

GronsfeldRUS

Создано системой Doxygen 1.9.4

1 GronsfeldRUS-Mod	1
2 Иерархический список классов	3
2.1 Иерархия классов	3
3 Алфавитный указатель классов	5
3.1 Классы	5
4 Список файлов	7
4.1 Файлы	7
5 Классы	9
5.1 Класс modAlphaCipher	9
5.1.1 Подробное описание	10
5.1.2 Конструктор(ы)	10
5.1.2.1 modAlphaCipher()	10
5.1.3 Методы	10
5.1.3.1 convert() [1/2]	10
5.1.3.2 convert() [2/2]	11
5.1.3.3 decrypt()	11
5.1.3.4 encrypt()	11
5.1.3.5 getValidKey()	12
5.1.3.6 getValidText()	12
5.2 Класс modAlphaCipher_error	13
5.2.1 Подробное описание	13
6 Файлы	15
6.1 Файл modAlphaCipher.h	15
6.1.1 Подробное описание	16
6.2 modAlphaCipher.h	16
Предметный указатель	17

Глава 1

GronsfeldRUS-Mod

Глава 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
modAlphaCipher_error	13
modAlphaCipher	9

Глава 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

modAlphaCipher	
Шифрование методом Гронсфельда	9
modAlphaCipher_error	
Класс исключений	13

Глава 4

Список файлов

4.1 Файлы

Полный список документированных файлов.

modAlphaCipher.h	
Заголовочный файл для модуля modAlphaCipher.cpp	15

Глава 5

Классы

5.1 Класс modAlphaCipher

Шифрование методом Гронсфелда

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

Открытые члены

- `modAlphaCipher ()=delete`
запретим конструктор без параметров
- `modAlphaCipher (const std::wstring &skey)`
Конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`
Функция зашифрования текста
- `std::wstring decrypt (const std::wstring &cipher_text)`
Функция расшифрования текста
- `std::wstring checkValid (std::wstring &open_text, std::wstring &cipher_text)`

Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`
Метод преобразования строки в вектор
- `std::wstring convert (const std::vector< int > &v)`
Метод преобразования вектора в строку
- `std::wstring getValidKey (const std::wstring &key)`
Метод получения валидного ключа
- `std::wstring getValidText (const std::wstring &text)`
Метод получения валидного входного текста

Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Русский алфавит по порядку
- `std::map< wchar_t, int > alphaNum`
Ассоциативный массив "номер по символу".
- `std::vector< int > key`
ключ

5.1.1 Подробное описание

Шифрование методом Гронсфелда

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

Предупреждения

Реализация только для русского языка

5.1.2 Конструктор(ы)

5.1.2.1 `modAlphaCipher()`

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

Конструктор для установки ключа

Записывает в переменную класса вектор позиции букв полученные из строки с помощью `convert`

Аргументы

in	skey	Строка-ключ
----	------	-------------

5.1.3 Методы

5.1.3.1 `convert()` [1/2]

```
std::wstring modAlphaCipher::convert (
    const std::vector< int > & v ) [inline], [private]
```

Метод преобразования вектора в строку

Каждой букве присваивается номер согласно алфавиту по порядку

Аргументы

in	v	Вектор
----	---	--------

Возвращает

Строка

5.1.3.2 convert() [2/2]

```
std::vector< int > modAlphaCipher::convert (
    const std::wstring & s ) [inline], [private]
```

Метод преобразования строки в вектор

Каждой букве присваивается номер согласно алфавиту по порядку

Аргументы

in	s	Строка
----	---	--------

Возвращает

Вектор букв со своими позициями

5.1.3.3 decrypt()

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

Функция расшифрования текста

Функция расшифровывает строку зашифрованную методом Гронсфельда

Аргументы

in	cipher_text	Строка с зашифрованным текстом
----	-------------	--------------------------------

5.1.3.4 encrypt()

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

Функция зашифрования текста

Функция шифрует строку методом Гронсфельда

Аргументы

in	open_text	Строка с текстом для зашифрования
----	-----------	-----------------------------------

5.1.3.5 getValidKey()

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

Метод получения валидного ключа

Аргументы

in	key	Ключ
----	-----	------

Возвращает

Строка с валидным ключом

Исключения

modAlphaCipher_error , если	ключ невалидный
---	-----------------

5.1.3.6 getValidText()

```
std::wstring modAlphaCipher::getValidText (
    const std::wstring & text ) [inline], [private]
```

