

Шифрование методом Гронсвельда

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 CipherError	2
4.1.1 Подробное описание	3
4.1.2 Конструктор(ы)	3
4.2 Класс modAlphaCipher	3
4.2.1 Подробное описание	4
4.2.2 Конструктор(ы)	4
4.2.3 Методы	4
5 Файлы	7
5.1 Файл CipherError.h	7
5.1.1 Подробное описание	8
5.2 Файл modAlphaCipher.h	8
5.2.1 Подробное описание	9
Предметный указатель	11

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

invalid_argument	
CipherError	2
modAlphaCipher	3

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

CipherError	
Класс для обработки ошибок, которые могут возникнуть при взаимодействии пользователя с программой	2

[modAlphaCipher](#)

Класс, предназначенный для шифрования сообщений методом Гронсвельда

[3](#)

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

[CipherError.h](#)

Описание класса [CipherError](#)

[7](#)[modAlphaCipher.h](#)

Описание класса [modAlphaCipher](#)

[8](#)

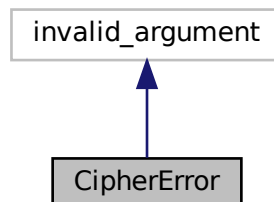
4 Классы

4.1 Класс CipherError

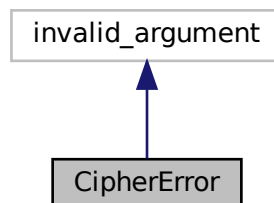
Класс для обработки ошибок, которые могут возникнуть при взаимодействии пользователя с программой

```
#include <CipherError.h>
```

Граф наследования: CipherError:



Граф связей класса CipherError:



Открытые члены

- [CipherError](#) ()=delete
Запрещающий конструктор без параметров
- [CipherError](#) (const string &what_arg)
Конструктор с параметром

4.1.1 Подробное описание

Класс для обработки ошибок, которые могут возникнуть при взаимодействии пользователя с программой

Класс является наследником существующего класса обработки исключений с именем `invalid_` ← `argument`

4.1.2 Конструктор(ы)

4.1.2.1 `CipherError()` `CipherError::CipherError` (
const string & what_arg) [inline], [explicit]

Конструктор с параметром

Аргументы

std::string	what_arg - строка, хранящая описание ошибки. Данный параметр перегружается конструктором класса <code>invalid_argument</code>
-------------	---

Объявления и описания членов класса находятся в файле:

- [CipherError.h](#)

4.2 Класс modAlphaCipher

Класс, предназначенный для шифрования сообщений методом Гронсвельда

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

Открытые члены

- [modAlphaCipher](#) ()=delete
Запрещающий конструктор без параметров
- [modAlphaCipher](#) (const wstring &skey)
Конструктор для ключа
- wstring [encrypt](#) (const wstring &open_text)
Метод, предназначенный для шифрования
- wstring [decrypt](#) (const wstring &cipher_text)
Метод, предназначенный для расшифрования

Закрытые члены

- `vector< int > convert (const wstring &s)`
Преобразование строки в вектор(чисел)
- `wstring convert (const vector< int > &v)`
Преобразование вектора(чисел) в строку
- `wstring getValidKey (const wstring &s)`
Метод, предназначенный для валидации ключа
- `wstring getValidText (const wstring &s)`
Метод, предназначенный для валидации строки при шифровании или расшифровании

Закрытые данные

- `wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Используемый алфавит для сообщений
- `map< char, int > alphaNum`
Ассоциативный массив "номер по символу".
- `vector< int > key`
Атрибут для ключа

4.2.1 Подробное описание

Класс, предназначенный для шифрования сообщений методом Гронсвельда

Предупреждения

Работает только с русскоязычными сообщениями

4.2.2 Конструктор(ы)

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

Конструктор для ключа

Аргументы

<code>std::wstring</code>	- ключ в виде строки
---------------------------	----------------------

4.2.3 Методы

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

Преобразование вектора(чисел) в строку

Аргументы

<code>std::vector<int></code>	v - вектор чисел, которые преобразуются в строку по индексам
-------------------------------------	--

Возвращает

строка текста типа `wstring`

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

Преобразование строки в вектор(чисел)

Аргументы

<code>std::wstring</code>	s - строка для преобразования в вектор(чисел)
---------------------------	---

