

Шифрование методом Гронсвельда

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 <code>Error</code>	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 <code>Error()</code>	8
4.2 Класс <code>modAlphaCipher</code>	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 <code>modAlphaCipher()</code>	9
4.2.3 Методы	9
4.2.3.1 <code>convert()</code> [1/2]	10
4.2.3.2 <code>convert()</code> [2/2]	10
4.2.3.3 <code>decrypt()</code>	10
4.2.3.4 <code>encrypt()</code>	11
4.2.3.5 <code>getValidKey()</code>	11
4.2.3.6 <code>getValidOpenCipherText()</code>	12
5 Файлы	13
5.1 Файл <code>Error.h</code>	13
5.1.1 Подробное описание	14
5.2 Файл <code>modAlphaCipher.h</code>	14
5.2.1 Подробное описание	15
Предметный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

invalid_argument	
Error	7
modAlphaCipher	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Error

Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument` 7

modAlphaCipher

Класс, который предназначен для зашифрования и расшифрования сообщений методом Гронсвельда 8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Error.h	
Описание класса Error	13
modAlphaCipher.h	
Описание класса modAlphaCipher	14

Глава 4

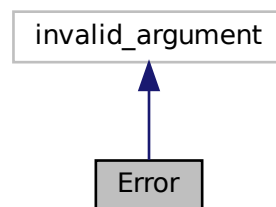
Классы

4.1 Класс Error

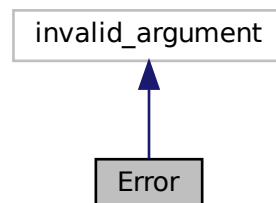
Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

```
#include <Error.h>
```

Граф наследования:Error:



Граф связей класса Error:



Открытые члены

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

4.1.1 Подробное описание

Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

4.1.2 Конструктор(ы)

4.1.2.1 Error()

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

Конструктор с параметром

Аргументы

what_arg	- строка, которая хранит описание ошибки
----------	------------------------------------------

Параметр конструктора перегружается конструктором базового класса с названием `invalid_↔argument`

Объявления и описания членов класса находятся в файле:

- [Error.h](#)

4.2 Класс modAlphaCipher

Класс, который предназначен для зашифрования и расшифрования сообщений методом Гронсфельда

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

Открытые члены

- [modAlphaCipher](#) ()=delete
Запрещающий конструктор без параметров
- [modAlphaCipher](#) (const wstring &skey)
Конструктор с параметром
- wstring [encrypt](#) (const wstring &open_text)
Метод для зашифрования русскоязычных сообщений методом Гронсфельда
- wstring [decrypt](#) (const wstring &cipher_text)
Метод для расшифрования русскоязычных сообщений методом Гронсфельда

Закрытые члены

- `vector< int > convert (const wstring &s)`
Преобразование строки типа `wstring` в вектор(чисел)
- `wstring convert (const vector< int > &v)`
Преобразование вектора (чисел) в строку типа `wstring`.
- `wstring getValidKey (const wstring &ws)`
Метод валидации ключа
- `wstring getValidOpenCipherText (const wstring &ws)`
Метод валидации строки для зашифрования или расшифрования

Закрытые данные

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

4.2.1 Подробное описание

Класс, который предназначен для зашифрования и расшифрования сообщений методом Гронсвельда

Работает только с русскоязычными сообщениями

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

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

Конструктор с параметром

Аргументы

skey	- ключ в виде строки типа <code>wstring</code>
------	------------------------------------------------

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 типа `wstring`.

Аргументы

<code>v</code>	- вектор(чисел) для преобразования в строку по индексам атрибута <code>numAlpha</code>
----------------	----------------------------------------------------------------------------------------

Возвращает

строка текста типа `wstring`

4.2.3.2 `convert()` [2/2]

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

Преобразование строки типа `wstring` в вектор(чисел)

Аргументы

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

Возвращает

вектор чисел

4.2.3.3 `decrypt()`

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

Метод для расшифрования русскоязычных сообщений методом Гронсфельда

Аргументы

<code>cipher_text</code>	- строка, которую нужно расшифровать
--------------------------	--------------------------------------

Возвращает

расшифрования строка типа wstring

4.2.3.4 encrypt()

