

Шифр маршрутной перестановки

1

Создано системой Doxygen 1.9.1



---

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_errgr . . . . .	7
4.1.1 Подробное описание . . . . .	8
4.2 Класс Table . . . . .	8
4.2.1 Подробное описание . . . . .	9
4.2.2 Конструктор(ы) . . . . .	9
4.2.2.1 Table() . . . . .	9
4.2.3 Методы . . . . .	10
4.2.3.1 decrypt() . . . . .	10
4.2.3.2 encrypt() . . . . .	10
4.2.3.3 is_low_rus() . . . . .	11
4.2.3.4 is_rus() . . . . .	11
4.2.3.5 toValidtext() . . . . .	11
4.2.4 Данные класса . . . . .	12
4.2.4.1 lnumAlpha . . . . .	12
4.2.4.2 numAlpha . . . . .	12
5 Файлы	13
5.1 Файл table.cpp . . . . .	13
5.1.1 Подробное описание . . . . .	13
5.2 Файл table.h . . . . .	14
5.2.1 Подробное описание . . . . .	14
Предметный указатель	17



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
Table . . . . .	8



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	Пользовательский класс исключений . . . . .	<a href="#">7</a>
<a href="#">Table</a>	Шифрование методом маршрутной перестановки . . . . .	<a href="#">8</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">table.cpp</a>	СРР файл для модуля шифрования методом маршрутной перестановки . . . . .	<a href="#">13</a>
<a href="#">table.h</a>	Заголовочный файл для модуля шифрования методом перестановки . . . . .	<a href="#">14</a>



## Глава 4

# Классы

### 4.1 Класс `cipher_error`

Пользовательский класс исключений

```
#include <table.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

### 4.1.1 Подробное описание

#### Пользовательский класс исключений

Класс-наследник класса `invalid_argument`. Создан для того чтобы специально отслеживать исключения, возбуждаемые в процессе шифрования

Объявления и описания членов класса находятся в файле:

- [table.h](#)

## 4.2 Класс Table

#### Шифрование методом маршрутной перестановки

```
#include <table.h>
```

#### Открытые члены

- `Table ()=delete`  
Запрет конструктора без параметров
- `Table (const int skey)`  
Конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`  
Функция зашифрования
- `std::wstring decrypt (const std::wstring &cipher_text)`  
Функция расшифрования

#### Закрытые члены

- `bool is_rus (wchar_t wc)`  
Проверка символа на принадлежность к русскому алфавиту
- `int is_low_rus (wchar_t wch)`  
Проверка символа на принадлежность к строчным буквам р. алфавита
- `std::wstring toValidtext (const std::wstring &s)`  
Метод валидации строки

## Закрытые данные

- `std::wstring numAlpha`  
Русский алфавит заглавными буквами
- `std::wstring lnumAlpha`  
Русский алфавит заглавными буквами
- `uint key`  
ключ
- `std::map< int, std::wstring > subNum`  
словарь "субстрока по номеру" (для зашифровки)
- `std::map< int, std::wstring > resubNum`  
словарь "субстрока по номеру" (для расшифровки)

### 4.2.1 Подробное описание

#### Шифрование методом маршрутной перестановки

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

#### Предупреждения

Используется только для русского текста

### 4.2.2 Конструктор(ы)

#### 4.2.2.1 Table()

```
Table::Table (
    const int skey )
```

Конструктор для установки ключа

#### Шифрование методом маршрутной перестановки

Записывает в переменную класса ключ в виде цифры

#### Аргументы

in	skey	Целочисленный ключ
----	------	--------------------

#### Исключения

<code>cipher_error</code> , если	ключ меньше, либо равен единице
----------------------------------	---------------------------------

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены

методы encrypt и decrypt.

Предупреждения

Используется только для русского текста

### 4.2.3 Методы

#### 4.2.3.1 decrypt()

```
std::wstring Table::decrypt (  
    const std::wstring & cipher_text )
```

Функция расшифрования

Расшифровывает строку по методу маршрутной перестановки

Аргументы

in	cipher_text	Строка с зашифрованным текстом
----	-------------	--------------------------------

Исключения

<a href="#">cipher_error</a> ,если	возбуждается исключение в методе валидации toValidtext
------------------------------------	--

#### 4.2.3.2 encrypt()

```
std::wstring Table::encrypt (  
    const std::wstring & open_text )
```

Функция зашифрования

Шифрует строку методом маршрутной перестановки

Аргументы

in	open_text	Строка с текстом для зашифрования
----	-----------	-----------------------------------

Исключения

<a href="#">cipher_error</a> ,если	возбуждается исключение в методе валидации toValidtext
------------------------------------	--

## 4.2.3.3 is\_low\_rus()

