

TP Compiladores - Parte II  
Analisador Sintático  
23/10/2023

Guilherme de Assis Lima

## Código Fonte e arquivo .jar

O código fonte e arquivo .jar estão disponíveis em:

<https://github.com/guicompeng/compilador/tree/sintatico>

## Forma de uso do compilador

Para executar o compilador, basta rodar o comando:

```
java -jar AnalisadorSintatico.jar [file-path]
```

Exemplo:

```
java -jar AnalisadorSintatico.jar testes/1-corrigido-sintatico.txt
```

## Alterações realizadas na gramática

### 1ª alteração:

#### Antes

`program ::= class identifier [decl-list] body`

#### Depois

`program ::= programAux $`

`programAux ::= class identifier [decl-list] body`

### 2ª alteração:

#### Antes

`if-stmt ::= if "(" condition ")" "{" stmt-list "}" | if "(" condition ")" "{" stmt-list "}" else "{" stmt-list "}"`

#### Depois

`if-stmt ::= if "(" condition ")" "{" stmt-list "}" else-stmt`

`else-stmt ::= else "{" stmt-list "}" |  $\lambda$`

### 3ª alteração:

#### Antes:

`simple-expr ::= term | simple-expr addop term`

#### Depois:

`simple-expr ::= term simple-expr-aux`

`simple-expr-aux ::= addop term simple-expr-aux |  $\lambda$`

### 4ª alteração

#### Antes:

`term ::= factor-a | term mulop factor-a`

### Depois:

$\text{term} ::= \text{factor-a term-aux}$

$\text{term-aux} ::= \text{mulop factor-a term-aux} \mid \lambda$

## Abordagem

A abordagem utilizada foi baseada no livro texto da disciplina, com algumas alterações no código para adaptar a especificação do compilador.

### Alterações realizadas na tabela de símbolos

Na etapa 1, foi implementado a tabela de símbolos juntamente com a classe do analisador léxico. Após a alteração realizada na etapa 2, foi criada a classe “SymbolTable.java” que é do tipo *tabela única*, sendo instanciada pela classe do analisador sintático “Parser.java”.

A classe SymbolTable contém:

- Dois atributos:
  - countLevel: contador de nível que começa com 1
  - symbolTable: array list de RowSymbolTable.
    - RowSymbolTable foi outra classe implementada, que representa uma linha na tabela de símbolos. É uma classe bem simples, que contém 3 atributos (lexema, nível e tipo), além dos métodos de get e set para cada atributo.
- Cinco métodos:
  - insertRowSymbolTable: método para inserir uma linha na TS
  - getCountLevel: obter o valor do contador de nível
  - blockInput: método que pode ser chamado na entrada de um bloco
  - blockOutput: método chamado na saída do bloco. Esse método, além de atualizar o contador de nível, remove todas as linhas do nível antigo.
  - findRow: método para encontrar uma linha na TS, e tem como parâmetro o lexema. Esse método, vai procurando pelo maior nível, e se não encontrar em nenhum dos níveis, retorna *null*.

Portanto, na classe “Parser.java”, nas entradas dos blocos “*if, else, do*” foi chamado o método blockInput. Consequentemente, na saída, o método blockOutput.

## Abordagem do Parser

O parser contém:

- Atributos:
  - lexer: instancia do analisador léxico, é possível chamar o método estático lexer.scan() para obter o próximo token.
  - tok: instância da classe Token

- symbolTable: instância da tabela de símbolos.
- Métodos:
  - main: faz a leitura do arquivo código fonte, e chama o método program() do parser.
  - error: mostra a linha do erro, a causa e encerra a o programa.
  - advance: lê o próximo token.
  - eat: se for o token esperado, avança, se não, chama o método error().
  - Um método para cada não terminal.

## Testes

Para realizar os testes, foram utilizados os programas corrigidos do analisador léxico, e depois gerado um novo programa que contém as correções para o analisador sintático.

