table_cipher\Солдатенков 22ПТ2

Создано системой Doxygen 1.9.4

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument

table_cipher_error 6

table_cipher 2

UI 6

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

4 Классы

4.1 Класс table_cipher

Класс для шифрования и дешифрования методом табличной перестановки.

#include <table_cipher.h>

Открытые члены

- table_cipher (std::string text, uint k)

  Конструктор класса table_cipher Происходит установка ключа, обработка ошибок текста и ключа
- std::string encrypt ()

  Функция шифрования текста Шифрование происходит путем считывания с матрицы, заполненной открытым текстом, в порядке сверху вниз справа налево
- std::string decrypt ()

  Функция расшифрования текста Расшифрование просходит путем заполнения матрицы в порядке слева направо сверху вниз Затем происходит "транспонирование матрицы" - последний столбец становится первой строкой и т.д После матрица считывается в порядке слева направо сверху вниз
- void fillVector (std::vector< std::vector< std::string > > &v, const std::string &text)

  Заполняет матрицу символами для шифрования в порядке слева направо сверху вниз
- void fillVector_for_decrypt (std::vector< std::vector< std::string > > &v, const std::string &text)

  Заполняет матрицу символами для дешифрования. Просходит транспонирование. Последний столбец становится первой строкой и т.д.
- void text_validation (std::string &text)

  Проверяет и преобразует текст перед шифрованием. Проверяется наличие посторонних символов и является ли строка пустой
- void key_validation (uint &key, std::string &text)

  Проверяет корректность ключа шифрования. Проверяется длина ключа

Открытые атрибуты

- uint key

  Ключ шифрования (количество столбцов в таблице)
- double rows

  Количество строк в таблице
- std::string t

  Исходный текст без пробелов
- std::vector< std::vector< std::string > > str_matrix

  Двумерный вектор для хранения таблицы шифрования

4.1.1 Подробное описание

Класс для шифрования и дешифрования методом табличной перестановки.

4.1.2 Конструктор(ы)

4.1.2.1 table_cipher() table_cipher::table_cipher (std::string text, uint k)

Конструктор класса table_cipher Происходит установка ключа, обработка ошибок текста и ключа

Аргументы

text	Исходный текст
k	Ключ (количество столбцов)

Исключения

table_cipher_error	в случае некорректного ключа или текста

4.1.3 Методы

4.1.3.1 decrypt() std::string table_cipher::decrypt ( )

Функция расшифрования текста Расшифрование просходит путем заполнения матрицы в порядке слева направо сверху вниз Затем происходит "транспонирование матрицы" - последний столбец становится первой строкой и т.д После матрица считывается в порядке слева направо сверху вниз

Возвращает

Расшифрованный текст

4.1.3.2 encrypt() std::string table_cipher::encrypt ( )

Функция шифрования текста Шифрование происходит путем считывания с матрицы, заполненной открытым текстом, в порядке сверху вниз справа налево

Возвращает

Зашифрованный текст

4.1.3.3 fillVector() void table_cipher::fillVector (std::vector< std::vector< std::string > > & v, const std::string & text)

Заполняет матрицу символами для шифрования в порядке слева направо сверху вниз

Аргументы

v	Двумерный вектор, в который записывается текст
text	Исходный текст для заполнения матрицы

4.1.3.4 fillVector_for_decrypt() void table_cipher::fillVector_for_decrypt (std::vector< std::vector< std::string > > & v, const std::string & text)

Заполняет матрицу символами для дешифрования. Просходит транспонирование. Последний столбец становится первой строкой и т.д.

Аргументы

v	Двумерный вектор, в который записывается текст
text	Зашифрованный текст для заполнения матрицы

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

Проверяет корректность ключа шифрования. Проверяется длина ключа

Аргументы

key	Ключ шифрования
text	Исходный текст

Исключения

table_cipher_error	в случае некорректного ключа

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

Проверяет и преобразует текст перед шифрованием. Проверяется наличие посторонних символов и является ли строка пустой

Аргументы

text	Исходный текст

Исключения

table_cipher_error	в случае наличия недопустимых символов

Объявления и описания членов классов находятся в файлах:

- table_cipher.h
- table_cipher.cpp

4.2 Класс table_cipher_error

Класс исключений для обработки ошибок в table_cipher.

#include <table_cipher.h>

Граф наследования:table_cipher_error:

4.3 Класс UI

Класс для обработки аргументов командной строки и предоставления параметров для шифрования/дешифрования.

