
**Centro Federal de Educação Tecnológica
CEFET-MG**

**Documentação do Compilador
2ª Entrega - Analisador Sintático**

Lucas Silvestre Viana
Victor Balbo de Oliveira

Sumário

1	Implementação	3
2	Execução	3
3	Testes	4
3.1	1.test	4
3.1.1	4
3.1.2	4
3.1.3	5
3.1.4	5
3.1.5	6
3.1.6	6
3.2	2.test	7
3.2.1	7
3.2.2	7
3.2.3	8
3.2.4	8
3.2.5	9
3.2.6	9
3.2.7	10
3.2.8	11
3.3	3.test	11
3.3.1	11
3.3.2	12
3.3.3	12
3.3.4	13
3.4	4.test	14
3.4.1	14
3.4.2	14
3.4.3	15
3.4.4	16
3.4.5	16
3.4.6	17
3.5	5.test	18
3.5.1	18
3.5.2	18
3.5.3	19
3.5.4	20
3.6	6.test	20
3.6.1	20
3.6.2	21
3.6.3	21

1 Implementação

O analisador léxico recebe caracteres do arquivo fonte para gerar tokens. O analisador sintático recebe os tokens gerados pelo analisador léxico para verificar se a sintaxe do programa está correta.

A implementação do analisador sintático foi feita através de um parser LL(1) que toma a decisão de qual regra da gramática aplicar de acordo com apenas o primeiro token.

Para utilizar o parser LL(1) foram necessárias algumas alterações na gramática original. As alterações feitas foram:

- A regra *if-stmt* foi alterada para ter apenas uma regra de derivação que passa a ter os tokens **else stmt-list** como opcionais antes do token final **end**.
- A regra *expression* foi alterada para ter apenas uma regra de derivação que possui uma sequência de **relop simple-expr** como opcionais no final da regra.
- A regra *simple-expr* foi alterada para começar com um **fator-a** seguida de uma expressão de multiplicação **mulop** seguida de uma expressão de adição **addop**.
- A regra **term** foi removida por não ser mais utilizada.

A gramática resultante é a seguinte:

```
program      ::= program [decl-list] stmt-list end
decl-list    ::= decl {decl}
decl         ::= type ident-list ";"
ident-list   ::= identifier {"," identifier}
type         ::= int | string
stmt-list    ::= stmt {stmt}
stmt         ::= assign-stmt ";" | if-stmt | while-stmt | read-stmt ";" | write-stmt ";"
assign-stmt  ::= identifier "=" simple-expr
if-stmt      ::= if condition then stmt-list [else stmt-list] end
condition    ::= expression
while-stmt   ::= do stmt-list stmt-suffix
stmt-suffix  ::= while condition end
read-stmt    ::= scan "(" identifier ")"
write-stmt   ::= print "(" writable ")"
writable     ::= simple-expr | literal
expression   ::= simple-expr [relop simple-expr]
simple-expr   ::= fator-a {mulop fator-a} {addop fator-a {mulop fator-a}}
fator-a     ::= factor | ! factor | "-" factor
factor       ::= identifier | constant | "(" expression ")"
relop        ::= "==" | ">" | ">=" | "<" | "<=" | "!="
addop        ::= "+" | "-" | "||"
mulop        ::= "*" | "/" | "&&"
constant     ::= integer_const | literal
integer_const ::= digit {digit}
literal      ::= " \" {caractere} ""
identifier   ::= letter {letter | digit }
letter       ::= [A-Za-z]
digit        ::= [0-9]
caractere    ::= um dos caracteres ASCII, exceto \" e quebra de linha
```

2 Execução

Primeiramente, é necessário que o Java esteja instalado. Caso não tenha o Java instalado na máquina, baixe o Java JDK e faça sua instalação. A compilação do código é feita pelo comando no terminal:

```
$ javac *.java
```

E a execução é realizada via Terminal, inserindo o comando a seguir:

```
$ java Compiler <arquivo_de_entrada>
```

