

Шифрование методом табличной маршрутной перестановки.

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 CIPHER . . . . .	2
4.1.1 Подробное описание . . . . .	3
4.1.2 Конструктор(ы) . . . . .	3
4.1.3 Методы . . . . .	3
4.2 Класс cipher_error . . . . .	6
4.2.1 Конструктор(ы) . . . . .	7
5 Файлы	8
5.1 Файл Routechange.cpp . . . . .	8
5.1.1 Подробное описание . . . . .	8
5.2 Файл Routechange.h . . . . .	8
5.2.1 Подробное описание . . . . .	9
Предметный указатель	11

## 1 Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Cipher	2
invalid_argument	
cipher_error	6

## 2 Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

Cipher	
Описание класса Cipher	2
cipher_error	6

## 3 Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">Routechange.cpp</a>	
Описание класса <a href="#">cipher_error</a>	8
<a href="#">Routechange.h</a>	
Шифрование методом табличной перестановки	8

## 4 Классы

### 4.1 Класс Cipher

Описание класса [Cipher](#).

```
#include <Routechange.h>
```

Открытые члены

- [Cipher](#) ()=delete  
Запрещающий конструктор без параметров
- [Cipher](#) (std::wstring &ws\_key)  
Конструктор принимает ключ
- std::wstring [encrypt](#) (std::wstring &ws\_open\_text)  
Метод использующийся для зашифрования
- std::wstring [decrypt](#) (const std::wstring &ws\_cipher\_text)  
Метод использующийся для расшифрования
- void [set\\_text](#) (const std::wstring &ws\_text)  
Формирует информацию о таблице
- void [set\\_key](#) (std::wstring &ws\_key)  
Установка нового ключа
- int [getValidKey](#) (std::wstring &ws\_key)  
Проверка на правильность ключа
- std::wstring [getValidOpenText](#) (const std::wstring &ws\_open\_text)  
Проверка на правильность текста для зашифровки
- std::wstring [getValidCipherText](#) (const std::wstring &ws\_cipher\_text)  
Проверка на правильность текста для расшифровки

Закрытые данные

- std::wstring\_convert< std::codecvt\_utf8< wchar\_t >, wchar\_t > [codec](#)  
для преобразования в широкий формат строки и обратно
- int [pillar](#)  
Ключ, обозначающий количество столбцов в таблице
- int [rad](#)  
Количество строк в таблице
- int [len\\_text](#)  
Количество символов в строке

#### 4.1.1 Подробное описание

Описание класса [Cipher](#).

Ключ устанавливается в конструкторе, Для зашифрования и расшифрования предназначены методы `encrypt` и `decrypt`.

Предупреждения

Реализация производится только для русского языка. С использованием `wstring`.

#### 4.1.2 Конструктор(ы)

4.1.2.1 `Cipher()` `Cipher::Cipher (`  
`std::wstring & ws_key )`

Конструктор принимает ключ

Конструктор, принимающий на вход ключ, который устанавливает кол-во столбцов

Аргументы

<code>ws_key</code>	
---------------------	--

Возвращает

Ничего не возвращает

#### 4.1.3 Методы

4.1.3.1 `decrypt()` `std::wstring Cipher::decrypt (`  
`const std::wstring & cipher_text )`

Метод использующийся для расшифрования

Метод `decrypt` расшифровывает текст.

Аргументы

<code>cipher_text</code>	
--------------------------	--

Возвращает

Возвращает расшифрованный текст.

4.1.3.2 `encrypt()` `std::wstring Cipher::encrypt (`  
`std::wstring & open_text )`

Метод использующийся для зашифрования

Метод `encrypt` зашифровывает принятый текст.

Аргументы

<code>open_text</code>	
------------------------	--

Возвращает

Возвращает зашифрованный текст

4.1.3.3 `getValidCipherText()` `std::wstring Cipher::getValidCipherText (`  
`const std::wstring & ws_cipher_text )` `[inline]`

Проверка на правильность текста для расшифровки

Данный метод проверяет зашифрованный текст на правильность.

Аргументы

<code>ws_cipher_text</code>	
-----------------------------	--

Возвращает

Зашифрованный текст

Исключения

<code>cipher_error</code> ,если	текст пустой или невалидный
---------------------------------	-----------------------------

4.1.3.4 `getValidKey()` `int Cipher::getValidKey (`  
`std::wstring & ws_key )` `[inline]`

Проверка на правильность ключа

Метод проверяющий ключ на правильность.

Аргументы

ws_key	
--------	--

Возвращает

Ключ

Исключения

<a href="#">cipher_error</a> ,если	ключ пустой или невалидный
------------------------------------	----------------------------

4.1.3.5 `getValidOpenText()` `std::wstring Cipher::getValidOpenText (`  
`const std::wstring & ws_open_text )` `[inline]`

Проверка на правильность текста для зашифровки

Данный метод проверяет принятый текст на правильность. В данном методе строчные буквы превращаются в прописные. Когда встречаются знаки, цифры и пробелы они удаляются.

