

Table\_cipher

1.0

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 table_cipher . . . . .	7
4.1.1 Подробное описание . . . . .	8
4.1.2 Методы . . . . .	8
4.1.2.1 decrypt() . . . . .	8
4.1.2.2 encrypt() . . . . .	8
4.1.2.3 key_validation() . . . . .	8
4.1.2.4 text_validation() . . . . .	9
4.1.2.5 write_str_to_v() [1/2] . . . . .	9
4.1.2.6 write_str_to_v() [2/2] . . . . .	10
4.2 Класс table_error . . . . .	10
4.2.1 Подробное описание . . . . .	11
4.3 Класс UI . . . . .	12
5 Файлы	13
5.1 Файл main.cpp . . . . .	13
5.1.1 Подробное описание . . . . .	14
5.2 Файл table_cipher.cpp . . . . .	14
5.2.1 Подробное описание . . . . .	15
5.3 Файл table_cipher.h . . . . .	15
5.3.1 Подробное описание . . . . .	16
Предметный указатель	17



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
table_error . . . . .	10
table_cipher . . . . .	7
UI . . . . .	12



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">table_cipher</a>	Шифрование методом табличной маршрутной перестановки . . . . .	<a href="#">7</a>
<a href="#">table_error</a>	Класс ошибок для модуля Table_cipher . . . . .	<a href="#">10</a>
<a href="#">UI</a>	. . . . .	<a href="#">12</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">main.cpp</a>	Главный файл для модуля Table_cipher . . . . .	13
<a href="#">table_cipher.cpp</a>	Исполняемый файл для модуля Table_cipher . . . . .	14
<a href="#">table_cipher.h</a>	Заголовочный файл для модуля Table_cipher . . . . .	15



## Глава 4

# Классы

### 4.1 Класс `table_cipher`

Шифрование методом табличной маршрутной перестановки

```
#include <table_cipher.h>
```

Открытые члены

- `std::string encrypt ()`  
Шифрование
- `std::string decrypt ()`  
Расшифрование
- `void text_validation (std::string &text)`  
Обработка ошибок в строке для операции
- `void key_validation (uint &key, std::string &text)`  
Обработка ошибок ключа
- `table_cipher (std::string text, double k, std::string &initial_value)`  
Конструктор для установки обрабатываемой строки, ключа и начального значения двумерного вектора

Закрытые члены

- `void write_str_to_v (std::string st, std::vector< std::vector< std::string >> &v)`  
Запись строки в двумерный массив для метода `encrypt`.
- `void write_str_to_v (std::vector< std::vector< std::string >> &v)`  
Запись строки в двумерный массив для метода `decrypt`.

Закрытые данные

- `uint key`
- `double rows`
- `std::string i_value`
- `std::string t`
- `std::vector< std::vector< std::string > > st_to_vec`

### 4.1.1 Подробное описание

Шифрование методом табличной маршрутной перестановки

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

Предупреждения

Реализация только для английского языка

### 4.1.2 Методы

#### 4.1.2.1 `decrypt()`

```
std::string table_cipher::decrypt ( )
```

Расшифрование

Строка для операции инициализируется в конструкторе класса

Возвращает

Функция возвращает расшифрованную строку

#### 4.1.2.2 `encrypt()`

```
std::string table_cipher::encrypt ( )
```

Шифрование

Строка для операции инициализируется в конструкторе класса

Возвращает

Функция возвращает зашифрованную строку

#### 4.1.2.3 `key_validation()`

```
void table_cipher::key_validation (
    uint & key,
    std::string & text )
```

Обработка ошибок ключа

Просиходит проверка длины ключа

## Аргументы

in	key	Обрабатываемый ключ
in	text	Обрабатываемая строка

## Возвращает

Функция ничего не возвращает

## Исключения

<code>table_error</code> , если	ключ = 1 или ключ больше длины строки
---------------------------------	---------------------------------------

## 4.1.2.4 text\_validation()

```
void table_cipher::text_validation (
    std::string & text )
```

Обработка ошибок в строке для операции

Строчные буквы в строке преобразует в заглавные

## Аргументы

in	text	Обрабатываемая строка
----	------	-----------------------

## Возвращает

Функция ничего не возвращает

## Исключения

<code>table_error</code> , если	текст пустой или присутствуют символы не-бувы
---------------------------------	---

## 4.1.2.5 write\_str\_to\_v() [1/2]

```
void table_cipher::write_str_to_v (
    std::string st,
    std::vector< std::vector< std::string >> & v ) [private]
```

Запись строки в двумерный массив для метода encrypt.

Строчные символы автоматически преобразуются к прописным.

Вектор, переданный по ссылке заполняется значениями

## Аргументы

in	open_text	Открытый текст. Не должен быть пустой строкой.
in	vector	Двумерный массив для записи строки

## Возвращает

Функция ничего не возвращает

## Исключения

<a href="#">table_error</a> ,если	текст пустой или присутствуют символы не-бувы
-----------------------------------	---

## 4.1.2.6 write\_str\_to\_v() [2/2]

```
void table_cipher::write_str_to_v (
    std::vector< std::vector< std::string >> & v ) [private]
```

Запись строки в двумерный массив для метода decrypt.