3 Testes

3.1 1.test

3.1.1

Utilizando o 1º teste aceito na entrega anterior:

```
program
    int a, b;
    int result;
    float a,x,total;
    a = 2;
    x = 1;
    scan (b);
    scan (y)
    result = (a*b ++ 1) / 2;
    print "Resultado: ";
    print (result);
    print ("Total: ");
    total = y / x;
    print ("Total: ";
    print (total);
end
```

Saída:

```
File 1.test: java.lang.Exception: Expected token 'ASSIGN' not found. Found 'IDENTIFIER' instead on line 4.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:69)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.1.2

Corrigindo o erro causado por não haver haver um tipo float na linha 4:

```
program
    int a, b;
    int result;
    int a,x,total;
    a = 2;
    x = 1;
    scan (b);
    scan (y)
    result = (a*b ++ 1) / 2;
    print "Resultado: ";
    print (result);
    print ("Total: ");
    total = y / x;
    print ("Total: ";
    print (total);
end
```

Saída:

```
File 1.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'IDENTIFIER' instead on line 9.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:96)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
```

```
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)
```

3.1.3

Corrigindo a falta de ';' na linha 8:

```
program
    int a, b;
    int result;
    int a,x,total;
    a = 2;
    x = 1;
    scan (b);
    scan (y);
    result = (a*b ++ 1) / 2;
    print "Resultado: ";
    print (result);
    print ("Total: ");
    total = y / x;
    print ("Total: ";
    print (total);
end
```

Saída:

```
File 1.test: java.lang.Exception: Unexpected token found 'ADDITION' on line 9. Expected IDENTIFIER, INT_CONSTAN
    at SyntacticalAnalyzer.factor(SyntacticalAnalyzer.java:177)
    at SyntacticalAnalyzer.addop(SyntacticalAnalyzer.java:157)
    at SyntacticalAnalyzer.simpleExpr(SyntacticalAnalyzer.java:137)
    at SyntacticalAnalyzer.expression(SyntacticalAnalyzer.java:120)
    at SyntacticalAnalyzer.factor(SyntacticalAnalyzer.java:173)
    at SyntacticalAnalyzer.simpleExpr(SyntacticalAnalyzer.java:135)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:70)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.1.4

Corrigindo os dois sinais de adição '++' na linha 9:

```
program
    int a, b;
    int result;
    int a,x,total;
    a = 2;
    x = 1;
    scan (b);
    scan (y);
    result = (a*b + 1) / 2;
    print "Resultado: ";
    print (result);
    print ("Total: ");
    total = y / x;
    print ("Total: ";
    print (total);
end
```

Saída:

```
File 1.test: java.lang.Exception: Expected token 'PARENTHESES_OPEN' not found. Found 'LITERAL' instead on line
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:100)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.1.5

Corrigindo a falta de parenteses na função **print** da linha 10:

```

program
    int a, b;
    int result;
    int a,x,total;
    a = 2;
    x = 1;
    scan (b);
    scan (y);
    result = (a*b + 1) / 2;
    print ("Resultado: ");
    print (result);
    print ("Total: ");
    total = y / x;
    print ("Total: ";
    print (total);
end

```

Saída:

```
File 1.test: java.lang.Exception: Expected token 'PARENTHESES_CLOSE' not found. Found 'SEMICOLON' instead on line 1
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:103)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.1.6

Corrigindo a falta de fecha parenteses na função **print** da linha 14:

```
program
    int a, b;
    int result;
    int a,x,total;
    a = 2;
```

```

x = 1;
scan (b);
scan (y);
result = (a*b + 1) / 2;
print ("Resultado: ");
print (result);
print ("Total: ");
total = y / x;
print ("Total: ");
print (total);
end

```

Saída:

File 1.test compiled with success.

