Шифр табличной маршрутной перестановки

Создано системой Doxygen 1.9.4

1 Алфавитный указатель классов 1
1.1 Классы . . . 1

2 Список файлов 3
2.1 Файлы . . . 3

3 Классы 5
3.1 Класс tablecipher . . . 5
3.1.1 Подробное описание . . . 6
3.1.2 Методы . . . 6
3.1.2.1 decrypt() . . . 6
3.1.2.2 encrypt() . . . 6
3.1.2.3 getValidOpenText() . . . 7
3.1.2.4 isPlusKey() . . . 7
3.1.2.5 isValidKey() . . . 8

4 Файлы 9
4.1 tablecipher.h . . . 9

Предметный указатель 11

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

tablecipher
Шифрование методом табличной маршрутной перестановки 5

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

tablecipher.h . . . . . ??

Глава 3

Классы

3.1 Класс tablecipher

Шифрование методом табличной маршрутной перестановки

#include <tablecipher.h>

Открытые члены

- tablecipher (int key)
  конструктор
- tablecipher ()=delete
  запрект конструктора без параметров
- string encrypt (string text)
  Зашифровывание
- string decrypt (string text)
  Расшифровывание

Закрытые члены

- int isValidKey (int key, string s)
  Проверка валидации ключа
- bool isPlusKey (int key)
  Проверка символов строки
- string getValidOpenText (const std::string &s)
  Валидация зашифрованного текста

Закрытые данные

- int st
  количество столбцов

3.1.1 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

3.1.2 Методы

3.1.2.1 decrypt()

```
string tablecipher::decrypt (
                string text )
```

Расшифровывание

Аргументы

in	text	Шифрованный текст должен быть пустой строкой. Все пробелы убираются из текста Все не-буквы удаляются

Возвращает

расшифрованная строка

Исключения

cipher_error,если	текст пустой

3.1.2.2 encrypt()

```
string tablecipher::encrypt (
                string text )
```

Зашифровывание

Аргументы

in	text	Открытый текст. Не должен быть пустой строкой. Пробелы удаляются

Возвращает

Зашифрованная строка

Исключения

cipher_error,если	текст пустой

3.1.2.3 getValidOpenText()

std::string tablecipher::getValidOpenText (
const std::string & s) [inline], [private]

Валидация зашифрованного текста

Убирает пробелы из текста

Аргументы

ws	Входная строка, представляющая текст.

Исключения

cipher_error	если строка пуста

3.1.2.4 isPlusKey()

bool tablecipher::isPlusKey (
int key) [private]

Проверка символов строки

Проверяет, есть ли в сообщении символы, отличные от букв английского алфавита

Аргументы

s	Входная строка, представляющая текст.

Исключения

cipher_error,если	присутствуют некорректные символы

3.1.2.5 isValidKey()

```
int tablecipher::isValidKey (
                int key,
                string s )   [inline], [private]
```

Проверка валидации ключа

Проверяет, что ключ не более половины длины шифруемого сообщения.

Аргументы

s	Входная строка, представляющая ключ, строка с текстом.

Возвращает

ключ, который равен половины длины строки.

Объявления и описания членов классов находятся в файлах:

- tablecipher.h
- tablecipher.cpp

Глава 4

Файлы

4.1 tablecipher.h

```
1
7 #pragma once
8 #include <string>
9 #include <locale>
10 #include <iostream>
11 #include <map>
12 #include <cmath>
13 using namespace std;
19 class tablecipher {
20 private:
21     int st;
30     int isValidKey(int key,string s);
39     bool isPlusKey(int key);
48     string getValidOpenText(const std::string& s);
49 public:
50     tablecipher(int key);
51     tablecipher() = delete;
59     string encrypt(string text);
68     string decrypt(string text);
75 }
76 class cipher_error : public std::invalid_argument
77 {
78 public:
84     explicit cipher_error(const std::string& what_arg):
85     std::invalid_argument(what_arg){}
91     explicit cipher_error(const char* what_arg):
92     std::invalid_argument(what_arg){}
93 };
```

Предметный указатель

decrypt
    tablecipher, 6

encrypt
    tablecipher, 6

getValidOpenText
    tablecipher, 7

isPlusKey
    tablecipher, 7
isValidKey
    tablecipher, 7

tablecipher, 5
    decrypt, 6
    encrypt, 6
    getValidOpenText, 7
    isPlusKey, 7
    isValidKey, 7