



UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO

CONSTRUÇÃO DE COMPILADORES

Documentação Interna - T2

Áquila Oliveira, 759313

Carlos Eduardo Fontaneli , 769949

Ingrid Lira dos Santos, 790888

São Carlos - SP

29 de maio de 2023

1 Documentação Interna

1.1 Documentação interna da main.py

```
1 import sys
2 from antlr4 import *
3 from LASyntaxLexer import LASyntaxLexer
4 from LASyntaxParser import LASyntaxParser
5 from LASyntaxParserErrorListener import LASyntaxParserErrorListener
6 from LASyntaxLexerErrorListener import LASyntaxLexerErrorListener
7
8 # Leitura dos nomes e abertura dos arquivos
9 input_file_name = sys.argv[1]
10 output_file_name = sys.argv[2]
11 input_stream = FileStream(input_file_name, encoding="utf-8")
12 output = open(output_file_name, "w")
13
14 # Analisador élxico
15 lexer = LASyntaxLexer(input_stream)
16
17 # Tokens Stream
18 stream = CommonTokenStream(lexer)
19
20 # Analisador á sinttico
21 parser = LASyntaxParser(stream)
22
23 # Atualizando os detectores de erros pelos personalizados
24 lexer.removeErrorListeners()
25 parser.removeErrorListeners()
26 lexer.addErrorListener(LASyntaxLexerErrorListener())
27 parser.addErrorListener(LASyntaxParserErrorListener())
28
29 # Lista de erros
30 errors = []
31
32 try:
33     # Chamando o parser
34     parser.programa()
35
36 except Exception as e:
```

```

37     errors.append(str(e))
38     errors.append("Fim da compilacao")
39     for error in errors:
40         output.write(f"{error}\n")
41
42     output.close()

```

1.2 Documentação interna do LASyntaxParserErrorListener.py

```

1  # Criando o detector de erro de sintaxe personalizado
2  from antlr4 import *
3  from antlr4.error.ErrorListener import ErrorListener
4
5  class LASyntaxParserErrorListener(ErrorListener):
6      def syntaxError(self, recognizer, offendingSymbol, line, column, msg
7          , e):
8          # Erro óprximo ao fim do arquivo
9          if offendingSymbol.text == "<EOF>":
10             raise Exception(
11                 f"Linha {offendingSymbol.line}: erro sintatico proximo a
12                 EOF"
13             )
14             # Erro localizado durante o programa
15             else:
16                 raise Exception(
17                     f"Linha {offendingSymbol.line}: erro sintatico proximo a
18                     {offendingSymbol.text}"
19                 )

```

1.3 Documentação interna do LASyntaxLexerErrorListener.py

```

1  from antlr4 import *
2  from antlr4.error.ErrorListener import ErrorListener
3
4  # Criando o detector de erro élxico personalizado
5  class LASyntaxLexerErrorListener(ErrorListener):
6      def syntaxError(self, recognizer, offendingSymbol, line, column, msg
7          , e):
8          error = str(e.input)[e.startIndex]
9
10         # Erro de fechamento de ácomentrio
11         if error == "{":

```

```
11         raise Exception(f"Linha {line}: comentario nao fechado")
12
13     # Erro de fechamento de cadeia literal
14     elif error == '":
15         raise Exception(f"Linha {line}: cadeia literal nao fechada")
16
17     # Erro de simbolo desconhecido
18     else:
19         raise Exception(f"Linha {line}: {error} - simbolo nao
identificado")
```