### Fonte 1

Comando: `java -jar AnalisadorSintatico.jar testes/1-corrigido-lexico.txt`

#### Corrigido léxico

```

1. class Teste1
2.     int a,b,c;
3.     float result;
4.     {
5.         write("Digite o valor de a:");
6.         read (a);
7.         write("Digite o valor de c:");
8.         read (c);
9.         b = 10;
10.        result = (a * c)/(b 5 - 345);
11.        write("O resultado e: ");
12.        write(result);
13.    }
```

#### Corrigido

```

1. class Teste1
2.     int a,b,c;
3.     float result;
4.     {
5.         write("Digite o valor de a:");
6.         read (a);
7.         write("Digite o valor de c:");
8.         read (c);
9.         b = 10;
10.        result = (a * c)/(b - 345);
11.        write("O resultado e: ");
12.        write(result);
13.    }
```

Erro sintático na linha 10  
Causado por: lexema não esperado: 5

Sintático: ok

## Fonte 2

Comando: `java -jar AnalisadorSintatico.jar testes/2-corrigido-lexico.txt`

### Corrigido léxico

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. a, 9, valor, b_1, b_2, int;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    Write (b2);
15. }
```

### Parcialmente Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, 9, valor, b_1, b_2, int;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    Write (b2);
15. }
```

### Parcialmente Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2, int;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    Write (b2);
15. }
```

Esperava o tipo das variáveis antes de "a"

Erro sintático na linha 4  
Causado por: lexema não esperado: a

Esperava um identificador, não número.

Erro sintático na linha 4  
Causado por: lexema não esperado: 9

3º: Não era esperado o "int" no final da linha 4.

Erro sintático na linha 4  
Causado por: lexema não esperado: int

### Parcialmente Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    Write (b2);
15. }
```

### Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    write (b2);
15. }
```

Não era esperado "Write" (identificador) e sim "write" com letra minúscula.

Erro sintático na linha 14  
Causado por: lexema não esperado: (

Sintático: ok

### Fonte 3

Comando: `java -jar AnalisadorSintatico.jar testes/3-corrigido-lexico.txt`

#### Corrigido léxico

```
1.  classe Teste3
2.  /** Verificando fluxo de controle
3.  Programa com if e while aninhados **/
4.  int i;
5.  int media, soma;
6.  {
7.      soma = 0;
8.
9.      write("Quantos dados deseja informar?" );
10.     read (qtd);
11.     IF (qtd>=2){
12.         i=0;
13.         do{
14.             write("Altura: ");
15.             read (altura);
16.             soma = soma+altura;
17.             i = i + 1;
18.         }while( i < qtd);
19.         media = soma / qtd;
20.         write("Media: ");
21.         write (media);
22.     }
23.     else{
24.         write("Quantidade inválida.");
25.     }
26. }
```

#### Parcialmente Corrigido

```
1.  class Teste3
2.  /** Verificando fluxo de controle
3.  Programa com if e while aninhados **/
4.  int i;
5.  int media, soma;
6.  {
7.      soma = 0;
8.
9.      write("Quantos dados deseja informar?" );
10.     read (qtd);
11.     IF (qtd>=2){
12.         i=0;
13.         do{
14.             write("Altura: ");
15.             read (altura);
16.             soma = soma+altura;
17.             i = i + 1;
18.         }while( i < qtd);
19.         media = soma / qtd;
20.         write("Media: ");
21.         write (media);
22.     }
23.     else{
24.         write("Quantidade inválida.");
25.     }
26. }
```

#### Era esperado “class” ao invés de “classe”

Erro sintático na linha 1  
Causado por: lexema não esperado: classe

#### Na linha 11, “IF” foi tratado como identificador, não como “if”

Erro sintático na linha 11  
Causado por: lexema não esperado: (

### Parcialmente Corrigido

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int media, soma;
6. {
7.     soma = 0;
8.
9.     write("Quantos dados deseja informar?" );
10.    read (qtd);
11.    if(qtd>=2){
12.        i=0;
13.        do{
14.            write("Altura: ");
15.            read (altura);
16.            soma = soma+altura;
17.            i = i + 1;
18.        }while( i < qtd);
19.        media = soma / qtd;
20.        write("Media: ");
21.        write (media);
22.    }
23.    else{
24.        write("Quantidade inválida.");
25.    }
26. }
```

### Corrigido sintático

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int media, soma;
6. {
7.     soma = 0;
8.
9.     write("Quantos dados deseja informar?" );
10.    read (qtd);
11.    if (qtd>=2){
12.        i=0;
13.        do{
14.            write("Altura: ");
15.            read (altura);
16.            soma = soma+altura;
17.            i = i + 1;
18.        }while( i < qtd);
19.        media = soma / qtd;
20.        write("Media: ");
21.        write (media);
22.    }
23.    else{
24.        write("Quantidade inválida.");
25.    };
26. }
```

Na linha 26, era esperado um “;”, pois após o if-stmt, é esperado um “;”

*stmt-list ::= stmt ";" { stmt ";" }*

Erro sintático na linha 26

Causado por: lexema não esperado: }

Sintático: ok

## Fonte 4

Comando: `java -jar AnalisadorSintatico.jar testes/4-corrigido-lexico.txt`

### Corrigido léxico

