

XML_processor_with_binary_images

Generated by Doxygen 1.8.15

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 structures Namespace Reference	7
5 Class Documentation	9
5.1 Coordinates Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Constructor & Destructor Documentation	9
5.1.2.1 Coordinates() [1/2]	9
5.1.2.2 Coordinates() [2/2]	10
5.1.3 Member Function Documentation	10
5.1.3.1 getI()	10
5.1.3.2 getJ()	10
5.1.3.3 operator[]()	10
5.1.3.4 setI()	10
5.1.3.5 setJ()	11
5.2 structures::LinkedList< T > Class Template Reference	11
5.2.1 Detailed Description	11
5.2.2 Constructor & Destructor Documentation	11
5.2.2.1 LinkedList()	12
5.2.2.2 ~LinkedList()	12
5.2.3 Member Function Documentation	12
5.2.3.1 clear()	12
5.2.3.2 empty()	12
5.2.3.3 pop()	13
5.2.3.4 push()	13
5.2.3.5 size()	13
5.2.3.6 top()	13
6 File Documentation	15
6.1 XML_processor_with_binary_images.cpp File Reference	15
6.1.1 Macro Definition Documentation	16
6.1.1.1 STRUCTURES_XML_PROCESSOR	16
6.1.2 Function Documentation	17
6.1.2.1 doFirstPart()	17
6.1.2.2 doSecondPart()	17

6.1.2.3 <code>getData()</code>	18
6.1.2.4 <code>getFile()</code>	18
6.1.2.5 <code>getHeight()</code>	18
6.1.2.6 <code>getImage()</code>	19
6.1.2.7 <code>getImgTagCount()</code>	19
6.1.2.8 <code>getName()</code>	19
6.1.2.9 <code>getTag()</code>	20
6.1.2.10 <code>getWidth()</code>	20
6.1.2.11 <code>main()</code>	20
6.1.2.12 <code>print_array()</code>	21

Index	23
--------------	-----------

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

structures	7
--------------------------------------	---

Chapter 2

Class Index

2.1 Class List

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

Coordinates	9
structures::LinkedStack< T >	11

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

XML_processor_with_binary_images.cpp	15
--	----

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::LinkedList< T > Class Template Reference

Public Member Functions

- [LinkedList](#) ()
Constructor method;.
- [~LinkedList](#) ()
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::LinkedList< 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::LinkedList< 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::LinkedList< T >::push (
    const T & data )
```

Inserts an element in the stack.

Parameters

<i>data</i>	data to be inserted
-------------	---------------------

5.2.3.5 size()

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

Returns the size of the stack.

Returns

stack's current size

5.2.3.6 top()

```
template<typename T >
T & structures::LinkedList< 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](#)

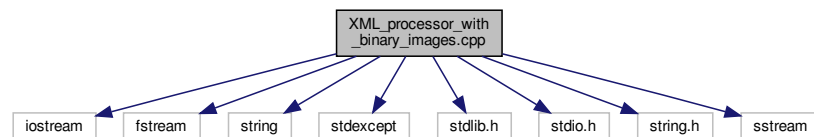
Chapter 6

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 opened but not closed 2 - A tag is opened but the last tag in the stack is not the opened tag 3 - The file ends and there is still some tag in the stack

Parameters

<i>xmlfilename</i>	the name of the file
<i>lines</i>	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 continuously, until all the image is visited.

Parameters

<i>xmlfilename</i>	the name of the file
<i>image</i>	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

<i>image</i>	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

<i>filename</i>	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

<i>image</i>	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

<i>xmlfilename</i>	the file's name
<i>index</i>	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

<i>xmlfilename</i>	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

<i>image</i>	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

<i>line</i>	the tag's line
<i>pos</i>	the tag's index

Returns

the tag found

6.1.2.10 getWidth()

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

Gets the binary image width.

Parameters

<i>image</i>	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

<i>image</i>	the image where the data will be extracted
--------------	--

Index

- ~LinkedList
 - structures::LinkedList, [12](#)
- clear
 - structures::LinkedList, [12](#)
- Coordinates, [9](#)
 - Coordinates, [9](#)
 - getI, [10](#)
 - getJ, [10](#)
 - operator[], [10](#)
 - setI, [10](#)
 - setJ, [10](#)
- doFirstPart
 - XML_processor_with_binary_images.cpp, [17](#)
- doSecondPart
 - XML_processor_with_binary_images.cpp, [17](#)
- empty
 - structures::LinkedList, [12](#)
- getData
 - XML_processor_with_binary_images.cpp, [17](#)
- getFile
 - XML_processor_with_binary_images.cpp, [18](#)
- getHeight
 - XML_processor_with_binary_images.cpp, [18](#)
- getImage
 - XML_processor_with_binary_images.cpp, [19](#)
- getImgTagCount
 - XML_processor_with_binary_images.cpp, [19](#)
- getName
 - XML_processor_with_binary_images.cpp, [19](#)
- getTag
 - XML_processor_with_binary_images.cpp, [20](#)
- getWidth
 - XML_processor_with_binary_images.cpp, [20](#)
- getI
 - Coordinates, [10](#)
- getJ
 - Coordinates, [10](#)
- LinkedList
 - structures::LinkedList, [11](#)
- main
 - XML_processor_with_binary_images.cpp, [20](#)
- operator[]
 - Coordinates, [10](#)
- pop
 - structures::LinkedList, [12](#)
- print_array
 - XML_processor_with_binary_images.cpp, [20](#)
- push
 - structures::LinkedList, [13](#)
- STRUCTURES_XML_PROCESSOR
 - XML_processor_with_binary_images.cpp, [16](#)
- setI
 - Coordinates, [10](#)
- setJ
 - Coordinates, [10](#)
- size
 - structures::LinkedList, [13](#)
- structures, [7](#)
- structures::LinkedList
 - ~LinkedList, [12](#)
 - clear, [12](#)
 - empty, [12](#)
 - LinkedList, [11](#)
 - pop, [12](#)
 - push, [13](#)
 - size, [13](#)
 - top, [13](#)
- structures::LinkedList< T >, [11](#)
- top
 - structures::LinkedList, [13](#)
- XML_processor_with_binary_images.cpp, [15](#)
 - doFirstPart, [17](#)
 - doSecondPart, [17](#)
 - getData, [17](#)
 - getFile, [18](#)
 - getHeight, [18](#)
 - getImage, [19](#)
 - getImgTagCount, [19](#)
 - getName, [19](#)
 - getTag, [20](#)
 - getWidth, [20](#)
 - main, [20](#)
 - print_array, [20](#)
 - STRUCTURES_XML_PROCESSOR, [16](#)