

Шифрование методом Гронсфельда.

1.0

Создано системой Doxygen 1.8.17

1 Иерархический список классов	1
1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	1
2.1 Классы	1
3 Список файлов	2
3.1 Файлы	2
4 Классы	2
4.1 Класс <code>cipher_error</code>	2
4.1.1 Подробное описание	3
4.1.2 Конструктор(ы)	3
4.2 Класс <code>modAlphaCipher</code>	3
4.2.1 Подробное описание	4
4.2.2 Конструктор(ы)	4
4.2.3 Методы	5
5 Файлы	7
5.1 Файл <code>modAlphaCipher.h</code>	7
5.1.1 Подробное описание	8
Предметный указатель	9

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

<code>invalid_argument</code>	
<code>cipher_error</code>	2
<code>modAlphaCipher</code>	3

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

<code>cipher_error</code>	
Собственный класс исключений, наследуемый от класса <code>"invalid_argument"</code>	2
<code>modAlphaCipher</code>	
Класс шифрования и расшифрования текста алгоритмом Гронсфельда	3

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

[modAlphaCipher.h](#)

Описание класса [modAlphaCipher](#)

[7](#)

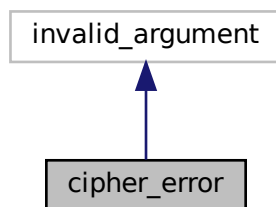
4 Классы

4.1 Класс cipher_error

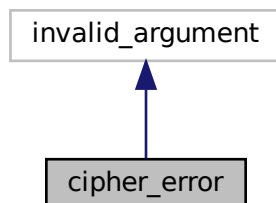
Собственный класс исключений, наследуемый от класса "invalid_argument".

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

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- [cipher_error](#) (const string &what_arg)
Конструктор инициализации ошибки строкой.
- [cipher_error](#) (const char *what_arg)
Конструктор инициализации ошибки Си-строкой.

4.1.1 Подробное описание

Собственный класс исключений, наследуемый от класса "invalid_argument".

4.1.2 Конструктор(ы)

4.1.2.1 `cipher_error()` [1/2] `cipher_error::cipher_error (const string & what_arg)` [inline], [explicit]

Конструктор инициализации ошибки строкой.

Аргументы

what_arg	- описание ошибки.
----------	--------------------

4.1.2.2 `cipher_error()` [2/2] `cipher_error::cipher_error (const char * what_arg)` [inline], [explicit]

Конструктор инициализации ошибки Си-строкой.

Аргументы

what_arg	- описание ошибки.
----------	--------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Класс зашифрования и расшифрования текста алгоритмом Гронсфельда.

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

Открытые члены

- `modAlphaCipher ()=delete`
Запрет конструктора без параметров.
- `modAlphaCipher (wstring skey)`
Конструктор для установки ключа.
- `wstring encrypt (const wstring &open_text)`
Метод зашифрования текста.
- `wstring decrypt (const wstring &cipher_text)`
Метод расшифрования текста.
- `bool language (wstring &skey)`
Метод проверки ключа на соответствие алфавиту.

Закрытые члены

- `vector< int > convert (const wstring &s)`
Преобразование строка-вектор.
- `wstring convert (const vector< int > &v)`
Преобразование вектор-строка.
- `wstring getValidKey (const wstring &s)`
Метод валидации ключа.
- `wstring getValidOpenText (const wstring &s)`
Метод валидации текста, который необходимо зашифровать.
- `wstring getValidCipherText (const wstring &s)`
Метод валидации текста, который необходимо расшифровать.

Закрытые данные

- `wstring numAlpha2 =L"ABCDEFGHIJKLMNOPQRSTUVWXYZ"`
Алфавит для работы с английским текстом
- `wstring numAlpha =L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Алфавит для работы с русским текстом
- `map< char, int > alphaNum`
ассоциативный массив "номер по символу".
- `vector< int > key`
ключ

4.2.1 Подробное описание

Класс зашифрования и расшифрования текста алгоритмом Гронсфельда.

4.2.2 Конструктор(ы)

4.2.2.1 `modAlphaCipher()` `modAlphaCipher::modAlphaCipher (wstring skey)`

Конструктор для установки ключа.

