

## Table Cipher

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 . . . . .	8
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 getValidCipherText() . . . . .	11
4.2.3.4 getValidKey() . . . . .	11
4.2.3.5 getValidOpenText() . . . . .	12
5 Файлы	13
5.1 Файл modTableCipher.h . . . . .	13
5.1.1 Подробное описание . . . . .	14
5.2 modTableCipher.h . . . . .	14
Предметный указатель	15



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
TableCipher . . . . .	8



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	Исключение для ошибок в классе modAlphaCipher . . . . .	7
<a href="#">TableCipher</a>	Шифрование методом табличной маршрутной перестановки . . . . .	8





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">modTableCipher.h</a>	
Описание класса <a href="#">TableCipher</a>	13



## Глава 4

# Классы

### 4.1 Класс `cipher_error`

Исключение для ошибок в классе `modAlphaCipher`.

```
#include <modTableCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- [cipher\\_error](#) (const std::string &what\_arg)  
Конструктор исключения с сообщением об.
- [cipher\\_error](#) (const char \*what\_arg)  
Конструктор исключения с сообщением об.

### 4.1.1 Подробное описание

Исключение для ошибок в классе modAlphaCipher.

### 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 исключения с сообщением об.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

#### 4.1.2.2 cipher\_error() [2/2]

```

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

```

Конструктор исключения с сообщением об.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

Объявления и описания членов класса находятся в файле:

- [modTableCipher.h](#)

## 4.2 Класс TableCipher

Шифрование методом табличной маршрутной перестановки

```
#include <modTableCipher.h>
```

### Открытые члены

- TableCipher ()=delete  
Конструктор по умолчанию запрещен.
- TableCipher (const std::wstring &key\_str)  
Конструктор класса TableCipher.
- std::wstring encrypt (const std::wstring &text)  
Метод шифрования текста с помощью табличной маршрутной перестановки.
- std::wstring decrypt (const std::wstring &encrypted\_text)  
Расшифровывает текст, зашифрованный с помощью табличной маршрутной перестановки.
- int getValidKey (const std::wstring &key\_str)  
Валидация ключа.
- std::wstring getValidOpenText (const std::wstring &s)  
Валидация открытого текста.
- std::wstring getValidCipherText (const std::wstring &s)  
Валидация зашифрованного текста.

### Закрытые данные

- int key  
Ключ для шифрования (Количество столбцов таблицы)

#### 4.2.1 Подробное описание

##### Шифрование методом табличной маршрутной перестановки

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

##### Предупреждения

Реализация только для русского языка

Класс, реализующий табличный шифр. Класс реализует шифр табличной маршрутной перестановки.

#### 4.2.2 Конструктор(ы)

##### 4.2.2.1 TableCipher()

```
TableCipher::TableCipher (  
    const std::wstring & key_str )
```

Конструктор класса TableCipher.

Создает объект TableCipher с заданным ключом. Ключ количество столбцов.

Аргументы

key_str	Строка, содержащая ключ шифра.
---------	--------------------------------

### 4.2.3 Методы

#### 4.2.3.1 decrypt()

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

Расшифровывает текст, зашифрованный с помощью табличной маршрутной перестановки.

Аргументы

encrypted_text	Текст для расшифровки.
----------------	------------------------

Возвращает

Расшифрованный текст.

#### 4.2.3.2 encrypt()

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

Метод шифрования текста с помощью табличной маршрутной перестановки.

Записывает текст в таблицу на основе полученного ключа и вычисленного значения количества строк таблицы. Маршрут записи: по горизонтали слева направо, сверху вниз. Маршрут считывания: сверху вниз, справа налево.

Аргументы

text	Открытый текст для шифрования.
------	--------------------------------

Возвращает

Зашифрованный текст.

## 4.2.3.3 getValidCipherText()

```
std::wstring TableCipher::getValidCipherText (
    const std::wstring & s ) [inline]
```

Валидация зашифрованного текста.

Функция проверяет, что строка не пуста и содержит только символы прописных букв и звездочки (\*).

Аргументы

s	Входная строка зашифрованного текста.
---	---------------------------------------

Возвращает

Строка зашифрованного текста, если она корректна.

Исключения

<a href="#">cipher_error</a>	Если строка пуста или содержит недопустимые символы.
------------------------------	--

## 4.2.3.4 getValidKey()