```
int Table::is_low_rus (
    wchar_t wch ) [private]
```

Проверка символа на принадлежность к строчным буквам р. алфавита

Находит позицию, на которой стоит в русском алфавите указанный символ

Аргументы

in	wch	Символ wchar
----	-----	--------------

Возвращает

Позиция символа

## 4.2.3.4 is\_rus()

```
bool Table::is_rus (
    wchar_t wc ) [private]
```

Проверка символа на принадлежность к русскому алфавиту

Аргументы

in	wc	Символ wchar
----	----	--------------

Возвращает

Да или нет

## 4.2.3.5 toValidtext()

```
std::wstring Table::toValidtext (
    const std::wstring & s ) [private]
```

Метод валидации строки

Может преобразовывать строчные буквы в заглавные

Аргументы

in	s	Строка текста. Не может быть пустой. Также в ней не должны присутствовать пробелы, цифры и знаки пунктуации
----	---	---

Возвращает

Валидированная строка

Исключения

<a href="#">cipher_error</a> , если	строка не соответствует требованиям, описанным выше
-------------------------------------	---

#### 4.2.4 Данные класса

##### 4.2.4.1 lnumAlpha

```
std::wstring Table::lnumAlpha [private]
```

Инициализатор

```
= L"абвгдеёжзийклмнопрстуфхцчшщъыьэюя"
```

Русский алфавит заглавными буквами

##### 4.2.4.2 numAlpha

```
std::wstring Table::numAlpha [private]
```

Инициализатор

```
= L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
```

Русский алфавит заглавными буквами

Объявления и описания членов классов находятся в файлах:

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



## Глава 5

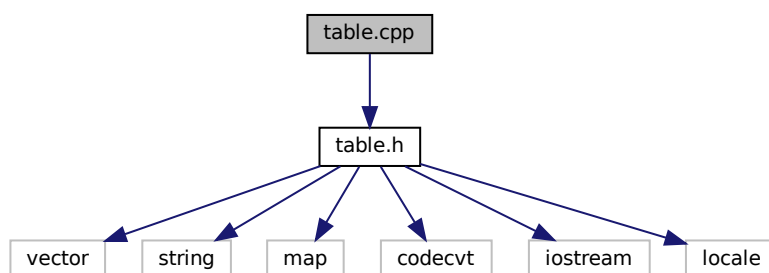
# Файлы

### 5.1 Файл table.cpp

CPP файл для модуля шифрования методом маршрутной перестановки

```
#include "table.h"
```

Граф включаемых заголовочных файлов для table.cpp:



#### 5.1.1 Подробное описание

CPP файл для модуля шифрования методом маршрутной перестановки

Автор

Гастиев Артур

Версия

1.0

Дата

13.12.2023

Предупреждения

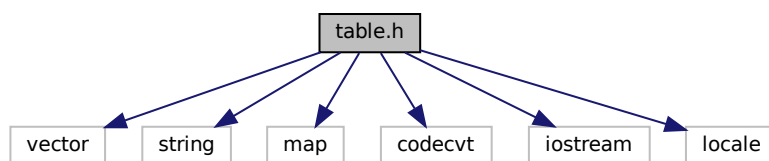
Тренировочная работа

## 5.2 Файл table.h

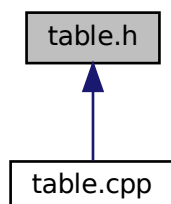
Заголовочный файл для модуля шифрования методом перестановки

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

Граф включаемых заголовочных файлов для table.h:



Граф файлов, в которые включается этот файл:



### Классы

- class [Table](#)  
Шифрование методом маршрутной перестановки
- class [cipher\\_error](#)  
Пользовательский класс исключений

### 5.2.1 Подробное описание

Заголовочный файл для модуля шифрования методом перестановки

Автор

Гастиев Артур

Версия

1.0

Дата

13.12.2023

Предупреждения

Тренировочная работа



# Предметный указатель

cipher\_error, [7](#)

decrypt  
Table, [10](#)

encrypt  
Table, [10](#)

is\_low\_rus  
Table, [10](#)

is\_rus  
Table, [11](#)

lnumAlpha  
Table, [12](#)

numAlpha  
Table, [12](#)

Table, [8](#)  
decrypt, [10](#)  
encrypt, [10](#)  
is\_low\_rus, [10](#)  
is\_rus, [11](#)  
lnumAlpha, [12](#)  
numAlpha, [12](#)  
Table, [9](#)  
toValidtext, [11](#)  
table.cpp, [13](#)  
table.h, [14](#)  
toValidtext  
Table, [11](#)