

Compiladores

Relatório do trabalho prático – Analisador Sintático

Alline Santos Ferreira
Arthur Miranda Moreira

1. Introdução

Quando utilizamos um software não pensamos no que acontece em um computador para que o resultado final seja o programa e suas funções. Na disciplina de Compiladores, estamos aprendendo que existe um processo por trás do software que a partir de um código fonte, construído por um programador, há um programa que faz a tradução desse código, em um formato que o computador seja capaz de executar e, como efeito final dessa execução é exibido o programa para que o usuário final possa utilizar.

Na disciplina de Compiladores, aprendemos mais detalhadamente como essa tradução ocorre. Sabemos que a estrutura deste software é dividida em análise e síntese. Na primeira parte, o programa fonte passa por uma fase de análise da estrutura gramatical. Se ela estiver incorreta, conforme é definido pela linguagem de programação utilizada para sua construção, o compilador exibe mensagem de erro para que o programador possa identificar e corrigir. Uma vez que essa fase de análise é feita com sucesso, então, na segunda etapa, denominada síntese, o compilador constrói o programa a partir do reconhecimento das informações analisadas e, em seguida, gera o executável que é o programa utilizado pelo usuário final.

Este trabalho visa, entendermos na prática, o funcionamento do compilador. Para isso, o trabalho proposto na disciplina é dividido em três etapas: Analisador Léxico, Analisador Sintático, Analisador Semântico e Tabelas de Símbolos.

Neste relatório, focaremos na construção da **segunda etapa** do trabalho: o **Analisador Sintático (parser)**. O papel crucial do analisador sintático é verificar se o programa-fonte está sintaticamente correto. Ele recebe do analisador léxico (desenvolvido na etapa 1) uma sequência de tokens, que representam os elementos básicos do programa (como palavras-chave, identificadores, símbolos, entre outros). Em seguida, o parser detecta erros de sintaxe e emite mensagens apropriadas, garantindo que a ordem dos tokens siga a estrutura esperada da linguagem, de acordo com suas regras gramaticais. Na terceira fase do trabalho, será desenvolvido o analisador semântico, e este analisador sintático será fundamental.

A seguir, detalharemos a construção do analisador sintático e as modificações necessárias no analisador léxico, desenvolvido na primeira etapa.

2. Desenvolvimento

2.2 Descrição da implementação

Para a implementação do Analisador Sintático, utilizamos a linguagem de programação Java. E para execução do programa é necessário informar o caminho onde se encontra o arquivo de teste. Ao digitar o caminho válido, o arquivo é aberto pelo analisador sintático. Ele instancia um analisador léxico e realiza chamadas para coletar o fluxo de tokens, verificando se estão coerentes com as regras gramaticais da linguagem de programação dada. Caso contrário, ele enviará mensagem de erro identificando que há um erro de sintaxe. Como este relatório trata da segunda etapa, é importante que se leia o relatório da 1ª etapa, pois aqui apresentaremos apenas as novas classes implementadas para a construção do analisador sintático. A seguir, explicaremos essas classes.

2.2.1 Classes para o Analisador

Para desenvolver o Analisador Sintático, criamos as classes **Sintax** e **SintaxException**. As classes existentes – App, Lexer, Word, Num, Token e Tag – foram atualizadas para permitir a comunicação entre o analisador léxico (desenvolvido na Etapa 1) e o analisador sintático.

A classe **Sintax** recebe os tokens do analisador léxico e trata seus nomes como símbolos terminais de uma gramática livre de contexto (que inclui símbolos terminais, não terminais, produções e um símbolo inicial). Este analisador constrói uma árvore de derivação para as sequências de tokens de entrada, verificando se estão corretas de acordo com essa gramática. Ele foi construído seguindo o método do parser recursivo descendente visto em sala. Para cada regra, foi implementado um procedimento que casa símbolos não terminais e faz chamadas a outras regras, quando necessário. Se um token não estiver correto conforme as regras gramaticais, é lançada uma **SintaxException** e a execução é interrompida.

A lógica do analisador léxico está implementada na classe **Lexer**. É nela que ocorre a leitura do código-fonte de teste e a verificação de sua conformidade sintática e semântica com a estrutura gramatical definida. Se algum erro léxico for detectado, o programa exibirá uma mensagem detalhada com a descrição do problema e a linha onde ocorreu, interrompendo a execução.

