

modAlphaCipher

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
4.2 Класс modAlphaCipher.....	8
4.2.1 Подробное описание.....	9
4.2.2 Конструктор(ы).....	9
4.2.2.1 modAlphaCipher().....	9
4.2.3 Методы.....	9
4.2.3.1 decrypt().....	9
4.2.3.2 encrypt().....	10
4.2.3.3 getValidCipherText().....	10
4.2.3.4 getValidKey().....	11
4.2.3.5 getValidOpenText().....	11
5 Файлы	13
5.1 Файл modAlphaCipher.h.....	13
5.1.1 Подробное описание.....	13
Предметный указатель	15

1.1

Иерархия классов.

invalid_argument	
cipher_error	7
modAlphaCipher.....	8

2.1

Классы с их кратким описанием.

cipher_error	7
modAlphaCipher	
Шифрование методом Гронсфельда	8

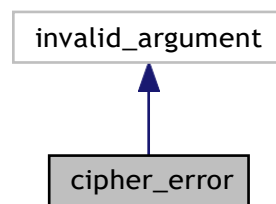
3.1

Полный список документированных файлов.

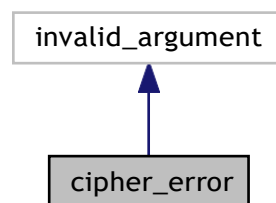
modAlphaCipher.h	
Заголовочный файл для модуля modAlphaCipher	13

4.1 cipher_error

Граф наследования: cipher_error:



Граф связей класса cipher_error:



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

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 modAlphaCipher

Шифрование методом Гронсфельда

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

- [modAlphaCipher](#) ()=delete
- [modAlphaCipher](#) (const string &skey)
- string [encrypt](#) (const string &open_text)
- string [decrypt](#) (const string &cipher_text)
- vector< int > [convert](#) (const string &s)
- string [convert](#) (const vector< int > &v)
- string [getValidKey](#) (const string &s)
- string [getValidOpenText](#) (const string &s)
- string [getValidCipherText](#) (const string &s)
- wstring [numAlpha](#) = L"АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЪЭЮЯ"
- map< char, int > [alphaNum](#)
- vector< int > [key](#)

4.2.1

Шифрование методом Гронсфелда

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

Предупреждения

Реализация только для русского языка.

4.2.2 ()

4.2.2.1 modAlphaCipher()

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

Конструктор для установки ключа.

Аргументы

skey	Строка для установки ключа. Должен содержать только строчные и прописные буквы кириллицы.
------	---

4.2.3

4.2.3.1 decrypt()

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

Расшифрование.

Аргументы

cipher_text	Зашифрованный текст на русском языке.
-------------	---------------------------------------

Предупреждения

Зашифрованный текст не должен быть пустой строкой, не должен содержать пробелы, символы пунктуации, цифры и буквы нижнего регистра.

Возвращает

Расшифрованный текст

4.2.3.2 encrypt()

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

Зашифрование.

Аргументы

open_text	Текст на русском языке.
-----------	-------------------------

Предупреждения

Текст не должен быть пустой строкой, не должен содержать пробелы, символы пунктуации и цифры.

Возвращает

Зашифрованный текст

4.2.3.3 getValidCipherText()

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

Валидация зашифрованного текста.

Аргументы

s	Зашифрованный текст на русском языке. Не должен быть пустым. Буквы нижнего регистра, символы пунктуации, цифры и пробелы приводят к возбуждению исключений.
---	---

Возвращает

Зашифрованный текст, прошедший валидацию.

Исключения

cipher_error	пустая строка, некорректные символы.
------------------------------	--------------------------------------

4.2.3.4 getValidKey()

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

Валидация ключа.

Аргументы

s	Ключ. Должен содержать только строчные и прописные буквы кириллицы. Если необходимо, буквы переводятся в верхний регистр.
---	---

Возвращает

Ключ, прошедший валидацию.

Исключения

cipher_error	пустая строка, некорректные символы в строке.
------------------------------	---

4.2.3.5 getValidOpenText()

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

Валидация текста.

Аргументы

s	Текст на русском языке. Не должен быть пустым. Символы пунктуации, цифры и пробелы приводят к возбуждению исключений. Если необходимо, буквы переводятся в верхний регистр.
---	---

Возвращает

Текст, прошедший валидацию.

Исключения

cipher_error	пустая строка, некорректные символы.
------------------------------	--------------------------------------

Объявления и описания членов классов находятся в файлах:

- [modAlphaCipher.h](#)

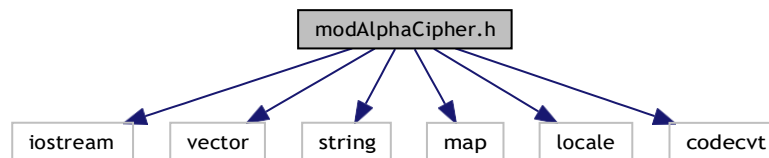
- modAlphaCipher.cpp

5.1 modAlphaCipher.h

Заголовочный файл для модуля [modAlphaCipher](#).

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



- class [modAlphaCipher](#)
- class [cipher_error](#)

5.1.1

Заголовочный файл для модуля [modAlphaCipher](#).

Автор

Попов М.С.

Версия

1.0.0

Дата

05.09.2023

Авторство

ИБСТ ПГУ

- cipher_error, [7](#)
- decrypt
 - modAlphaCipher, [9](#)
- encrypt
 - modAlphaCipher, [10](#)
- getValidCipherText
 - modAlphaCipher, [10](#)
- getValidKey
 - modAlphaCipher, [11](#)
- getValidOpenText
 - modAlphaCipher, [11](#)
- modAlphaCipher, [8](#)
 - decrypt, [9](#)
 - encrypt, [10](#)
 - getValidCipherText, [10](#)
 - getValidKey, [11](#)
 - getValidOpenText, [11](#)
 - modAlphaCipher, [9](#)
- modAlphaCipher.h, [13](#)