DataStructures4Beamer

1.0

Generated by Doxygen 1.8.1.2

Sun Oct 6 2013 13:42:31

Contents

Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

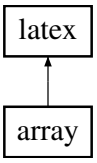
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 array Class Reference

Inheritance diagram for array:



Public Member Functions

- int delete_element ()

  Deletes the first element in the array.
- void delete_element_in_pos (int)

  Deletes the element in the position "pos" of the array.
- void insert_element (int)

  Inserts an element with value "elem" to the first position of the array.
- void insert_element_in_pos (int, int)

  Inserts an element with value "elem" to the position "pos" of the array.
- void fill_vector (int)

  Fills an empty array, assigns zero to each position.
- array (int n=10)

  Constructor initializes an array of size "n".
- int sum_vector ()

  Returns the sum of all elements in the array.
- int max ()

  Returns the element with the highest value in the array.
- int min ()

  Returns the element with the lowest value in the array.
- void invest_vector ()

  Reverses the positions of the elements in the array.
- void exchange_elements2 (int, int)

  Change the position of two elements.
- void print_vector ()

  Displays the array values right through console.

- int get_amount ()

  Returns current number of elements contained in the array.
- void clean_vector ()

  Deletes all elements in the array.
- int get_objet (int)

  Returns the element in the position "n" in the array.
- void exchange_elements (int, int)

  Change the position of two elements.
- int frequency (int)

  Calculate and return the frequency of element "elem".
- int mode ()

  Calculate and return the mode of the array.
- int arithmetic_mean ()

  Calculate and return the average of the values in the array.
- bool search_element (int)

  Confirms if exist the element "elem" in the array.
- void order ()

  Orders the array from lowest to highest.
- virtual ~array (void)

  Destructor, deletes the array.
- string **getCadena** ()

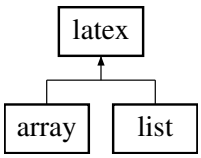
Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/array.h
- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/array.cpp

3.2 latex Class Reference

Inheritance diagram for latex:



Public Member Functions

- virtual string **getCadena** ()

Protected Attributes

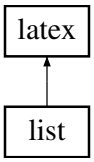
- string **cadena**

The documentation for this class was generated from the following files:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/latex.h
- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/latex.cpp

3.3 list Class Reference

Inheritance diagram for list:



Public Member Functions

- list ()

  Default constructor.
- ~list ()

  Destructor.
- bool search (int v)

  Confirms if the value "v" is in the list.
- void insert_at_beginning (int v)

  Inserts a new node with value "v" at beginning of the list.
- void insert_at_end (int v)

  Inserts a new node at end of the list.
- void insert_in_position (int v, int pos)

  Inserts a new node with value "v" in the position "pos" of the list.
- void delete_first_node ()

  Deletes the first node of the list.
- void delete_last_node ()

  Deletes the last node of the list.
- void delete_in_position (int pos)

  Deletes the node in the position "pos"of the list.
- void next_node ()

  Set the current node in the next node of the list.
- void go_to_first_node ()

  Set the current node in the first node of the list.
- void go_to_last_node ()

  Set the current node in the last node of the list.
- bool get_current_node ()

  Returns the current node, if this is not NULL.
- int current_value ()

  Returns the value of the current node.
- string **getCadena** ()
- void begin_tex (string)

  Writes the headers and required latex libraries and packages on the .tex file.
- void end_tex ()

  Writes the footers on the .tex file.
- string to_string (int v)

  Convert an int "v" to string.

Public Attributes

- int size

  Size of the list.

Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/list.h
- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/list.cpp

3.4 list_node Class Reference

Public Member Functions

- **list_node** (int v, list_node *)
- void **set_value** (int)
- int **get_value** ()
- void **set_next** (list_node *)
- list_node * **get_next** ()

The documentation for this class was generated from the following files:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/list_node.h
- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/list_node.cpp

3.5 node Struct Reference

Public Attributes

- int **key_value**
- node * **left**
- node * **right**

The documentation for this struct was generated from the following file:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/tree_node.h

3.6 tree Class Reference

Public Member Functions

- void **insert_root** (int key)
- void **insert1** (int key)
- void **insert2** (int key)
- node * **search** (int key)
- void **finish_tree** ()

The documentation for this class was generated from the following files:

- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/tree.h
- /home/mtorres/Dropbox/Universidad/Estructuras/Proyecto1/FoxHound/PROYECTO/tree.cpp