Шифр маршрутной перестановки

Создано системой Doxygen 1.9.4

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс cipher

Шифрование методом табличной маршрутной перестановки

#include <cipher.h>

Открытые члены

- cipher (int key)
  конструктор
- cipher ()=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 знака ключа
- bool isValidText (const string &text)
  Проверка символов строки
- string getValidOpenText (const std::string &s)
  Валидация зашифрованного текста

Закрытые данные

- int stolb
  количество столбцов

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()

```
string cipher::decrypt (
            string text )
```

Расшифровывание

Аргументы

in	ctext	Шифрованный текст должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Возвращает

Зашифрованная строка

Исключения

cipher_error,если	текст пустой

4.1.2.2 encrypt()

```
string cipher::encrypt (
            string text )
```

Зашифровывание

Аргументы

in	text	Открытый текст. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Возвращает

Зашифрованная строка

Исключения

cipher_error,если	текст пустой

4.1.2.3 getValidOpenText()

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

Валидация зашифрованного текста

Приводит символы строки к верхнему регистру

Аргументы

ws	Входная строка, представляющая текст.

Исключения

cipher_error	если строка пуста

4.1.2.4 isPlusKey()

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

Проверка знака ключа

Проверяет, что ключ более нуля

Аргументы

s	Входная строка, представляющая ключ.

Исключения

cipher_error,если	ключ <= 0

4.1.2.5 isValidKey()

```
int cipher::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.

4.1.2.6 isValidText()

```
bool cipher::isValidText (
                const string & text )   [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

Объявления и описания членов классов находятся в файлах:

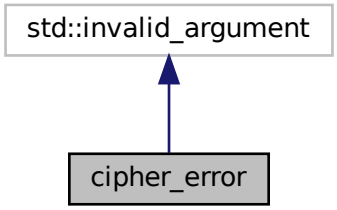
- cipher.h
- cipher.cpp

4.2 Класс cipher_error

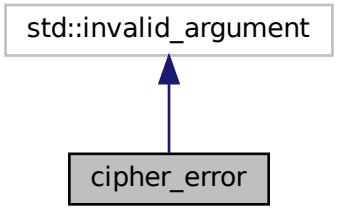
Обработка исключений

#include <cipher.h>

Граф наследования:cipher_error:



Граф связей класса cipher_error:



Открытые члены

- cipher_error (const std::string &what_arg)
  Конструктор с аргументом типа std::string.
- cipher_error (const char *what_arg)
  Конструктор с аргументом типа const char*.

4.2.1 Подробное описание

Обработка исключений

Класс, созданный для обработки ошибок

4.2.2 Конструктор(ы)

4.2.2.1 cipher_error() [1/2]

cipher_error::cipher_error (
const std::string & what_arg) [inline], [explicit]

Конструктор с аргументом типа std::string.

Аргументы

what_arg	Сообщение об ошибке.

4.2.2.2 cipher_error() [2/2]

cipher_error::cipher_error (
const char * what_arg) [inline], [explicit]

Конструктор с аргументом типа const char*.

Аргументы

what_arg	Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

- cipher.h

Глава 5

Файлы

5.1 cipher.h

```
1
7 #pragma once
8 #include <cmath>
9 #include <iostream>
10 #include <locale>
11 #include <map>
12 #include <stdexcept>
13 #include <string>
14 using namespace std;
20 class cipher
21 {
22 private:
23     int stolb;
32     int isValidKey(int key, string s);
41     bool isPlusKey(int key);
50     bool isValidText(const string& text);
59     string getValidOpenText(const std::string& s);
60
61 public:
62     cipher(int key);
63     cipher() = delete;
72     string encrypt(string text);
81     string decrypt(string text);
82 };
89 class cipher_error : public std::invalid_argument
90 {
91 public:
97     explicit cipher_error(const std::string& what_arg)
98         : std::invalid_argument(what_arg)
99     {
100     }
106     explicit cipher_error(const char* what_arg)
107         : std::invalid_argument(what_arg)
108     {
109     }
110 };
```

Предметный указатель