

Table_cipher

1.0

Создано системой Doxygen 1.9.1

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 <code>table_cipher</code>	2
4.1.1 Подробное описание	3
4.1.2 Методы	3
4.2 Класс <code>table_error</code>	5
4.2.1 Подробное описание	6
4.3 Класс <code>UI</code>	7
4.3.1 Конструктор(ы)	7
4.3.2 Методы	8
5 Файлы	9
5.1 Файл <code>main.cpp</code>	9
5.1.1 Подробное описание	9
5.2 Файл <code>table_cipher.cpp</code>	10
5.2.1 Подробное описание	10
5.3 Файл <code>table_cipher.h</code>	11
5.3.1 Подробное описание	11
Предметный указатель	13

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

<code>std::invalid_argument</code>	
<code>table_error</code>	5
<code>table_cipher</code>	2
<code>UI</code>	7

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

table_cipher	Шифрование методом табличной маршрутной перестановки	2
table_error	Класс ошибок для модуля Table_cipher	5
UI		7

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный файл для модуля Table_cipher	9
table_cipher.cpp	Исполняемый файл для модуля Table_cipher	10
table_cipher.h	Заголовочный файл для модуля Table_cipher	11

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-буквы
--------------------------------	------------------------------------------------

```
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	<code>open_text</code>	Открытый текст. Не должен быть пустой строкой.
in	<code>vector</code>	Двумерный массив для записи строки

Возвращает

Функция ничего не возвращает

Исключения

<code>table_error</code> , 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	<code>vector</code>	Двумерный массив для записи строки
----	---------------------	------------------------------------

Возвращает

Функция ничего не возвращает

Исключения

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

Объявления и описания членов классов находятся в файлах:

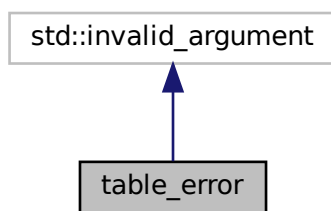
- [table_cipher.h](#)
- [table_cipher.cpp](#)

4.2 Класс `table_error`

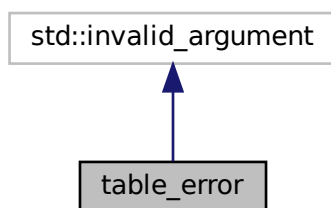
Класс ошибок для модуля `Table_cipher`.

```
#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[])`
Конструктор инициализации
- `uint get_action ()`
Метод получения типа операции
- `double get_key ()`
Метод получения ключа
- `std::string get_string ()`
Метод получения строки

Открытые атрибуты

- `po::options_description desc`
- `po::variables_map vm`
- `std::string st`
Обрабатываемая строка
- `uint action`
Тип операции
- `double key`
Ключ

4.3.1 Конструктор(ы)

4.3.1.1 `UI() UI: UI (`
`int argc,`
`char * argv[])`

Конструктор инициализации

Происходит разбор комстроки, инициализируются `st`, `action`, `key` На основе полученных данных вызывается соответствующий метод `table_cipher`

Аргументы

in	argc	Количество параметров комстроки
in	argv	Значения параметров комстроки

Исключения

<code>table_error</code> , если	создании оъекта класса будет выявлена ошибка: текст содержит не буквы, ключ слишком длинный или короткий
---------------------------------	----------------------------------------------------------------------------------------------------------

4.3.2 Методы

4.3.2.1 `get_action()` `uint UI::get_action ()`

Метод получения типа операции

Если значение параметра не введено, то выводится справка

Возвращает

Численное значение типа операции

4.3.2.2 `get_key()` `double UI::get_key ()`

Метод получения ключа

Если значение параметра не введено, то выводится справка

Возвращает

Значение ключа

4.3.2.3 `get_string()` `std::string UI::get_string ()`

Метод получения строки

Если значение параметра не введено, то выводится справка

Возвращает

Строка для обработки

Объявления и описания членов класса находятся в файле:

- [main.cpp](#)

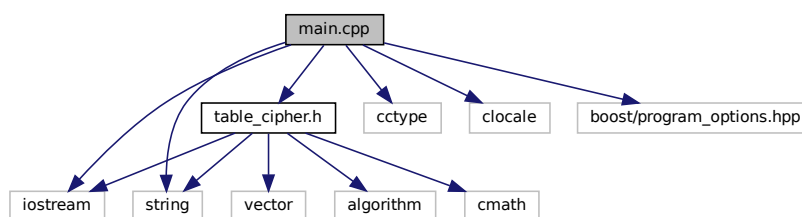
5 Файлы

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

Черный М.В.

Версия

1.0

Дата

22.12.2023

Авторство

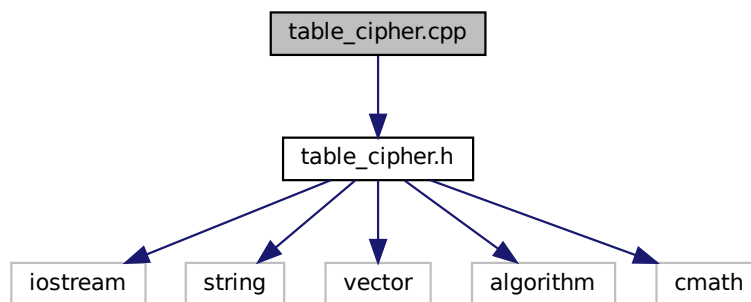
ИБСТ ПГУ

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

Черный М.В.

Версия

1.0

Дата

22.12.2023

Авторство

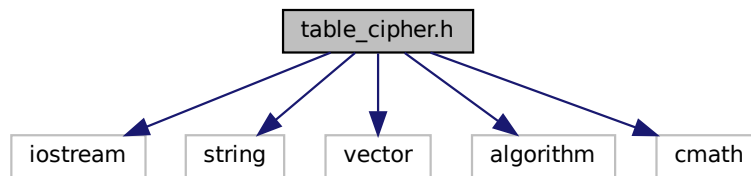
ИБСТ ПГУ

5.3 Файл table_cipher.h

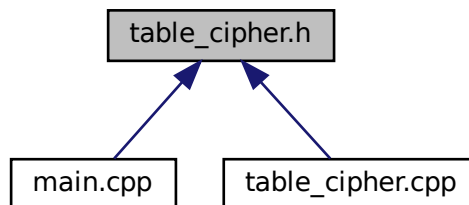
Заголовочный файл для модуля Table_cipher.

```
#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

Черный М.В.

Версия

1.0

Дата

22.12.2023

Авторство

ИБСТ ПГУ

Предметный указатель

- decrypt
 - table_cipher, [3](#)
- encrypt
 - table_cipher, [3](#)
- get_action
 - UI, [8](#)
- get_key
 - UI, [8](#)
- get_string
 - UI, [8](#)
- key_validation
 - table_cipher, [3](#)
- main.cpp, [9](#)
- table_cipher, [2](#)
 - decrypt, [3](#)
 - encrypt, [3](#)
 - key_validation, [3](#)
 - text_validation, [4](#)
 - write_str_to_v, [4](#), [5](#)
- table_cipher.cpp, [10](#)
- table_cipher.h, [11](#)
- table_error, [5](#)
- text_validation
 - table_cipher, [4](#)
- UI, [7](#)
 - get_action, [8](#)
 - get_key, [8](#)
 - get_string, [8](#)
 - UI, [7](#)
- write_str_to_v
 - table_cipher, [4](#), [5](#)