```
int TableCipher::getValidKey (
    const std::wstring & key_str )
```

Валидация ключа.

Функция проверяет, что строка ключа не пуста и содержит только цифры. Если строка валидна, она преобразуется в целое число. В случае ошибки выбрасывается исключение.

Аргументы

key_str	Строка, содержащая ключ.
---------	--------------------------

Возвращает

Целое число, представляющее ключ.

Исключения

<a href="#">cipher_error</a>	Если ключ пустой, содержит нецифровые символы или <code>key &lt;= 0</code> .
------------------------------	--

#### 4.2.3.5 getValidOpenText()

```
std::wstring TableCipher::getValidOpenText (
    const std::wstring & s ) [inline]
```

Валидация открытого текста.

Функция обрабатывает входную строку, оставляя только буквы и пробелы. Буквы приводятся к строчному формату. Если после обработки строка пуста, выбрасывается исключение.

Аргументы

s	Входная строка открытого текста.
---	----------------------------------

Возвращает

Очищенная строка, содержащая только буквы верхнего регистра и пробелы.

Исключения

<a href="#">cipher_error</a>	Если после обработки строка оказывается пустой.
------------------------------	---

Объявления и описания членов классов находятся в файлах:

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



## Глава 5

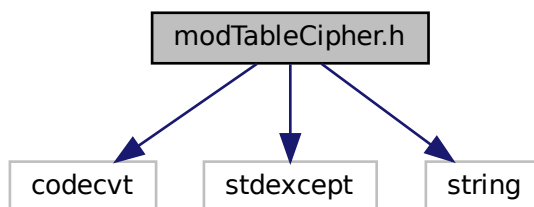
### Файлы

#### 5.1 Файл modTableCipher.h

Описание класса [TableCipher](#).

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

Граф включаемых заголовочных файлов для modTableCipher.h:



#### Классы

- class [TableCipher](#)  
Шифрование методом табличной маршрутной перестановки
- class [cipher\\_error](#)  
Исключение для ошибок в классе modAlphaCipher.

### 5.1.1 Подробное описание

Описание класса [TableCipher](#).

Автор

Грачев В.В.

Версия

1.0

Дата

18.11.2023

## 5.2 modTableCipher.h

[См. документацию.](#)

```
1
9 #pragma once
10 #include <codecvt>
11 #include <stdexcept>
12 #include <string>
13
25 class TableCipher
26 {
27 private:
28     int key;
29 public:
30     TableCipher() = delete;
31
39     TableCipher(const std::wstring& key_str);
40
51     std::wstring encrypt(const std::wstring& text);
52
58     std::wstring decrypt(const std::wstring& encrypted_text);
59
70     int getValidKey(const std::wstring& key_str);
71
82     std::wstring getValidOpenText(const std::wstring& s);
83
93     std::wstring getValidCipherText(const std::wstring& s);
94 };
95
100 class cipher_error : public std::invalid_argument
101 {
102 public:
107     explicit cipher_error(const std::string& what_arg)
108         : std::invalid_argument(what_arg)
109     {
110     }
111
116     explicit cipher_error(const char* what_arg)
117         : std::invalid_argument(what_arg)
118     {
119     }
120 };
```

# Предметный указатель

`cipher_error`, [7](#)  
    `cipher_error`, [8](#)

`decrypt`  
    `TableCipher`, [10](#)

`encrypt`  
    `TableCipher`, [10](#)

`getValidCipherText`  
    `TableCipher`, [10](#)

`getValidKey`  
    `TableCipher`, [11](#)

`getValidOpenText`  
    `TableCipher`, [11](#)

`modTableCipher.h`, [13](#)

`TableCipher`, [8](#)  
    `decrypt`, [10](#)  
    `encrypt`, [10](#)  
    `getValidCipherText`, [10](#)  
    `getValidKey`, [11](#)  
    `getValidOpenText`, [11](#)  
    `TableCipher`, [9](#)