Nesta implementação, erros são considerados fatais e não há recuperação: O programador precisa corrigir o erro antes de tentar compilar novamente.

Uma alteração realizada foi a criação de algumas tags que faltaram na primeira etapa: INT_CONST, FLOAT_CONST e CHAR_CONST. Na primeira etapa, tanto as constantes quanto as palavras reservadas Int, Float e Char compartilhavam a mesma tag, o que não era correto.

Por fim, a classe **App** contém o método main. Sua principal função é verificar a localização do arquivo de teste. Se o arquivo for encontrado, ela cria uma instância do analisador sintático e inicia o processo de análise.

Os códigos desenvolvidos nas classes Tag, Word, Token, Num, Lexer, Sintax e SintaxException foram inspirados na obra dos autores Alfred Aho, Monica Lam, Ravi Sethi e Jeffrey Ullman.

2.2.2 Modificações na gramática

Para implementarmos o Analisador Sintático, foi necessário realizarmos algumas modificações na gramática disponibilizada pela professora, removendo recursão à esquerda e prefixos comuns. As alterações foram destacadas em vermelho:

Gramática original:

```
program ::= program [decl-list] begin stmt-list end
decl-list ::= decl {decl}
decl ::= type “:” ident-list “;”
ident-list ::= identifier {“,” identifier}
type ::= int | float | char

stmt-list ::= stmt {“;” stmt}
stmt ::= assign-stmt | if-stmt | while-stmt | repeat-stmt | read-stmt | write-stmt
assign-stmt ::= identifier “=” simple_expr
if-stmt ::= if condition then [decl-list] stmt-list end | if condition then [decl-list] stmt-list else
declaration stmt- list end
condition ::= expression
```

repeat-stmt ::= **repeat** [decl-list] stmt-list stmt-suffix
stmt-suffix ::= **until** condition
while-stmt ::= stmt-prefix [decl-list] stmt-list **end**
stmt-prefix ::= **while** condition **do**

read-stmt ::= **in** "(" identifier ")"
write-stmt ::= **out** "(" writable ")"
writable ::= simple-expr | literal

expression ::= simple-expr | simple-expr relop simple-expr
simple-expr ::= term | simple-expr addop term
term ::= factor-a | term mulop factor-a
fator-a ::= factor | ! factor | "-" factor
factor ::= identifier | constant | "(" expression ")"
relop ::= "==" | ">" | ">=" | "<" | "<=" | "!="
addop ::= "+" | "-" | ||
mulop ::= "*" | "/" | &&
constant ::= integer_const | float_const | char_const

Gramática alterada (alterações marcadas em azul):

program ::= **program** [decl-list] **begin** stmt-list **end**
decl-list ::= decl decl-list-suffix
decl-list-suffix ::= decl decl-list-suffix | λ
decl ::= type “,” ident-list “,”
ident-list ::= identifier {" , " identifier}
type ::= **int** | **float** | **char**

stmt-list ::= stmt stmt-suffix
stmt-list-suffix ::= stmt stmt-list-suffix | λ
stmt ::= assign-stmt “,” | if-stmt “,” | while-stmt “,” | repeat-stmt “,” | read-stmt “,” | write-stmt “,”

assign-stmt ::= identifier "=" simple_expr
if-stmt ::= **if** condition **then** [decl-list] stmt-list **if-suffix**
if-suffix ::= **end** | **else** declaration stmt- list **end**
condition ::= expression
repeat-stmt ::= **repeat** [decl-list] stmt-list stmt-suffix
stmt-suffix ::= **until** condition
while-stmt ::= stmt-prefix [decl-list] stmt-list **end**
stmt-prefix ::= **while** condition **do**
read-stmt ::= **in** "(" identifier ")"
write-stmt ::= **out** "(" writable ")"
writable ::= simple-expr | literal

expression ::= simple-expr [relop simple-expr]
simple-expr ::= term {addop simple-expr}
term ::= factor-a {mulop term}
fator-a ::= factor | ! factor | "-" factor
factor ::= identifier | constant | "(" expression ")"
relop ::= "==" | ">" | ">=" | "<" | "<=" | "!="
addop ::= "+" | "-" | ||
mulop ::= "*" | "/" | &&

constant ::= integer_const | float_const | char_const

2.2.3 First e Follow na Implementação do Analisador Sintático

Para a construção do Analisador Sintático, foi essencial a elaboração prévia das tabelas First e Follow. A seguir, apresentamos essas tabelas.

