Projet C++ : Cryptage et decryptage

0.1.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 ArbreB Class Reference

```
#include <ArbreB.hpp>
```

Public Member Functions

- ArbreB ()
- ArbreB (const char &data, const double &freq)
- ArbreB (const Sommet &node)
- ArbreB (const ArbreB &other)
- ∼ArbreB ()
- ArbreB & operator= (const ArbreB &other)
- Sommet & get_m_Root ()
- void insert (Sommet &new_node)
- void insert (const char &data, const double &freq)
- bool search (const char &data)
- bool bst_search (const char &data, std::string &path)
- ArbreB & remove (const char &data)
- ArbreB operator+ (const ArbreB &other)
- std::tuple< ArbreB, ArbreB > decompose ()

Public Attributes

- Sommet ∗ m_Root

Friends

- std::ostream & operator<< (std::ostream &stream, ArbreB &tree)

2.1.1 Detailed Description

The class ArbreB represents a binary tree.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020/11/17

2.1.2 Constructor & Destructor Documentation

2.1.2.1 ArbreB() [1/4]

```
ArbreB::ArbreB ( )
```

Creates a default object ArbreB. Sets `m_Root` at `nullptr`.

2.1.2.2 ArbreB() [2/4]

```
ArbreB::ArbreB (
            const char & data,
            const double & freq )
```

Creates an object ArbreB from the specified parameters.

Parameters

data	The character to store in the root of the ArbreB.
freq	The character's frequency to store in the root of the ArbreB.

2.1.2.3 ArbreB() [3/4]

```
ArbreB::ArbreB (
            const Sommet & node )
```

Creates an object ArbreB from the specified `Sommet`.

Parameters

node	The `Sommet` to initialize `m_Root` from.

2.1.2.4 ArbreB() [4/4]

```
ArbreB::ArbreB (
             const ArbreB & other )
```

Creates a copy of the specified `ArbreB`.

Parameters

other	The `ArbreB` to copy.

2.1.2.5 ∼ArbreB()

```
ArbreB::∼ArbreB ( )
```

Frees the memory of an `ArbreB`.

2.1.3 Member Function Documentation

2.1.3.1 bst_search()

```
bool ArbreB::bst_search (
             const char & data,
             std::string & path )
```

Searches for the specified character in the object `ArbreB`. Internaly calls private method `bst_search()`.

Parameters

data	The character to search for.
path	A string that stores the path to the character. '0's mean the path takes a left branch, '1's means it takes a right branch.

Returns

True if the character was found, False otherwise.

2.1.3.2 decompose()

```
std::tuple<ArbreB, ArbreB> ArbreB::decompose ( )
```

Decomposes one object ArbreB into two.

Returns

An std::tuple that holds two ArbreB. The first one is the left branch of the original tree. The second one is the right branch of the original tree.

2.1.3.3 get_m_Root()

```
Sommet& ArbreB::get_m_Root ( )
```

Gets the root of the object ArbreB.

Returns

A reference to the root of the ArbreB.

2.1.3.4 insert() [1/2]

```
void ArbreB::insert (
             Sommet & new_node )
```

Inserts a new node in the object ArbreB. Internaly calls private method insert().

Parameters

new_node	The Sommet to insert in the ArbreB.

2.1.3.5 insert() [2/2]

```
void ArbreB::insert (
            const char & data,
            const double & freq )
```

Inserts a new node in the object ArbreB. Internaly calls private method `insert()`.

Parameters

data	The character to insert in the ArbreB.
freq	The frequency of the character to insert.

2.1.3.6 operator+()

```
ArbreB ArbreB::operator+ (
            const ArbreB & other )
```

Overloads the operator + to redefine its behavior. Fuses two ArbreB together to creates a new one. Sets `m_Root->m_Left` as `this`, `m_Root->m_Right` as `other`. Sets `m_Root->m_Data` as `\0`, `m_↩Root->m_Freq` as `this.m_Root->m_Freq + other.m_Root->m_Freq`.

Parameters

other	The ArbreB to fuse.

Returns

The fusion of the two ArbreBs.

2.1.3.7 operator=()

```
ArbreB& ArbreB::operator= (
            const ArbreB & other )
```

Overloads the operator = to redefine its behavior.

Parameters

other	The ArbreB to assign the values from.

Returns

A reference to a copy of `other`.

2.1.3.8 remove()

```
ArbreB& ArbreB::remove (
            const char & data )
```

Removes a `Sommet` from the object `ArbreB`. Internaly calls private method `remove()`.

Parameters

data	The character to delete.

Returns

A reference of the `ArbreB` with the removed `Sommet`.

2.1.3.9 search()

```
bool ArbreB::search (
            const char & data )
```

Searches for the specified character in the object `ArbreB`. Internaly calls private method `search()`.

Parameters

data	The character to search for.

Returns

`True` if the character was found, `False` otherwise.