```
1. {
2.     // Outro programa de teste
3.
4.
5.     int idade, j, k, total;
6.     string nome, texto;
7.
8.     write("Digite o seu nome: ");
9.     read(nome);
10.    write("Digite o seu sobrenome");
11.    read(sobrenome);
12.    write("Digite a sua idade: ");
13.    read (idade);
14.    k = i * (5-i * 50 / 10;
15.    j = i * 10;
16.    k = i * j / k;
17.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
18.    write (text);
19.    write(j);
20.    write(k);
21. }
```

### Parcialmente corrigido

```
1.    Class Teste4 {
2.        // Outro programa de teste
3.
4.
5.        int idade, j, k, total;
6.        string nome, texto;
7.
8.        write("Digite o seu nome: ");
9.        read(nome);
10.       write("Digite o seu sobrenome");
11.       read(sobrenome);
12.       write("Digite a sua idade: ");
13.       read (idade);
14.       k = i * (5-i * 50 / 10;
15.       j = i * 10;
16.       k = i * j / k;
17.       texto = nome + " " + sobrenome + ", os
        números gerados sao: ";
18.       write (text);
19.       write(j);
20.       write(k);
21. }
```

### Parcialmente corrigido

```
1.    Class Teste4
2.        // Outro programa de teste
3.
4.
5.        int idade, j, k, total;
```

### Não tem "class Teste4" no começo

Erro sintático na linha 1  
Causado por: lexema não esperado: {

### "{" da linha 1 deveria estar abaixo das declarações de variáveis.

Erro sintático na linha 5  
Causado por: lexema não esperado: int

### Não fechou ")" na linha 14

Erro sintático na linha 14  
Causado por: lexema não esperado: ;



```
6.    string nome, texto;
7.    {
8.    write("Digite o seu nome: ");
9.    read(nome);
10.   write("Digite o seu sobrenome");
11.   read(sobrenome);
12.   write("Digite a sua idade: ");
13.   read (idade);
14.   k = i * (5-i * 50 / 10);
15.   j = i * 10;
16.   k = i * j / k;
17.   texto = nome + " " + sobrenome + ", os
números gerados sao: ";
18.   write (text);
19.   write(j);
20.   write(k);
21. }
```

### Corrigido sintático

```
1.  class Teste4
2.  // Outro programa de teste
3.
4.
5.  int idade, j, k, total;
6.  string nome, texto;
7.  {
8.    write("Digite o seu nome: ");
9.    read(nome);
10.   write("Digite o seu sobrenome");
11.   read(sobrenome);
12.   write("Digite a sua idade: ");
13.   read (idade);
14.   k = i * (5-i * 50 / 10);
15.   j = i * 10;
16.   k = i * j / k;
17.   texto = nome + " " + sobrenome + ", os
números gerados sao: ";
18.   write (text);
19.   write(j);
20.   write(k);
21. }
```

Sintático: ok

## Fonte 5

Comando: `java -jar AnalisadorSintatico.jar testes/5-corrigido-lexico.txt`

### Corrigido léxico

```
1. class MinhaClasse
2. {
3.     float a, b, c;
4.
5.
6.     write("Digite um número");
7.     read(a);
8.     write("Digite outro número: ");
9.     read(b);
10.    write("Digite mais um número: ");
11.    read(c);
12.
13.    maior = 0;
14.    if ( a>b && a>c )
15.        maior = a;
16.    else
17.        if (b>c)
18.            maior = b;
19.        else
20.            maior = c;
21.
22.    write("O maior número é: ");
23.    write(maior);
```

### Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.
6.     write("Digite um número");
7.     read(a);
8.     write("Digite outro número: ");
9.     read(b);
10.    write("Digite mais um número: ");
11.    read(c);
12.
13.    maior = 0;
14.    if ( a>b && a>c )
15.        maior = a;
16.    else
17.        if (b>c)
18.            maior = b;
19.        else
20.            maior = c;
21.
22.    write("O maior número é: ");
23.    write(maior);
```

### Declaração de variável deveria ser antes de "{"

Erro sintático na linha 3  
Causado por: lexema não esperado: float

### Faltou fechar ")"

Erro sintático na linha 11  
Causado por: lexema não esperado: ;

### Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.
6.     write("Digite um número");
7.     read(a);
8.     write("Digite outro número: ");
9.     read(b);
10.    write("Digite mais um número: ");
11.    read(c);
12.
13.    maior = 0;
14.    if ( a>b && a>c )
15.        maior = a;
16.    else
17.        if (b>c)
18.            maior = b;
19.        else
20.            maior = c;
21.
22.    write("O maior número é: ");
23.    write(maior);
```

Faltou parênteses para separar as expressões da linha 14.  
Sem parênteses, "b && a" tornou-se uma expressão

Erro sintático na linha 14  
Causado por: lexema não esperado: >

### Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.
6.     write("Digite um número");
7.     read(a);
8.     write("Digite outro número: ");
9.     read(b);
10.    write("Digite mais um número: ");
11.    read(c);
12.
13.    maior = 0;
14.    if ( (a>b) && (a>c) )
15.        maior = a;
16.    else
17.        if (b>c)
18.            maior = b;
19.        else
20.            maior = c;
21.
22.    write("O maior número é: ");
23.    write(maior);
```

Não era esperado "maior", pois deveria ser "{".

Erro sintático na linha 15  
Causado por: lexema não esperado: maior

### Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.     write("Digite um número");
6.     read(a);
7.     write("Digite outro número: ");
8.     read(b);
9.     write("Digite mais um número: ");
10.    read(c);
11.
12.
13.    maior = 0;
14.    if ( (a>b) && (a>c) ) {
15.        maior = a;
16.    }else{
17.        if (b>c){
18.            maior = b;
19.        }
20.        else{
21.            maior = c;
22.        }
23.    }
24.
25.    write("O maior número é: ");
26.    write(maior);
```

Faltou “;” depois do “else”

Erro sintático na linha 23  
Causado por: lexema não esperado: }

### Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.     write("Digite um número");
6.     read(a);
7.     write("Digite outro número: ");
8.     read(b);
9.     write("Digite mais um número: ");
10.    read(c);
11.
12.
13.    maior = 0;
14.    if ( (a>b) && (a>c) ) {
15.        maior = a;
16.    }else{
17.        if (b>c){
18.            maior = b;
19.        }
20.        else{
21.            maior = c;
22.        };
23.    };
24.
25.    write("O maior número é: ");
26.    write(maior);
27.
```

Faltou “;”

Erro sintático na linha 27  
Causado por: lexema não esperado:

### Corrigido sintático

```
1. class MinhaClasse
2.     float a, b, c;
3. {
4.
5.     write("Digite um número");
6.     read(a);
7.     write("Digite outro número: ");
8.     read(b);
9.     write("Digite mais um número: ");
10.    read(c);
11.
12.
13.    maior = 0;
14.    if ( (a>b) && (a>c) ) {
15.        maior = a;
16.    }else{
17.        if (b>c){
18.            maior = b;
19.        }
20.        else{
21.            maior = c;
22.        };
23.    };
24.
25.    write("O maior número é: ");
26.    write(maior);
27. }
```

Sintático: ok

## Fonte 6

Comando: `java -jar AnalisadorSintatico.jar testes/6.txt`

### Corrigido léxico

```
1.  classe Teste6
2.  int a, b, resultado;
3.  {
4.      write("Digite o valor de a:");
5.      read (a);
6.      write("Digite o valor de b:");
7.      read (b);
8.      resultado = a + b;
9.      write("A soma e: ");
10.     write(resultado);
11. }
```

Erro sintático na linha 1  
Causado por: lexema não esperado: classe

### Corrigido sintático

```
28. class Teste6
29. int a, b, resultado;
30. {
31.     write("Digite o valor de a:");
32.     read (a);
33.     write("Digite o valor de b:");
34.     read (b);
35.     resultado = a + b;
36.     write("A soma e: ");
37.     write(resultado);
38. }
```

Sintático: ok

## Fonte 7

Comando: `java -jar AnalisadorSintatico.jar testes/7.txt`

### Corrigido léxico

```
1.  classe Teste6
2.  int peso, altura, resultado;
3.  {
4.      write("Digite o peso:");
5.      read (peso);
6.      write("Digite a altura:");
7.      read (altura);
8.      resultado = peso/(altura*altura);
9.      write("O IMC e: ");
10.     write(resultado);
11. }
```

### Corrigido sintático

```
1.  class Teste6
2.  int peso, altura, resultado;
3.  {
4.      write("Digite o peso:");
5.      read (peso);
6.      write("Digite a altura:");
7.      read (altura);
8.      resultado = peso/(altura*altura);
9.      write("O IMC e: ");
10.     write(resultado);
11. }
```

1º

Erro sintático na linha 1  
Causado por: lexema não esperado: classe

Sintático: ok