Regra	First	Follow
program	"program"	"\$"
decl-list	"int", "float", "char"	"begin", id, "if", "int", "float", "char", "while", "repeat", "in", "out"
decl	"int", "float", "char"	";", "begin", id, "if", "int", "float", "char", "while", "repeat", "in", "out"
ident-list	id	;", ", "
type	"int", "float", "char"	":"
stmt-list	id, "if", "int", "float", "char", "while", "repeat", "in", "out"	"end", "until"
stmt	id, "if", "int", "float", "char", "while", "repeat", "in", "out"	"end", "until" , ";"
assign-stmt	id	"end", "until" , ";"
if-stmt	"if"	"end", "until" , ";"
condition	"!", "-", id, número, "("	"then", "do", "end", "until" , ";"
repeat-stmt	"repeat"	"end", "until" , ";"
stmt-suffix	"until"	"end", "until" , ";"
while-stmt	"while"	"end", "until" , ";"
stmt-prefix	"while"	"int", "float", "char", id, "if", "int", "float", "char", "while", "repeat", "in", "out"
read-stmt	"in"	"end", "until" , ";"
write-stmt	"out"	"end", "until" , ";"
writable	"!", "-", id, número, "(" , literal)"
expression	"!", "-", id, número, "(")", "then", "do", "end", "until" , ";"
simple-expr	"!", "-", id, número, "(")", "then", "do", "end", "until" , ";" , "==" , ">" , ">=" , "<" , "<=" , "!=" , "+", "-", " "
term	"!", "-", id, número, "(")", "then", "do", "end", "until" , ";" , "==" , ">" , ">=" , "<" , "<=" , "!=" , "+", "-", " " , "*", "/" , "&&"
fator-a	"!", "-", id, número, "(")", "then", "do", "end", "until" , ";" , "==" , ">" , ">=" , "<" , "<=" , "!=" , "+", "-", " " , "*", "/" , "&&"
factor	id, número, "(")", "then", "do", "end", "until" , ";" , "==" , ">" , ">=" , "<" , "<=" , "!=" , "+", "-", " " , "*", "/" , "&&"
relop	"==" , ">" , ">=" , "<" , "<=" , "!="	!", "-", id, número, "("
addop	"+" , "-" , " "	!", "-", id, número, "("
mulop	"*" , "/" , "&&"	!", "-", id, número, "("
constant	número)", "then", "do", "end", "until" , ";" , "==" , ">" , ">=" , "<" , "<=" , "!=" , "+", "-", " " , "*", "/" , "&&"

No próximo capítulo, serão apresentados os resultados dos testes especificados.

3. Resultados

Para executarmos o programa a fim de avaliarmos a criação do Analisador Sintático, foram disponibilizados pela professora 5 arquivos testes. O objetivo era verificar se o compilador construído, reconheceria os erros, léxicos, semânticos e sintáticos existentes, identificando através de uma mensagem em qual linha houve o erro e abortando a execução. Em seguida, os erros seriam corrigidos pelo usuário e ao repetir a execução, ele deveria avaliar novamente e com sucesso. Nessa etapa, os únicos erros identificados devem ser os erros léxicos e sintáticos, destacados de vermelho nos programas de teste.

A seguir mostraremos os resultados de execução de todos os arquivos testes. E outro teste feito pelos autores do trabalho.

3.1 Início da execução

Informar o caminho onde está salvo o arquivo teste.

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\C
ode\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....
```

3.2 Programas de entrada e saída do compilador

Para a segunda etapa do trabalho, foram coletados os códigos de teste com os erros léxicos já corrigidos, produzidos ao fim da primeira etapa do trabalho. Dessa forma, os testes são feitos com ênfase nos erros sintáticos.

Teste do Programa 1:

```
program
  int: a,b,c;
  float: result;
  char: ch;

begin
  out("Digite o valor de a:");
  in (a);
  out("Digite o valor de c:");
  read(ch);
  b = 10;
  result = (a * ch)/(b 5 - 345.27);
  out("O resultado e: ");
  out(result);
  result = result + ch;
end
```

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\C
ode\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa
s_teste\programa1.txt

Inicilizando a análise.....
Erro sintático na linha 10. Esperado token 272 ('ASSIGN'), lido 289 '('')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
```

É detectado um erro na linha 10:

```
read(ch);
```

Read é um identificador, e nessa situação era esperado um sinal de atribuição, mas é lido um parêntese. Trocando read por in:

```
in(ch);
```

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'c:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa1.txt

Inicilizando a análise.....
Erro sintático na linha 12. Esperado token 290 (')'), lido 294 ('CONST_CHAR')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado um erro na linha 12:

```
result = (a * ch)/(b5 - 345.27);
```

Removendo o 5, de forma que fique:

```
result = (a * ch)/(b - 345.27);
```

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'c:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa1.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

O programa é lido e aceito.