2.1.4 Friends And Related Function Documentation

2.1.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & stream,
            ArbreB & tree )  [friend]
```

Overloads the operator >> and redefines its behavior.

Parameters

stream	The output stream.
tree	The `ArbreB`.

Returns

The output stream to print to `std::cout`.

2.1.5 Member Data Documentation

2.1.5.1 m_Root

```
Sommet* ArbreB::m_Root
```

Represents a pointer to the root of the object `ArbreB`.

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

- src/lib/ArbreB.hpp

2.2 PartOneTests Class Reference

```
#include <PartOneTests.hpp>
```

Public Member Functions

- PartOneTests ()
- bool should_create_default_Sommet ()
- bool should_create_parameterized_Sommet ()
- bool should_create_copy_Sommet ()
- bool should_set_Sommet_values ()
- bool should_assign_Sommet ()
- bool should_not_link_Sommet_copies ()
- bool should_create_default_ArbreB ()
- bool should_create_parameterized_ArbreB ()
- bool should_create_ArbreB_from_Sommet ()
- bool should_create_copy_ArbreB ()
- bool should_assign_ArbreB ()
- bool should_not_link_ArbreB_copies ()
- bool should_insert_Sommet_into_ArbreB ()
- bool should_update_freq_if_char_already_in_ArbreB ()
- bool should_find_character_c ()
- bool should_not_find_character_z ()
- bool should_remove_leaf ()
- bool should_remove_Sommet_with_one_child ()
- bool should_remove_Sommet_with_two_children ()
- bool should_fuse_two_ArbreB ()
- bool should_decompose_one_ArbreB_into_two ()
- bool should_find_character_y_with_bfs ()
- bool should_not_find_character_s_with_bfs ()

Public Attributes

- unsigned int tests_run
- unsigned int tests_failed

Static Public Attributes

- static unsigned int total_tests

Friends

- class **Sommet**
- class **ArbreB**

2.2.1 Detailed Description

The class `PartOneTests` implements tests to assert that the functions in classes `Sommet` and `ArbreB` have the expected behavior.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020/11/17

2.2.2 Constructor & Destructor Documentation

2.2.2.1 PartOneTests()

```
PartOneTests::PartOneTests ( )
```

Creates an object `PartOneTests`.

2.2.3 Member Function Documentation

2.2.3.1 should_assign_ArbreB()

```
bool PartOneTests::should_assign_ArbreB ( )
```

Asserts that the overload of `operator=` for `ArbreB` assign the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.2 should_assign_Sommet()

```
bool PartOneTests::should_assign_Sommet ( )
```

Asserts that the overload of `operator=` for `Sommet` assign the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.3 should_create_ArbreB_from_Sommet()

```
bool PartOneTests::should_create_ArbreB_from_Sommet ( )
```

Asserts that the constructor of `ArbreB` from a `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.4 should_create_copy_ArbreB()

```
bool PartOneTests::should_create_copy_ArbreB ( )
```

Asserts that the copy constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.5 should_create_copy_Sommet()

```
bool PartOneTests::should_create_copy_Sommet ( )
```

Asserts that the copy constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.6 should_create_default_ArbreB()

```
bool PartOneTests::should_create_default_ArbreB ( )
```

Asserts that the default constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.7 should_create_default_Sommet()

```
bool PartOneTests::should_create_default_Sommet ( )
```

Asserts that the default constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.8 should_create_parameterized_ArbreB()

```
bool PartOneTests::should_create_parameterized_ArbreB ( )
```

Asserts that the parameterized constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.9 should_create_parameterized_Sommet()

```
bool PartOneTests::should_create_parameterized_Sommet ( )
```

Asserts that the parameterized constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.10 should_decompose_one_ArbreB_into_two()

```
bool PartOneTests::should_decompose_one_ArbreB_into_two ( )
```

Asserts that decomposing an `ArbreB` returns a tuple holding two `ArbreB`s with the expected values.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.11 should_find_character_c()

```
bool PartOneTests::should_find_character_c ( )
```

Asserts that the character `c` is found in an `ArbreB` that contains it.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.12 should_find_character_y_with_bfs()

```
bool PartOneTests::should_find_character_y_with_bfs ( )
```

Asserts that using the method `search()` (BFS algorithm), the character `c` is found in an `ArbreB` that contains it.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.13 should_fuse_two_ArbreB()

```
bool PartOneTests::should_fuse_two_ArbreB ( )
```

Asserts that the overload of `operator+` for `ArbreB` fuses two `ArbreB`s into one and has the expected values at its root.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.14 should_insert_Sommet_into_ArbreB()

```
bool PartOneTests::should_insert_Sommet_into_ArbreB ( )
```

Asserts that a `Sommet` is correctly inserted into an `ArbreB`. This method tests for both `insert(const Sommet&)` and `insert(const char&, const double&)`.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.15 should_not_find_character_s_with_bfs()

```
bool PartOneTests::should_not_find_character_s_with_bfs ( )
```

Asserts that using the method `search()` (BFS algorithm), the character `s` is not found in an `ArbreB` that does not contain it.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.16 should_not_find_character_z()

```
bool PartOneTests::should_not_find_character_z ( )
```

Asserts that the character `z` is not found in an `ArbreB` that does not contain it.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.17 should_not_link_ArbreB_copies()