Вектор, переданный по ссылке заполняется значениями

## Аргументы

in	vector	Двумерный массив для записи строки
----	--------	------------------------------------

## Возвращает

Функция ничего не возвращает

## Исключения

<a href="#">table_error</a> ,если	текст пустой или присутствуют символы не-бувы
-----------------------------------	---

Объявления и описания членов классов находятся в файлах:

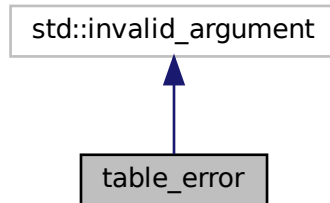
- [table\\_cipher.h](#)
- [table\\_cipher.cpp](#)

## 4.2 Класс table\_error

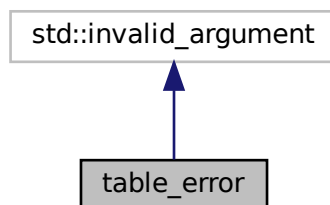
Класс ошибок для модуля Table\_cipher.

```
#include <table_cipher.h>
```

Граф наследования: `table_error`:



Граф связей класса `table_error`:



### Открытые члены

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

#### 4.2.1 Подробное описание

Класс ошибок для модуля `Table_cipher`.

Класс ошибок для специфичных исключений, возникающих при работе шифра

#### Предупреждения

Реализация только для английского языка

Объявления и описания членов класса находятся в файле:

- [table\\_cipher.h](#)

## 4.3 Класс UI

### Открытые члены

- `UI (int argc, char *argv[])`
- `uint get_action ()`
- `double get_key ()`
- `std::string get_string ()`

### Открытые атрибуты

- `po::options_description desc`
- `po::variables_map vm`
- `std::string st`
- `uint action`
- `double key`

Объявления и описания членов класса находятся в файле:

- [main.cpp](#)



## Глава 5

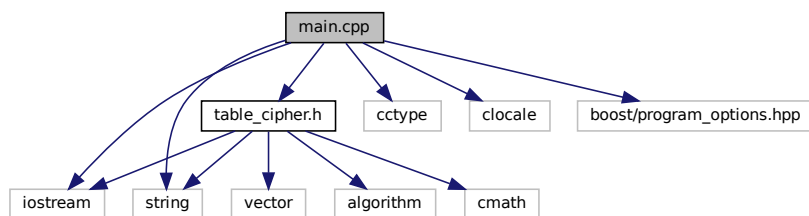
# Файлы

### 5.1 Файл main.cpp

Главный файл для модуля Table\_cipher.

```
#include <iostream>
#include <cctype>
#include <clocale>
#include <string>
#include "table_cipher.h"
#include <boost/program_options.hpp>
```

Граф включаемых заголовочных файлов для main.cpp:



### Классы

- class [UI](#)

### Функции

- int main (int argc, char \*\*argv)

### 5.1.1 Подробное описание

Главный файл для модуля Table\_cipher.

Автор

Стригин А.В.

Версия

1.0

Дата

22.12.2019

Авторство

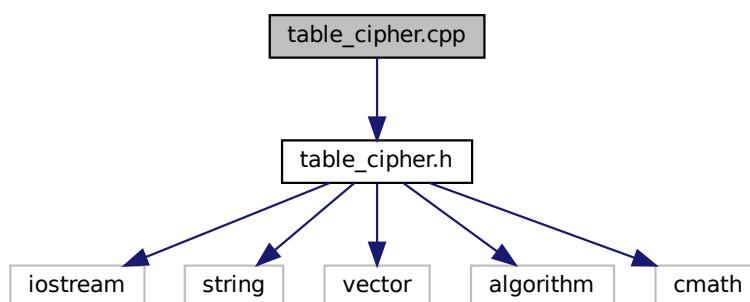
ИБСТ ПГУ

## 5.2 Файл table\_cipher.cpp

Исполняемый файл для модуля Table\_cipher.

```
#include "table_cipher.h"
```

Граф включаемых заголовочных файлов для table\_cipher.cpp:



### 5.2.1 Подробное описание

Исполняемый файл для модуля Table\_cipher.

Автор

Стригин А.В.

Версия

1.0

Дата

22.12.2019

Авторство

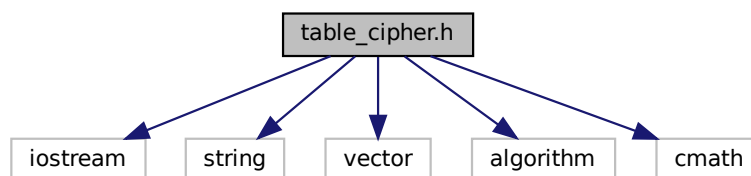
ИБСТ ПГУ

## 5.3 Файл table\_cipher.h

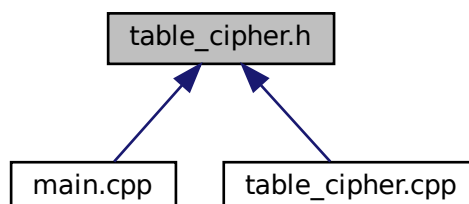
Заголовочный файл для модуля Table\_cipher.

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <cmath>
```

Граф включаемых заголовочных файлов для table\_cipher.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [table\\_cipher](#)  
Шифрование методом табличной маршрутной перестановки
- class [table\\_error](#)  
Класс ошибок для модуля Table\_cipher.

### 5.3.1 Подробное описание

Заголовочный файл для модуля Table\_cipher.

Автор

Стригин А.В.

Версия

1.0

Дата

22.12.2019

Авторство

ИБСТ ПГУ

# Предметный указатель

- decrypt
  - table\_cipher, [8](#)
- encrypt
  - table\_cipher, [8](#)
- key\_validation
  - table\_cipher, [8](#)
- main.cpp, [13](#)
- table\_cipher, [7](#)
  - decrypt, [8](#)
  - encrypt, [8](#)
  - key\_validation, [8](#)
  - text\_validation, [9](#)
  - write\_str\_to\_v, [9](#), [10](#)
- table\_cipher.cpp, [14](#)
- table\_cipher.h, [15](#)
- table\_error, [10](#)
- text\_validation
  - table\_cipher, [9](#)
- UI, [12](#)
- write\_str\_to\_v
  - table\_cipher, [9](#), [10](#)