

Удалов 23ПТ1

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_eggot . . . . .	7
4.1.1 Подробное описание . . . . .	8
4.2 Класс modAlphaCipher . . . . .	8
4.2.1 Подробное описание . . . . .	8
4.2.2 Конструктор(ы) . . . . .	8
4.2.2.1 modAlphaCipher() . . . . .	8
5 Файлы	11
5.1 Файл 1.h . . . . .	11
5.1.1 Подробное описание . . . . .	12
5.2 1.h . . . . .	12
Предметный указатель	13



# 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.

<a href="#">cipher_error</a>	Класс для обработки исключений . . . . .	7
<a href="#">modAlphaCipher</a>	Класс для шифрования и расшифрования шифра табличной маршрутной перестановки . . . . .	8





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">1.h</a>	Описание класса <a href="#">modAlphaCipher</a> . . . . .	<a href="#">11</a>
---------------------	----------------------------------------------------------	--------------------



## Глава 4

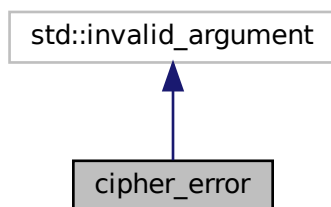
# Классы

### 4.1 Класс `cipher_error`

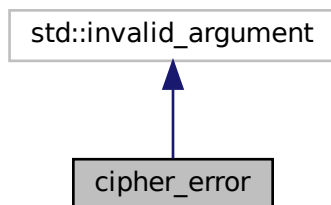
Класс для обработки исключений

```
#include <1.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

### 4.1.1 Подробное описание

Класс для обработки исключений

Объявления и описания членов класса находятся в файле:

- [1.h](#)

## 4.2 Класс modAlphaCipher

Класс для шифрования и расшифрования шифра табличной маршрутной перестановки

```
#include <1.h>
```

## Открытые члены

- `modAlphaCipher ()=delete`  
Запрещающий конструктор без параметров
- [modAlphaCipher](#) (`const std::wstring &skey`)  
Конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`  
Метод, предназначенный для шифрования шифром табличной маршрутной перестановки
- `std::wstring decrypt (const std::wstring &cipher_text)`  
Метод, предназначенный для расшифрования шифра табличной маршрутной перестановки

### 4.2.1 Подробное описание

Класс для шифрования и расшифрования шифра табличной маршрутной перестановки

### 4.2.2 Конструктор(ы)

#### 4.2.2.1 modAlphaCipher()

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

Конструктор для установки ключа

Аргументы

целочисленное	число ключ
---------------	------------

число, которое пришло на вход записывается в "private" атрибут с названием "key"

Объявления и описания членов класса находятся в файле:

- [1.h](#)



## Глава 5

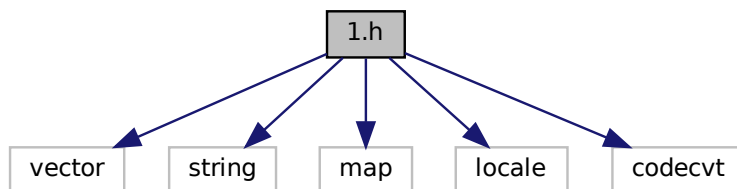
# Файлы

### 5.1 Файл 1.h

Описание класса `modAlphaCipher`.

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

Граф включаемых заголовочных файлов для 1.h:



### Классы

- class `modAlphaCipher`  
Класс для шифрования и расшифрования шифра табличной маршрутной перестановки
- class `cipher_error`  
Класс для обработки исключений

### 5.1.1 Подробное описание

Описание класса `modAlphaCipher`.

Автор

Удалов В.В.

Версия

1.0

Дата

20.11.2024

## 5.2 1.h

[См. документацию.](#)

```
1
8 #pragma once
9 #include <vector>
10 #include <string>
11 #include <map>
12 #include <locale>
13 #include <codecvt>
14 class modAlphaCipher
15 {
16 private:
17     std::wstring_convert<std::codecvt_utf8<wchar_t>, wchar_t> codec;
18     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"; //алфавит по порядку
19     std::map<wchar_t, int> alphaNum; //ассоциативный массив "номер по символу"
20     std::vector<int> key; //ключ
21     std::vector<int> convert(const std::wstring& s); //преобразование строка-вектор
22     std::wstring convert(const std::vector<int>& v); //преобразование вектор-строка
23     std::wstring getValidKey(const std::wstring & s);
24     std::wstring getValidOpenText(const std::wstring & s);
25     std::wstring getValidCipherText(const std::wstring & s);
26 public:
27     modAlphaCipher()=delete; //запретим конструктор без параметров
28     modAlphaCipher(const std::wstring& skey); //конструктор для установки ключа
29     std::wstring encrypt(const std::wstring& open_text); //зашифрование
30     std::wstring decrypt(const std::wstring& cipher_text); //расшифрование
31 };
32 class cipher_error: public std::invalid_argument
33 {
34 public:
35     explicit cipher_error(const std::string& what_arg):
36         std::invalid_argument(what_arg) {}
37     explicit cipher_error(const char* what_arg):
38         std::invalid_argument(what_arg) {}
39 };
```



# Предметный указатель

1.h, [11](#)

cipher\_error, [7](#)

modAlphaCipher, [8](#)

    modAlphaCipher, [8](#)