

Table_Cipher Демина
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>	6
4.2.1 Подробное описание	6
5 Файлы	7
5.1 Файл <code>main.cpp</code>	7
5.1.1 Подробное описание	7
5.2 Файл <code>table_cipher.cpp</code>	8
5.2.1 Подробное описание	8
5.3 Файл <code>table_cipher.h</code>	8
5.3.1 Подробное описание	9
Предметный указатель	11

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

<code>std::invalid_argument</code>	
<code>table_error</code>	6
<code>table_cipher</code>	2

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

<code>table_cipher</code>	
Шифрование методом табличной маршрутной перестановки	2

[table_error](#)

Класс ошибок для модуля Table_cipher

6

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

[main.cpp](#)

Главный файл для модуля Table_cipher

7

[table_cipher.cpp](#)

Исполняемый файл для модуля Table_cipher

8

[table_cipher.h](#)

Заголовочный файл для модуля Table_cipher

8

4 Классы

4.1 Класс table_cipher

Шифрование методом табличной маршрутной перестановки

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

Открытые члены

- `std::string encrypt (std::string st)`
Зашифрование
- `std::string decrypt (std::string st)`
Расшифрование
- `void write_str_to_v (std::string st, std::vector< std::vector< std::string >> &v)`
Запись строки в двумерный вектор для метода encrypt.
- `void write_str_to_v (std::vector< std::vector< std::string >> &v, std::string st)`
Запись строки в двумерный вектор для метода decrypt.
- `void text_validation (std::string &text)`
Обработка ошибок открытого текста и шифротекста
- `void key_validation (uint &key, std::string &text)`
Обработка ошибок ключа для шифротекста и открытого текста
- `table_cipher (double k, std::string &initial_value)`
Конструктор для установки ключа и значения инициализации двумерного вектора

Открытые атрибуты

- uint `key`
Ключ, заданный пользователем
- double `rows`
Значение строк, вычисленное по ключу
- std::string `i_value`
Значение инициализации для двумерного вектора
- std::vector< std::vector< std::string > > `st_to_vec`
Вектор для преобразования строки в таблицу

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 (std::string st)

Расшифрование

Аргументы

in	st	Шифротекст
----	----	------------

В методе производится вычисление числа строк в таблице по введенному шифротексту. Происходит проверка ключа и введенного шифротекста на наличие ошибок. Далее методом `write_str_to_v` происходит запись шифротекста в двумерный вектор. Затем по заданному в условии порядку происходит считывание символов из вектора и запись их в результирующую строку. На этапе ввода шифротекста и возвращения результирующей строки удаляются пробелы.

Возвращает

Функция возвращает расшифрованную строку

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

Зашифрование

Аргументы

in	st	Открытый текст
----	----	----------------

В методе производится вычисление числа строк в таблице по введенному открытому тексту. Происходит проверка ключа и введенного открытого текста на наличие ошибок. Далее методом `write_` ← `str_to_v` происходит запись открытого текста в двумерный вектор. Затем по заданному в условии порядку происходит считывание символов из вектора и запись их в результирующую строку. На этапе ввода открытого текста и возвращения результирующей строки удаляются пробелы.

Возвращает

Функция возвращает зашифрованную строку.

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

Обработка ошибок ключа для шифротекста и открытого текста

Происходит 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, ключ больше длины строки или равен длине строки
---------------------------------	---

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

Обработка ошибок открытого текста и шифротекста

Строчные буквы в строке преобразуют в заглавные

Аргументы

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

Возвращает

Функция ничего не возвращает

Исключения

table_error ,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)

Запись строки в двумерный вектор для метода encrypt.

Вектор, переданный по ссылке заполняется посимвольно значениями открытого текста

Аргументы

in	open_text	Открытый текст
in	vector	Двумерный вектор для записи строки

Возвращает

Функция ничего не возвращает

4.1.2.6 write_str_to_v() [2/2] void table_cipher::write_str_to_v (
std::vector< std::vector< std::string >> & v,
std::string st)

Запись строки в двумерный вектор для метода decrypt.