Аргументы

skey	- ключ, введенный пользователем.
------	----------------------------------

После проверки ключа с помощью метода "getValidKey" конвертирует строку-ключ в вектор.

4.2.3 Методы

4.2.3.1 convert() [1/2] wstring modAlphaCipher::convert (
const vector< int > & v) [private]

Преобразование вектор-строка.

Аргументы

v	- вектор для преобразования в строку.
---	---------------------------------------

4.2.3.2 convert() [2/2] vector<int> modAlphaCipher::convert (
const wstring & s) [private]

Преобразование строка-вектор.

Аргументы

s	- строка для преобразования в вектор.
---	---------------------------------------

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

Метод расшифрования текста.

Аргументы

cipher_text	- текст для расшифрования.
-------------	----------------------------

Возвращает

Расшифрованная строка.

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

Метод зашифрования текста.

Аргументы

open_text	- текст для зашифрования.
-----------	---------------------------

Возвращает

Зашифрованная методом Гронсфельда строка.

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

Метод валидации текста, который необходимо расшифровать.

Аргументы

s	- текст-строка, введенный пользователем.
---	--

Метод проверяет корректность ввода текста-строки и выбрасывает исключение если: Введен пустой текст, текст содержит символы нижнего регистра или символы, которых нет в используемом алфавите.

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

Метод валидации ключа.

Аргументы

s	- ключ-строка.
---	----------------

Метод проверяет правильность ввода ключа-строки и выбрасывает исключение если: Введен пустой ключ, ключ содержит символы, не являющиеся символами русского или английского алфавита. Все символы ключа переводятся в верхний регистр.

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

Метод валидации текста, который необходимо зашифровать.

Аргументы

s	- текст-строка, введенный пользователем.
---	--

Метод проверяет корректность ввода текста-строки и выбрасывает исключение если: Введен пустой текст, текст содержит символы, которых нет в используемом алфавите. Все символы текста переводятся в верхний регистр.

4.2.3.8 `language()` `bool modAlphaCipher::language (`
`wstring & skey)`

Метод проверки ключа на соответствие алфавиту.

Метод выбрасывает исключение, если в ключе находятся символы русского и английского алфавита одновременно.

Аргументы

<code>skey</code>	- ключ-строка.
-------------------	----------------

Возвращает

Значение `true`, если проверка прошла.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

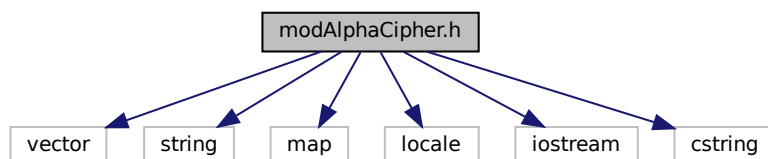
5 Файлы

5.1 Файл `modAlphaCipher.h`

Описание класса [modAlphaCipher](#).

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

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Классы

- class [modAlphaCipher](#)
Класс зашифрования и расшифрования текста алгоритмом Гронсфельда.
- class [cipher_error](#)
Собственный класс исключений, наследуемый от класса "invalid_argument".

5.1.1 Подробное описание

Описание класса [modAlphaCipher](#).

Автор

Рясков.С.С.

Версия

1.0

Дата

29.05.2021

Авторство

ИБСТ ПГУ

Предметный указатель

- cipher_error, [2](#)
 - cipher_error, [3](#)
- convert
 - modAlphaCipher, [5](#)
- decrypt
 - modAlphaCipher, [5](#)
- encrypt
 - modAlphaCipher, [5](#)
- getValidCipherText
 - modAlphaCipher, [6](#)
- getValidKey
 - modAlphaCipher, [6](#)
- getValidOpenText
 - modAlphaCipher, [6](#)
- language
 - modAlphaCipher, [7](#)
- modAlphaCipher, [3](#)
 - convert, [5](#)
 - decrypt, [5](#)
 - encrypt, [5](#)
 - getValidCipherText, [6](#)
 - getValidKey, [6](#)
 - getValidOpenText, [6](#)
 - language, [7](#)
 - modAlphaCipher, [4](#)
- modAlphaCipher.h, [7](#)