

modAlphaCipher

1.0

Создано системой Doxygen 1.8.17

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 Конструктор(ы)	9
4.2.2.1 modAlphaCipher()	9
4.2.3 Методы	9
4.2.3.1 convert() [1/2]	9
4.2.3.2 convert() [2/2]	10
4.2.3.3 decrypt()	10
4.2.3.4 encrypt()	11
4.2.3.5 getValidKey()	11
4.2.3.6 getValidText()	12
5 Файлы	13
5.1 Файл modAlphaCipher.h	13
5.1.1 Подробное описание	14
Предметный указатель	15

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.

<code>cipher_error</code>	7
<code>modAlphaCipher</code> Класс, реализующий шифрование сообщений методом "Гронсвельда"	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

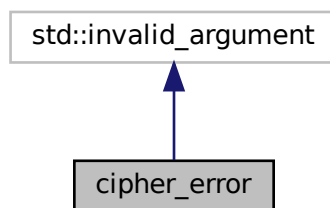
modAlphaCipher.h	
Описание класса modAlphaCipher	13

Глава 4

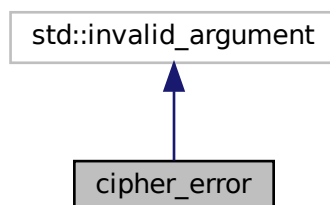
Классы

4.1 Класс `cipher_error`

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

предназначенный для возбуждения исключений.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Класс, реализующий шифрование сообщений методом "Гронсвельда".

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

Открытые члены

- [modAlphaCipher](#) ()=delete
Конструктор без параметров.
- [modAlphaCipher](#) (const std::wstring &key)
Конструктор для ключа
- std::wstring [encrypt](#) (const std::wstring &open_text)
Метод, который предназначен для зашифрования
- std::wstring [decrypt](#) (const std::wstring &cipher_text)
Метод, который предназначен для расшифрования

Закрытые члены

- std::vector< int > [convert](#) (const std::wstring &s)
Преобразование строки в вектор(чисел)
- std::wstring [convert](#) (const std::vector< int > &v)
Преобразование вектора(чисел) в строку
- std::wstring [getValidKey](#) (const std::wstring &s)
Валидация ключа
- std::wstring [getValidText](#) (const std::wstring &s)
Валидация текста при шифровании или расшифровании

Закрытые данные

- std::wstring [numAlpha](#) = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
Алфавит для сообщений, который используется в данной программе
- std::map< char, int > [alphaNum](#)
Ассоциативный массив "номер по символу".
- std::vector< int > [key](#)
Атрибут для ключа

4.2.1 Подробное описание

Класс, реализующий шифрование сообщений методом "Гронсвельда".

Предупреждения

Предназначен для русскоязычных сообщений.

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

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

Конструктор для ключа

Цикл `for` построен по строке-алфавиту и на каждом шаге добавляет в ассоциативный массив символ и его номер.

```
for (unsigned i=0; i<numAlpha.size(); i++) {
    alphaNum[numAlpha[i]]=i;
}
```

Аргументы

строка	текста типа "wstring"
--------	-----------------------

4.2.3 Методы

4.2.3.1 convert() [1/2]

```
std::wstring modAlphaCipher::convert (
    const std::vector< int > & v ) [inline], [private]
```

Преобразование вектора(чисел) в строку

В переменную типа "wstring" с именем "result" формируется строка по индексам алфавита "num←Alpha".

```
wstring result;
for(auto i:v) {
    result.push_back(numAlpha[i]);
}
```

Возвращает

"result" - строка текста типа "wstring"

4.2.3.2 convert() [2/2]

```
std::vector< int > modAlphaCipher::convert (
    const std::wstring & s ) [inline], [private]
```

Преобразование строки в вектор(чисел)

В вектор типа "int" с именем "result" формируются числа, являющиеся индексами алфавита "num←Alpha" из строки, которая пришла на вход.

```
vector<int> result;
for(auto c:s) {
    result.push_back(alphaNum[c]);
}
```

Возвращает

std::vector <int>, в котором хранятся индексы букв сообщения из алфавита

4.2.3.3 decrypt()

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

Метод, который предназначен для расшифрования

Формируется вектор work из строки зашифрованного текста с помощью метода [convert\(\)](#). А также происходит проверка зашифрованного текста на наличие ошибки при помощи метода [getValid←AlphabetText\(\)](#).

```
vector<int> work = convert(getValidAlphabetText(cipher_text));
```

