

Шифр табличной маршрутной перестановки

Создано системой Doxygen 1.9.4



---

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
4.1.2 Конструктор(ы) . . . . .	8
4.1.2.1 cipher_error() [1/2] . . . . .	8
4.1.2.2 cipher_error() [2/2] . . . . .	8
4.2 Класс TableCipher . . . . .	9
4.2.1 Подробное описание . . . . .	9
4.2.2 Конструктор(ы) . . . . .	9
4.2.2.1 TableCipher() . . . . .	9
4.2.3 Методы . . . . .	10
4.2.3.1 decrypt() . . . . .	10
4.2.3.2 encrypt() . . . . .	10
4.2.3.3 getValidKey() . . . . .	11
4.2.3.4 getValidText() . . . . .	11
5 Файлы	13
5.1 Файл tableCipher.h . . . . .	13
5.1.1 Подробное описание . . . . .	14
5.2 tableCipher.h . . . . .	14
Предметный указатель	15



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
TableCipher . . . . .	9



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	
Обработка исключений	7
<a href="#">TableCipher</a>	
Шифрование методом маршрутной перестановки	9





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

[tableCipher.h](#)

Заголовочный файл для модуля маршрутной перестановки . . . . . 13



## Глава 4

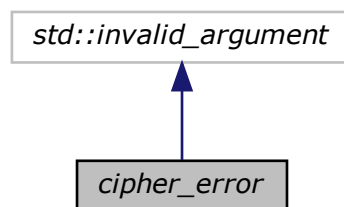
# Классы

### 4.1 Класс `cipher_error`

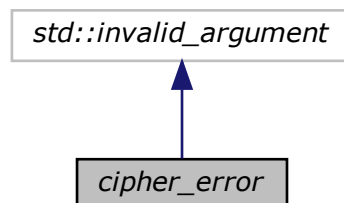
Обработка исключений.

```
#include <tableCipher.h>
```

Граф наследования:`cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- `cipher_error` (`const std::string &what_arg`)  
Конструктор с параметром типа `const std::string`.
- `cipher_error` (`const char *what_arg`)  
Конструктор с параметром типа `const char`.

### 4.1.1 Подробное описание

Обработка исключений.

Класс `cipher_error` наследуется от `std::invalid_argument` и используется для обработки ошибок, возникающих при некорректных входных данных (например, невалидный ключ или текст).

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 `cipher_error()` [1/2]

```

cipher_error::cipher_error (
    const std::string & what_arg )  [inline], [explicit]

```

Конструктор с параметром типа `const std::string`.

Создает объект исключения с указанным сообщением об ошибке.

Аргументы

<code>what_arg</code>	Сообщение об ошибке.
-----------------------	----------------------

#### 4.1.2.2 `cipher_error()` [2/2]

```

cipher_error::cipher_error (
    const char * what_arg )  [inline], [explicit]

```

Конструктор с параметром типа `const char`.

Создает объект исключения с указанным сообщением об ошибке.

Аргументы

<code>what_arg</code>	Сообщение об ошибке.
-----------------------	----------------------

Объявления и описания членов класса находятся в файле:

- [tableCipher.h](#)

## 4.2 Класс TableCipher

Шифрование методом маршрутной перестановки

```
#include <tableCipher.h>
```

Открытые члены

- [TableCipher](#) (const int &key)  
Конструктор с параметрами
- std::wstring [encrypt](#) (const std::wstring &encrypt\_text)  
Метод для шифрования текста.
- std::wstring [decrypt](#) (const std::wstring &decrypt\_text)  
Метод для расшифровки текста.

Закрытые члены

- int [getValidKey](#) (const int &key)  
Проверка ключа на валидность.
- std::wstring [getValidText](#) (const std::wstring &text)  
Проверка текста на валидность.

Закрытые данные

- int stolb  
Ключ, используемый для зашифрования и расшифрования

### 4.2.1 Подробное описание

Шифрование методом маршрутной перестановки

Класс [TableCipher](#) предоставляет методы для зашифрования и расшифрования текста с использованием табличного шифра. Ключ для зашифрования и расшифрования должен быть положительным целым числом.

### 4.2.2 Конструктор(ы)

#### 4.2.2.1 TableCipher()

```
TableCipher::TableCipher (  
    const int & key )
```

Конструктор с параметрами

Создает объект [TableCipher](#) с указанным ключом.