3.2 2.test

3.2.1

Utilizando o 2º teste aceito na entrega anterior:

```

program
    int a, c;
    float d, e;
    a = 0; d = 35
    c = d / 12;

    Scan (a);
    Scan (c);
    b = a * a;
    c = b + a * (1 + a*c);
    print ("Resultado: ");
    print c;
    a = b + c + d)/2;
    e = val + c + a;
    print ("E: ");
    print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'ASSIGN' not found. Found 'IDENTIFIER' instead on line 3.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:69)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.2

Substituindo o tipo das variáveis definidas na linha 3 para *int*:

```

program
    int a, c;
    int d, e;
    a = 0; d = 35
    c = d / 12;

    Scan (a);
    Scan (c);
    b = a * a;
    c = b + a * (1 + a*c);
    print ("Resultado: ");

```

```

print c;
a = b + c + d)/2;
e = val + c + a;
print ("E: ");
print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'IDENTIFIER' instead on line 5.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:71)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.3

Corrigindo a falta de ';' na linha 4:

program

```

int a, c;
int d, e;
a = 0; d = 35;
c = d / 12;

Scan (a);
Scan (c);
b = a * a;
c = b + a * (1 + a*c);
print ("Resultado: ");
print c;
a = b + c + d)/2;
e = val + c + a;
print ("E: ");
print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'ASSIGN' not found. Found 'PARENTHESSES_OPEN' instead on line 7
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:69)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.4

Corrigindo a chamada da função **scan** na linha 7 removendo 'S' maiúsculo:

program

```

int a, c;
int d, e;
a = 0; d = 35;
c = d / 12;

scan (a);
Scan (c);
b = a * a;
c = b + a * (1 + a*c);

```



```

print ("Resultado: ");
print c;
a = b + c + d)/2;
e = val + c + a;
print ("E: ");
print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'ASSIGN' not found. Found 'PARENTHESSES_OPEN' instead on line 8
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:69)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.5

Corrigindo a chamada da função **scan** na linha 8 removendo 'S' maiúsculo:

```

program
    int a, c;
    int d, e;
    a = 0; d = 35;
    c = d / 12;

    scan (a);
    scan (c);
    b = a * a;
    c = b + a * (1 + a*c);
    print ("Resultado: ");
    print c;
    a = b + c + d)/2;
    e = val + c + a;
    print ("E: ");
    print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'PARENTHESSES_OPEN' not found. Found 'IDENTIFIER' instead on li
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:100)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.6

Corrigindo a chamada da função **print** na linha 12 adicionando os parenteses:

```

program
    int a, c;
    int d, e;
    a = 0; d = 35;
    c = d / 12;

    scan (a);
    scan (c);
    b = a * a;
    c = b + a * (1 + a*c);
    print ("Resultado: ");
    print (c);
    a = b + c + d)/2;
    e = val + c + a;
    print ("E: ");
    print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'PARENTHESSES_CLOSE' instead on li
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:71)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.7

Corrigindo a expressão na linha 13 adicionando o abre parenteses:

```

program
    int a, c;
    int d, e;
    a = 0; d = 35;
    c = d / 12;

    scan (a);
    scan (c);
    b = a * a;
    c = b + a * (1 + a*c);
    print ("Resultado: ");
    print (c);
    a = (b + c + d)/2;
    e = val + c + a;
    print ("E: ");
    print (e);

```

Saída:

```

File 2.test: java.lang.Exception: Expected token 'END' not found. Found 'EOF' instead on line 16.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:47)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.2.8

Corrigindo a falta do token **end** no fim do programa:

```
program
  int a, c;
  int d, e;
  a = 0; d = 35;
  c = d / 12;

  scan (a);
  scan (c);
  b = a * a;
  c = b + a * (1 + a*c);
  print ("Resultado: ");
  print (c);
  a = (b + c + d)/2;
  e = val + c + a;
  print ("E: ");
  print (e);
end
```

Saída:

File 2.test compiled with sucess.

3.3 3.test

3.3.1

Utilizando o 3º teste aceito na entrega anterior:

