Gronsfeld

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

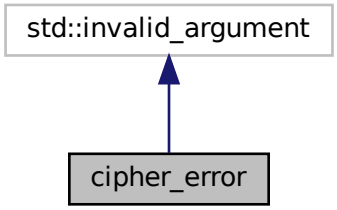
Классы

4.1 Класс cipher_error

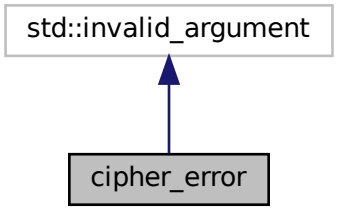
Исключение для ошибок в классе modAlphaCipher.

#include <modAlphaCipher.h>

Граф наследования:cipher_error:



Граф связей класса cipher_error:



Открытые члены

- cipher_error (const std::string &what_arg)
  Конструктор исключения с сообщением об.
- cipher_error (const char *what_arg)
  Конструктор исключения с сообщением об ошибке.

4.1.1 Подробное описание

Исключение для ошибок в классе modAlphaCipher.

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 исключения с сообщением об.

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 исключения с сообщением об ошибке.

Аргументы

what_arg	Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

- modAlphaCipher.h

4.2 Класс modAlphaCipher

Шифрование методом Гронсфельда

#include <modAlphaCipher.h>

Открытые члены

- modAlphaCipher ()=delete
  Конструктор по умолчанию запрещен.
- modAlphaCipher (const std::wstring &skey)
  Конструктор для ключа.
- std::wstring encrypt (const std::wstring &open_text)
  Метод шифрования открытого текста методом Гронсфельда.
- std::wstring decrypt (const std::wstring &cipher_text)
  Метод для расшифрования текса, зашифрованного методом Гронсфельда.

Закрытые члены

- std::vector< int > convert (const std::wstring &s)
  Преобразование "строка-вектор".
- std::wstring convert (const std::vector< int > &v)
  Преобразование "вектор-строка".
- std::wstring getValidKey (const std::wstring &s)
  Валидация ключа
- std::wstring getValidOpenText (const std::wstring &ws)
  Метод валидации открытого текста.
- std::wstring getValidCipherText (const std::wstring &ws)
  Валидация зашифрованного текста.

Закрытые данные

- std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
  Русский алфавит по порядку
- std::map< char, int > alphaNum
  Ассоциативный массив "номер по символу
- std::vector< int > key
  Ключ для шифрования

4.2.1 Подробное описание

Шифрование методом Гронсфельда

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

Предупреждения

Реализация только для русского языка

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

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

Конструктор для ключа.

Инициализирует алфавит, ассоциативный массив "номер по символу" и ключ.

Аргументы

skey	Ключ для шифрования типа "wstring".

4.2.3 Методы

4.2.3.1 decrypt()

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

Метод для расшифрования текса, зашифрованного методом Гронсфельда.

Аргументы

cipher_text	Строка зашифрованного текста для расшифровки.

Возвращает

std::wstring Расшифрованный текст.

Исключения

cipher_error	Если зашифрованный текст некорректен (пустой или содержит недопустимые символы).

4.2.3.2 encrypt()

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

Метод шифрования открытого текста методом Гронсфельда.

Аргументы

open_text	Строка открытого текста для шифрования.

Возвращает

std::wstring Зашифрованный текст.

Исключения

cipher_error	Если открытый текст некорректен (пустой или содержит недопустимые символы).

4.2.3.3 getValidCipherText()

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

Валидация зашифрованного текста.

Функция проверяет, что строка не пуста и содержит только символы прописных букв.

Аргументы

ws	Входная строка зашифрованного текста.

Возвращает

std::wstring Строка зашифрованного текста (копия входной строки), если она корректна.

Исключения

cipher_error	Если строка пуста или содержит символы, отличные от прописных букв.

4.2.3.4 getValidKey()

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

Валидация ключа

Исключения

cipher_error,если	ключ пустой или в ключе находится символ не принадлежащий алфавиту

4.2.3.5 getValidOpenText()

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

Метод валидации открытого текста.

Эта функция принимает строку открытого текста в качестве входных данных и возвращает новую строку, содержащую только буквы верхнего регистра из исходной строки. Строка очищается от всех символов, отличных от букв русского алфавита. Если результирующая строка пуста, выбрасывается исключение.

Аргументы

ws	Входная строка открытого текста.

Возвращает

std::wstring Строка,содержащая только буквы верхнего регистра.

Исключения

cipher_error	Если строка пришла пустая или осталась пустой после валидации.

Объявления и описания членов классов находятся в файлах:

- modAlphaCipher.h
- modAlphaCipher.cpp

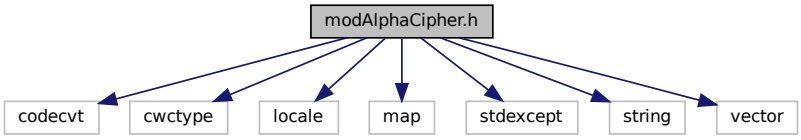
Глава 5

Файлы

5.1 Файл modAlphaCipher.h

Описание класса modAlphaCipher.

#include <codecvt>
#include <cwctype>
#include <locale>
#include <map>
#include <stdexcept>
#include <string>
#include <vector>
Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class modAlphaCipher

  Шифрование методом Гронсфельда
- class cipher_error

  Исключение для ошибок в классе modAlphaCipher.

5.1.1 Подробное описание

Описание класса modAlphaCipher.

Автор

Грачев В.В.

Версия

1.0

Дата

18.11.2024

Авторство

ИБСТ ПГУ

5.2 modAlphaCipher.h

См. документацию.

```
1
10 #pragma once
11 #include <codecvt>
12 #include <cwctype>
13 #include <locale>
14 #include <map>
15 #include <stdexcept>
16 #include <string>
17 #include <vector>
18
24 class modAlphaCipher
25 {
26 private:
27     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
28     std::map<char, int> alphaNum;
29     std::vector<int> key;
30     std::vector<int> convert(const std::wstring& s);
31     std::wstring convert(const std::vector<int>& v);
32
37     std::wstring getValidKey(const std::wstring& s);
38
50     std::wstring getValidOpenText(const std::wstring& ws);
51
61     std::wstring getValidCipherText(const std::wstring& ws);
62
63 public:
64     modAlphaCipher() = delete;
70     modAlphaCipher(const std::wstring& skey);
71
78     std::wstring encrypt(const std::wstring& open_text);
85     std::wstring decrypt(const std::wstring& cipher_text);
86 };
87
93 class cipher_error : public std::invalid_argument
94 {
95 public:
100     explicit cipher_error(const std::string& what_arg)
101         : std::invalid_argument(what_arg)
102     {
103     }
108     explicit cipher_error(const char* what_arg)
109         : std::invalid_argument(what_arg)
110     {
111     }
112 };
```

Предметный указатель