Teste do Programa 2:

program

```
float: raio, area = 0.0;
```

begin

```
repeat
```

```
in(raio);
```

```
char: resposta;
```

```
if (raio > 0.0) then
```

```
    area = 3.0 * raio * raio;
```

```
    out (area);
```

```
end;
```

```
out ("Deseja continuar?");
```

```
in (resp);
```

```
until (resp == 'N' || resp == 'n');
```

end

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa_s_teste\\programa2.txt

Inicilizando a análise.....
Erro sintático na linha 2. Esperado token 286 (';'), lido 272 ('ASSIGN')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É obtido um erro na linha 2:

float: raio, area = 0.0;

Não é permitido atribuição no ato de declaração de variáveis, nessas circunstâncias o token esperado era vírgula, caso houvessem mais variáveis, ou ponto e vírgula. Removendo a atribuição, de forma que fique:

float: raio, area;

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa_s_teste\\programa2.txt

Inicilizando a análise.....
Erro sintático na linha 7. Esperado token 267 ('UNTIL'), lido 258 ('CHAR')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado um erro na linha 7:

char: resposta;

Não é permitido realizar declarações após o primeiro statement do bloco Repeat. Movendo essa linha para cima, antes do comando In:

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa_s_teste\\programa2.txt

Inicilizando a análise.....
Erro sintático na linha 14. Esperado token 290 (')'), lido 273 ('==')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 14:

until (resp == 'N' || resp == 'n');

Nessa linguagem, essa derivação é impossível. Ao derivar expression para conseguir os “(“ e “)”, é chegada na regra “(“ expression “)”. A partir daí, é possível obter uma operação relacional: simple-expr relop simple-expr, porém, para só é possível conseguir mais operações relacionais se utilizar mais parênteses. Substituindo por uma derivação possível e equivalente:

until ((resp == 'N') || (resp == 'n'));

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa2.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> |
```

O programa é lido e aceito.

Teste do Programa 3:

program

int: a, b, aux;

begin
 in (a);
 in(b);
 if (a>b) then
 int aux;
 aux = b;
 b = a;
 a = aux
 end;
 out(a);
 out(b)
end

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa3.txt

Inicilizando a análise.....
Erro sintático na linha 9. Esperado token 287 (':'), lido 259 ('ID')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> |
```

É encontrado erro na linha 9:

int aux;

É necessário o uso de dois pontos após o tipo das variáveis a serem declaradas. Adicionando os dois pontos:

int: aux;

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa3.txt

Inicilizando a análise.....
Erro sintático na linha 13. Esperado token 286 (';'), lido 262 ('END')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> |
```

É obtido erro na linha 13:

a = aux

Adicionando ponto e vírgula ao final da linha:

a = aux;

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa3.txt

Inicilizando a análise.....
Erro sintático na linha 14. Esperado token 290 (')'), lido 286 (';')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 14:

out(a;

Fechando o parêntese que faltava:

out(a);

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa3.txt

Inicilizando a análise.....
Erro sintático na linha 16. Esperado token 286 (';'), lido 262 ('END')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 16:

out(b)

Adicionando o ponto e vírgula ao final do comando:

out(b);

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa3.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

O programa é lido até o final e aceito.

Teste do Programa 4:

program

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

begin

repeat

out("A");

in(a);

out("B");

in(b);

out("C");

in(c);

%Verifica o maior

