**RELATÓRIO – LEXER**

Aluno 1: Caio de Souza Célio

Aluno 2: Gabriel Henrique Dias Werneck

**CLASSES, MÉTODOS E ATRIBUTOS**

1. <*classe*> **Compiler:** Classe principal, responsável por iniciar o Lexer, bem como as futuras partes do compilador.
2. <*classe*> **TabelaSim:** Classe que manipula a Tabela de Símbolos, com o uso de um HashMap abrange o Token e o Identifier das palavras lidas.
   * + - <*atributo*> symbolTable: Um HashMap responsável por armazenar os elementos da tabela de símbolos;
   1. <*método*> Token returnToken(String lexeme): Pesquisa na tabela de símbolos se há algum token com determinado lexema;
   2. <*método*> Identifier getIdentifier(Token w): Retorna a identificação do token;
   3. <*método*> String toString(): Sobrescreve o método toString para imprimir as informações da tabela de símbolos;
   4. <*método*> put(Token w, Identifier i): Utilizado para adicionar elementos no HashMap.
3. <*classe*> **Token**: Classe responsável por abranger a impressão de um token e o controle de linhas e colunas.

Inclui os getters e setters das seguintes variáveis:

* + - * <*atributo*> lexeme: Recebe o lexema, ou seja, a palavra lida no arquivo que está sendo compilado;
      * <*atributo*> Type classType: Recebe o tipo do lexema, ou seja, identifica se ela é uma Key Word, um char e etc;
      * <*atributo*> line: Recebe o valor da linha;
      * <*atributo*> column: Recebe o valor da coluna.
  1. <*método*> Token(Type classType, String lexeme, int line, int column): Imprime o tipo da palavra lida, a palavra lida, e o seu posicionamento no aspecto de linha e coluna;
  2. <*método*> toString(): Formata a saída de parte do Token;
  3. <*método*> resetColumn (int column): Resetar a coluna para o valor inicial.
  4. <*método*> ColumnAdd (int column): Adicionar uma coluna;
  5. <*método*> ColumnAddTabMode (int column): Adicionar 3 colunas, pois \t é igual a 3 espaços;
  6. <*método*> ColumnReturn (int column): Subtrair uma coluna;
  7. <*método*> resetLine (int line): Resetar uma linha;
  8. <*método*> LineAdd (int line): Adicionar uma linha.

1. <*classe*> **Type**: Classe que enumera os tipos de dados lidos no arquivo pelo lexer.
2. <*classe*> **Identifier**: Classe que contém uma string para ser identificadora no HashMap na Tabela de Símbolos (<*classe*> **SymbolTable**).
3. <*classe*> **ErrorMessages**: Classe responsável por fornecer métodos com mensagens de erro.
   1. <*método*> openFileError() : Informa erro de abertura do arquivo.
   2. <*método*> programError() : Informa erro no programa ou falha na tabela de símbolos.
   3. <*método*> closeFileError() : Informa erro ao fechar o programa.
   4. <*método*> readFile() : Informa erro na leitura do arquivo.
   5. <*método*> lexerError(String message) : Informa erro léxico.
      1. <*String*> message: Parâmetro para a mensagem de erro.
4. <classe> **Lexer**: Classe responsável pela manipulação do Lexer. Os atributos são:
   * + <*atributo*> EOF: Aponta o final do arquivo.
     + <*atributo*> lastChar: O último caractere lido.
     + <*atributo*> line: Número de linhas.
     + <*atributo*> column: Número de colunas.
     + <*atributo*> TS:
     + <*atributo*> file\_reference:

As instâncias realizadas são:

TK = <*classe*> **Token**: Retorna os métodos para manipular as linhas e colunas. TK. <*método*> resetColumn (int column) e outros.

EM = <*classe*> **ErrorMessages**: Retorna as mensagens de erro. EM. <*método*> openFileError(), EM. <*método*> programError() e outros.

* 1. <*método*> Lexer(String file): Abertura do arquivo;
     1. <*String*> file: Recebe como parâmetro o nome do arquivo a ser aberto.
  2. <*método*> closeFile(): Encerra a instância do arquivo;
  3. <*método*> returnCharPosition(): Retorna uma posição de leitura de char;
  4. <*método*> nextChar(char c) : Avança a leitura de um char;
     1. <*char*> c: Parâmetro de abrange o char lido.
  5. <*método*> proxToken(): Avança a leitura do token.