



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
Engenharia de Computação

## **Trabalho Prático - Compiladores**

Filipe da Silva Rocha (20193003809)  
Matheus Freire Henrique Fonseca (20203002786)  
Vitor Laguardia Xavier (201712060554)

Belo Horizonte  
Janeiro de 2025

## 1) Forma de uso do compilador

Para executar o compilador basta compilar o projeto (javac Compiler.java) e executá-lo passando como argumento o nome de algum arquivo teste (e.g.: java Compiler tests/test1.txt).

A função executada pela classe Compiler será responsável por:

- Construir o analisador léxico, sintático e semântico a partir do arquivo de teste
- Retornar uma mensagem de erro, caso haja algum. Se a compilação for bem sucedida, não haverá reports de erro

## 2) Descrição da abordagem utilizada na implementação, indicando as principais classes da aplicação e seus respectivos propósitos

O projeto do analisador semântico foi implementado com base nas seguintes classes principais:

- **Parser:** o código foi consideravelmente alterado com relação à etapa anterior. Agora, cada procedimento passou a ser uma classe, possuindo um atributo type que representa o tipo do procedimento (INTEGER, FLOAT, STRING, VOID) e um método exec(), que irá executar o procedimento sintático. Em alguns procedimentos, ocorre a verificação semântica de tipos compatíveis e se uma variável já foi declarada previamente. Cabe ressaltar que agora a classe do parser tem uma própria hashtable para verificar se algum identificador já foi declarado previamente
- **Semantic:** essa classe possui dois métodos auxiliares. O primeiro é o resultType, que recebe dois tipos e uma tag representando a operação. O método retorna o tipo resultante com base no operador, por exemplo: caso '+', se o tipo 1 for string e o tipo 2 for string, o tipo resultante será string. Já o segundo método apenas verifica se um tipo é numérico, ou seja, se é Integer ou Float.
- **Type:** essa classe é um enum contendo os tipos assumidos pelos procedimentos do analisador sintático. São eles: integer, float, string e void.
- **SemanticException:** classe para reportar os erros semânticos. Ela estende RuntimeException, logo caso haja um erro, será lançada uma exceção. Quem irá tratar ela é a Main (classe Compiler), printando a mensagem para o usuário e abortando o programa.

### 3) Apresentação dos testes

- Teste 1 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```
1  start
2      float a;
3      int b, result;
4      float a,x,total;
5
6      a = 2.0;
7      x = 0.1;
8      scan (b);
9      scan (y);
10     result = (a*b + 1) % a;
11     print ({Resultado: });
12     print (result);
13     print ({Total: });
14     total = y / x;
15     print ({Total: });
16     print (total);
17 exit
```

Saída do programa para cada erro:

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test1.txt
Semantic Error at line 4: Variable already declared: a
```

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test1.txt
Semantic Error at line 9: Undefined variable in scan: y
```

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test1.txt
Semantic Error at line 10: Incompatible types for operator '%'
```

Arquivo txt corrigido:

```
start
float a;
int b, result;
float c,x,total;
a = 2.0;
x = 0.1;
scan (b);
scan (c);
result = (a*b + 1) % a;
print ({Resultado: });
print (result);
print ({Total: });
total = y / x;
print ({Total: });
print (total);
exit
```

- Teste 2 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```

1  start
2      int a, b;
3      float c_, d, _e, val;
4
5      a = 0; d = 3.5;
6      c_ = d / 1.2; //Erro
7
8      scan (a);
9      scan (c_);
10     b = a * a;
11     c_ = b + a * (1 + a*c_);
12     print ({Resultado: });
13     print (c_);
14     d = 34.2;
15     _e = val + 2.2;
16     print ({E: });
17     print (_e);
18     a = b + a/2;
19     exit
20

```

Resultado do Parser para cada erro:

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 6: Undefined variable in assignment: c
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 6: Type mismatch in assignment. Variable 'c_' is INTEGER but expression is FLOAT
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 9: Undefined variable in scan: c
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 10: Undefined variable in assignment: b
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 11: Undefined variable in assignment: c
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 11: Undefined variable: c
vitor@FRMECMNOT002699:~/compiladores$ clear

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 13: Undefined variable: c
vitor@FRMECMNOT002699:~/compiladores$ clear

```

```

Semantic Error at line 15: Undefined variable: c
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 15: Undefined variable in assignment: e
• vitor@FRMECMNOT002699:~/compiladores$
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 15: Undefined variable: val
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 17: Undefined variable: e
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 18: Undefined variable: c
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
Semantic Error at line 18: Type mismatch in assignment. Variable 'a' is INTEGER but expression is FLOAT
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test2.txt
• vitor@FRMECMNOT002699:~/compiladores$

```

Arquivo corrigido:

```

1  start
2      int a, b;
3      float c_, d, _e, val;
4
5      a = 0; d = 3.5;
6      c_ = d / 1.2; //Erro
7
8      scan (a);
9      scan (c_);
10     b = a * a;
11     c_ = b + a * (1 + a*c_);
12     print ({Resultado: });
13     print (c_);
14     d = 34.2;
15     _e = val + 2.2;
16     print ({E: });
17     print (_e);
18     a = b + a/2;
19     exit

