

Шифр Гронсвельда

Создано системой Doxygen 1.9.4

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Классы	7
4.1 Класс cipher_error	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 cipher_error() [1/2]	8
4.1.2.2 cipher_error() [2/2]	8
4.2 Класс modAlphaCipher	9
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	10
4.2.2.1 modAlphaCipher()	10
4.2.3 Методы	10
4.2.3.1 decrypt()	10
4.2.3.2 encrypt()	11
4.2.3.3 getValidCipherText()	11
4.2.3.4 getValidKey()	11
4.2.3.5 getValidOpenText()	12
5 Файлы	13
5.1 Файл modAlphaCipher.h	13
5.1.1 Подробное описание	14
5.2 modAlphaCipher.h	14
Предметный указатель	15

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	7
modAlphaCipher	9

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Обработка исключений Класс, созданный для обработки ошибок	7
modAlphaCipher	Шифрование методом Гронсвельда	9

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.h	
Заголовочный файл для модуля Гронсвельда	13

Глава 4

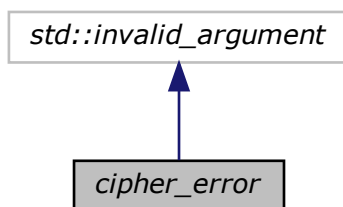
Классы

4.1 Класс cipher_error

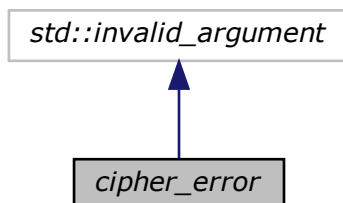
Обработка исключений Класс, созданный для обработки ошибок

```
#include <modAlphaCipher.h>
```

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- [cipher_error](#) (const std::string &what_arg)
Конструктор с параметром типа const std::string.
- [cipher_error](#) (const char *what_arg)
Конструктор с параметром типа const char.

4.1.1 Подробное описание

Обработка исключений Класс, созданный для обработки ошибок

Наследуется от std::invalid_argument

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 параметром типа const std::string.

Аргументы

what_arg	Описание ошибки
----------	-----------------

4.1.2.2 cipher_error() [2/2]

```

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

```

Конструктор с параметром типа const char.

Аргументы

what_arg	Описание ошибки
----------	-----------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Шифрование методом Гронсвельда

```
#include <modAlphaCipher.h>
```

Открытые члены

- modAlphaCipher ()=delete
Запрет на использование конструктора по умолчанию
- modAlphaCipher (const std::wstring &skey)
Конструктор с параметром
- std::wstring encrypt (const std::wstring &open_text)
Метод для зашифровывания
- std::wstring decrypt (const std::wstring &cipher_text)
Метод для расшифровывания

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 по порядку
- std::map< wchar_t, int > alphaNum
Ассоциативный массив "номер по символу".
- std::vector< int > key
Ключ

4.2.1 Подробное описание

Шифрование методом Гронсвельда

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 с параметром

Аргументы

skey	Ключ для шифрования/расшифрования
------	-----------------------------------

Исключения

cipher_error	Если ключ невалидный
------------------------------	----------------------

4.2.3 Методы

4.2.3.1 decrypt()

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

Метод для расшифрования

Аргументы

cipher_text	Зашифрованный текст
-------------	---------------------

Исключения

cipher_error	Если зашифрованный текст невалидный
------------------------------	-------------------------------------

Возвращает

Расшифрованный текст

4.2.3.2 encrypt()

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

Метод для зашифровывания

Аргументы

open_text	Открытый текст Строчные символы автоматически преобразуются к прописным. Все символы, не являющиеся буквами, удаляются
-----------	--

Исключения

cipher_error	Если открытый текст невалидный
------------------------------	--------------------------------

Возвращает

Зашифрованный текст

4.2.3.3 getValidCipherText()

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

Валидация зашифрованного текста

Проверяет, что зашифрованный текст содержит только допустимые символы алфавита.

Аргументы

s	Входная строка, представляющая зашифрованный текст.
---	---

Исключения

cipher_error	Если зашифрованный текст содержит недопустимые символы.
------------------------------	---

4.2.3.4 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.
------------------------------	---

4.2.3.5 getValidOpenText()

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

Валидация открытого текста

Проверяет, что открытый текст содержит только допустимые символы алфавита.

Аргументы

s	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

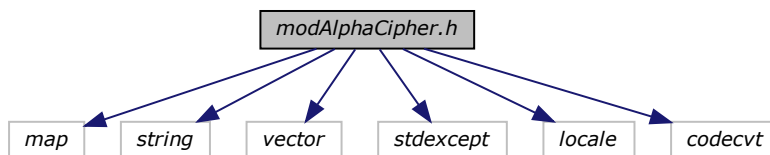
Файлы

5.1 Файл modAlphaCipher.h

Заголовочный файл для модуля Гронсвельда

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class [modAlphaCipher](#)
Шифрование методом Гронсвельда
- class [cipher_error](#)
Обработка исключений Класс, созданный для обработки ошибок

5.1.1 Подробное описание

Заголовочный файл для модуля Гронсвельда

Автор

Ермаков Н.А.

Версия

1.0

Дата

19.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.2 modAlphaCipher.h

[См. документацию.](#)

```
1
9 #pragma once
10 #include <map>
11 #include <string>
12 #include <vector>
13 #include <stdexcept>
14 #include <locale>
15 #include <codecvt>
21 class modAlphaCipher
22 {
23 private:
24     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
25     std::map<wchar_t, int> alphaNum;
26     std::vector<int> key;
27     std::vector<int> convert(const std::wstring& s);
28     std::wstring convert(const std::vector<int>& v);
37     std::wstring getValidKey(const std::wstring& s);
46     std::wstring getValidOpenText(const std::wstring& s);
56     std::wstring getValidCipherText(const std::wstring& s);
57 public:
58     modAlphaCipher() = delete;
64     modAlphaCipher(const std::wstring& skey);
73     std::wstring encrypt(const std::wstring& open_text);
80     std::wstring decrypt(const std::wstring& cipher_text);
81 };
87 class cipher_error : public std::invalid_argument
88 {
89 public:
94     explicit cipher_error(const std::string& what_arg):
95         std::invalid_argument(what_arg) {}
100     explicit cipher_error(const char* what_arg):
101         std::invalid_argument(what_arg) {}
102 };
```

Предметный указатель

- cipher_error, [7](#)
 - cipher_error, [8](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [10](#)
- getValidCipherText
 - modAlphaCipher, [11](#)
- getValidKey
 - modAlphaCipher, [11](#)
- getValidOpenText
 - modAlphaCipher, [12](#)
- modAlphaCipher, [9](#)
 - decrypt, [10](#)
 - encrypt, [10](#)
 - getValidCipherText, [11](#)
 - getValidKey, [11](#)
 - getValidOpenText, [12](#)
 - modAlphaCipher, [10](#)
- modAlphaCipher.h, [13](#)