

NotLab4

1.5

Создано системой Doxygen 1.9.1

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 cipher_error	7
4.1.1 Подробное описание	8
4.2 Класс modAlphaCipher	8
4.2.1 Подробное описание	9
4.2.2 Методы	9
4.2.2.1 decrypt()	9
4.2.2.2 encrypt()	10
4.2.2.3 getValidCipherText()	10
4.2.2.4 getValidKey()	11
4.2.2.5 getValidOpenText()	11
5 Файлы	13
5.1 Файл modAlphaCipher.cpp	13
5.1.1 Подробное описание	13
Предметный указатель	15

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	7
modAlphaCipher	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Класс ошибок для шифрования методом Гронсфельда	7
modAlphaCipher	Шифрование методом Гронсфельда	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.cpp	
Исполняемый файл для модуля Gronsfeld	13
modAlphaCipher.h	??

Глава 4

Классы

4.1 Класс `cipher_error`

Класс ошибок для шифрования методом Гронсфельда

```
#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 ошибок для шифрования методом Гронсфельда

Используется для указания специфических ошибок при работе с шифром

Предупреждения

Вывод сообщений об ошибках только на английском языке

Объявления и описания членов класса находятся в файле:

- `modAlphaCipher.h`

4.2 Класс `modAlphaCipher`

Шифрование методом Гронсфельда

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

Открытые члены

- `modAlphaCipher ()=delete`
Запретим конструктор без параметров
- `modAlphaCipher (const std::wstring &skey)`
Конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`
Зашифрование
- `std::wstring decrypt (const std::wstring &cipher_text)`
Расшифрование

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)`
Обработка ошибок ключа
- `std::wstring getValidOpenText (const std::wstring &s)`
Обработка ошибок открытого текста
- `std::wstring getValidCipherText (const std::wstring &s)`
Обработка ошибок шифро текста

Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Алфавит по порядку
- `std::map< wchar_t, int > alphaNum`
Ассоциативный массив "номер по символу".
- `std::vector< int > key`
Ключ

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 Методы

4.2.2.1 `decrypt()`

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

Расшифрование

Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Аргументы

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

Возвращает

Расшифрованная строка

Исключения

<code>cipher_error</code> , если	текст пустой или присутствуют символы, не входящие в алфавит
----------------------------------	--

4.2.2.2 encrypt()

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

Зашифрование

Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Аргументы

in	open_text	Открытый текст. Не должен быть пустой строкой.
----	-----------	--

Возвращает

Зашифрованная строка

Исключения

cipher_error ,если	текст пустой или присутствуют символы, не входящие в алфавит
------------------------------------	--

4.2.2.3 getValidCipherText()

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

Обработка ошибок шифро текста

Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Аргументы

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

Возвращает

Обработанный шифро текст

Исключения

cipher_error ,если	шифро текст пустой или присутствуют символы, не входящие в алфавит
------------------------------------	--

4.2.2.4 getValidKey()

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

Обработка ошибок ключа

Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Аргументы

in	key	Ключ. Не должен быть пустой строкой
----	-----	-------------------------------------

Возвращает

Обработанный ключ

Исключения

cipher_error ,если	ключ пустой или присутствуют символы, не входящие в алфавит
------------------------------------	---

4.2.2.5 getValidOpenText()

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

Обработка ошибок открытого текста

Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Аргументы

in	open_text	Открытый текст. Не должен быть пустой строкой.
----	-----------	--

Возвращает

Обработанный открытый текст

Исключения

cipher_error ,если	открытый текст пустой или присутствуют символы, не входящие в алфавит
------------------------------------	---

Объявления и описания членов классов находятся в файлах:

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

Глава 5

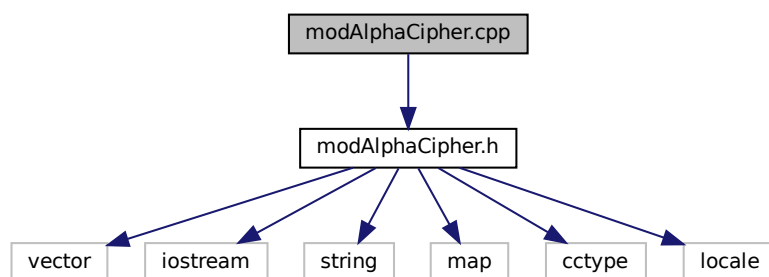
Файлы

5.1 Файл modAlphaCipher.cpp

Исполняемый файл для модуля Gronsfeld.

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

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



5.1.1 Подробное описание

Исполняемый файл для модуля Gronsfeld.

Автор

Полехин Д.А.

Версия

1.0

Дата

22.12.2019

Авторство

ИБСТ ПГУ

Предметный указатель

`cipher_error`, [7](#)

`decrypt`

`modAlphaCipher`, [9](#)

`encrypt`

`modAlphaCipher`, [9](#)

`getValidCipherText`

`modAlphaCipher`, [10](#)

`getValidKey`

`modAlphaCipher`, [10](#)

`getValidOpenText`

`modAlphaCipher`, [11](#)

`modAlphaCipher`, [8](#)

`decrypt`, [9](#)

`encrypt`, [9](#)

`getValidCipherText`, [10](#)

`getValidKey`, [10](#)

`getValidOpenText`, [11](#)

`modAlphaCipher.cpp`, [13](#)