

Мой проект, Олег Максимов
1.0.0

Создано системой Doxygen 1.8.13

Оглавление

1	Иерархический список классов	1
1.1	Иерархия классов	1
2	Алфавитный указатель классов	3
2.1	Классы	3
3	Список файлов	5
3.1	Файлы	5
4	Классы	7
4.1	Класс <code>cipher_error</code>	7
4.1.1	Подробное описание	8
4.2	Класс <code>modAlphaCipher</code>	8
4.2.1	Подробное описание	8
4.2.2	Конструктор(ы)	9
4.2.2.1	<code>modAlphaCipher()</code> [1/2]	9
4.2.2.2	<code>modAlphaCipher()</code> [2/2]	9
4.2.3	Методы	9
4.2.3.1	<code>decrypt()</code>	9
4.2.3.2	<code>encrypt()</code>	10
4.2.3.3	<code>fromwstr()</code>	11
4.2.3.4	<code>towstr()</code>	11
5	Файлы	13
5.1	Файл <code>main.cpp</code>	13
5.1.1	Подробное описание	14
5.1.2	Функции	14
5.1.2.1	<code>main()</code>	14
5.2	Файл <code>modAlphaCipher.cpp</code>	14
5.2.1	Подробное описание	15
5.3	Файл <code>modAlphaCipher.h</code>	15
5.3.1	Подробное описание	16
	Алфавитный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

invalid_argument	
cipher_error	7
modAlphaCipher	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	
Класс-исключение	7
modAlphaCipher	
Шифрование методом Гронсфельда	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	
Заголовочный файл для модуля main.cpp	13
modAlphaCipher.cpp	
Заголовочный файл для модуля modAlphaCipher.cpp	14
modAlphaCipher.h	
Заголовочный файл для модуля Gronsfeld	15

Глава 4

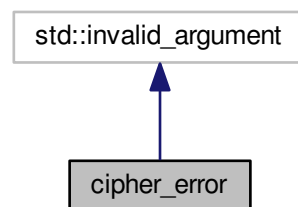
Классы

4.1 Класс `cipher_error`

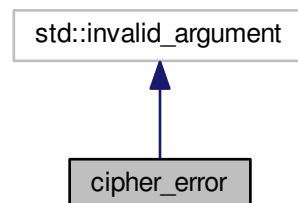
Класс-исключение.

```
#include <modAlphaCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

4.1.1 Подробное описание

Класс-исключение.

Природный от класса `invalid_argument`. При перегрузке конструкторов явно указаны вызов конструктора базового класса с параметром.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс `modAlphaCipher`

Шифрование методом Гронсфельда

```
#include <modAlphaCipher.h>
```

Открытые члены

- [modAlphaCipher](#) `()=delete`
Конструктор без параметров
- [modAlphaCipher](#) `(const std::string &skey)`
Конструктор для установки ключа
- `std::string encrypt (const std::string &open_text)`
Зашифровывание
- `std::string decrypt (const std::string &cipher_text)`
Расшифровывание
- `std::wstring towstr (const std::string &s)`
Перевод из кодировки UTF-8 в UTF-32.
- `std::string fromwstr (const std::wstring &ws)`
Перевод из кодировки UTF-32 в UTF-8.

4.2.1 Подробное описание

Шифрование методом Гронсфельда

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

Предупреждения

Реализация только для русского языка

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher() [1/2]

```
modAlphaCipher::modAlphaCipher ( ) [delete]
```

Конструктор без параметров

Запрещен

4.2.2.2 modAlphaCipher() [2/2]

```
modAlphaCipher::modAlphaCipher (
    const std::string & skey )
```

Конструктор для установки ключа

Перевод из кодировки UTF-32 в UTF-8.

Ключ проверяется на валидность. Переводится в кодировку UTF-32. Формируется вектор-ключ.

Аргументы

in	skey	Ключ в кодировке UTF-8
----	------	------------------------

Исключения

cipher_error	ключ вырожденный
------------------------------	------------------

Аргументы

in	s	Строка в кодировке UTF-32
----	---	---------------------------

Возвращает

Строка в кодировке UTF-8

4.2.3 Методы

4.2.3.1 decrypt()

```
std::string modAlphaCipher::decrypt (
    const std::string & cipher_text )
```

Расшифровывание

Аргументы

in	cipher_text	Зашифрованный текст в кодировке UTF-8. Не должен быть пустой строкой. Не должен содержать строчные символы и не-буквы.
----	-------------	--

Возвращает

Расифрованная строка в кодировке UTF-8

Исключения

cipher_error	текст пустой
------------------------------	--------------

Аргументы

in	cipher_text	Зашифрованный текст в кодировке UTF-8. Не должен быть пустой строкой. Не должен содержать строчные символы и небуквы.
----	-------------	---

Возвращает

Расифрованная строка в кодировке UTF-8

Исключения

cipher_error	текст пустой
------------------------------	--------------

4.2.3.2 encrypt()

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

Зашифровывание

Аргументы

in	open_text	Открытый текст в кодировке UTF-8. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются.
----	-----------	--

Возвращает

Зашифрованная строка в кодировке UTF-8

Исключения

<code>cipher_error</code>	текст пустой
---------------------------	--------------

11*

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

<code>cipher_error</code> ,если	текст пустой
---------------------------------	--------------

4.2.3.3 fromwstr()