Возвращает

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

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

Метод, предназначенный для расшифрования

Аргументы

<code>std::wstring</code>	<code>cipher_text</code> - текст для расшифрования
---------------------------	--

Возвращает

Расшифрованная методом Гронсфельда строка типа `wstring`

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

Метод, предназначенный для шифрования

Аргументы

<code>std::wstring</code>	<code>open_text</code> - текст для зашифрования
---------------------------	---

Возвращает

Зашифрованная методом Гронсфельда строка типа `wstring`

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

Метод, предназначенный для валидации ключа

Аргументы

<code>std::wstring</code>	<code>s</code> - ключ, который нужно проверить на наличие ошибок
---------------------------	--

Предупреждения

Письменные буквы алфавита переводятся в прописные

Исключения

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

Возвращает

Ключ типа `wstring`, который успешно прошёл валидацию

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

Метод, предназначенный для валидации строки при шифровании или расшифровании

Аргументы

<code>std::wstring</code>	<code>s</code> - строка, которую нужно проверить на наличие ошибок
---------------------------	--

Предупреждения

Письменные буквы алфавита переводятся в прописные

Исключения

CipherError , если	строка является пустой или в ней присутствуют недопустимые символы
------------------------------------	--

Возвращает

строка типа `wstring`, которая успешно прошла валидацию

Объявления и описания членов классов находятся в файлах:

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

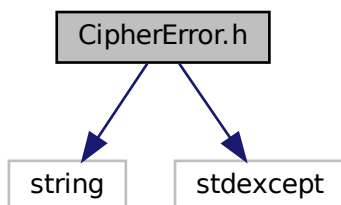
5 Файлы

5.1 Файл `CipherError.h`

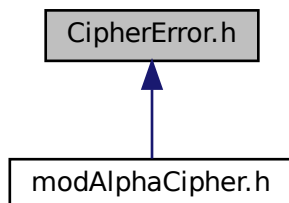
Описание класса [CipherError](#).

```
#include <string>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для `CipherError.h`:



Граф файлов, в которые включается этот файл:



Классы

- class [CipherError](#)

Класс для обработки ошибок, которые могут возникнуть при взаимодействии пользователя с программой

5.1.1 Подробное описание

Описание класса [CipherError](#).

Автор

Антонов И.С.

Версия

1.0

Авторство

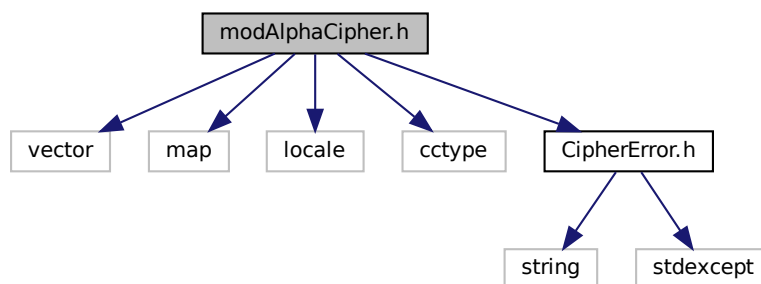
ИБСТ ПГУ

5.2 Файл modAlphaCipher.h

Описание класса [modAlphaCipher](#).

```
#include <vector>
#include <map>
#include <locale>
#include <cctype>
#include "CipherError.h"
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class [modAlphaCipher](#)

Класс, предназначенный для шифрования сообщений методом Гронсвельда

5.2.1 Подробное описание

Описание класса [modAlphaCipher](#).

Автор

Антонов И.С.

Версия

1.0

Авторство

ИБСТ ПГУ

Предметный указатель

- CipherError, [2](#)
 - CipherError, [3](#)
- CipherError.h, [7](#)
- convert
 - modAlphaCipher, [4](#), [5](#)
- decrypt
 - modAlphaCipher, [5](#)
- encrypt
 - modAlphaCipher, [5](#)
- getValidKey
 - modAlphaCipher, [6](#)
- getValidText
 - modAlphaCipher, [6](#)
- modAlphaCipher, [3](#)
 - convert, [4](#), [5](#)
 - decrypt, [5](#)
 - encrypt, [5](#)
 - getValidKey, [6](#)
 - getValidText, [6](#)
 - modAlphaCipher, [4](#)
- modAlphaCipher.h, [8](#)