CENTRO UNIVERSITÁRIO DE BELO HORIZONTE

CIÊNCIA DA COMPUTAÇÃO

31 de março, 2018

Aluno: Cássio Ribeiro

Disciplina: Compiladores

Sumário

ANALISADOR LÉXICO E TABELA DE SÍMBOLOS

PROGRAM.CS:
Main() -> Busca o caminho da pasta de testes e inicializa o analisador léxico.

LEXER.CS:
Lexer() -> Inicializa a análise léxica, abre o arquivo, faz a chamada para o método de impressão dos tokens, fecha o arquivo e exibe uma mensagem de termino.

ImprimeToken() -> Busca e depois exibe todos os token que foram encontrados na leitura do arquivo.

FechaArquivo() -> Fecha o arquivo instance_file de input_data

SinalizaErro() -> Chama o método ErrorLexer para exibir a mensagem de erro.

RetornaPonteiro() -> Volta uma posição do buffer de leitura.

ProxToken() -> Obtém próximo token.

TABLEOFSYMBOLS.CS:
TableOfSymbols() -> Cria a tabela de símbolos.

Add() -> Adiciona um token na tabela de símbolos.

ShowAllTableSymbols() -> Mostra todos os símbolos contidos na tabela de simbolos.

ReturnToken() -> Verifica se já existe o lexema dentro da tabela de símbolos. Vai ser usado esse método somente para diferenciar ID e KW.

ToStrinf() -> Sobrepondo o ToString() do c#.

TOKEN.CS:
Token() -> Construtor da classe, atribui valores para os atributos.

ToString() -> Sobrepondo o ToString() do c#.

TOKENENUM.CS:
Tem todos os nomes para os tokens do compilador PasC.

MESSAGE.CS:
ShowToken() -> Mostra o token.

ShowTableSymbols() -> Mostra o simbolo da tabela.

EndOfBuild() -> Mostra a mensagem de termino da compilação do arquivo lido.

Print() -> Mostra a mensagem no console.

ERROMESSAGE.CS:
Error() -> Adiciona o erro na lista de erros encontrados.

ErrorCloseArchive() -> Erro ao fechar o arquivo.

ErrorLexer() -> Erro léxico.

ErroRead() -> Erro na leitura do caracter ou retorno de um caracter.

ErrorStart() -> Erro ao inicializar o compilador.

AddError() -> Adiciona a mensagem de erro dentro da lista de erros.

ShowErrorFound() -> Faz um foreach na lista de erros, exibindo todos os erros encontrados.

Print() -> Mostra o erro no console.

TESTES:
Além dos testes feitos ao longo da construção do compilador, foi criado três trechos de código PasC que são aceitos pelo compilador e três que não são. Os arquivos no formato .txt se encontra dentro da solução do projeto dentro da pasta Testes.

Para poder testar os 6 arquivos que estão no projeto, siga os seguintes passos:

CASOS DE SUCESSO:
Basta trocar o número da string, que corresponde ao nome do arquivo.

Deixe a linha de baixo comentada para os testes de sucesso.

```
file = file.Replace(@"bin\Debug\netcoreapp2.0", @"Tests\Test_Success_3.txt");
//file = file.Replace(@"bin\Debug\netcoreapp2.0", @"Testes\Test_Error_3.txt");
```

CASOS DE ERROS:
Basta trocar o número da string, que corresponde ao nome do arquivo.

Deixe a linha de cima comentada para os testes de erros.

```
//file = file.Replace(@"bin\Debug\netcoreapp2.0", @"Tests\Test_Success_3.txt");
file = file.Replace(@"bin\Debug\netcoreapp2.0", @"Testes\Test_Error_3.txt"); //
```

Essas duas linhas de código fazem referência a pasta de Test que está no arquivo Program.cs dentro do solução do projeto.

Qualquer problema que tiver na leitura do arquivo, você poderá adicionar o caminho (diretório da pasta no computador) para uma pasta na área de trabalho para a realização dos testes.

Exemplo:

Tire o atributo file e coloque dentro dos parênteses o @"", e entre as aspas duplas coloque o diretório local para a pasta de testes.

```
Lexer lexer = new Lexer(@"C:\Users\Cassio Ribeiro\Desktop\Compilers\Tests\Test_Error_3.txt");
```

EXECUTANDO APLICAÇÃO

Passos:

1 – Instalar o .NET (dotnet) core na máquina.

Link: <https://www.microsoft.com/net/learn/get-started/windows>.

2 – Instalar o Visual Studio Code (Editor de texto).

Link: <https://code.visualstudio.com/download>.

3 – Abrir o projeto dentro do Visual Studio Code.

Menu: File-> Open Folder -> Selecione a pasta do projeto do diretório da máquina.

4 – Após abrir o projeto no Visual Studio Code, aperte as teclas CRTL + ', vai abrir o terminal do VSCode, com isso siga os passos abaixo para pode executar a aplicação:

Escreva no terminal **dotnet restore,** para poder restaurar as dependências da aplicação, após o termino da restauração, digite no terminal **dotnet run,** para poder executar o compilador. Com isso pode se seguir os passos para poder testar a aplicação, como mudança de arquivo de sucesso e erro, que são mostrados no modulo de **Testes**.

Caso não consiga executar a aplicação mesmo seguindo os passos acima e possível ver melhores detalhes para a instalar no Youtube. Segue links:

Instalação do .NET Core <https://www.youtube.com/watch?v=9MBnWy04wRI>.

Instalação do VSCode <https://www.youtube.com/watch?v=ocYgwQfTexY>.