## Аргументы

key	Ключ для шифрования и дешифрования.
-----	-------------------------------------

## Исключения

<a href="#">cipher_error</a>	Если ключ не является положительным целым числом.
------------------------------	---

## 4.2.3 Методы

## 4.2.3.1 decrypt()

```
std::wstring TableCipher::decrypt (  
    const std::wstring & decrypt_text )
```

Метод для расшифровки текста.

Расшифровывает текст с использованием табличного шифра и ключа, указанного при создании объекта.

## Аргументы

decrypt_text	Текст, который нужно расшифровать.
--------------	------------------------------------

## Возвращает

Расшифрованный текст.

## Исключения

<a href="#">cipher_error</a>	Если текст содержит недопустимые символы.
------------------------------	---

## 4.2.3.2 encrypt()

```
std::wstring TableCipher::encrypt (  
    const std::wstring & encrypt_text )
```

Метод для шифрования текста.

Шифрует текст с использованием табличного шифра и ключа, указанного при создании объекта.

## Аргументы

encrypt_text	Текст, который нужно зашифровать.
--------------	-----------------------------------

## Возвращает

Зашифрованный текст.

## Исключения

<a href="#">cipher_error</a>	Если текст содержит недопустимые символы.
------------------------------	---

## 4.2.3.3 getValidKey()

```
int TableCipher::getValidKey (  
    const int & key ) [private]
```

Проверка ключа на валидность.

Метод проверяет, является ли ключ положительным целым числом.

## Аргументы

key	Ключ, который нужно проверить.
-----	--------------------------------

## Возвращает

Валидный ключ.

## Исключения

<a href="#">cipher_error</a>	Если ключ не является положительным целым числом.
------------------------------	---

## 4.2.3.4 getValidText()

```
std::wstring TableCipher::getValidText (  
    const std::wstring & text ) [private]
```

Проверка текста на валидность.

Метод проверяет, содержит ли текст только буквы алфавита.

## Аргументы

text	Текст, который нужно проверить.
------	---------------------------------

## Возвращает

Валидный текст.

## Исключения

<a href="#">cipher_error</a>	Если текст содержит недопустимые символы.
------------------------------	---

Объявления и описания членов классов находятся в файлах:

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



## Глава 5

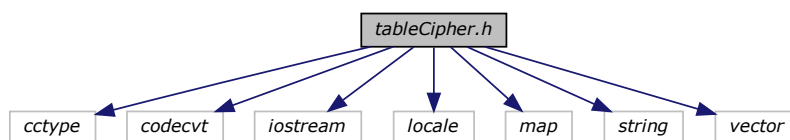
# Файлы

### 5.1 Файл tableCipher.h

Заголовочный файл для модуля маршрутной перестановки

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

Граф включаемых заголовочных файлов для tableCipher.h:



## Классы

- class [TableCipher](#)  
Шифрование методом маршрутной перестановки
- class [cipher\\_error](#)  
Обработка исключений.

### 5.1.1 Подробное описание

Заголовочный файл для модуля маршрутной перестановки

Автор

Ермаков Н.А.

Версия

1.0

Дата

19.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

## 5.2 tableCipher.h

[См. документацию.](#)

```
1
9 #pragma once
10 #include <cctype>
11 #include <codecvt>
12 #include <iostream>
13 #include <locale>
14 #include <map>
15 #include <string>
16 #include <vector>
24 class TableCipher
25 {
26 private:
27     int stolb;
28     int getValidKey(const int& key); // проверка ключа на валидности
29     std::wstring getValidText(const std::wstring& text); // проверка текста на валидности
30 public:
31     TableCipher() = delete; // < Запрет на конструктор по умолчанию.
32     TableCipher(const int& key);
33     std::wstring encrypt(const std::wstring& encrypt_text);
34     std::wstring decrypt(const std::wstring& decrypt_text);
35 };
36 class cipher_error : public std::invalid_argument
37 {
38 public:
39     explicit cipher_error(const std::string& what_arg):
40         std::invalid_argument(what_arg) {}
41     explicit cipher_error(const char* what_arg):
42         std::invalid_argument(what_arg) {}
43 };
```

# Предметный указатель

- cipher\_error, [7](#)
  - cipher\_error, [8](#)
- decrypt
  - TableCipher, [10](#)
- encrypt
  - TableCipher, [10](#)
- getValidKey
  - TableCipher, [11](#)
- getValidText
  - TableCipher, [11](#)
- TableCipher, [9](#)
  - decrypt, [10](#)
  - encrypt, [10](#)
  - getValidKey, [11](#)
  - getValidText, [11](#)
  - TableCipher, [9](#)
- tableCipher.h, [13](#)