```
program
  int pontuacao, pontuacaoMaxima, disponibilidade;
  string pontuacaoMinima;
  disponibilidade = "Sim";
  pontuacaoMinima = 50;
  pontuacaoMaxima = 100;

  /* Entrada de dados
  Verifica aprovação de candidatos */
  do
    print("Pontuacao Candidato: ");
    scan(pontuacao);
    print("Disponibilidade Candidato: ");
    scan(disponibilidade);
    if ((pontuacao > pontuacaoMinima) and (disponibilidade=="Sim") then
      print("Candidato aprovado");
    else
      print("Candidato reprovado")
    end
  while (pontuacao >= 0)end
end
```

Saída:

File 3.test: java.lang.Exception: Expected token 'PARENTHESES_CLOSE' not found. Found 'IDENTIFIER' instead on 1
at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
at SyntacticalAnalyzer.factor(SyntacticalAnalyzer.java:174)
at SyntacticalAnalyzer.simpleExpr(SyntacticalAnalyzer.java:135)
at SyntacticalAnalyzer.expression(SyntacticalAnalyzer.java:120)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:75)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)

```

at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:86)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)

```

3.3.2

Para corrigir o programa é necessário substituir o identificador **and** pelo operador logico '&&' na linha 15:

program

```

int pontuacao, pontuacaoMaxima, disponibilidade;
string pontuacaoMinima;
disponibilidade = "Sim";
pontuacaoMinima = 50;
pontuacaoMaxima = 100;

/* Entrada de dados
Verifica aprovação de candidatos */
do
    print("Pontuacao Candidato: ");
    scan(pontuacao);
    print("Disponibilidade Candidato: ");
    scan(disponibilidade);
    if ((pontuacao > pontuacaoMinima) && (disponibilidade=="Sim") then
        print("Candidato aprovado");
    else
        print("Candidato reprovado")
    end
while (pontuacao >= 0)end
end

```

end

Saída:

File 3.test: java.lang.Exception: Expected token 'PARENTHESES_CLOSE' not found. Found 'THEN' instead on line 15

```

at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
at SyntacticalAnalyzer.factor(SyntacticalAnalyzer.java:174)
at SyntacticalAnalyzer.simpleExpr(SyntacticalAnalyzer.java:135)
at SyntacticalAnalyzer.expression(SyntacticalAnalyzer.java:120)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:75)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:86)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)

```

3.3.3

Para corrigir o programa é necessário fechar o parenteses da expressão da linha 15:

program

```

int pontuacao, pontuacaoMaxima, disponibilidade;
string pontuacaoMinima;
disponibilidade = "Sim";
pontuacaoMinima = 50;
pontuacaoMaxima = 100;

/* Entrada de dados
Verifica aprovação de candidatos */
do
    print("Pontuacao Candidato: ");
    scan(pontuacao);
    print("Disponibilidade Candidato: ");
    scan(disponibilidade);
    if ((pontuacao > pontuacaoMinima) && (disponibilidade=="Sim")) then
        print("Candidato aprovado");
    else
        print("Candidato reprovado")
    end
while (pontuacao >= 0)end
end

```

Saída:

```

File 3.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'END' instead on line 19.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:104)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:80)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:86)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.3.4

Para corrigir o programa é necessário adicionar ';' após o print da linha 18:

```

program
    int pontuacao, pontuacaoMaxima, disponibilidade;
    string pontuacaoMinima;
    disponibilidade = "Sim";
    pontuacaoMinima = 50;
    pontuacaoMaxima = 100;

    /* Entrada de dados
    Verifica aprovação de candidatos */
    do
        print("Pontuacao Candidato: ");
        scan(pontuacao);
        print("Disponibilidade Candidato: ");
        scan(disponibilidade);
        if ((pontuacao > pontuacaoMinima) && (disponibilidade=="Sim")) then
            print("Candidato aprovado");
        else
            print("Candidato reprovado");
        end
    end
end

```

```
        end
    while (pontuacao >= 0)end
end
```

Saída:

File 3.test compiled with sucess.

