Projeto II

Gerado por Doxygen 1.9.1

Capítulo 1

Índice dos componentes

1.1 Lista de componentes

Lista de classes, estruturas, uniões e interfaces com uma breve descrição:

Capítulo 2

Documentação da classe

2.1 Referência à classe structures::LinkedList

Lista encadeada genérica.

`#include <linked_list.h>`

Membros públicos

- LinkedList ()
  Construtor padrão, inicializa uma lista vazia.
- ~LinkedList ()
  Destrutor.
- void clear ()
  Deleta todos os elementos.
- void push_back (const T &data)
  Insere no final.
- void push_front (const T &data)
  Insere no inicio.
- void insert (const T &data, std::size_t index)
  Insere no índice especificado.
- void insert_sorted (const T &data)
  Insere em ordem.
- T & at (std::size_t index)
  Acessa o elemento na posição, lançando exceções se necessário.
- T & operator[ ] (std::size_t index)
  Acessa o elemento na posição.
- void pop (std::size_t index)
  Retirar da posição.
- void pop_back ()
  Retirar do final.
- void pop_front ()
  Retirar do inicio.
- void remove (const T &data)
  Remover o elemento.

- bool empty () const

  Verifica se a lista está vazia.
- bool contains (const T &data)

  Verifica se o elemento está na lista.
- std::size_t find (const T &data) const

  A posição do elemento específico.
- std::size_t size () const

  Tamanho da lista.
- std::size_t binary_search (const T &data)

  Busca binária na lista, retorna size se o elemento não estiver presente.
- std::size_t binary_search (const T &data, std::size_t left, std::size_t right)

  Busca binária na lista, dado um range.

2.1.1 Descrição detalhada

Lista encadeada genérica.

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- linked_list.h

2.2 Referência à classe parser::Parser

Percorre o arquivo identificando prefixos e indexando as palavras.

`#include <parser.h>`

Membros públicos

- Parser (std::string path)

  Construtor com o nome do arquivo.
- void identify ()

  Identifica as palavras e adiciona na árvore Trie.
- void verify (std::string words)

  Verifica se as palavras são prefixos.

2.2.1 Descrição detalhada

Percorre o arquivo identificando prefixos e indexando as palavras.

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- parser.h

2.3 Referência à estrutura structures::TrieNode

Nó da trie, armazena letra, os seus filhos, posição e comprimento se for palavra.

`#include <trie_node.h>`

Membros públicos

- TrieNode (char letter)
  Construtor.
- ∼TrieNode ()
  Destrutor.
- bool operator== (const TrieNode &node)
  Operador de igualdade, compara as letras de 2 nós.
- bool operator> (const TrieNode &node)
  Operador maior, compara as letras de 2 nós.
- bool operator!= (const TrieNode &node)
  Operador diferente, compara as letras de 2 nós.

Atributos Públicos

- char letter
  Letra do nó
- structures::TrieNodeLinkedList< structures::TrieNode * > * children
  Filhos do nó
- unsigned long position
  Posição de ínico da palavra, 0 se não for final de uma palavra.
- unsigned long length
  Tamanho da palavra, 0 se não for final de uma palavra.

2.3.1 Descrição detalhada

Nó da trie, armazena letra, os seus filhos, posição e comprimento se for palavra.

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- trie_node.h

2.4 Referência à classe structures::TrieNodeLinkedList

Lista encadeada especializada em ponteiros.

`#include <trie_node_linked_list.h>`

Membros públicos

- TrieNodeLinkedList ()

  Construtor padrão, inicializa uma lista vazia.
- ~TrieNodeLinkedList ()

  Destrutor.
- void clear ()

  Deleta todos os elementos.
- void push_back (const T &data)

  Insere no final.
- void push_front (const T &data)

  Insere no inicio.
- void insert (const T &data, std::size_t index)

  Insere no índice especificado.
- void insert_sorted (const T &data)

  Insere em ordem.
- std::size_t insert_sorted_unique (const T &data)

  Insere em ordem se o elemento não estiver na lista, considerando uma lista ordenada.
- T & at (std::size_t index)

  Acessa o elemento na posição, lançando exceções se necessário.
- T & operator[ ] (std::size_t index)

  Acessa o elemento na posição.
- void pop (std::size_t index)

  Retirar da posição.
- void pop_back ()

  Retirar do final.
- void pop_front ()

  Retirar do inicio.
- void remove (const T &data)

  Remover o elemento.
- bool empty () const

  Verifica se a lista está vazia.
- bool contains (const T &data)

  Verifica se o elemento está na lista.
- std::size_t find (const T &data) const

  A posição do elemento específico.
- std::size_t size () const

  Tamanho da lista.
- std::size_t binary_search (const T &data)

  Busca binária na lista, retorna size se o elemento não estiver presente.
- std::size_t binary_search (const T &data, std::size_t left, std::size_t right)

  Busca binária na lista, dado um range.

2.4.1 Descrição detalhada

Lista encadeada especializada em ponteiros.

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- trie_node_linked_list.h

Índice