XML_processor_with_binary_images

Generated by Doxygen 1.8.15

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 structures Namespace Reference

Classes

- class LinkedStack

Chapter 5

Class Documentation

5.1 Coordinates Class Reference

Public Member Functions

- Coordinates ()
  Empty constructor method.
- Coordinates (const int i, const int j)
  Constructor method.
- int getI ()
  Coordinate I getter.
- int getJ ()
  Coordinate J getter.
- void setI (const int i)
  Coordinate I setter.
- void setJ (const int j)
  Coordinate J setter.
- const int operator[ ] (std::size_t index) const
  Overloads the [] operator.

5.1.1 Detailed Description

Coordinates class

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Coordinates() [1/2]

```
Coordinates::Coordinates ( )  [inline]
```

Empty constructor method.

5.1.2.2 Coordinates() [2/2]

```
Coordinates::Coordinates (
            const int i,
            const int j )  [inline]
```

Constructor method.

5.1.3 Member Function Documentation

5.1.3.1 getI()

```
int Coordinates::getI ( )  [inline]
```

Coordinate I getter.

5.1.3.2 getJ()

```
int Coordinates::getJ ( )  [inline]
```

Coordinate J getter.

5.1.3.3 operator[]()

```
const int Coordinates::operator[] (
            std::size_t index ) const  [inline]
```

Overloads the [] operator.

5.1.3.4 setI()

```
void Coordinates::setI (
            const int i )  [inline]
```

Coordinate I setter.

5.1.3.5 setJ()

```
void Coordinates::setJ (
            const int j )  [inline]
```

Coordinate J setter.

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

- XML_processor_with_binary_images.cpp

5.2 structures::LinkedStack< T > Class Template Reference

Public Member Functions

- LinkedStack ()
  Constructor method;.
- ~LinkedStack ()
  Destructor method;.
- void clear ()
  Wipes the list.
- void push (const T &data)
  Inserts an element in the stack.
- T pop ()
  Removes an element.
- T & top () const
  Returns the first element.
- bool empty () const
  Returns true if empty and false otherwise.
- std::size_t size () const
  Returns the size of the stack.

5.2.1 Detailed Description

template<typename T>
class structures::LinkedStack< T >

Linked Stack class

5.2.2 Constructor & Destructor Documentation

5.2.2.1 LinkedStack()

```
template<typename T >
structures::LinkedStack< T >::LinkedStack ( )
```

Constructor method;.

Constructor method.

5.2.2.2 ~LinkedStack()

```
template<typename T >
structures::LinkedStack< T >::~LinkedStack ( )
```

Destructor method;.

Destructor method.

5.2.3 Member Function Documentation

5.2.3.1 clear()

```
template<typename T >
void structures::LinkedStack< T >::clear ( )
```

Wipes the list.

Wipe the list.

5.2.3.2 empty()

```
template<typename T >
bool structures::LinkedStack< T >::empty ( ) const
```

Returns true if empty and false otherwise.

Returns

true if empty

5.2.3.3 pop()

```
template<typename T >
T structures::LinkedStack< T >::pop ( )
```

Removes an element.

Returns the first element.

Returns

the data inside the removed element

5.2.3.4 push()

```
template<typename T >
void structures::LinkedStack< T >::push (
            const T & data )
```

Inserts an element in the stack.

Parameters

data	data to be inserted

5.2.3.5 size()

```
template<typename T >
std::size_t structures::LinkedStack< T >::size ( ) const
```

Returns the size of the stack.

Returns

stack's current size

5.2.3.6 top()

```
template<typename T >
T & structures::LinkedStack< T >::top ( ) const
```

Returns the first element.

Removes an element.

