

Программа для шифрования/расшифрования методом маршрутной
перестановки

1.0

Создано системой Doxygen 1.9.1

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	1
2.1 Классы	1
3 Классы	2
3.1 Класс cipher_error	2
3.1.1 Подробное описание	2
3.1.2 Конструктор(ы)	3
3.2 Класс coder	3
3.2.1 Подробное описание	4
3.2.2 Конструктор(ы)	4
3.2.3 Методы	4
Предметный указатель	7

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

coder	3
std::invalid_argument	
cipher_error	2

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	
Класс вызова исключений	2
coder	
Класс для шифрования и расшифрования методом маршрутной перестановки	3

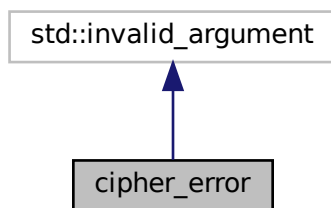
3 Классы

3.1 Класс cipher_error

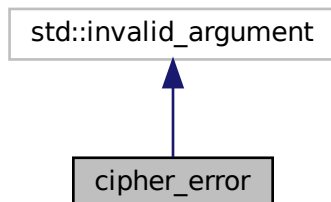
Класс вызова исключений

```
#include <coder.h>
```

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- [cipher_error](#) (const std::string &what_arg)
Обработка ошибки
- [cipher_error](#) (const char *what_arg)
Обработка ошибки

3.1.1 Подробное описание

Класс вызова исключений

3.1.2 Конструктор(ы)

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

Обработка ошибки

Аргументы

in	what_arg	Открытый текст на русском/английском языке
----	----------	--

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

Обработка ошибки

Аргументы

in	what_arg	Открытый текст на русском/английском языке
----	----------	--

Объявления и описания членов класса находятся в файле:

- `coder.h`

3.2 Класс coder

Класс для шифрования и расшифрования методом маршрутной перестановки

```
#include <coder.h>
```

Открытые члены

- `coder ()=delete`
Запрет конструктора без параметров
- `coder (std::wstring &ws_key)`
Конструктор класса
- `std::wstring encrypt (const std::wstring &open_text)`
Зашифрование
- `std::wstring decrypt (const std::wstring &cipher_text)`
Расшифрование

Закрытые члены

- `int getValidKey (std::wstring &ws_key)`
- `std::wstring getValidOpenText (const std::wstring &ws_open_text)`
Метод класса, проверяющий текст на валидность
- `std::wstring getValidCipherText (const std::wstring &ws_cipher_text)`
Метод класса, проверяющий расшифрованный текст на валидность

Закрытые данные

- `std::wstring_convert< std::codecvt_utf8< wchar_t >, wchar_t > codec`
- `int key1`

3.2.1 Подробное описание

Класс для шифрования и расшифрования методом маршрутной перестановки

Ключ Для шифрования - метод `encrypt` для расшифровки - метод `decrypt`

Предупреждения

Реализация для русского и английского языка!

3.2.2 Конструктор(ы)

3.2.2.1 `coder()` `coder::coder (`
`std::wstring & ws_key)`

Конструктор класса

Аргументы

in	ws_key	Ключ для шифрования
----	--------	---------------------

Предупреждения

Ключ должен быть целым числом

Исключения

Invalide	key, при неверном ключе
----------	-------------------------

3.2.3 Методы

3.2.3.1 `decrypt()` `wstring coder::decrypt (`
`const std::wstring & cipher_text)`

Расшифрование

Аргументы

in	cipher_text	Зашифрованный текст на русском/английском языке
----	-------------	---

Предупреждения

Текст не должен содержать цифры, специальные знаки.

Возвращает

result Расшифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	--

3.2.3.2 `encrypt()` `wstring coder::encrypt (`
`const std::wstring & open_text)`

Зашифрование

Аргументы

in	open_text	Открытый текст на русском/английском языке
----	-----------	--

Предупреждения

Текст не должен содержать цифры, специальные знаки.

Возвращает

result Зашифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	--

3.2.3.3 `getValidCipherText()` `std::wstring coder::getValidCipherText (`
`const std::wstring & ws_cipher_text)` `[inline], [private]`

Метод класса, проверяющий расшифрованный текст на валидность

Аргументы

in	s	Зашифрованный текст на русском/английском языке
----	---	---

Возвращает

result

3.2.3.4 `getValidOpenText()` `std::wstring coder::getValidOpenText (`
`const std::wstring & ws_open_text)` `[inline], [private]`

Метод класса, проверяющий текст на валидность

Аргументы

in	s	Текст на русском/английском языке
----	---	-----------------------------------

Возвращает

result

Объявления и описания членов классов находятся в файлах:

- `coder.h`
- `coder.cpp`

Предметный указатель

- cipher_error, [2](#)
 - cipher_error, [3](#)
- coder, [3](#)
 - coder, [4](#)
 - decrypt, [4](#)
 - encrypt, [5](#)
 - getValidCipherText, [5](#)
 - getValidOpenText, [6](#)
- decrypt
 - coder, [4](#)
- encrypt
 - coder, [5](#)
- getValidCipherText
 - coder, [5](#)
- getValidOpenText
 - coder, [6](#)