3.4 4.test

3.4.1

Utilizando o 4º teste aceito na entrega anterior:

```
int a, aux, b;
string nome, sobrenome, msg;
print("Nome: ");
scan (nome);
print("Sobrenome: ");
scan (sobrenome);
msg = "Ola, " + nome + " " +
sobrenome + "!";
msg = msg + 1;
print (msg);
scan (a);
scan(b);
if (a>b) then
    aux = b;
    b = a;
    a = aux;
end;
print ("Apos a troca: ");
out(a);
out(b)
end
```

Saída:

File 4.test: java.lang.Exception: Expected token 'PROGRAM' not found. Found 'INT' instead on line 1.
at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:44)
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)

3.4.2

Para corrigir o programa basta adicionar o token **program** ao inicio:

```
program
    int a, aux, b;
    string nome, sobrenome, msg;
    print("Nome: ");
    scan (nome);
    print("Sobrenome: ");
    scan (sobrenome);
    msg = "Ola, " + nome + " " +
sobrenome + "!";
    msg = msg + 1;
    print (msg);
    scan (a);
    scan(b);
    if (a>b) then
        aux = b;
```


3.4.4

Para corrigir o programa é necessário trocar a palavra `out` por **print** na linha 20:

```

program
    int a, aux, b;
    string nome, sobrenome, msg;
    print("Nome: ");
    scan (nome);
    print("Sobrenome: ");
    scan (sobrenome);
    msg = "Ola, " + nome + " " +
    sobrenome + "!";
    msg = msg + 1;
    print (msg);
    scan (a);
    scan(b);
    if (a>b) then
        aux = b;
        b = a;
        a = aux;
    end
    print ("Apos a troca: ");
    print(a);
    out(b)
end

```

Saída:

```
File 4.test: java.lang.Exception: Expected token 'ASSIGN' not found. Found 'PARENTHESES_OPEN' instead on line 2
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:69)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.4.5

Para corrigir o programa é necessário trocar a palavra out por **print** na linha 21:

```
program
    int a, aux, b;
    string nome, sobrenome, msg;
    print("Nome: ");
    scan (nome);
    print("Sobrenome: ");
    scan (sobrenome);
    msg = "Ola, " + nome + " " +
    sobrenome + "!";
    msg = msg + 1;
```



```

print (msg);
scan (a);
scan(b);
if (a>b) then
    aux = b;
    b = a;
    a = aux;
end
print ("Apos a troca: ");
print(a);
print(b)
end

```

Saída:

```

File 4.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'END' instead on line 22.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:104)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)

```

3.4.6

Para corrigir o programa é necessário adicionar o ';' ao final da expressão **print** na linha 21:

```

program
    int a, aux, b;
    string nome, sobrenome, msg;
    print("Nome: ");
    scan (nome);
    print("Sobrenome: ");
    scan (sobrenome);
    msg = "Ola, " + nome + " " +
    sobrenome + "!";
    msg = msg + 1;
    print (msg);
    scan (a);
    scan(b);
    if (a>b) then
        aux = b;
        b = a;
        a = aux;
    end
    print ("Apos a troca: ");
    print(a);
    print(b);
end

```

Saída:

File 4.test compiled with sucess.

3.5 5.test

3.5.1

Utilizando o 5º teste aceito na entrega anterior:

```
program
  int a, b, c, maior, outro;

  do
    print("A");
    scan(a);
    print("B");
    scan(b);
    print("C");
    scan(c);
    //Realizacao do teste
    if ( (a>b) && (a>c) )
      maior = a

    else
      if (b>c) then
        maior = b;

      else
        maior = c;
      end
    end
    print("Maior valor:");
    print (maior);
    print ("Outro? ");
    scan(outro);
    while (outro >= 0)
  end
```

File 5.test: java.lang.Exception: Expected token 'THEN' not found. Found 'IDENTIFIER' instead on line 13.