Метод получения валидного входного текста

Аргументы

in	text	Текст для зашифрования/расшифрования
----	------	--------------------------------------

Возвращает

Строка с валидным текстом

Исключения

modAlphaCipher_error , если	текст невалидный
---	------------------

Объявления и описания членов классов находятся в файлах:

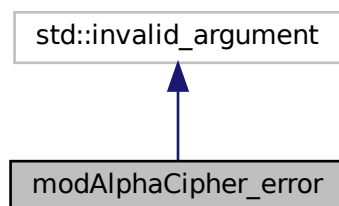
- [modAlphaCipher.h](#)
- modAlphaCipher.cpp

5.2 Класс modAlphaCipher_error

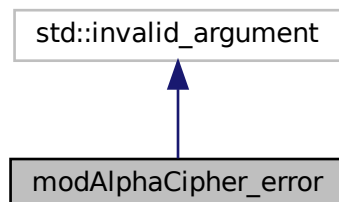
Класс исключений

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

Граф наследования:modAlphaCipher_error:



Граф связей класса modAlphaCipher_error:



Открытые члены

- modAlphaCipher_error (const std::string &what_arg)
- modAlphaCipher_error (const char *what_arg)

5.2.1 Подробное описание

Класс исключений

Класс для отлова исключений

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

Глава 6

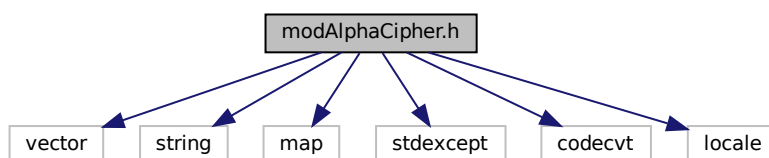
Файлы

6.1 Файл modAlphaCipher.h

Заголовочный файл для модуля modAlphaCipher.cpp.

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class `modAlphaCipher`
Шифрование методом Гронсфельда
- class `modAlphaCipher_error`
Класс исключений

6.1.1 Подробное описание

Заголовочный файл для модуля modAlphaCipher.cpp.

Автор

Храбров А.А.

Версия

1.0

Дата

09.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Тестовый

6.2 modAlphaCipher.h

[См. документацию.](#)

```
1
10 #pragma once
11 #include <vector>
12 #include <string>
13 #include <map>
14 #include <stdexcept>
15 #include <codecvt>
16 #include <locale>
17
24 class modAlphaCipher {
25 private:
26     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
27     std::map<wchar_t, int> alphaNum;
28     std::vector<int> key;
29
36     std::vector<int> convert(const std::wstring& s);
37
44     std::wstring convert(const std::vector<int>& v);
45
52     std::wstring getValidKey(const std::wstring& key);
53
60     std::wstring getValidText(const std::wstring& text);
61
62 public:
63     modAlphaCipher() = delete;
64
70     modAlphaCipher(const std::wstring& skey);
71
76     std::wstring encrypt(const std::wstring& open_text);
77
83     std::wstring decrypt(const std::wstring& cipher_text);
84     std::wstring checkValid(std::wstring& open_text, std::wstring& cipher_text);
85 };
86
90 class modAlphaCipher_error: public std::invalid_argument
91 {
92 public:
93     explicit modAlphaCipher_error (const std::string& what_arg):
94         std::invalid_argument(what_arg) {}
95     explicit modAlphaCipher_error (const char* what_arg):
96         std::invalid_argument(what_arg) {}
97 };
```

Предметный указатель

- convert
 - modAlphaCipher, [10](#), [11](#)
- decrypt
 - modAlphaCipher, [11](#)
- encrypt
 - modAlphaCipher, [11](#)
- getValidKey
 - modAlphaCipher, [12](#)
- getValidText
 - modAlphaCipher, [12](#)
- modAlphaCipher, [9](#)
 - convert, [10](#), [11](#)
 - decrypt, [11](#)
 - encrypt, [11](#)
 - getValidKey, [12](#)
 - getValidText, [12](#)
 - modAlphaCipher, [10](#)
- modAlphaCipher.h, [15](#)
- modAlphaCipher_error, [13](#)