Вектор, переданный по ссылке заполняется посимвольно значениями шифротекста

Аргументы

in	cipher_text	Шифротекст
in	vector	Двумерный вектор для записи строки

Возвращает

Функция ничего не возвращает

Объявления и описания членов классов находятся в файлах:

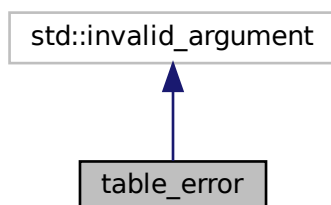
- [table_cipher.h](#)
- [table_cipher.cpp](#)

4.2 Класс `table_error`

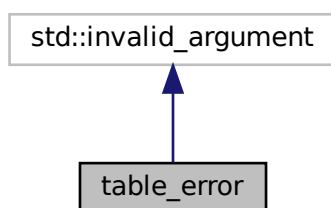
Класс ошибок для модуля `Table_cipher`.

```
#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`.

Используется для обозначения специфических ошибок, возникающих при работе с модулем

Предупреждения

Реализация только для английского языка

Объявления и описания членов класса находятся в файле:

- [table_cipher.h](#)

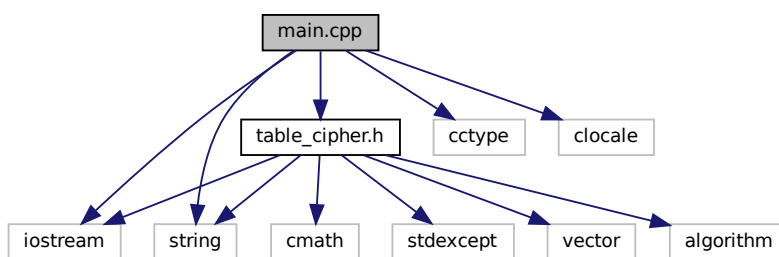
5 Файлы

5.1 Файл main.cpp

Главный файл для модуля Table_cipher.

```
#include <iostream>
#include <cctype>
#include <clocale>
#include <string>
#include "table_cipher.h"
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

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

5.1.1 Подробное описание

Главный файл для модуля Table_cipher.

Автор

Демина К.С.

Версия

1.0

Дата

18.01.24

Авторство

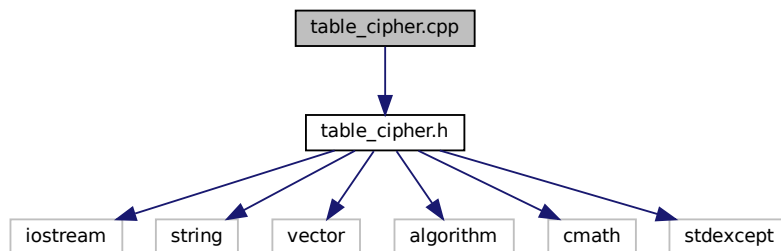
ИБСТ ПГУ

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

Демина К.С.

Версия

1.0

Дата

18.01.24

Авторство

ИБСТ ПГУ

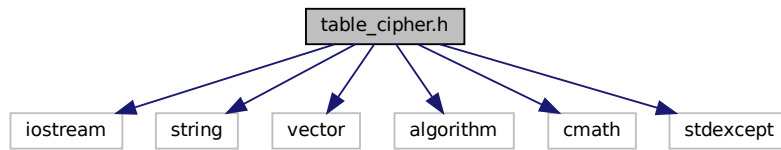
5.3 Файл table_cipher.h

Заголовочный файл для модуля Table_cipher.

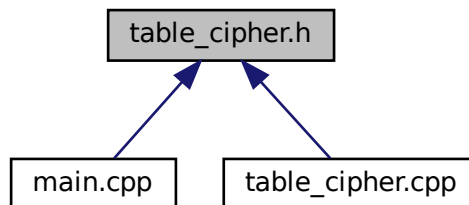
```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <cmath>
```

```
#include <stdexcept>
```

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

Демина К.С.

Версия

1.0

Дата

18.01.24

Авторство

ИБСТ ПГУ

Предметный указатель

- decrypt
 - table_cipher, [3](#)
- encrypt
 - table_cipher, [3](#)
- key_validation
 - table_cipher, [4](#)
- main.cpp, [7](#)
- table_cipher, [2](#)
 - decrypt, [3](#)
 - encrypt, [3](#)
 - key_validation, [4](#)
 - text_validation, [4](#)
 - write_str_to_v, [5](#)
- table_cipher.cpp, [8](#)
- table_cipher.h, [8](#)
- table_error, [6](#)
- text_validation
 - table_cipher, [4](#)
- write_str_to_v
 - table_cipher, [5](#)