

AnalizadorDeStrings

Autoria

João Pedro Ferreira Pedreira - 202076009

Miguel Sales de Almeida Lopes - 202076024

Como executar?

Primeiramente, é necessário instalar o php versão ≥ 8.0 em sua máquina. Para isso, entre no site <https://windows.php.net/download#php-8.1> e baixe o zip da versão VS16 x64 Non Thread Safe. Após isso, descompacte o arquivo no diretório raiz de sua máquina. (C:). Então, altere o nome do arquivo "php.ini-development" para "php.ini". É necessário também adicionar o php em suas variáveis de ambiente. Acesse o painel de controle, entre em sistema, clique em avançado e em variáveis de ambiente. Na seção Variáveis do sistema, selecione Path, agora clique em Editar, em Valor da variável, vá até o final do campo de texto, agora iremos colocar o caminho onde o nosso PHP está, acrescente antes um ; (ponto e vírgula) para finalizar os caminhos anteriores e coloque c:\php, então, ficará assim:;c:\php. Reinicie o PC para salvar as mudanças. Finalmente, utilize o comando "php {caminho do arquivo main.php}" para executar o programa.

Estrutura do projeto

O arquivo "main.php" é o principal do projeto, ele é o único que precisa ser executado diretamente para executar o programa. Em seu início, são instanciadas os objetos necessários para o funcionamento do programa, os quais serão descritos nesse documento. Após isso, é feito um loop do tipo do While, o qual é executado enquanto o usuário não digita uma tag ou comando inválido. No final do arquivo, são definidos os comandos, com suas funções respectivas ou mensagens de erro caso não estejam implementados corretamente.

O arquivo "autoload.php" é o responsável por permitir o uso de namespaces e importação de classes no php de forma mais simples ao usuário, para que não seja necessário digitar o caminho real do arquivo na máquina ou utilizar a função require do php.

Além disso, na pasta raiz do projeto estão presentes alguns arquivos .txt com definições relevantes. Seus nomes são auto-explicativos para seus conteúdos.

Os arquivos e pastas mais relevantes do projeto estão localizados na pasta src e serão descritos abaixo. Todos os arquivos descritos a seguir seguem a arquitetura de orientação a objetos (classes, objetos, herança, etc). Tomamos essa decisão estrutural para que se tornasse mais fácil a manipulação, alteração e criação das tags, strings e derivados, pois o php funciona muito bem com

esse tipo de padrão de código, tornando o programa de fácil entendimento para qualquer um com acesso ao código fonte.

Pasta/arquivo Arquivo

Essa classe é responsável pela manipulação de arquivos. É capaz de abrir, ler, escrever e fechar arquivos com informações relevantes, como arquivos com Tags e comandos. Além disso, possui métodos de auxílio, como a que retira espaços em branco de uma string digitada pelo usuário ou carregada de um arquivo.

Pasta/arquivo Aviso

Essa classe realiza exclusivamente a exibição de avisos na tela do usuário, seja de warning, informação ou erro. É passado o tipo de aviso e a mensagem para seu método estático "mostrarAviso", o qual printa na tela a mensagem com seu tipo de aviso que foi enviado como argumento.

Pasta/arquivo String

Essa classe está ainda em fase de implementação para a entrega da segunda parte do trabalho.

Pasta/arquivo Tags_Comandos

Em seu construtor, carrega os arquivos com os comandos comuns, comandos unários e tags, instanciando um novo objeto Arquivo para isso. Além dos métodos de exibição de comandos e tags, salva as Tags em um arquivo recebido por parâmetro, chamando a função "salvaTags" do próprio objeto Arquivo que foi instanciado. O método "isCommand" verifica se, o que foi digitado pelo usuário é de fato um comando, verificando pelo atributo "comandos", que foi definido no construtor. O método "isTag" verifica se as tags digitadas pelo usuário, também salvas no atributo "tags" do construtor, são tags válidas. O método "defineTagsDoUsuário" define um array, com as chaves sendo as tags do usuário, e o valor sendo a definição em si. O "isUnaryTag" verifica se a tag unária digitada pelo usuário é válida, e por último, a classe carregaTagsExternas, recebendo o caminho do arquivo como parâmetro, carrega tags de arquivos externos.