Returns

the data of the node at the stack's top

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

- XML_processor_with_binary_images.cpp

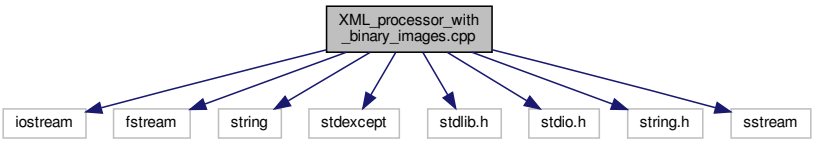
Chapter 6

File Documentation

6.1 XML_processor_with_binary_images.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <stdexcept>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sstream>
```

Include dependency graph for XML_processor_with_binary_images.cpp:



Classes

- class Coordinates
- class structures::LinkedStack< T >

Namespaces

- structures

Macros

- #define STRUCTURES_XML_PROCESSOR

  XML_processor_with_binary_images.cpp.

Functions

- string getFile (string filename)
  Gets a file by name.
- string getTag (string line, int pos)
  Gets a complete tag.
- size_t getImgTagCount (string xmlfilename)
  Counts how many "imgs" tags there are.
- string getImage (string xmlfilename, size_t index)
  Gets the complete image, containing tags and the binary data.
- string getName (string image)
  Gets the name of the image.
- size_t getHeight (string image)
  Gets the binary image height.
- size_t getWidth (string image)
  Gets the binary image width.
- string getData (string image)
  Gets the binary image data.
- bool doFirstPart (string xmlfilename, size_t lines)
  Checks if all the tags are correctly formatted.
- void print_array (string image)
  Gets the binary image data.
- size_t doSecondPart (string xmlfilename, string image)
  Counts how many blocks of 1's there are.
- int main ()

6.1.1 Macro Definition Documentation

6.1.1.1 STRUCTURES_XML_PROCESSOR

```
#define STRUCTURES_XML_PROCESSOR
```

XML_processor_with_binary_images.cpp.

Author

Alan Djon Lüdke & Matheus Schaly

Since

04/10/2018

Version

1.0

Copyright

2018

Warning

dataset06.xml with error

6.1.2 Function Documentation

6.1.2.1 doFirstPart()

```
bool doFirstPart (
            string xmlfilename,
            size_t lines )
```

Checks if all the tags are correctly formatted.

Returns false and prints error if: 1 - A tag is openned but not closed 2 - A tag is openned but the last tag in the stack is not the openned tag 3 - The file ends and there is still some tag in the stack

Parameters

xmlfilename	the name of the file
lines	the line where the the procedure should start

Returns

true if there was an error, false otherwise

6.1.2.2 doSecondPart()

```
size_t doSecondPart (
            string xmlfilename,
            string image )
```

Counts how many blocks of 1's there are.

Goes through the list, element by element, from left to right until a neighbour of such element is a 1. When the number 1 is found, it checks all the other neighbours and add them to the stack, giving them a label. It does it continously, until all the image is visited.

Parameters

xmlfilename	the name of the file
image	the image containing the tags and the binary data

Returns

the number of clusters of numbers 1

6.1.2.3 getData()

```
string getData (
            string image )
```

Gets the binary image data.

Parameters

image	the image where the data will be extracted

Returns

the image's binary data

6.1.2.4 getFile()

```
string getFile (
            string filename )
```

Gets a file by name.

Parameters

filename	the filename to return

Returns

filename as a string

6.1.2.5 getHeight()

```
size_t getHeight (
            string image )
```

Gets the binary image height.

Parameters

image	the image to be measured

Returns

the image's height

6.1.2.6 getImage()

```
string getImage (
            string xmlfilename,
            size_t index )
```

Gets the complete image, containing tags and the binary data.

Parameters

xmlfilename	the file's name
index	the image's index

Returns

the image as a string

6.1.2.7 getImgTagCount()

```
size_t getImgTagCount (
            string xmlfilename )
```

Counts how many "imgs" tags there are.

Parameters

xmlfilename	the document's filename

Returns

quantity of tags in the file

6.1.2.8 getName()

```
string getName (
            string image )
```

Gets the name of the image.

Parameters

image	the image that contains the name

Returns

the image's name

6.1.2.9 getTag()

```
string getTag (
            string line,
            int pos )
```

Gets a complete tag.

Parameters

line	the tag's line
pos	the tag's index

Returns

the tag found

6.1.2.10 getWidth()

```
size_t getWidth (
            string image )
```

Gets the binary image width.

Parameters

image	the image to be measured

Returns

the image's width

6.1.2.11 main()

```
int main ( )
```

6.1.2.12 print_array()

```
void print_array (
             string image )
```

Gets the binary image data.

Parameters

image	the image where the data will be extracted

Index