Linguagem 2020.1 - T4

Alunos: Benjamin Mezger e Pedro Prado

Aplicação: Executar arquivo **TheCompilerApp**

Lista de erros da linguagem 2020.1

Erros Sintaticos:

1. Erro: Forma geral do programa incorreto
2. Erro: Descricao do programa incorreto
3. Erro: Declaração do corpo incorreto
4. Erro: Erro de atribuição (designate)
5. Erro: Erro no corpo do write
6. Erro: Erro no corpo do write
7. Erro: no comando write all
8. Erro: Comando write incorreto
9. Erro: Declaração do comando avaliate incorreta
10. Erro: Declaração do comando repeat incorreta
11. Erro: Expressão incorreta
12. Erro: Declaração de comando incorreta
13. Erro: Clausula de teste incorreta.
14. Erro: Verificação de resultado logico incorreto
15. Erro: Forma geral de declaração de constante e variável incorreta.
16. Erro: Lista de identificadores incorreto
17. Erro: declaracao de enum incorreta.
18. Erro: declaracao de enum interna incorreta

Erros lexicos:

1. Símbolo inválido, linha X; Coluna Y <token> <ID>
2. Identificador inválido, linha X; Coluna Y <token> <id>
3. Token inválido, linha X; Coluna Y <token> <id>
4. Error lexico: Comentário de bloco não encerrado <id>

Erros semanticos:

1. Error Semantico 1: identificador ja declarado
2. Error Semantico 2: identificador de variavel indexada
3. Error Semantico 3: identificador de variavel - faltou indice
4. Erro Semantico 4: identificador não declarado ou identificador usado como identificador do programa ou identificador é uma constante/tipo enumerado
5. Error Semantico 5: tipo inválido para constante de tipo logic
6. Error Semantico 6: tipo inválido para constante de tipo enumerado
7. Error Semantico 7: identificador não declarado ou identificador usado como identificador do programa ou identificador é um tipo enumerado
8. Erro Semântico 8: identificador de constante ou variável não indexada

BNF da linguagem 2020.1

<type> ::= integer | real | string | logic

<programa> ::= do this *IDENTIFICADOR* [ ]
<declaration_type_enum>
<declaration_constants_and_variables>
body [
<list_of_commands>
]
<description>

<list_of_commands> ::= <repeat><list_of_commands_cont> |
<avaliate><list_of_commands_cont> | <write><list_of_commands_cont> |
<write_all><list_of_commands_cont> | <designate><list_of_commands_cont> |
<read><list_of_commands_cont>

<list_of_commands_cont> ::= <list_of_commands> | Epsilon

<description> ::= description STRING_LITERAL | Epsilon

<declaration_type_enum> ::= declaration type [<inner_enum_declaration>] | Epsilon

<inner_enum_declaration> ::= IDENTIFIER is <enum_values>
<enum_continuation>.<inner_enum_decla_cont>

<enum_values> ::= NUMBER_REAL | NUM | STRING_LITERAL | <identifiers>

<enum_continuation> ::= , <enum_values> | Epsilon

<inner_enum_decla_cont> ::= <inner_enum_declaration> | Epsilon

<declaration_constants_and_variables> ::= declaration constant and variable [
<inner_declarations>
] | Epsilon

<inner_declarations> ::= as <start_declaration>

<start_declaration> ::= <start_variable> | <start_constant>

<start_constant> ::= constant
<constant_declaration>
<end_variable>
<start_variable> ::= variable
<variable_declaration>
<end_constant>

<end_variable> ::= as variable
<variable_declaration> | Epsilon

<end_constant> ::= as constant
<constant_declaration> | Epsilon

<constant_declaration> ::= <identifiers_list> is <type> = <result> . <constant_decla_cont>

<constant_decla_cont> ::= <constant_declaration> | Epsilon

<variable_declaration> ::= <identifiers_list> is <type> . <variable_decla_cont>

<variable_decla_cont> ::= <variable_declaration> | Epsilon

<identifiers_list> ::= <identifiers> <identifier_cont>

<identifier_cont> ::= , <identifiers_list> | Epsilon

<identifiers> ::= IDENTIFICADOR <index>

<type> ::= INTEGER | REAL | STRING | LOGIC

<result> ::=STRING_LITERAL | NUM | NUMBER_REAL

<repeat> ::= repeat this <expression>

[<list_of_commands>] .

<avaliate> ::= avaliate this <expression>
<logic_result> .

<logic_result> ::= true result [<list_of_commands>] <true_result_cont> |
untrue result [<list_of_commands>] <untrue_result_cont>

<true_result_cont> ::= untrue result [<list_of_commands>] | Epsilon

<untrue_result_cont ::= true result [<list_of_commands>] | Epsilon

<expression ::= <arithmetic_or_logic_expression> <expression_cont>

<expression_cont> ::= == <arithmetic_or_logic_expression>
| != <arithmetic_or_logic_expression>
| << <arithmetic_or_logic_expression>
| >> <arithmetic_or_logic_expression>
| <<= <arithmetic_or_logic_expression>
| >>= <arithmetic_or_logic_expression>
| Epsilon

<arithmetic_or_logic_expression> ::= <second_term> <lesser_priority>

<lesser_priority> ::= + <second_term> <lesser_priority>
| - <second_term> <lesser_priority>
| **|(simbolo do OU)** <second_term> <lesser_priority>
| Epsilon

<second_term> ::= <first_term> <medium_priority>

<medium_priority> ::= * <first_term> <medium_priority>
| / <first_term> <medium_priority>
| % <first_term> <medium_priority>
| %% <first_term> <medium_priority>
| & <first_term> <medium_priority>
| Epsilon

<first_term> ::= <element> <top_priority>

<top_priority> ::= ** <element> <top_priority>
| Epsilonlist_of_

<element> ::= IDENTIFIER <index>
| NUM
| NUMBER_REAL

| STRING_LITERAL
| true
| untrue
| (<expression>)
| ! (<expression>)

<index> ::= [NUM] | Epsilon

<write> ::= write this [<write_body>] .

<write_all> ::= write all this [<write_body>] .

<write_body> ::= <enum_values> <write_body_cont>

<write_body_cont> ::= , <write_body> | Epsilon

<designate> ::= designate this <list_of_identifiers> as <expression> .

<read> ::= read this [<identifiers_list>] .