

NotLab4

1.5

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 <code>table_cipher</code>	7
4.1.1 Подробное описание	8
4.1.2 Методы	8
4.1.2.1 <code>decrypt()</code>	8
4.1.2.2 <code>encrypt()</code>	8
4.1.2.3 <code>key_validation()</code>	8
4.1.2.4 <code>text_validation()</code>	9
4.1.2.5 <code>write_str_to_v()</code> [1/2]	9
4.1.2.6 <code>write_str_to_v()</code> [2/2]	10
4.2 Класс <code>table_error</code>	10
4.2.1 Подробное описание	11
4.3 Класс <code>UI</code>	12
4.3.1 Конструктор(ы)	12
4.3.1.1 <code>UI()</code>	12
4.3.2 Методы	13
4.3.2.1 <code>get_action()</code>	13
4.3.2.2 <code>get_key()</code>	13
4.3.2.3 <code>get_string()</code>	13
5 Файлы	15
5.1 Файл <code>main.cpp</code>	15
5.1.1 Подробное описание	16
5.2 Файл <code>table_cipher.cpp</code>	16
5.2.1 Подробное описание	17
5.3 Файл <code>table_cipher.h</code>	17
5.3.1 Подробное описание	18
Предметный указатель	19

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.

table_cipher	Шифрование методом табличной маршрутной перестановки	7
table_error	Класс ошибок для модуля Table_cipher	10
UI	12

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный файл для модуля Table_cipher	15
table_cipher.cpp	Исполняемый файл для модуля Table_cipher	16
table_cipher.h	Заголовочный файл для модуля Table_cipher	17

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

table_error ,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

table_error ,если	текст пустой или присутствуют символы не-бувы
-----------------------------------	---

Объявления и описания членов классов находятся в файлах:

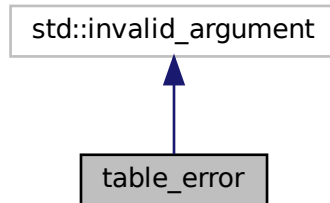
- [table_cipher.h](#)
- [table_cipher.cpp](#)

4.2 Класс table_error

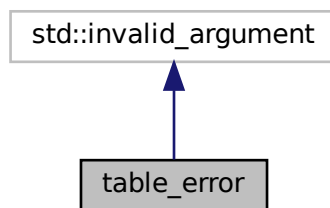
Класс ошибок для модуля Table_cipher.

```
#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

table_error ,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](#)

Глава 5

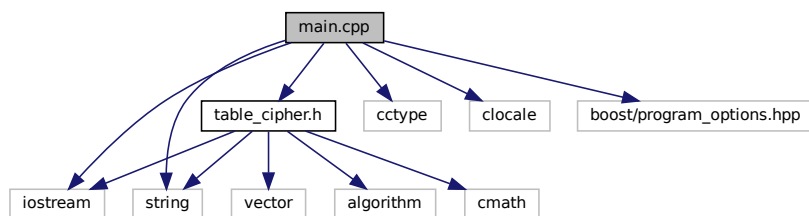
Файлы

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

Юртаев В.И.

Версия

1.0

Дата

08.11.2024

Авторство

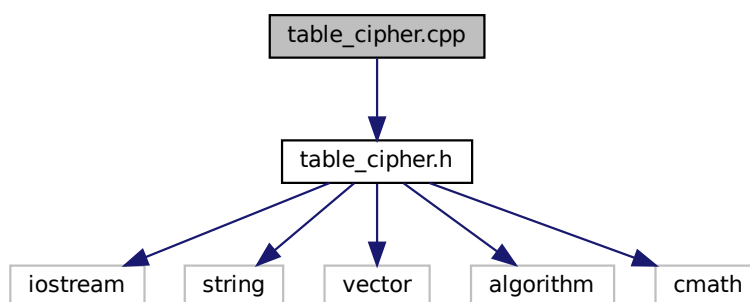
ИБСТ ПГУ

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

Юртаев В.И.

Версия

1.0

Дата

08.11.2024

Авторство

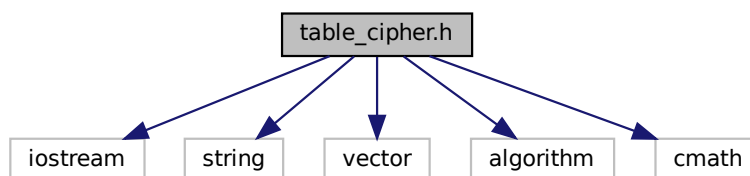
ИБСТ ПГУ

5.3 Файл table_cipher.h

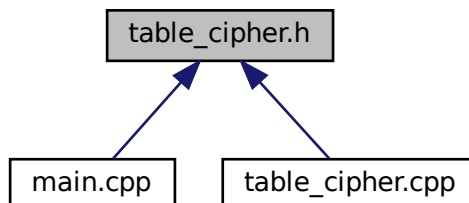
Заголовочный файл для модуля Table_cipher.

```
#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

Юртаев В.И.

Версия

1.0

Дата

08.11.2024

Авторство

ИБСТ ПГУ

Предметный указатель

- decrypt
 - table_cipher, [8](#)
- encrypt
 - table_cipher, [8](#)
- get_action
 - UI, [13](#)
- get_key
 - UI, [13](#)
- get_string
 - UI, [13](#)
- key_validation
 - table_cipher, [8](#)
- main.cpp, [15](#)
- table_cipher, [7](#)
 - decrypt, [8](#)
 - encrypt, [8](#)
 - key_validation, [8](#)
 - text_validation, [9](#)
 - write_str_to_v, [9](#), [10](#)
- table_cipher.cpp, [16](#)
- table_cipher.h, [17](#)
- table_error, [10](#)
- text_validation
 - table_cipher, [9](#)
- UI, [12](#)
 - get_action, [13](#)
 - get_key, [13](#)
 - get_string, [13](#)
 - UI, [12](#)
- write_str_to_v
 - table_cipher, [9](#), [10](#)