Если при зашифровывании мы прибавляли значение ключа, то при расшифровывании значения ключа надо вычитать. А чтобы не получить отрицательных значений, выполняется еще прибавление значения модуля, так как такое прибавление не влияет на результат модулю.

```
for(unsigned i=0; i < work.size(); i++) {
    work[i] = (work[i] + alphaNum.size() - key[i % key.size()]) % alphaNum.size();
}
```

Аргументы

wstring	cipher_text - сообщение, которое нужно расшифровать
---------	---

Исключения

cipher_error	- строка, которая пришла на вход оказывается пустой или в ней есть недопустимые символы
------------------------------	---

Возвращает

строка расшифрованного текста типа "wstring"

4.2.3.4 encrypt()

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

Метод, который предназначен для зашифрования

Формируется вектор `work` из строки открытого текста с помощью метода `convert()`. Происходит проверка текста на наличие ошибки при помощи метода `getValidAlphabetText()`.
`vector<int> work = convert(getValidAlphabetText(open_text));`

В цикле к каждому элементу вектора прибавляется элемент ключа по модулю размера алфавита. Если ключ короче текста, то при индексации ключа выполняется операция по модулю размера ключа.

```
for(unsigned i=0; i < work.size(); i++) {
    work[i] = (work[i] + key[i % key.size()]) % alphaNum.size();
}
```

При возврате значения, вектор `work` опять преобразуется в строку.

Аргументы

<code>wstring</code>	<code>open_text</code> - сообщение, которое нужно зашифровать
----------------------	---

Исключения

<code>cipher_error</code>	- строка, которая пришла на вход оказывается пустой или в ней есть недопустимые символы
---------------------------	---

Возвращает

строка зашифрованного текста типа "wstring"

4.2.3.5 getValidKey()

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

Валидация ключа

Ключ проверяется на пустоту при помощи обычного условия. Если проверка закончилась успешно, то ключ проверяется на наличие недопустимых символов.

Предупреждения

Строчные буквы алфавита переводятся в прописные.

Аргументы

<code>wstring</code>	<code>s</code> - ключ, который нужно проверить на наличие ошибок, в виде строки.
----------------------	--

Исключения

cipher_error ,если	ключ является пустым или в нём присутствуют недопустимые символы
------------------------------------	--

Возвращает

"result" - строка текста типа "wstring"

4.2.3.6 getValidText()

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

Валидация текста при шифровании или расшифровании

Сначала текст проверяется на пустоту при помощи обычного условия. Если проверка закончилась успешно, то текст проверяется на наличие недопустимых символов.

Предупреждения

Строчные буквы алфавита переводятся в прописные.

Аргументы

wstring	s - строка текста для шифрования или расшифрования, которую нужно проверить на наличие ошибок
---------	---

Исключения

cipher_error ,если	текст является пустым или в нём присутствуют недопустимые символы
------------------------------------	---

Возвращает

строка текста типа "wstring"

Объявления и описания членов классов находятся в файлах:

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

Глава 5

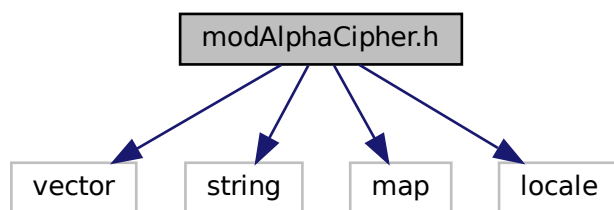
Файлы

5.1 Файл modAlphaCipher.h

Описание класса `modAlphaCipher`.

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

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Классы

- class `modAlphaCipher`

Класс, реализующий шифрование сообщений методом "Гронсвельда".

- class `cipher_error`

5.1.1 Подробное описание

Описание класса [modAlphaCipher](#).

Автор

Корзин А.А.

Версия

1.0

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предметный указатель

- cipher_error, [7](#)
- convert
 - modAlphaCipher, [9](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [10](#)
- getValidKey
 - modAlphaCipher, [11](#)
- getValidText
 - modAlphaCipher, [12](#)
- modAlphaCipher, [8](#)
 - convert, [9](#)
 - decrypt, [10](#)
 - encrypt, [10](#)
 - getValidKey, [11](#)
 - getValidText, [12](#)
 - modAlphaCipher, [9](#)
- modAlphaCipher.h, [13](#)