Аргументы

ws_open_text	
--------------	--

Возвращает

Текст для расшифрования

Исключения

<a href="#">cipher_error</a> ,если	текст пустой
------------------------------------	--------------

4.1.3.6 `set_key()` `void Cipher::set_key (`  
`std::wstring & ws_key )`

Установка нового ключа

Метод, принимающий на вход ключ, который устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

4.1.3.7 `set_text()` `void Cipher::set_text (`  
`const std::wstring & open_text )`

Формирует информацию о таблице

Метод, принимающий на вход текст для зашифровки

Данный метод принимает на вход длину текста и формируется кол-во строк в таблице.

Аргументы

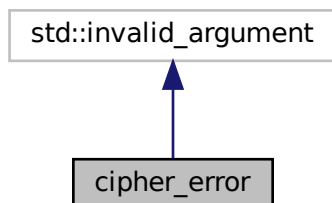
<code>open_text</code>	
------------------------	--

Объявления и описания членов классов находятся в файлах:

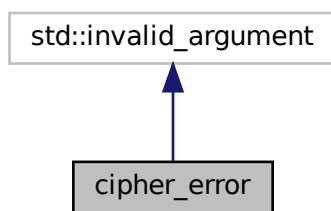
- [Routechange.h](#)
- [Routechange.cpp](#)

## 4.2 Класс `cipher_error`

Граф наследования:`cipher_error`:



Граф связей класса cipher\_error:



Открытые члены

- `cipher_error` (`const std::string &what_arg`)  
Принимает на вход строку, вызывает исключение
- `cipher_error` (`const char *what_arg`)  
Принимает на вход си строку, вызывает исключение

#### 4.2.1 Конструктор(ы)

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

Принимает на вход строку, вызывает исключение

Аргументы

what_arg	
----------	--

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

Принимает на вход си строку, вызывает исключение

Аргументы

what_arg	
----------	--

Объявления и описания членов класса находятся в файле:



- [Routechange.h](#)

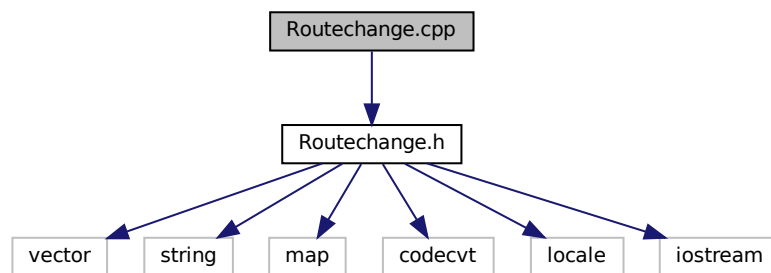
## 5 Файлы

### 5.1 Файл Routechange.cpp

Описание класса [cipher\\_error](#).

```
#include "Routechange.h"
```

Граф включаемых заголовочных файлов для Routechange.cpp:



#### 5.1.1 Подробное описание

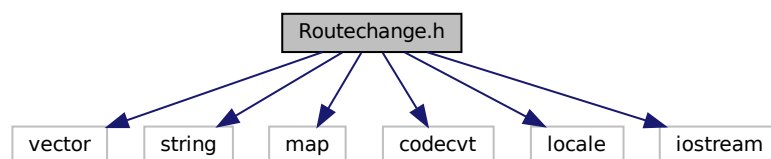
Описание класса [cipher\\_error](#).

### 5.2 Файл Routechange.h

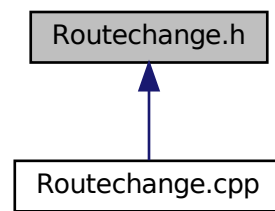
Шифрование методом табличной перестановки

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

Граф включаемых заголовочных файлов для Routechange.h:



Граф файлов, в которые включается этот файл:



### Классы

- class [Cipher](#)  
Описание класса [Cipher](#).
- class [cipher\\_error](#)

#### 5.2.1 Подробное описание

### Шифрование методом табличной перестановки

#### Автор

Иванщина А.К.

#### Версия

1.0

#### Дата

21.05.2021

#### Авторство

ИБСТ ПГУ



## Предметный указатель

- Cipher, [2](#)
  - Cipher, [3](#)
  - decrypt, [3](#)
  - encrypt, [4](#)
  - getValidCipherText, [4](#)
  - getValidKey, [4](#)
  - getValidOpenText, [5](#)
  - set\_key, [5](#)
  - set\_text, [6](#)
- cipher\_error, [6](#)
  - cipher\_error, [7](#)
- decrypt
  - Cipher, [3](#)
- encrypt
  - Cipher, [4](#)
- getValidCipherText
  - Cipher, [4](#)
- getValidKey
  - Cipher, [4](#)
- getValidOpenText
  - Cipher, [5](#)
- Routechange.cpp, [8](#)
- Routechange.h, [8](#)
- set\_key
  - Cipher, [5](#)
- set\_text
  - Cipher, [6](#)