

modAlphaCipher.h

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()	12
4.2.3.7 getValidOpenText()	12
5 Файлы	15
5.1 Файл modAlphaCipher.h	15
5.1.1 Подробное описание	15
5.2 modAlphaCipher.h	16
Предметный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

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 обработки ошибок, связанных с шифрованием	7
modAlphaCipher	Класс для реализации шифра на основе модифицированного алфавита	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.h	15
----------------------------------	----

Глава 4

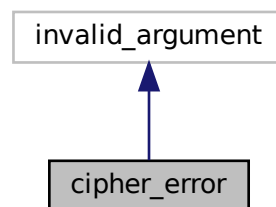
Классы

4.1 Класс `cipher_error`

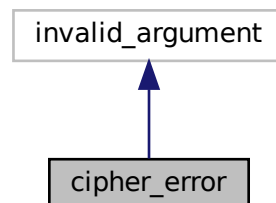
Класс исключений для обработки ошибок, связанных с шифрованием.

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

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- [cipher_error](#) (const string &what_arg)
Конструктор, принимающий строку в качестве аргумента.
- [cipher_error](#) (const char *what_arg)
Конструктор, принимающий C-строку в качестве аргумента.

4.1.1 Подробное описание

Класс исключений для обработки ошибок, связанных с шифрованием.

4.1.2 Конструктор(ы)

4.1.2.1 cipher_error() [1/2]

```

cipher_error::cipher_error (
    const string & what_arg ) [inline], [explicit]

```

Конструктор, принимающий строку в качестве аргумента.

Аргументы

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

4.1.2.2 cipher_error() [2/2]

```

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

```

Конструктор, принимающий C-строку в качестве аргумента.

Аргументы

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

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Класс для реализации шифра на основе модифицированного алфавита.

```
#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)`
Метод для шифрования текста.
- `string decrypt (const string &cipher_text)`
Метод для расшифрования текста.

Закрытые члены

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

Закрытые данные

- `wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Строка, представляющая алфавит.
- `map< char, int > alphaNum`
Сопоставление символов алфавита с их порядковыми номерами.
- `vector< int > key`
Вектор, представляющий ключ.

4.2.1 Подробное описание

Класс для реализации шифра на основе модифицированного алфавита.

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

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

Конструктор с параметром ключа.

Аргументы

skey	Ключ в виде строки.
------	---------------------

4.2.3 Методы

4.2.3.1 convert() [1/2]

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

Преобразует строку в вектор целых чисел.

Аргументы

s	Входная строка.
---	-----------------

Возвращает

Вектор целых чисел.

4.2.3.2 convert() [2/2]

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

Преобразует вектор целых чисел в строку.

Аргументы

v	Вектор целых чисел.
---	---------------------

Возвращает

Строка.

4.2.3.3 decrypt()

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

Метод для расшифрования текста.

Аргументы

cipher_text	Зашифрованный текст для расшифрования.
-------------	--

Возвращает

Расшифрованный текст.

Исключения

cipher_error	Если текст некорректен.
------------------------------	-------------------------

4.2.3.4 encrypt()

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

Метод для шифрования текста.

Аргументы

open_text	Открытый текст для шифрования.
-----------	--------------------------------

Возвращает

Зашифрованный текст.

Исключения

cipher_error	Если текст некорректен.
------------------------------	-------------------------

4.2.3.5 getValidCipherText()

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

Проверяет и получает корректный зашифрованный текст.

Аргументы

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

Возвращает

Корректный зашифрованный текст.

Исключения

cipher_error	Если текст некорректен.
------------------------------	-------------------------

4.2.3.6 getValidKey()

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

Проверяет и получает корректный ключ.

Аргументы

s	Ключ в виде строки.
---	---------------------

Возвращает

Корректная строка ключа.

Исключения

cipher_error	Если ключ некорректен.
------------------------------	------------------------

4.2.3.7 getValidOpenText()

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

Проверяет и получает корректный открытый текст.

Аргументы

s	Открытый текст.
---	-----------------

Возвращает

Корректный открытый текст.

Исключения

cipher_error	Если текст некорректен.
------------------------------	-------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

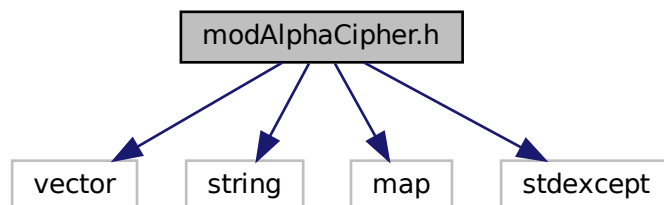
Глава 5

Файлы

5.1 Файл modAlphaCipher.h

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class `modAlphaCipher`
Класс для реализации шифра на основе модифицированного алфавита.
- class `cipher_error`
Класс исключений для обработки ошибок, связанных с шифрованием.

5.1.1 Подробное описание

Автор

Сверчков А.Д.

Версия

1.0

Дата

25.11.2024

5.2 modAlphaCipher.h

[См. документацию.](#)

```
1
6 #pragma once
7 #include <vector>
8 #include <string>
9 #include <map>
10 #include <stdexcept>
11
12 using namespace std;
13
14 class modAlphaCipher {
15     private:
16         wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
17
18         map <char,int> alphaNum;
19
20         vector <int> key;
21
22         vector<int> convert(const string& s);
23
24         string convert(const vector<int>& v);
25
26         string getValidKey(const string & s);
27
28         string getValidOpenText(const string & s);
29
30         string getValidCipherText(const string & s);
31
32     public:
33         modAlphaCipher()=delete;
34
35         modAlphaCipher(const string& skey);
36
37         string encrypt(const string& open_text);
38
39         string decrypt(const string& cipher_text);
40 };
41
42 class cipher_error: public invalid_argument {
43     public:
44         explicit cipher_error (const string& what_arg):
45             invalid_argument(what_arg) {}
46
47         explicit cipher_error (const char* what_arg):
48             invalid_argument(what_arg) {}
49 };
50
```

Предметный указатель

- cipher_error, [7](#)
 - cipher_error, [8](#)
- convert
 - modAlphaCipher, [10](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [11](#)
- getValidCipherText
 - modAlphaCipher, [11](#)
- getValidKey
 - modAlphaCipher, [12](#)
- getValidOpenText
 - modAlphaCipher, [12](#)
- modAlphaCipher, [8](#)
 - convert, [10](#)
 - decrypt, [10](#)
 - encrypt, [11](#)
 - getValidCipherText, [11](#)
 - getValidKey, [12](#)
 - getValidOpenText, [12](#)
 - modAlphaCipher, [9](#)
- modAlphaCipher.h, [15](#)