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

Метод для зашифрования русскоязычных сообщений методом Гронсфельда

Аргументы

open_text	- строка, которую нужно зашифровать
-----------	-------------------------------------

Возвращает

зашифрованная строка типа wstring

4.2.3.5 getValidKey()

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

Метод валидации ключа

Аргументы

ws	- ключ в виде строки, который нужно проверить на наличие ошибок
----	-----------------------------------------------------------------

Письменные буквы алфавита в строке ключа переводятся в прописные

Исключения

Error , если	ключ является пустым или в нём присутствуют запрещающие символы
---------------------	-----------------------------------------------------------------

Возвращает

Ключ в виде строки типа wstring, который успешно прошёл валидацию

4.2.3.6 `getValidOpenCipherText()`

```
wstring modAlphaCipher::getValidOpenCipherText (  
    const wstring & ws )    [inline], [private]
```

Метод валидации строки для зашифрования или расшифрования

Аргументы

ws	- строка для шифрования или расшифрования, которую нужно проверить на наличие ошибок
----	--------------------------------------------------------------------------------------

Письменные буквы алфавита в строке переводятся в прописные

Исключения

Error , если	строка является пустой или в ней присутствуют запрещающие символы
------------------------------	-------------------------------------------------------------------

Возвращает

строка типа `wstring`, которая успешно прошла валидацию

Объявления и описания членов классов находятся в файлах:

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

Глава 5

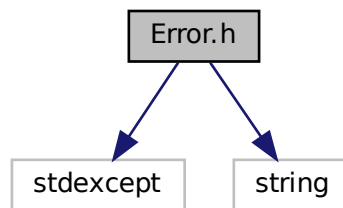
Файлы

5.1 Файл Error.h

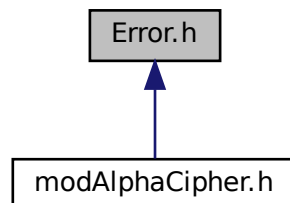
Описание класса [Error](#).

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

Граф включаемых заголовочных файлов для Error.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Error](#)

Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

5.1.1 Подробное описание

Описание класса [Error](#).

Автор

Мещеряков В.А.

Версия

1.0

Авторство

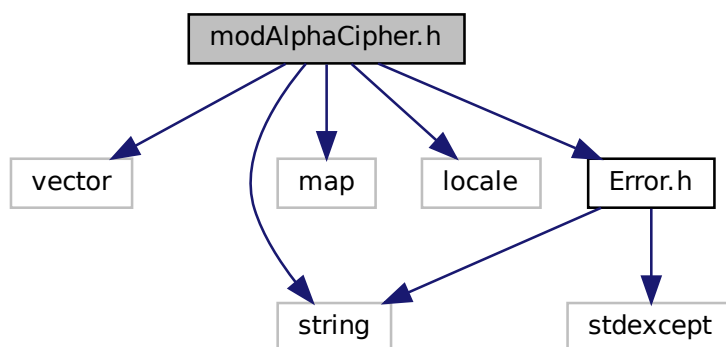
ИБСТ ПГУ

5.2 Файл `modAlphaCipher.h`

Описание класса [modAlphaCipher](#).

```
#include <vector>
#include <string>
#include <map>
#include <locale>
#include "Error.h"
```

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Классы

- class [modAlphaCipher](#)

Класс, который предназначен для зашифрования и расшифрования сообщений методом Гронс-вельда

5.2.1 Подробное описание

Описание класса [modAlphaCipher](#).

Автор

Мещеряков В.А.

Версия

1.0

Авторство

ИБСТ ПГУ

Предметный указатель

- convert
 - modAlphaCipher, [9](#), [10](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [11](#)
- Error, [7](#)
 - Error, [8](#)
- Error.h, [13](#)
- getValidKey
 - modAlphaCipher, [11](#)
- getValidOpenCipherText
 - modAlphaCipher, [11](#)
- modAlphaCipher, [8](#)
 - convert, [9](#), [10](#)
 - decrypt, [10](#)
 - encrypt, [11](#)
 - getValidKey, [11](#)
 - getValidOpenCipherText, [11](#)
 - modAlphaCipher, [9](#)
- modAlphaCipher.h, [14](#)