```
bool PartOneTests::should_not_link_ArbreB_copies ( )
```

Asserts that the copy of an object `ArbreB` is not linked with the original.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.18 should_not_link_Sommet_copies()

```
bool PartOneTests::should_not_link_Sommet_copies ( )
```

Asserts that the copy of an object `Sommet` is not linked with the original.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.19 should_remove_leaf()

```
bool PartOneTests::should_remove_leaf ( )
```

Asserts that removing a `Sommet` that is a leaf deletes it correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.20 should_remove_Sommet_with_one_child()

```
bool PartOneTests::should_remove_Sommet_with_one_child ( )
```

Asserts that removing a `Sommet` that has only one child (left or right) deletes it correctly and replaces it with its child.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.21 should_remove_Sommet_with_two_children()

```
bool PartOneTests::should_remove_Sommet_with_two_children ( )
```

Asserts that removing a `Sommet` that has two children deletes it correctly and replaces it with its inorder succesor.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.22 should_set_Sommet_values()

```
bool PartOneTests::should_set_Sommet_values ( )
```

Asserts that the setters for `Sommet` set the values correctly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.3.23 should_update_freq_if_char_already_in_ArbreB()

```
bool PartOneTests::should_update_freq_if_char_already_in_ArbreB ( )
```

Asserts that inserting a `Sommet` that already is in the `ArbreB` updates the `m_Freq` field of that `Sommet` accordingly.

Returns

`True` if the test passed, `false` if it failed.‘

2.2.4 Member Data Documentation

2.2.4.1 tests_failed

```
unsigned int PartOneTests::tests_failed
```

Represents the number of tests failed.

2.2.4.2 tests_run

```
unsigned int PartOneTests::tests_run
```

Represents the number of tests ran.

2.2.4.3 total_tests

```
unsigned int PartOneTests::total_tests [static]
```

Represents the total number of tests.

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

- src/test/PartOneTests.hpp

2.3 Sommet Class Reference

```
#include <Sommet.hpp>
```

Collaboration diagram for Sommet:

Public Member Functions

- Sommet ()
- Sommet (const char &data, const double &freq)
- Sommet (const Sommet &other)
- ∼Sommet ()
- Sommet & operator= (const Sommet &other)
- char & get_m_Data ()
- double & get_m_Freq ()
- Sommet & get_m_Left ()
- Sommet & get_m_Right ()
- void set_m_Data (const char &data)
- void set_m_Freq (const double &freq)

Public Attributes

- Sommet ∗ m_Left
- Sommet ∗ m_Right

Friends

- class **ArbreB**

2.3.1 Detailed Description

The class Sommet represents a node of the class ArbreB.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020/11/17

2.3.2 Constructor & Destructor Documentation

2.3.2.1 Sommet() [1/3]

```
Sommet::Sommet ( )
```

Creates a default object Sommet. Sets `m_Data` to `\0`, `m_Freq` to `0`, `m_Left` and `m_Right` to `nullptr`.

2.3.2.2 Sommet() [2/3]

```
Sommet::Sommet (
            const char & data,
            const double & freq )
```

Creates an object Sommet with the specified parameters.

Parameters

data	The character to store.
freq	The frequency of the stored character.

2.3.2.3 Sommet() [3/3]

```
Sommet::Sommet (
            const Sommet & other )
```

Creates a copy of the specified object `Sommet`.

Parameters

other	The Sommet to copy.

2.3.2.4 ∼Sommet()

```
Sommet::∼Sommet ( )
```

Frees the memory for of an object Sommet.

2.3.3 Member Function Documentation

2.3.3.1 get_m_Data()

```
char& Sommet::get_m_Data ( )
```

Gets the character.

Returns

A reference of the character.

2.3.3.2 get_m_Freq()

```
double& Sommet::get_m_Freq ( )
```

Gets the character's frequency.

Returns

A reference of the character's frequency.

2.3.3.3 get_m_Left()

```
Sommet& Sommet::get_m_Left ( )
```

Gets the left child.

Returns

A reference of the left child.

2.3.3.4 get_m_Right()

```
Sommet& Sommet::get_m_Right ( )
```

Gets the right child.

Returns

A reference of the right child.

2.3.3.5 operator=()

```
Sommet& Sommet::operator= (
            const Sommet & other )
```

Redefines the behavior of the operator =.

Parameters

other	The object Sommet to assign the values from.

Returns

A reference to a copy of `other`.

2.3.3.6 set_m_Data()

```
void Sommet::set_m_Data (
            const char & data )
```

Sets the value of the character.

Parameters

data	The character to assign to `m_Data`.

2.3.3.7 set_m_Freq()

```
void Sommet::set_m_Freq (
            const double & freq )
```

Sets the value of the frequency.

Parameters

freq	The value to assign to m_Freq.

2.3.4 Member Data Documentation

2.3.4.1 m_Left

```
Sommet* Sommet::m_Left
```

Represents the left child of an object Sommet.

2.3.4.2 m_Right

```
Sommet* Sommet::m_Right
```

Represents the right child of an object Sommet.

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

- src/lib/Sommet.hpp

Index