```
at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:76)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:86)
at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)
```

3.5.2

Para corrigir o programa basta adicionar o token **then** depois do **if** da linha 12:

```
program
  int a, b, c, maior, outro;

  do
    print("A");
    scan(a);
    print("B");
```

```

scan(b);
print("C");
scan(c);
//Realizacao do teste
if ( (a>b) && (a>c) ) then
    maior = a

else
    if (b>c) then
        maior = b;

    else
        maior = c;
    end
end
print("Maior valor:");
print (maior);
print ("Outro? ");
scan(outro);
while (outro >= 0)
end

```

File 5.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'ELSE' instead on line 15.

```

at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:71)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:77)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:114)
at SyntacticalAnalyzer.stmtList(SyntacticalAnalyzer.java:86)
at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:46)
at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
at Compiler.main(Compiler.java:12)

```

3.5.3

Para corrigir o programa basta adicionar um ';' a atribuição da linha 13:

```

program
    int a, b, c, maior, outro;

do
    print("A");
    scan(a);
    print("B");
    scan(b);
    print("C");
    scan(c);
    //Realizacao do teste
    if ( (a>b) && (a>c) ) then
        maior = a;

    else
        if (b>c) then
            maior = b;

        else
            maior = c;

```

```

        end
    end
    print("Maior valor:");
    print (maior);
    print ("Outro? ");
    scan(outro);
    while (outro >= 0)
end

```

File 5.test: java.lang.Exception: Expected token 'END' not found. Found 'EOF' instead on line 28.
 at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
 at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:47)
 at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
 at Compiler.main(Compiler.java:12)

3.5.4

Para corrigir o programa basta adicionar o token **end** depois do **while** da linha 27:

```

program
    int a, b, c, maior, outro;

    do
        print("A");
        scan(a);
        print("B");
        scan(b);
        print("C");
        scan(c);
        //Realizacao do teste
        if ( (a>b) && (a>c) ) then
            maior = a;

        else
            if (b>c) then
                maior = b;

            else
                maior = c;
            end
        end
        end
        print("Maior valor:");
        print (maior);
        print ("Outro? ");
        scan(outro);
        while (outro >= 0) end
    end

```

File 5.test compiled with sucess.

3.6 6.test

3.6.1

Utilizando o 6º teste aceito na entrega anterior:

```

program
    int a, b;
    scan(a);
    scan (b);
    int r = 2 * (a + b) / (b - a);
    print ("resultado: ");

```

```
    print(r);
end
```

Saída:

```
File 6.test: java.lang.Exception: Expected token 'END' not found. Found 'INT' instead on line 5.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:47)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.6.2

Este erro acontece pois a gramática não permite a definição de variáveis no meio do programa, todas devem acontecer logo após o token **program**. Para corrigir isso basta mover a linha 5 para a linha 3:

```
program
    int a, b;
    int r = 2 * (a + b) / (b - a);
    scan(a);
    scan (b);
    print ("resultado: ");
    print(r);
end
```

Saída:

```
File 6.test: java.lang.Exception: Expected token 'SEMICOLON' not found. Found 'ASSIGN' instead on line 3.
    at SyntacticalAnalyzer.eat(SyntacticalAnalyzer.java:39)
    at SyntacticalAnalyzer.declList(SyntacticalAnalyzer.java:59)
    at SyntacticalAnalyzer.program(SyntacticalAnalyzer.java:45)
    at SyntacticalAnalyzer.<init>(SyntacticalAnalyzer.java:12)
    at Compiler.main(Compiler.java:12)
```

3.6.3

Este erro acontece pois a gramática não permite a atribuição de variáveis durante a declaração. Para corrigir isso basta fazer a atribuição do identificador **r** na seguinte a sua definição:

```
program
    int a, b;
    int r;
    r = 2 * (a + b) / (b - a);
    scan(a);
    scan (b);
    print ("resultado: ");
    print(r);
end
```

Saída:

```
File 6.test compiled with sucess.
```