Ponomarev_TIMP_LB4

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

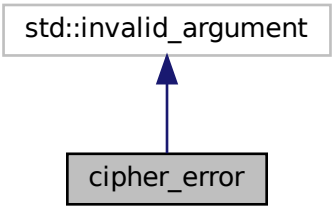
Классы

4.1 Класс cipher_error

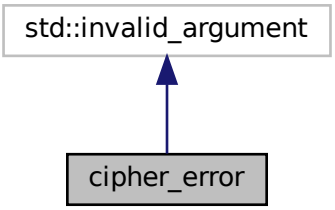
Класс для обработки ошибок шифрования.

#include <TableCipher.h>

Граф наследования:cipher_error:



Граф связей класса cipher_error:



Открытые члены

- cipher_error (const std::string &what_arg)

  Конструктор класса cipher_error с сообщением об ошибке.
- cipher_error (const char *what_arg)

  Конструктор класса cipher_error с сообщением об ошибке в виде C-строки.

4.1.1 Подробное описание

Класс для обработки ошибок шифрования.

Данный класс расширяет стандартное исключение std::invalid_argument для обработки ошибок, связанных с шифрованием и дешифрованием текста.

4.1.2 Конструктор(ы)

4.1.2.1 cipher_error() [1/2]

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

Конструктор класса cipher_error с сообщением об ошибке.

Аргументы

what_arg	Сообщение об ошибке в виде строки.

4.1.2.2 cipher_error() [2/2]

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

Конструктор класса cipher_error с сообщением об ошибке в виде C-строки.

Аргументы

what_arg	Сообщение об ошибке в виде C-строки.

Объявления и описания членов класса находятся в файле:

- TableCipher.h

4.2 Класс TableCipher

Класс для шифрования и дешифрования текста с использованием таблицы.

#include <TableCipher.h>

Открытые члены

- TableCipher ()=delete
  Конструктор класса TableCipher.
- TableCipher (const std::wstring &key_str)
- std::wstring encrypt (const std::wstring &text)
  Шифрует заданный текст.
- std::wstring decrypt (const std::wstring &encrypted_text)
  Дешифрует зашифрованный текст.
- int getValidKey (const std::wstring &key_str)
  Проверяет валидность ключа.
- std::wstring getValidOpenText (const std::wstring &s)
  Проверяет валидность открытого текста.
- std::wstring getValidCipherText (const std::wstring &s)
  Проверяет валидность зашифрованного текста.

Закрытые данные

- int key
  Ключ для шифрования и дешифрования.

4.2.1 Подробное описание

Класс для шифрования и дешифрования текста с использованием таблицы.

Данный класс реализует шифрование и дешифрование текста на основе заданного ключа. Ключ должен быть валидным и соответствовать определённым требованиям.

4.2.2 Конструктор(ы)

4.2.2.1 TableCipher()

TableCipher::TableCipher ( ) [delete]

Конструктор класса TableCipher.

Конструктор инициализирует объект с заданным ключом, который должен быть валидным. Если ключ не валиден, будет выброшено исключение.

Аргументы

key_str	Строка, представляющая ключ для шифрования.

Исключения

cipher_error	Если ключ недействителен.

4.2.3 Методы

4.2.3.1 decrypt()

```
std::wstring TableCipher::decrypt (
            const std::wstring & encrypted_text )
```

Дешифрует зашифрованный текст.

Метод принимает зашифрованный текст и возвращает открытый текст, используя тот же ключ, что и при шифровании.

Аргументы

encrypted_text	Зашифрованный текст, который необходимо расшифровать.

Возвращает

Открытый текст.

4.2.3.2 encrypt()

```
std::wstring TableCipher::encrypt (
            const std::wstring & text )
```

Шифрует заданный текст.

Метод принимает открытый текст и возвращает зашифрованный текст на основе заданного ключа.

Аргументы

text	Открытый текст, который необходимо зашифровать.

Возвращает

Зашифрованный текст.

4.2.3.3 getValidCipherText()

```
std::wstring TableCipher::getValidCipherText (
        const std::wstring & s )
```

Проверяет валидность зашифрованного текста.

Метод проверяет, является ли переданная строка валидным зашифрованным текстом. Если текст недействителен, будет выброшено исключение.

Аргументы

s	Строка, представляющая зашифрованный текст для проверки.

Возвращает

Валидный зашифрованный текст.

Исключения

cipher_error	Если зашифрованный текст недействителен.

4.2.3.4 getValidKey()

```
int TableCipher::getValidKey (
        const std::wstring & key_str )
```

Проверяет валидность ключа.

Метод проверяет, является ли переданная строка валидным ключом для шифрования. Если ключ недействителен, будет выброшено исключение.

Аргументы

key_str	Строка, представляющая ключ для проверки.

Возвращает

Валидный целочисленный ключ.

Исключения

cipher_error	Если ключ недействителен.

4.2.3.5 getValidOpenText()

```
std::wstring TableCipher::getValidOpenText (
                const std::wstring & s )
```

Проверяет валидность открытого текста.

Метод проверяет, является ли переданная строка валидным открытым текстом. Если текст недействителен, будет выброшено исключение.

Аргументы

s	Строка, представляющая открытый текст для проверки.

Возвращает

Валидный открытый текст.

Исключения

cipher_error	Если открытый текст недействителен.

Объявления и описания членов класса находятся в файле:

- TableCipher.h

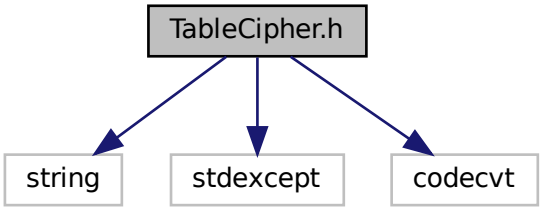
Глава 5

Файлы

5.1 Файл TableCipher.h

Заголовочный файл для модуля шифрования методом маршутной перестановки

#include <string>
#include <stdexcept>
#include <codecvt>
Граф включаемых заголовочных файлов для TableCipher.h:



Классы

- class TableCipher

  Класс для шифрования и дешифрования текста с использованием таблицы.
- class cipher_error

  Класс для обработки ошибок шифрования.

5.1.1 Подробное описание

Заголовочный файл для модуля шифрования методом маршутной перестановки

Автор

Пономарев А.А

Версия

1.0

Дата

04.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.2 TableCipher.h

См. документацию.

```
1
10 #pragma once
11 #include <string>
12 #include <stdexcept>
13 #include <codecvt>
14
22 class TableCipher
23 {
24 private:
25     int key;
26
27 public:
34     TableCipher() = delete;
35     TableCipher(const std::wstring& key_str);
36
43     std::wstring encrypt(const std::wstring& text);
44
51     std::wstring decrypt(const std::wstring& encrypted_text);
52
60     int getValidKey(const std::wstring& key_str);
61
69     std::wstring getValidOpenText(const std::wstring & s);
70
78     std::wstring getValidCipherText(const std::wstring & s);
79 };
80
88 class cipher_error : public std::invalid_argument
89 {
90 public:
94     explicit cipher_error(const std::string& what_arg)
95         : std::invalid_argument(what_arg)
96     {
97     }
98
102     explicit cipher_error(const char* what_arg)
103         : std::invalid_argument(what_arg)
104     {
105     }
106 };
```

Предметный указатель