Шифр табличной 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

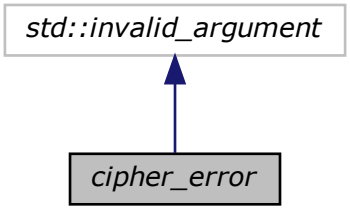
Классы

4.1 Класс cipher_error

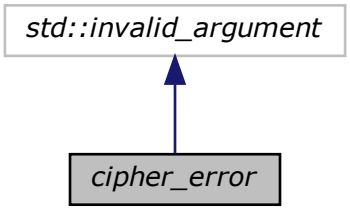
Обработка исключений

#include <modAlphaCipher.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*.

4.1.1 Подробное описание

Обработка исключений

Класс, созданный для обработки ошибок

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 с аргументом типа std::string.

Аргументы

what_arg	Сообщение об ошибке.

4.1.2.2 cipher_error() [2/2]

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

Конструктор с аргументом типа const char*.

Аргументы

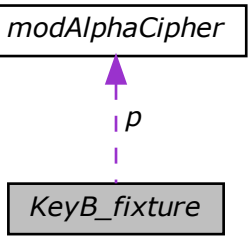
what_arg	Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

- modAlphaCipher.h

4.2 Структура KeyB_fixture

Граф связей класса KeyB_fixture:



Открытые атрибуты

- modAlphaCipher * p

Объявления и описания членов структуры находятся в файле:

- main.cpp

4.3 Класс modAlphaCipher

Шифрование методом Гронсфельда

#include <modAlphaCipher.h>

Открытые члены

- modAlphaCipher ()=delete
  Запрет на использование конструктора по умолчанию
- modAlphaCipher (const std::wstring &skey)
  Конструктор для установки ключа
- std::wstring encrypt (const std::wstring &open_text)
  Зашифровывание
- std::wstring decrypt (const std::wstring &cipher_text)
  Расшифровывание

Закрытые члены

- std::vector< int > convert (const std::wstring &s)

  Преобразование строки в вектор
- std::wstring convert (const std::vector< int > &v)

  Преобразование вектора в строку
- std::wstring getValidKey (const std::wstring &s)

  Валидация ключа
- std::wstring getValidOpenText (const std::wstring &s)

  Валидация открытого текста
- std::wstring getValidCipherText (const std::wstring &s)

  Валидация зашифрованного текста

Закрытые данные

- std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"

  Алфавит
- std::map< char, int > alphaNum

  Ассоциативный массив "номер по символу".
- std::vector< int > key

  Ключ

4.3.1 Подробное описание

Шифрование методом Гронсфельда

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы encrypt и decrypt.

Предупреждения

Реализация только для русского языка

Шифрование методом Гронсфельда

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы encrypt и decrypt.

Предупреждения

Реализация только для русского языка

4.3.2 Конструктор(ы)

4.3.2.1 modAlphaCipher()

```
modAlphaCipher::modAlphaCipher (
            const std::wstring & skey )
```

Конструктор для установки ключа

Аргументы

skey	Строка, представляющая ключ для шифрования.

4.3.3 Методы

4.3.3.1 decrypt()

```
std::wstring modAlphaCipher::decrypt (
                const std::wstring & cipher_text )
```

Расшифровывание

Аргументы

in	cipher_text	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

4.3.3.2 encrypt()

```
std::wstring modAlphaCipher::encrypt (
                const std::wstring & open_text )
```

Зашифровывание

Аргументы

in	open_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

4.3.3.3 getValidCipherText()

std::wstring modAlphaCipher::getValidCipherText (
const std::wstring & s) [inline], [private]

Валидация зашифрованного текста

Проверяет, что зашифрованный текст содержит только допустимые символы алфавита.

Аргументы

ws	Входная строка, представляющая зашифрованный текст.

Исключения

cipher_error	Если зашифрованный текст содержит недопустимые символы.

4.3.3.4 getValidKey()

std::wstring modAlphaCipher::getValidKey (
const std::wstring & s) [private]

Валидация ключа

Проверяет, что ключ не пустой и не содержит символов, не принадлежащих алфавиту.

Аргументы

s	Входная строка, представляющая ключ.

Исключения

cipher_error	Если ключ пустой или содержит недопустимые символы.

4.3.3.5 getValidOpenText()

std::wstring modAlphaCipher::getValidOpenText (
const std::wstring & s) [inline], [private]

Валидация открытого текста

Проверяет, что открытый текст содержит только допустимые символы алфавита.

Аргументы

ws	Входная строка, представляющая открытый текст.

Исключения

cipher_error	Если открытый текст содержит недопустимые символы.

Объявления и описания членов классов находятся в файлах:

- modAlphaCipher.h
- modAlphaCipher.cpp

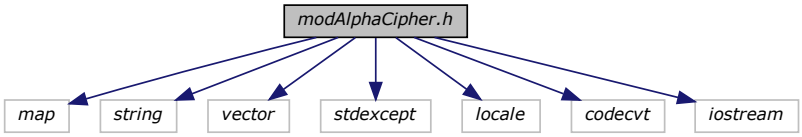
Глава 5

Файлы

5.1 Файл modAlphaCipher.h

```
#include <map>
#include <string>
#include <vector>
#include <stdexcept>
#include <locale>
#include <codecvt>
#include <iostream>
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class modAlphaCipher

  Шифрование методом Гронсфельда
- class cipher_error

  Обработка исключений

5.1.1 Подробное описание

Автор

Белов А.Р.

Версия

1.0

Дата

06.12.2024

Авторство

ИБСТ ПГУ

5.2 modAlphaCipher.h

См. документацию.

```
1
8 #pragma once
9 #include <map>
10 #include <string>
11 #include <vector>
12 #include <stdexcept>
13 #include <locale>
14 #include <codecvt>
15 #include <iostream>
26 class modAlphaCipher
27 {
28 private:
29     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
30     std::map<char, int> alphaNum;
31     std::vector<int> key;
32     std::vector<int> convert(const std::wstring& s);
33     std::wstring convert(const std::vector<int>& v);
42     std::wstring getValidKey(const std::wstring & s);
51     std::wstring getValidOpenText(const std::wstring& s);
60     std::wstring getValidCipherText(const std::wstring& s);
61
62 public:
63     modAlphaCipher() = delete;
69     modAlphaCipher(const std::wstring& skey);
78     std::wstring encrypt(const std::wstring& open_text);
87     std::wstring decrypt(const std::wstring& cipher_text);
88 };
95 class cipher_error : public std::invalid_argument
96 {
97 public:
103     explicit cipher_error(const std::string& what_arg):
104     std::invalid_argument(what_arg){}
110     explicit cipher_error(const char* what_arg):
111     std::invalid_argument(what_arg){}
112 };
```

Предметный указатель