```
if ( (a>b) and (a>c) ) end
maior = a

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

  else
    maior = c
  end
end;
out("Maior valor: ");
out (maior);
out ("Outro? ");
in(outro);
until (outro == 0)
end
```



É encontrado erro na linha 2:

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

O tipo deve estar antes das variáveis. Corrigindo:

Int: a, b, c, maior, outro;

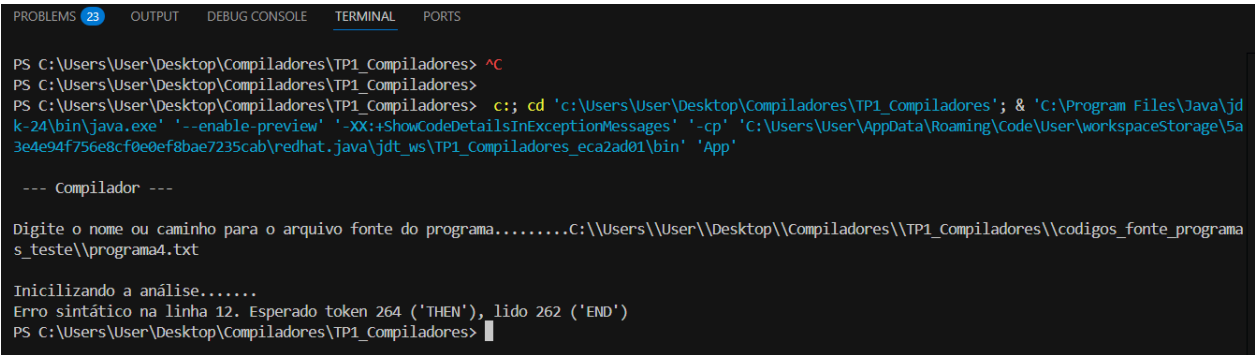


É encontrado erro na linha 12:

if ((a>b) and (a>c)) end

And é um identificador. Substituindo por &&:

if ((a>b) && (a>c)) end

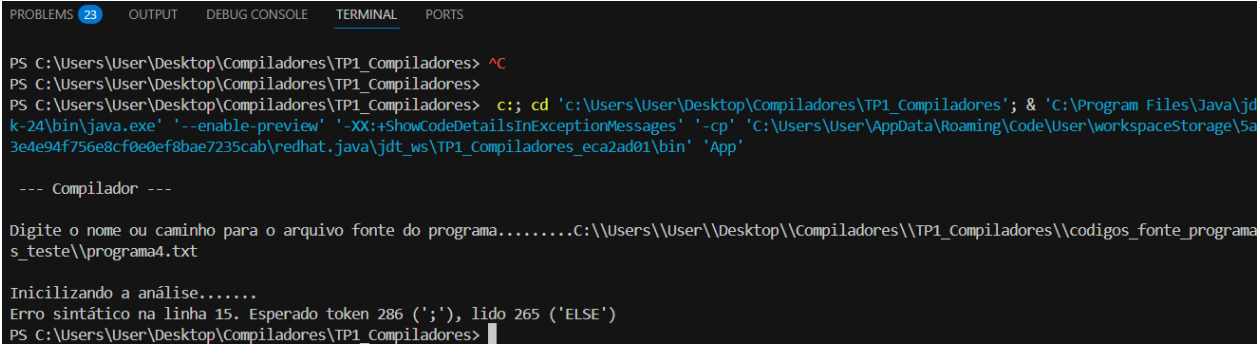


É encontrado outro na linha 12:

```
if ( (a>b) && (a>c) ) end
```

É esperado um then após o prefixo de um IF. Adicionando o then e removendo o end:

```
if ( (a>b) && (a>c) ) then
```

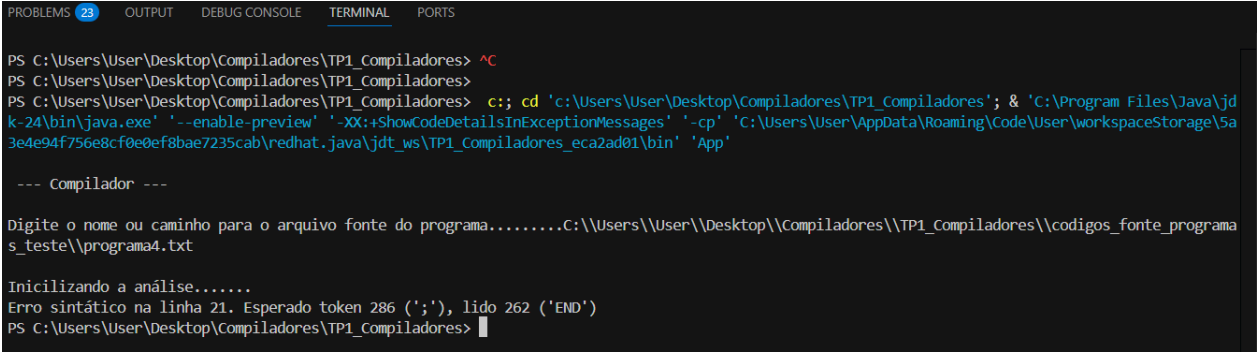


É encontrado erro na linha 15:

