

Шифрование методом Гронсфельда

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	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 modAlphaCipher()	9
4.2.3 Методы	10
4.2.3.1 convert() [1/2]	10
4.2.3.2 convert() [2/2]	10
4.2.3.3 decrypt()	10
4.2.3.4 encrypt()	11
4.2.3.5 getValidCipherText()	11
4.2.3.6 getValidKey()	11
4.2.3.7 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	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Класс исключений для ошибок шифрования	7
modAlphaCipher	Класс для шифрования и дешифрования с использованием модифицированного шифра Альфа	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.h	
Описание класса modAlphaCipher	13

Глава 4

Классы

4.1 Класс `cipher_error`

Класс исключений для ошибок шифрования.

```
#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.
- [cipher_error](#) (const char *what_arg)
Конструктор исключения с сообщением.

4.1.1 Подробное описание

Класс исключений для ошибок шифрования.

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.

Аргументы

what_arg	Сообщение исключения.
----------	-----------------------

4.1.2.2 cipher_error() [2/2]

```

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

```

Конструктор исключения с сообщением.

Аргументы

what_arg	Сообщение исключения (C-style строка).
----------	--

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Класс для шифрования и дешифрования с использованием модифицированного шифра Альфа.

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

Открытые члены

- `modAlphaCipher ()=delete`
Конструктор по умолчанию удален.
- `modAlphaCipher (const string &skey)`
Конструктор с параметром ключа.
- `string encrypt (const string &open_text)`
Шифрует открытый текст.
- `string decrypt (const string &cipher_text)`
Дешифрует зашифрованный текст.

Закрытые члены

- `vector< int > convert (const string &s)`
Преобразует строку в вектор чисел.
- `string convert (const vector< int > &v)`
Преобразует вектор чисел в строку.
- `string getValidKey (const string &s)`
Проверяет и возвращает корректный ключ.
- `string getValidOpenText (const string &s)`
Проверяет и возвращает корректный открытый текст.
- `string getValidCipherText (const string &s)`
Проверяет и возвращает корректный зашифрованный текст.

Закрытые данные

- `wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
- `map< char, int > alphaNum`
- `vector< int > key`

4.2.1 Подробное описание

Класс для шифрования и дешифрования с использованием модифицированного шифра Альфа.

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

```
modAlphaCipher::modAlphaCipher (  
    const string & skey )
```

Конструктор с параметром ключа.

Аргументы

skey	Ключ.
------	-------

4.2.3 Методы

4.2.3.1 convert() [1/2]

```
vector< int > modAlphaCipher::convert (
    const string & s ) [private]
```

Преобразует строку в вектор чисел.

Аргументы

s	Исходная строка.
---	------------------

Возвращает

Вектор числовых значений.

4.2.3.2 convert() [2/2]

```
string modAlphaCipher::convert (
    const vector< int > & v ) [private]
```

Преобразует вектор чисел в строку.

Аргументы

v	Вектор числовых значений.
---	---------------------------

Возвращает

Строка.

4.2.3.3 decrypt()

```
string modAlphaCipher::decrypt (
    const string & cipher_text )
```

Дешифрует зашифрованный текст.

Аргументы

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

Возвращает

Расшифрованный текст.

4.2.3.4 encrypt()

```
string modAlphaCipher::encrypt (  
    const string & open_text )
```

Шифрует открытый текст.

Аргументы

open_text	Текст для шифрования.
-----------	-----------------------

Возвращает

Зашифрованный текст.

4.2.3.5 getValidCipherText()

```
string modAlphaCipher::getValidCipherText (  
    const string & s ) [private]
```

Проверяет и возвращает корректный зашифрованный текст.

Аргументы

s	Зашифрованный текст.
---	----------------------

Возвращает

Корректный зашифрованный текст.

4.2.3.6 getValidKey()

```
string modAlphaCipher::getValidKey (  
    const string & s ) [private]
```

Проверяет и возвращает корректный ключ.

Аргументы

s	Исходный ключ.
---	----------------

Возвращает

Корректный ключ.

4.2.3.7 `getValidOpenText()`

```
string modAlphaCipher::getValidOpenText (  
    const string & s ) [private]
```

Проверяет и возвращает корректный открытый текст.

Аргументы

s	Исходный текст.
---	-----------------

Возвращает

Корректный открытый текст.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

Глава 5

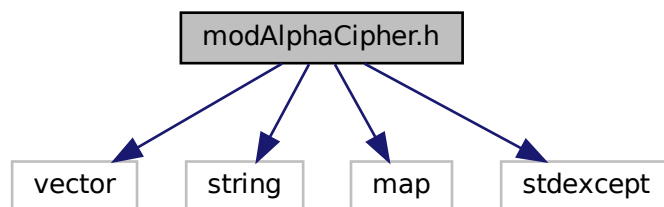
Файлы

5.1 Файл modAlphaCipher.h

Описание класса `modAlphaCipher`.

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class `modAlphaCipher`
Класс для шифрования и дешифрования с использованием модифицированного шифра Альфа.
- class `cipher_error`
Класс исключений для ошибок шифрования.

5.1.1 Подробное описание

Описание класса `modAlphaCipher`.

Автор

Беликов А.М.

Версия

1.0

Дата

02.12.2024

5.2 `modAlphaCipher.h`

[См. документацию.](#)

```
1
2
3 #pragma once
4 #include <vector>
5 #include <string>
6 #include <map>
7 #include <string>
8 #include <string>
9 #include <string>
10 #include <string>
11 #include <string>
12 #include <string>
13
14 using namespace std;
15
16 class modAlphaCipher {
17     private:
18         wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
19         map <char,int> alphaNum;
20         vector <int> key;
21         vector<int> convert(const string& s);
22         string convert(const vector<int>& v);
23         string getValidKey(const string & s);
24         string getValidOpenText(const string & s);
25         string getValidCipherText(const string & s);
26     public:
27         modAlphaCipher()=delete;
28         modAlphaCipher(const string& skey);
29         string encrypt(const string& open_text);
30         string decrypt(const string& cipher_text);
31 };
32
33 class cipher_error : public std::invalid_argument {
34     public:
35         explicit cipher_error(const std::string& what_arg) : std::invalid_argument(what_arg) {}
36         explicit cipher_error(const char* what_arg) : std::invalid_argument(what_arg) {}
37 };
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Предметный указатель

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