

ShifrGronsfield

1.3

Создано системой Doxygen 1.8.13

Оглавление

1	Алфавитный указатель классов	1
1.1	Классы	1
2	Список файлов	3
2.1	Файлы	3
3	Классы	5
3.1	Класс <code>modAlphaCipher</code>	5
3.1.1	Конструктор(ы)	6
3.1.1.1	<code>modAlphaCipher()</code> [1/2]	6
3.1.1.2	<code>modAlphaCipher()</code> [2/2]	6
3.1.2	Методы	6
3.1.2.1	<code>decrypt()</code>	6
3.1.2.2	<code>encrypt()</code>	7
3.1.2.3	<code>getValidCipherText()</code>	7
3.1.2.4	<code>getValidKey()</code>	8
3.1.2.5	<code>getValidOpenText()</code>	8
4	Файлы	11
4.1	Файл <code>main.cpp</code>	11
4.1.1	Подробное описание	12
4.1.2	Функции	12
4.1.2.1	<code>check()</code>	12
4.1.2.2	<code>main()</code>	13
4.2	Файл <code>modAlphaCipher.cpp</code>	13
4.2.1	Подробное описание	13
4.3	Файл <code>modAlphaCipher.h</code>	14
4.3.1	Подробное описание	15
	Алфавитный указатель	17

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

<code>modAlphaCipher</code>	5
-----------------------------	---

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

main.cpp	
Файл с проверками методов для модуля Gronsfeld(main.cpp)	11
modAlphaCipher.cpp	
Файл с описаниями методов для модуля Gronsfeld	13
modAlphaCipher.h	
Заголовочный файл для модуля Gronsfeld	14

Глава 3

Классы

3.1 Класс modAlphaCipher

Открытые члены

- `modAlphaCipher ()=delete`
Конструктор без параметра
- `modAlphaCipher (const std::wstring &skey)`
Конструктор с параметром
- `std::wstring encrypt (const std::wstring &open_text)`
Зашифровывание
- `std::wstring decrypt (const std::wstring &cipher_text)`
Расшифровывание

Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`
преобразование строка-вектор
- `std::wstring convert (const std::vector< int > &v)`
преобразование вектор-строка
- `std::wstring getValidKey (const std::wstring &s)`
Метод вызывающий исключение при не верном ключе
- `std::wstring getValidOpenText (const std::wstring &s)`
Метод вызывающий исключение при не верном введенном для шифрования тексте
- `std::wstring getValidCipherText (const std::wstring &s)`
Метод вызывающий исключение при не верном зашифрованном тексте

Закрытые данные

- `std::wstring numAlpha = L"ABCDEFGHIJKLMNOPQRSTUVWXYZ"`
Алфавит для шифра
- `std::map< char, int > alphaNum`
ассоциативный массив "номер по символу".
- `std::vector< int > key`
ключ

3.1.1 Конструктор(ы)

3.1.1.1 modAlphaCipher() [1/2]

modAlphaCipher::modAlphaCipher () [delete]

Конструктор без параметра

Возвращает

Удаление элемента класса

3.1.1.2 modAlphaCipher() [2/2]

modAlphaCipher::modAlphaCipher (
 const std::wstring & skey)

Конструктор с параметром

Аргументы

in	skey	Ключ для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	------	--

Возвращает

создание элемента класса с ключом

3.1.2 Методы

3.1.2.1 decrypt()

std::wstring modAlphaCipher::decrypt (
 const std::wstring & cipher_text)

Расшифровывание

Аргументы

in	cipher_text	Зашифрованный текст. Не должен быть пустой строкой.
----	-------------	---

Возвращает

Зашифрованная строка

Исключения

cipher_error,если	текст пустой или присутствуют строчные символы
-------------------	--

3.1.2.2 encrypt()

```
std::wstring modAlphaCipher::encrypt (
    const std::wstring & open_text )
```

Зашифровывание

Аргументы

in	open_text	Открытый текст. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	-----------	---

Возвращает

Зашифрованная строка

Исключения

cipher_error,если	текст пустой
-------------------	--------------

3.1.2.3 getValidCipherText()

```
std::wstring modAlphaCipher::getValidCipherText (
    const std::wstring & s ) [inline], [private]
```

Метод вызывающий исключение при не верном зашифрованном тексте

Аргументы

in	s	Ключ для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	--

Возвращает

зашифрованный текст

Исключения

cipher_error,если	зашифрованный текст не правильного типа
-------------------	---

3.1.2.4 getValidKey()

```
std::wstring modAlphaCipher::getValidKey (
    const std::wstring & s ) [inline], [private]
```

Метод вызывающий исключение при не верном ключе

Аргументы

in	s	Ключ для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	--

Возвращает

правильный ключ

Исключения

cipher_error,если	ключ не правильный
-------------------	--------------------

3.1.2.5 getValidOpenText()

```
std::wstring modAlphaCipher::getValidOpenText (
    const std::wstring & s ) [inline], [private]
```

Метод вызывающий исключение при не верном введенном для шифрования тексте

Аргументы

in	s	Ключ для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	--

Возвращает

правильный текст

Исключения

cipher_error,если	текст не верный
-------------------	-----------------

Объявления и описания членов классов находятся в файлах:

- [modAlphaCipher.h](#)
- [modAlphaCipher.cpp](#)

Глава 4

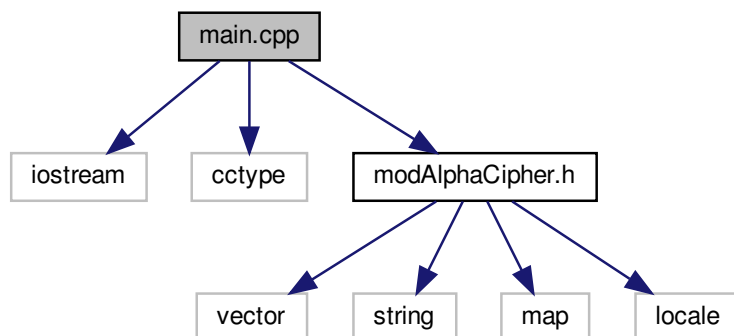
Файлы

4.1 Файл main.cpp

Файл с проверками методов для модуля Gronsfeld([main.cpp](#))

```
#include <iostream>
#include <cctype>
#include "modAlphaCipher.h"
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- void [check](#) (const wstring &Text, const wstring &key, const bool destructCipherText=false)

Функция для проверки с помощью UnitTest++.

- int [main](#) ()

Главная функция запуска метода Гринфильда

4.1.1 Подробное описание

Файл с проверками методов для модуля Gronsfeld([main.cpp](#))

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

4.1.2 Функции

4.1.2.1 check()

```
void check (
    const wstring & Text,
    const wstring & key,
    const bool destructCipherText = false )
```

Функция для проверки с помощью UnitTest++.

Аргументы

in	Text	текст для использования методов шифра.
in	key	ключ для использования методов шифра.

Возвращает

Зашифрованную и расшифрованную строку

Исключения

cipher_erroг,если	в используемых методах был не верный формат параметров
-------------------	--

4.1.2.2 main()

```
int main ( )
```

Главная функция запуска метода Гринфильда

Возвращает

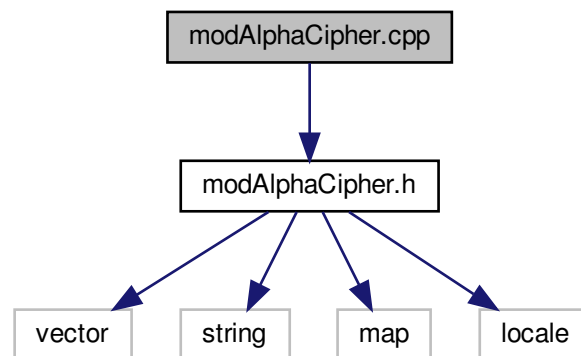
Количество пройденных программой тестирований

4.2 Файл modAlphaCipher.cpp

Файл с описаниями методов для модуля Gronsfeld.

```
#include "modAlphaCipher.h"
```

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



4.2.1 Подробное описание

Файл с описаниями методов для модуля Gronsfeld.

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

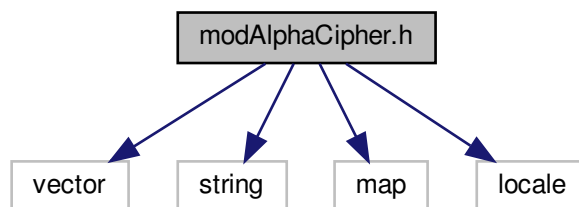
Это учебный пример

4.3 Файл modAlphaCipher.h

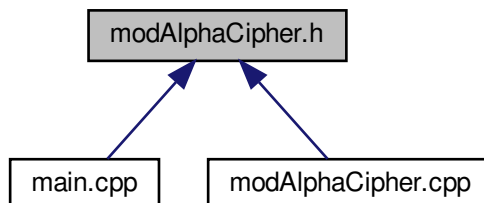
Заголовочный файл для модуля Gronsfeld.

```
#include <vector>
#include <string>
#include <map>
#include <locale>
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



Классы

- class [modAlphaCipher](#)

4.3.1 Подробное описание

Заголовочный файл для модуля Gronsfeld.

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

Предметный указатель

- check
 - main.cpp, [12](#)
- decrypt
 - modAlphaCipher, [6](#)
- encrypt
 - modAlphaCipher, [7](#)
- getValidCipherText
 - modAlphaCipher, [7](#)
- getValidKey
 - modAlphaCipher, [8](#)
- getValidOpenText
 - modAlphaCipher, [8](#)
- main
 - main.cpp, [13](#)
- main.cpp, [11](#)
 - check, [12](#)
 - main, [13](#)
- modAlphaCipher, [5](#)
 - decrypt, [6](#)
 - encrypt, [7](#)
 - getValidCipherText, [7](#)
 - getValidKey, [8](#)
 - getValidOpenText, [8](#)
 - modAlphaCipher, [6](#)
- modAlphaCipher.cpp, [13](#)
- modAlphaCipher.h, [14](#)