```
maior = a
```

Adicionando ponto e vírgula:

```
maior = a;
```

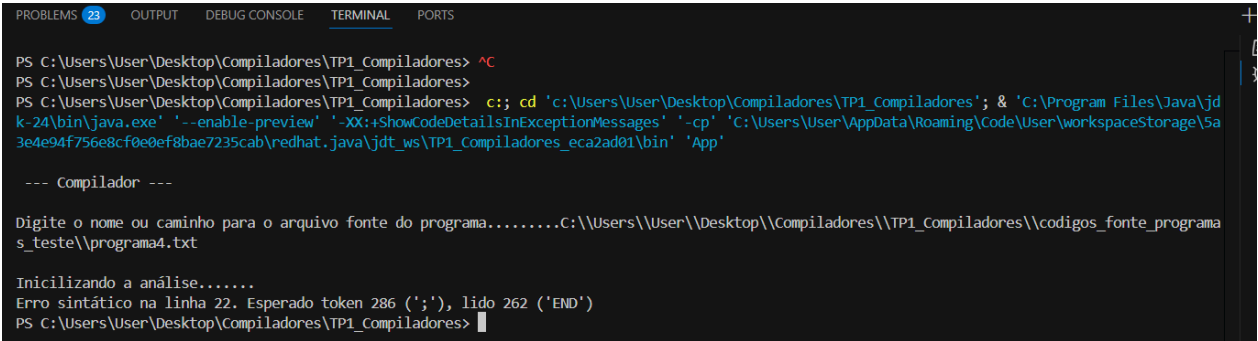


É encontrado erro na linha 21:

```
maior = c
```

Adicionando o ponto e vírgula:

```
maior = c;
```

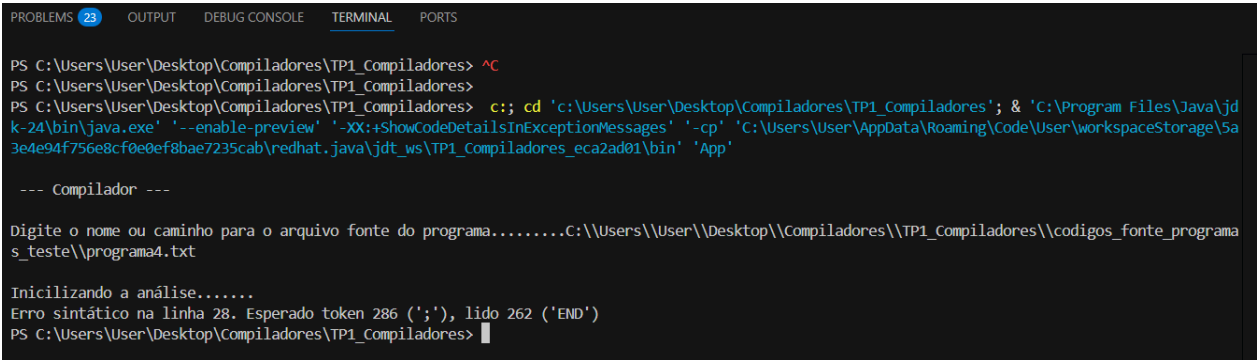


É encontrado erro na linha 22:

```
end
```

Adicionando o ponto e vírgula:

