LANGUAGE0

0

Generated by Doxygen 1.9.1

1 Class Index 1

1.1 Class List . . . 1

2 File Index 3

2.1 File List . . . 3

3 Class Documentation 5

3.1 Bigram Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Constructor & Destructor Documentation . . . 5

3.1.2.1 Bigram() [1/2] . . . 6

3.1.2.2 Bigram() [2/2] . . . 7

3.1.3 Member Function Documentation . . . 7

3.1.3.1 at() [1/2] . . . 7

3.1.3.2 at() [2/2] . . . 8

3.1.3.3 getText() . . . 8

3.1.3.4 toString() . . . 9

4 File Documentation 11

4.1 include/Bigram.h File Reference . . . 11

4.1.1 Detailed Description . . . 11

4.1.2 Function Documentation . . . 11

4.1.2.1 isValidCharacter() . . . 11

4.1.2.2 toUpper() . . . 12

4.2 src/Bigram.cpp File Reference . . . 12

4.2.1 Detailed Description . . . 13

4.2.2 Function Documentation . . . 13

4.2.2.1 isValidCharacter() . . . 13

4.2.2.2 toUpper() . . . 13

4.3 src/main.cpp File Reference . . . 14

4.3.1 Detailed Description . . . 14

4.3.2 Function Documentation . . . 14

4.3.2.1 toLower() . . . 14

Index 17

Chapter 1

Class Index

1.1 Class List

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

Bigram
It represents a pair of characters. It is used to store pairs of consecutive characters from a text. It uses a string to store the pair of characters 5

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

include/Bigram.h 11
src/Bigram.cpp 12
src/main.cpp 14

Chapter 3

Class Documentation

3.1 Bigram Class Reference

It represents a pair of characters. It is used to store pairs of consecutive characters from a text. It uses a string to store the pair of characters.

`#include <Bigram.h>`

Public Member Functions

- Bigram (const std::string &text="__")

  It builds a Bigram object with `text` as the text of the bigram. If the string `text` contains a number of characters other than two, then the text of the bigram will be initialized with "__".
- Bigram (char first, char second)

  It builds a Bigram object using the two characters passed as parameters of this constructor as the text of the bigram.
- std::string getText () const

  Obtains a copy of the text of this bigram as a string object.
- std::string toString () const

  Obtains a copy of the text of this bigram as a string object.
- const char & at (int index) const

  Gets a const reference to the character at the given position.
- char & at (int index)

  Gets a reference to the character at the given position.

3.1.1 Detailed Description

It represents a pair of characters. It is used to store pairs of consecutive characters from a text. It uses a string to store the pair of characters.

Definition at line 20 of file Bigram.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Bigram() [1/2]

```
Bigram::Bigram (
            const std::string & text = "__" )
```

It builds a Bigram object with `text` as the text of the bigram. If the string `text` contains a number of characters other than two, then the text of the bigram will be initialized with "__".

Parameters

text	the text for the bigram. It should be a string with just two characters.

3.1.2.2 Bigram() [2/2]

```
Bigram::Bigram (
            char first,
            char second )
```

It builds a Bigram object using the two characters passed as parameters of this constructor as the text of the bigram.

Parameters

first	the first character for the bigram
second	the second character for the bigram

Definition at line 24 of file Bigram.cpp.

```
24                                      {
25     _text += first;
26     _text += second;
27 }
```

3.1.3 Member Function Documentation

3.1.3.1 at() [1/2]

```
char & Bigram::at (
            int index )
```

Gets a reference to the character at the given position.

Parameters

index	the position to consider

Exceptions

std::out_of_range	Throws a std::out_of_range exception if the index is not equals to 0 or 1

Returns

A reference to the character at the given position

Definition at line 53 of file Bigram.cpp.

```
53                          {
54     if (index!=0 && index!=1){
55         string err;
56         err += "char& Bigram::at(int index)";
57         err += "\n";
58         err += "Invalid position " + to_string(index);
59 
60         throw std::out_of_range (err);
61     }
62 
63     return _text.at(index);
64 }
```

3.1.3.2 at() [2/2]

```
const char & Bigram::at (
            int index ) const
```

Gets a const reference to the character at the given position.

Parameters

index	the position to consider

Exceptions

std::out_of_range	Throws a std::out_of_range exception if the index is not equals to 0 or 1

Returns

A const reference to the character at the given position

Definition at line 40 of file Bigram.cpp.

```
40                                      {
41     if (index!=0 && index!=1){
42         string err;
43         err += "const char& Bigram::at(int index) const";
44         err += "\n";
45         err += "Invalid position " + to_string(index);
46 
47         throw std::out_of_range (err);
48     }
49 
50     return _text.at(index);
51 }
```

3.1.3.3 getText()

```
string Bigram::getText ( ) const
```

Obtains a copy of the text of this bigram as a string object.