```

- Teste 3 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```

1  start
2  ∨ int pontuacao, pontuacaoMaxima, disponibilidade;
3    string pontuacaoMinima;
4
5  {
6    disponibilidade = Sim;
7    pontuacaoMinima = 50;
8    pontuacaoMaxima = 100;
9
10   /* Entrada de dados
11   | Verifica aprovação de candidatos */
12   do
13     print({Pontuacao Candidato: });
14     scan(pontuacao);
15     print({Disponibilidade Candidato: });
16     scan(disponibilidade);
17
18     if ((pontuação > pontuacaoMinima) && (disponibilidade=={Sim})) then
19       print({Candidato aprovado});
20     else
21       print({Candidato reprovado});
22     end
23   while (pontuação >= 0)end
24   exit

```

Resultado do Parser para cada erro:

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
Semantic Error at line 5: Undefined variable: Sim

```

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
Semantic Error at line 7: Undefined variable in assignment: pontuacaoMinima

```

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
Semantic Error at line 8: Undefined variable in assignment: pontuacaoMaxima
• vitor@FRMECMNOT002699:~/compiladores$

```

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
Semantic Error at line 18: Undefined variable: pontuação

```

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
Semantic Error at line 23: Undefined variable: pontuação

```

```

• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test3.txt
• vitor@FRMECMNOT002699:~/compiladores$

```

Arquivo txt corrigido:

```
1  start
2  int pontuacao, pontuacaoMaxima, pontuacaoMinima;
3  string disponibilidade;
4
5
6  disponibilidade = {Sim};
7  pontuacaoMinima = 50;
8  pontuacaoMaxima = 100;
9
10 /* Entrada de dados
11  | Verifica aprovação de candidatos */
12 do
13     print({Pontuacao Candidato: });
14     scan(pontuacao);
15     print({Disponibilidade Candidato: });
16     scan(disponibilidade);
17
18     if ((pontuacao > pontuacaoMinima) && (disponibilidade=={Sim})) then
19         print({Candidato aprovado});
20     else
21         print({Candidato reprovado});
22     end
23 while (pontuacao >= 0)end
24 exit
```

- Teste 4 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```
tests > test4.txt
1  start
2      int a, aux, b;
3      string nome, sobrenome, msg;
4
5      print(Nome);
6      scan (nome);
7      print({Sobrenome: });
8      scan (sobrenome);
9      msg = {Ola, } + nome + { } +
10     sobrenome + {!};
11     msg = msg + 1;
12     print (msg);
13
14     scan (a);
15     scan(b);
16     if (a>b) then
17         aux = b;
18         b = a;
19         a = aux;
20     end
21     print ({Apos a troca: });
22     print(a);
23     print(b);
24     exit
25
```

Resultado do Parser para cada erro:

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test4.txt
Semantic Error at line 5: Undefined variable: Nome
○ vitor@FRMECMNOT002699:~/compiladores$
```

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test4.txt
Semantic Error at line 11: Operator '+' requires both operands to be numeric, or both to be strings
○ vitor@FRMECMNOT002699:~/compiladores$
```

```
• vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test4.txt
○ vitor@FRMECMNOT002699:~/compiladores$
```



Arquivo teste corrigido:

```
1  start
2      int a, aux, b;
3      string nome, sobrenome, msg;
4
5      print(nome);
6      scan (nome);
7      print({Sobrenome: });
8      scan (sobrenome);
9      msg = {Ola, } + nome + { } +
10     sobrenome + {!};
11     msg = msg + {uhu};
12     print (msg);
13
14     scan (a);
15     scan(b);
16     if (a>b) then
17         aux = b;
18         b = a;
19         a = aux;
20     end
21     print ({Apos a troca: });
22     print(a);
23     print(b);
24 exit
25
```

- Teste 5 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```
1  start
2      int a, b, c, maior, outro;
3
4  do
5      print({A});
6      scan(a);
7      print({B});
8      scan(b);
9      print({C});
10     scan(c);
11     //Realizacao do teste
12
13     if ( (a>b) && (a>c) )then
14         maior = a;
15
16     else
17         if (b>c) then
18             maior = b;
19
20         else
21             maior = c;
22         end
23     end
24     print({Maior valor:});
25     print (maior);
26     print ({Outro? });
27     scan(outro);
28     while (outro >= 0) end
29     exit
30
```

Resultado do Parser para cada erro:

```
vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test5.txt
vitor@FRMECMNOT002699:~/compiladores$
```

Não há erros.

- Teste 6 (já com as correções léxicas e sintáticas)

Arquivo txt original:

```

1  start
2      int par, impar;
3
4      int restoPar;
5      int restoImpar;
6
7      par = 20;
8      impar = 5;
9
10     restoPar = par % 2;
11     restoImpar = impar % 2;
12
13
14     if(restoPar == 0) then
15         print({Numero par!});
16     end
17
18     if(restoImpar != 0) then
19         print({Numero impar!});
20     else
21         a = 4;
22     end
23
24     exit

```

Resultado do Parser para cada erro:

```

vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test6.txt
Semantic Error at line 21: Undefined variable in assignment: a
vitor@FRMECMNOT002699:~/compiladores$
vitor@FRMECMNOT002699:~/compiladores$ java Compiler tests/test6.txt
vitor@FRMECMNOT002699:~/compiladores$

```

Arquivo txt corrigido:

```

1  start
2      int par, impar, a;
3
4      int restoPar;
5      int restoImpar;
6
7      par = 20;
8      impar = 5;
9
10     restoPar = par % 2;
11     restoImpar = impar % 2;
12
13
14     if(restoPar == 0) then
15         print({Numero par!});
16     end
17
18     if(restoImpar != 0) then
19         print({Numero impar!});
20     else
21         a = 4;
22     end
23
24     exit

```