#include <ui.h>

Открытые члены

- UI (int argc, char *argv[])
  Конструктор класса UI.
- uint get_action ()
  Геттер операции.
- double get_key ()
  Геттер ключа.
- std::string get_string ()
  Геттер строки для проведения операции.

Закрытые данные

- po::options_description desc
  Описание доступных аргументов командной строки.
- po::variables_map vm
  Карта параметров командной строки.

4.3.1 Подробное описание

Класс для обработки аргументов командной строки и предоставления параметров для шифрования/дешифрования.

4.3.2 Конструктор(ы)

4.3.2.1 UI()

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

Конструктор класса UI.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив аргументов командной строки.

Выполняет разбор аргументов командной строки с использованием библиотеки Boost.Program_↩ options.

4.3.3 Методы

4.3.3.1 get_action() uint UI::get_action ( )

Геттер операции.

Возвращает

Код действия (1 - шифрование, 2 - расшифрование).

Если аргумент не передан, программа завершится с выводом справки.

4.3.3.2 get_key() double UI::get_key ( )

Геттер ключа.

Возвращает

Количество столбцов в таблице (ключ шифрования).

Если аргумент не передан, программа завершится с выводом справки.

4.3.3.3 get_string() std::string UI::get_string ( )

Геттер строки для проведения операции.

Возвращает

Строка, которая будет зашифрована или расшифрована.

Если аргумент не передан, программа завершится с выводом справки.

Объявления и описания членов классов находятся в файлах:

- ui.h
- ui.cpp

5 Файлы

5.1 Файл table_cipher.cpp

Исполняемый файл класса table_cipher.

#include "table_cipher.h"
Граф включаемых заголовочных файлов для table_cipher.cpp:

5.1.1 Подробное описание

Исполняемый файл класса table_cipher.

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.2 Файл table_cipher.h

Определение класса table_cipher для шифрования методом табличной перестановки.

#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

  Класс для шифрования и дешифрования методом табличной перестановки.
- class table_cipher_error

  Класс исключений для обработки ошибок в table_cipher.

5.2.1 Подробное описание

Определение класса table_cipher для шифрования методом табличной перестановки.

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.3 table_cipher.h

См. документацию.

```
1
9 #pragma once
10 #include <iostream>
11 #include <string>
12 #include <vector>
13 #include <algorithm>
14 #include <cmath>
15
20 class table_cipher
21 {
22 public:
23     uint key;
24     double rows;
25     std::string t;
26     std::vector<std::vector<std::string» str_matrix;
27
35     table_cipher(std::string text, uint k);
36
42     std::string encrypt();
43
51     std::string decrypt();
52
58     void fillVector(std::vector<std::vector<std::string» &v, const std::string &text);
59
66     void fillVector_for_decrypt(std::vector<std::vector<std::string» &v, const std::string &text);
67
73     void text_validation(std::string &text);
74
81     void key_validation(uint &key, std::string &text);
82 };
83
88 class table_cipher_error: public std::invalid_argument
89 {
90 public:
95     explicit table_cipher_error (const std::string& what_arg):
96         std::invalid_argument(what_arg) {}
97
102     explicit table_cipher_error (const char* what_arg):
103         std::invalid_argument(what_arg) {}
104 };
```

5.4 Файл ui.cpp

Исполняемый файл класса ui.

#include <iostream>
#include <cctype>
#include <clocale>
#include <string>
#include <boost/program_options.hpp>
#include "ui.h"
Граф включаемых заголовочных файлов для ui.cpp:

5.4.1 Подробное описание

Исполняемый файл класса ui.

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.5 Файл ui.h

Определение класса UI для обработки аргументов командной строки.

#include <iostream>
#include <vector>
#include <string>
#include <boost/program_options.hpp>
#include "table_cipher.h"
Граф включаемых заголовочных файлов для ui.h: Граф файлов, в которые включается этот файл:

Классы

- class UI

  Класс для обработки аргументов командной строки и предоставления параметров для шифрования/дешифрования.

5.5.1 Подробное описание

Определение класса UI для обработки аргументов командной строки.

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.6 ui.h

См. документацию.

```
1
8 #pragma once
9
10 #include <iostream>
11 #include <vector>
12 #include <string>
13 #include <boost/program_options.hpp>
14 #include "table_cipher.h"
15
16 namespace po = boost::program_options;
17
22 class UI {
23 public:
31     UI(int argc, char* argv[]);
32
39     uint get_action();
40
47     double get_key();
48
55     std::string get_string();
56
57 private:
58     po::options_description desc;
59     po::variables_map vm;
60 };
```

Предметный указатель