ID]
Comi [VG]
identifier called [j]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [j]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [k]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [if]
ifstmt called [if]
Comi [IF]
condition called [()]
expression called [()]
simpleexpr called [()]
term called [()]
factor-a called [()]
factor called [()]
Comi [OPEN_PAR]
expression called [k]
simpleexpr called [k]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
Comi [ID]
relop called [!=]
Comi [NOT_EQUAL]
expression called [0]
simpleexpr called [0]
term called [0]
factor-a called [0]
factor called [0]
constant called [0]
integerconst called [0]
Comi [NUM_INT]
Comi [CLOSE_PAR]
Comi [THEN]
stmtlist called [a]
stmt called [a]
assignstmt called [a]
identifier called [a]
Comi [ID]
Comi [ATRIB]
simpleexpr called [j]
term called [j]
factor-a called [j]
factor called [j]
identifier called [j]
Comi [ID]
mulop called [/]
Comi [DIV]
term called [k]
factor-a called [k]
factor called [k]
identifier called [k]
```



```
Comi [ID]
Sai do assignstmt
stmt out [else]
stmtlist out [else]
Comi [ELSE]
stmtlist called [a]
stmt called [a]
assignstmt called [a]
identifier called [a]
Comi [ID]
Comi [ATRIB]
simpleexpr called [0]
term called [0]
factor-a called [0]
factor called [0]
constant called [0]
integerconst called [0]
Comi [NUM_INT]
Sai do assignstmt
stmt out [end]
stmtlist out [end]
Comi [END]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [a]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 6

```
execute called
program called [app]
Comi [APP]
identifier called [_teste6]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [a]
identlist called [a]
identifier called [a]
Comi [ID]
Comi [VG]
identifier called [b]
Comi [ID]
Comi [VG]
identifier called [c]
Comi [ID]
Comi [VG]
identifier called [_maior]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [a]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [b]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
stmt out [;]
Comi [PVG]
stmt called [_maior]
assignstmt called [_maior]
identifier called [_maior]
Comi [ID]
Comi [ATRIB]
simpleexpr called [0]
term called [0]
factor-a called [0]
factor called [0]
constant called [0]
integerconst called [0]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [if]
ifstmt called [if]
Comi [IF]
condition called [[]]
expression called [[]]
simpleexpr called [[]]
term called [[]]
factor-a called [[]]
factor called [[]]
Comi [OPEN_PAR]
expression called [a]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
relop called [>]
Comi [GREATER_THAN]
expression called [b]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
mulop called [&&]
Comi [AND]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
```



```

relop called [>]
Comi [GREATER_THAN]
expression called [c]
simpleexpr called [c]
term called [c]
factor-a called [c]
factor called [c]
identifier called [c]
Comi [ID]
Comi [CLOSE_PAR]
Comi [THEN]
stmtlist called [_maior]
stmt called [_maior]
assignstmt called [_maior]
identifier called [_maior]
Comi [ID]
Comi [ATRIB]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [else]
Not a valid statement, returning false
stmt out [else]
stmtlist out [else]
Comi [ELSE]
stmtlist called [if]
stmt called [if]
ifstmt called [if]
Comi [IF]
condition called [(]
expression called [(]
simpleexpr called [(]
term called [(]
factor-a called [(]
factor called [(]
Comi [OPEN_PAR]
expression called [b]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
relop called [>]
Comi [GREATER_THAN]
expression called [c]
simpleexpr called [c]
term called [c]
factor-a called [c]
factor called [c]
identifier called [c]
Comi [ID]
Comi [CLOSE_PAR]
Comi [THEN]
stmtlist called [_maior]
stmt called [_maior]
assignstmt called [_maior]
identifier called [_maior]
Comi [ID]
Comi [ATRIB]
simpleexpr called [b]
term called [b]
factor-a called [b]
factor called [b]
identifier called [b]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [else]
Not a valid statement, returning false
stmt out [else]
stmtlist out [else]
Comi [ELSE]
stmtlist called [_maior]
stmt called [_maior]
assignstmt called [_maior]
identifier called [_maior]
Comi [ID]
Comi [ATRIB]
simpleexpr called [c]
term called [c]
factor-a called [c]
factor called [c]
identifier called [c]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [end]
Not a valid statement, returning false
stmt out [end]
stmtlist out [end]
Comi [END]

```

Teste 7

```
execute called
program called [app]
Comi [APP]
identifier called [_testeiii]
Comi [ID]
body called [integer]
decllist called [integer]
decl called [integer]
type called [integer]
Comi [INTEGER]
type out [r]
identlist called [r]
identifier called [r]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [a]
identlist called [a]
identifier called [a]
Comi [ID]
identlist out [;]
decl out [;]
Comi [PVG]
decl called [real]
type called [real]
Comi [REAL]
type out [_pi]
identlist called [_pi]
identifier called [_pi]
Comi [ID]
identlist out [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [read]
stmt called [read]
readstmt called [read]
Comi [READ]
Comi [OPEN_PAR]
identifier called [r]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [_pi]
assignstmt called [_pi]
identifier called [_pi]
Comi [ID]
Comi [ATRIB]
simpleexpr called [3.1415926535]
term called [3.1415926535]
factor-a called [3.1415926535]
factor called [3.1415926535]
constant called [3.1415926535]
integerconst called [3.1415926535]
Comi [NUM_REAL]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [a]
assignstmt called [a]
identifier called [a]
Comi [ID]
Comi [ATRIB]
simpleexpr called [0]
term called [0]
factor-a called [0]
factor called [0]
constant called [0]
integerconst called [0]
Comi [NUM_INT]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [a]
assignstmt called [a]
identifier called [a]
Comi [ID]
Comi [ATRIB]
simpleexpr called [r]
term called [r]
factor-a called [r]
factor called [r]
identifier called [r]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [r]
factor-a called [r]
factor called [r]
identifier called [r]
Comi [ID]
mulop called [*]
Comi [MUL]
term called [_pi]
factor-a called [_pi]
```

```
factor called [_pi]
identifier called [_pi]
Comi [ID]
Sai do assignstmt
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [{a area do circulo e: }]
literal called [{a area do circulo e: }] [STRING]
Comi [STRING]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [a]
simpleexpr called [a]
term called [a]
factor-a called [a]
factor called [a]
identifier called [a]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [stop]
Not a valid statement, returning false
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

Teste 8

```
execute called
program called [app]
Comi [APP]
identifier called [_teste8]
Comi [ID]
body called [start]
decllist called [start]
decl called [start]
decl out [start]
decllist out [start]
Comi [START]
stmtlist called [stop]
stmt called [stop]
Not a valid statement, returning false
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```

```
stmt out [end]
stmtlist out [end]
Comi [END]
stmt out [write]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [{maior idade: }]
literal called [{maior idade: }] [STRING]
Comi [STRING]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [write]
writestmt called [write]
Comi [WRITE]
Comi [OPEN_PAR]
writable called [_maior]
simpleexpr called [_maior]
term called [_maior]
factor-a called [_maior]
factor called [_maior]
identifier called [_maior]
Comi [ID]
Comi [CLOSE_PAR]
stmt out [;]
Comi [PVG]
stmt called [stop]
Not a valid statement, returning false
stmt out [stop]
stmtlist out [stop]
Comi [STOP]
```