```
end;
```

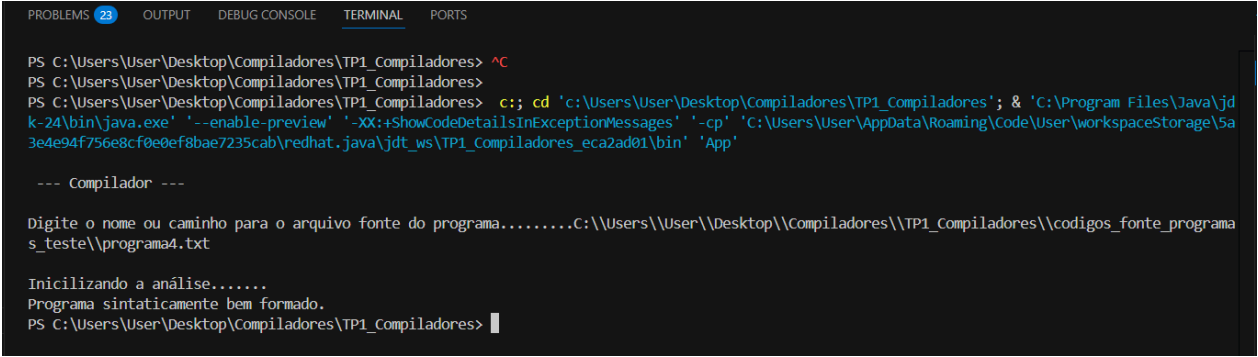


É encontrado erro na linha 28:

until (outro == 0)

Adicionando o ponto e vírgula:

until (outro == 0);



O programa é lido e aceito.

Teste do Programa 5:

programa

declare

inteiro: pontuacao, pontuacaoMaxina, disponibilidade;

char: pontuacaoMinima

begin

disponibilidade = 'S';
pontuacaoMinima = 50;
pontuacaoMaxima = 100;
out("Pontuacao Candidato: ");
in(pontuacao);
out("Disponibilidade Candidato: ");
in(disponibilidade);
{ Comentario
grande}

while (pontuacao>0 && (pontuação<=pontuacaoMaxima) do

int: cont;
cont = cont + 1;
if ((pontuação > pontuacaoMinima) && (disponibilidade==1)) then
out("Candidato aprovado")
else
out("Candidato reprovado")
end

out("Pontuacao Candidato: ");
in(pontuacao);
out("Disponibilidade Candidato: ");
in(disponibilidade);

end

out (cont);

end

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 1. Esperado token 260 ('PROGRAM'), lido 259 ('ID')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
```

Encontrado erro na linha 1:

programa

Substituindo PROGRAMA por PROGRAM:

program

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 2. Esperado token {int, float, char}, encontrado declare (ID)
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
```

É encontrado erro na linha 2:

declare

Removendo o identificador declare:

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 2. Esperado token {int, float, char}, encontrado inteiro (ID)
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
```

É encontrado erro na linha 2:

inteiro: pontuacao, pontuacaoMaxina, disponibilidade;

Substituindo o identificador inteiro por int:

int:pontuacao, pontuacaoMaxina, disponibilidade;

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 4. Esperado token 286 (';'), lido 261 ('BEGIN')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
```

É encontrado erro na linha 4:

char: pontuacaoMinima

Adicionando o ponto e vírgula:

char: pontuacaoMinima;

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 15. Esperado token 290 (')'), lido 269 ('DO')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 15:

while (pontuacao>0 && (pontuação<=pontuacaoMaxima) do

Fechando o parêntese que estava desbalanceado:

while (pontuacao>0 && (pontuação<=pontuacaoMaxima)) do

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 20. Esperado token 286 (';'), lido 265 ('ELSE')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 20:

out("Candidato aprovado")

Adicionando o ponto e vírgula:

out("Candidato aprovado");

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 22. Esperado token 286 (';'), lido 262 ('END')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 22:

out("Candidato reprovado")

Adicionando o ponto e vírgula:

out("Candidato reprovado");

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c++; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 24. Esperado token 286 (';'), lido 271 ('OUT')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 24:

end

Adicionando o ponto e vírgula:

end;

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Erro sintático na linha 30. Esperado token 286 (';'), lido 271 ('OUT')
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É encontrado erro na linha 30:

end

Adicionando o ponto e vírgula:

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa5.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

O programa é lido e aceito.

Teste do Programa 6:

O programa 6 foi alterado por um mais interessante:

```
program
    int: a,b;
begin
    out("Testando");
end
mais$ tokens$
```

A motivação deste programa é testar se os tokens após o end relacionados ao begin são lidos. É esperado que após o end haja um fim de arquivo.

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c;; cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa6.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

O programa é lido e aceito. Isso ocorre pois a análise termina após casar o último token “end”. Não é verificado se há um fim de arquivo em seguida.

Criando uma nova tag para o fim do arquivo:

```
END_OF_FILE = 295;
```

Na classe Lexer, alterando a regra de fim de arquivo:

```
// Fim de arquivo
```



```
        if (ch == (char) -1) {  
            return new Token(Tag.END_OF_FILE);  
        }  
    }
```

No analisador sintático, alterando a regra program, adicionando, após casar o último end, o casamento de um fim de arquivo:

```
// métodos do analisador sintático descendente:  
void program() throws SyntaxException, LexicalException {    //ok  
    eat(Tag.PROGRAM);  
    decl_list();  
    eat(Tag.BEGIN);  
    stmt_list();  
    eat(Tag.END);  
    eat(Tag.END_OF_FILE);  
}
```

Executando novamente o programa 6:

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c:: cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'  
  
--- Compilador ---  
  
Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa6.txt  
  
Inicilizando a análise.....  
Erro sintático na linha 6. Esperado token 295 ('END_OF_FILE'), lido 259 ('ID')  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

É obtido um erro na linha 6. É esperado um fim de arquivo, mas é lido um id. Removendo a última linha do programa 6 e executando novamente:

```
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> ^C  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores>  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> c:: cd 'c:\Users\User\Desktop\Compiladores\TP1_Compiladores'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\5a3e4e94f756e8cf0e0ef8bae7235cab\redhat.java\jdt_ws\TP1_Compiladores_eca2ad01\bin' 'App'  
  
--- Compilador ---  
  
Digite o nome ou caminho para o arquivo fonte do programa.....C:\Users\User\Desktop\Compiladores\TP1_Compiladores\codigos_fonte_programa_s_teste\programa6.txt  
  
Inicilizando a análise.....  
Programa sintaticamente bem formado.  
PS C:\Users\User\Desktop\Compiladores\TP1_Compiladores> █
```

O programa é lido e aceito.

Por fim, testando os programas anteriores já corrigidos, após essa alteração no código:


```
--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa
s_teste\\programa1.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores> c++; cd 'c:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores'; & 'C:\\Program Files\\Java\\jd
k-24\\bin\\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\\Users\\User\\AppData\\Roaming\\Code\\User\\workspaceStorage\\5a
3e4e94f756e8cf0e0ef8bae7235cab\\redhat.java\\jdt_ws\\TP1_Compiladores_eca2ad01\\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa
s_teste\\programa2.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores> c++; cd 'c:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores'; & 'C:\\Program Files\\Java\\jd
k-24\\bin\\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\\Users\\User\\AppData\\Roaming\\Code\\User\\workspaceStorage\\5a
3e4e94f756e8cf0e0ef8bae7235cab\\redhat.java\\jdt_ws\\TP1_Compiladores_eca2ad01\\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa
s_teste\\programa3.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores> c++; cd 'c:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores'; & 'C:\\Program Files\\Java\\jd
k-24\\bin\\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\\Users\\User\\AppData\\Roaming\\Code\\User\\workspaceStorage\\5a
3e4e94f756e8cf0e0ef8bae7235cab\\redhat.java\\jdt_ws\\TP1_Compiladores_eca2ad01\\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa
s_teste\\programa4.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores> c++; cd 'c:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores'; & 'C:\\Program Files\\Java\\jd
k-24\\bin\\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\\Users\\User\\AppData\\Roaming\\Code\\User\\workspaceStorage\\5a
3e4e94f756e8cf0e0ef8bae7235cab\\redhat.java\\jdt_ws\\TP1_Compiladores_eca2ad01\\bin' 'App'

--- Compilador ---

Digite o nome ou caminho para o arquivo fonte do programa.....C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores\\codigos_fonte_programa
s_teste\\programa5.txt

Inicilizando a análise.....
Programa sintaticamente bem formado.
PS C:\\Users\\User\\Desktop\\Compiladores\\TP1_Compiladores> 
```

Todos continuam sendo aceitos.

4. Conclusão

Compreender o funcionamento de um compilador é de suma importância para estudantes e futuros profissionais da área de computação. É fundamental saber que cada linguagem de programação possui sua própria gramática. Durante o processo de compilação, o programa é avaliado para garantir que sua estrutura esteja coerente com as regras de análise sintática e semântica dessa gramática. Essa compreensão é crucial não apenas para escrever código correto, mas também para entender a origem dos erros reportados durante a fase de compilação de um programa em desenvolvimento.

A elaboração deste trabalho proporcionou uma experiência prática valiosa, permitindo vivenciar como são realizadas as etapas de Análise Léxica e Análise Sintática, as duas primeiras fases essenciais para o funcionamento de um compilador.

5. Referência Bibliográfica

1. AHO, A. V. Jeffrey; Sethi, R.; Lam, MS (2008). **Compiladores. Princípios, técnicas e ferramentas.** São Paulo: Addison-Wesley, Pearson.