Returns

The text of this bigram as a string object

Definition at line 30 of file Bigram.cpp.

```
30                              {
31     return _text;
32 }
```

3.1.3.4 toString()

```
string Bigram::toString ( ) const
```

Obtains a copy of the text of this bigram as a string object.

Returns

The text of this bigram as a string object

Definition at line 35 of file Bigram.cpp.

```
35                              {
36     return _text;
37 }
```

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

- include/Bigram.h
- src/Bigram.cpp

Chapter 4

File Documentation

4.1 include/Bigram.h File Reference

```
#include <iostream>
#include <string>
```

Classes

- class Bigram

  It represents a pair of characters. It is used to store pairs of consecutive characters from a text. It uses a string to store the pair of characters.

Functions

- bool isValidCharacter (char character, const std::string &validCharacters)
- void toUpper (Bigram &bigram)

4.1.1 Detailed Description

Author

arturoolvrs

Date

2 de marzo de 2023, 10:52

4.1.2 Function Documentation

4.1.2.1 isValidCharacter()

```
bool isValidCharacter (
            char character,
            const std::string & validCharacters )
```

Checks if the given character is contained in `validCharacters`. That is, if the given character can be consireded as part of a word.

Parameters

character	The character to check
validCharacters	The set of characters that we consider as possible characters in a word. Any other character is considered as a separator.

Returns

true if the given character is contained in `validCharacters`; false otherwise

Definition at line 66 of file Bigram.cpp.

```
66                                                                          {
67
68     return validCharacters.find(character) != std::string::npos;
69 }
```

4.1.2.2 toUpper()

```
void toUpper (
             Bigram & bigram )
```

Converts lowercase letters in the given bigram to uppercase

Parameters

bigram	

Definition at line 72 of file Bigram.cpp.

```
72                                {
73
74     for (int i=0; i<2; i++)
75         bigram.at(i) = toupper(bigram.at(i));
76
77 }
```

4.2 src/Bigram.cpp File Reference

```
#include <iostream>
#include "Bigram.h"
```

Functions

- bool isValidCharacter (char character, const std::string &validCharacters)
- void toUpper (Bigram &bigram)

4.2.1 Detailed Description

Author

arturoolvrs

Date

2 de marzo de 2023, 10:52

4.2.2 Function Documentation

4.2.2.1 isValidCharacter()

```
bool isValidCharacter (
            char character,
            const std::string & validCharacters )
```

Checks if the given character is contained in `validCharacters`. That is, if the given character can be consireded as part of a word.

Parameters

character	The character to check
validCharacters	The set of characters that we consider as possible characters in a word. Any other character is considered as a separator.

Returns

true if the given character is contained in `validCharacters`; false otherwise

Definition at line 66 of file Bigram.cpp.

```
66                                                                  {
67
68     return validCharacters.find(character) != std::string::npos;
69 }
```

4.2.2.2 toUpper()

```
void toUpper (
            Bigram & bigram )
```

Converts lowercase letters in the given bigram to uppercase

Parameters

bigram	

Definition at line 72 of file Bigram.cpp.

```
72                                {
73
74     for (int i=0; i<2; i++)
75         bigram.at(i) = toupper(bigram.at(i));
76
77 }
```

4.3 src/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "Bigram.h"
```

Functions

- void toLower (string &cad)

  It converts a string to lowercase.

4.3.1 Detailed Description

Author

arturoolvrs

Date

2 de marzo de 2023, 10:52

4.3.2 Function Documentation

4.3.2.1 toLower()

```
void toLower (
             string & cad )
```

It converts a string to lowercase.

This program reads a text (without spaces) with a undefined number of characters and a text with two characters (bigram). It finds the bigrams contained in the first text, storing them in an array of Bigram. After that, the bigrams of the array are shown in the standard output. Then it converts to uppercase the bigrams in the array that are equals to the bigram of the second text. Finally the bigrams of the array are shown again in the standard output.

Parameters

cad	string to convert to lowercase

Definition at line 29 of file main.cpp.

```
29                              {
30
31     for (int i=0; i<cad.length(); i++)
32         cad.at(i)=tolower(cad.at(i));
33 }
```

Index

at
    Bigram, 7, 8

Bigram, 5
    at, 7, 8
    Bigram, 5, 7
    getText, 8
    toString, 8
Bigram.cpp
    isValidCharacter, 13
    toUpper, 13
Bigram.h
    isValidCharacter, 11
    toUpper, 12

getText
    Bigram, 8

include/Bigram.h, 11
isValidCharacter
    Bigram.cpp, 13
    Bigram.h, 11

main.cpp
    toLower, 14

src/Bigram.cpp, 12
src/main.cpp, 14

toLower
    main.cpp, 14
toString
    Bigram, 8
toUpper
    Bigram.cpp, 13
    Bigram.h, 12