```
std::string modAlphaCipher::fromwstr (
    const std::wstring & ws )
```

Перевод из кодировки UTF-32 в UTF-8.

Аргументы

in	s	Строка в кодировке UTF-32
----	---	---------------------------

Возвращает

Строка в кодировке UTF-8

4.2.3.4 towstr()

```
std::wstring modAlphaCipher::towstr (
    const std::string & s )
```

Перевод из кодировки UTF-8 в UTF-32.

Аргументы

in	s	Строка в кодировке UTF-8
----	---	--------------------------

Возвращает

Строка в кодировке UTF-32

Объявления и описания членов классов находятся в файлах:

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

Глава 5

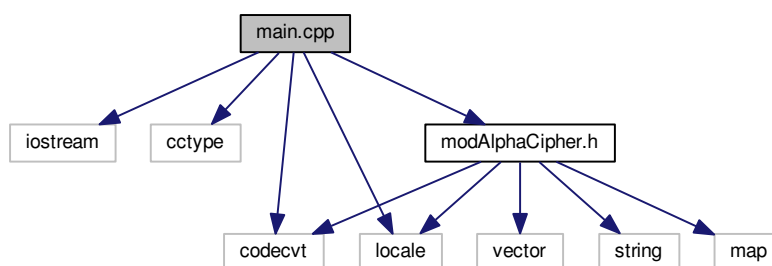
Файлы

5.1 Файл main.cpp

Заголовочный файл для модуля [main.cpp](#).

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- bool isValid (const string &s)
- int [main](#) (int argc, char **argv)

Интерфейс программы

5.1.1 Подробное описание

Заголовочный файл для модуля `main.cpp`.

Автор

Максимов О.В.

Версия

1.0.0

Дата

13.06.2019

5.1.2 Функции

5.1.2.1 `main()`

```
int main (  
    int argc,  
    char ** argv )
```

Интерфейс программы

Осуществление выбора ключа и операции 0, 1 или 2. В зависимости от выбора выполняются следующие действия: выход, зашифровка, расшифровка. ввод ключа

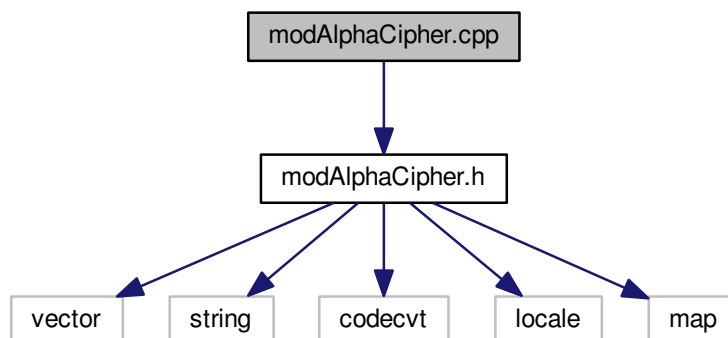
ввод числа

5.2 Файл `modAlphaCipher.cpp`

Заголовочный файл для модуля `modAlphaCipher.cpp`.

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

Граф включаемых заголовочных файлов для `modAlphaCipher.cpp`:



5.2.1 Подробное описание

Заголовочный файл для модуля `modAlphaCipher.cpp`.

Автор

Максимов О.В.

Версия

1.0.0

Дата

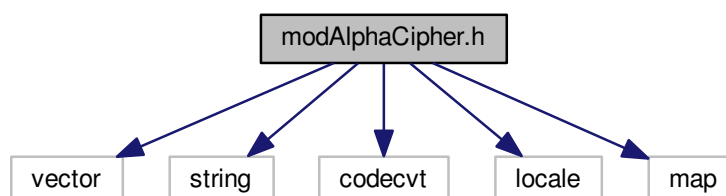
13.06.2019

5.3 Файл modAlphaCipher.h

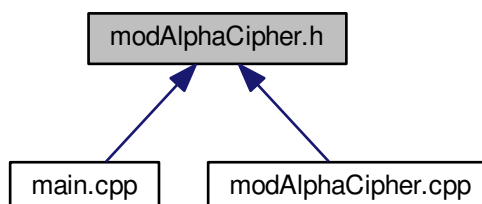
Заголовочный файл для модуля Gronsfeld.

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

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Граф файлов, в которые включается этот файл:



Классы

- class `modAlphaCipher`
Шифрование методом Гронсфельда
- class `cipher_error`
Класс-исключение.

5.3.1 Подробное описание

Заголовочный файл для модуля Gronsfeld.

Автор

Максимов О.В.

Версия

1.0.0

Дата

13.06.2019

Предметный указатель

cipher_error, [7](#)

decrypt

modAlphaCipher, [9](#)

encrypt

modAlphaCipher, [10](#)

fromwstr

modAlphaCipher, [11](#)

main

main.cpp, [14](#)

main.cpp, [13](#)

main, [14](#)

modAlphaCipher, [8](#)

decrypt, [9](#)

encrypt, [10](#)

fromwstr, [11](#)

modAlphaCipher, [9](#)

towstr, [11](#)

modAlphaCipher.cpp, [14](#)

modAlphaCipher.h, [15](#)

towstr

modAlphaCipher, [11](#)