

## Шифрование методом маршрутной перестановки

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() . . . . .	8
5 Файлы	9
5.1 modCipher.h . . . . .	9
Предметный указатель	11



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::runtime_error	
cipher_error . . . . .	7



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<code>cipher_error</code>	Исключение для ошибок шифрования . . . . .	7
---------------------------	--	---





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

[modCipher.h](#) . . . . . ??



## Глава 4

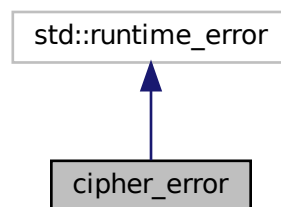
# Классы

### 4.1 Класс `cipher_error`

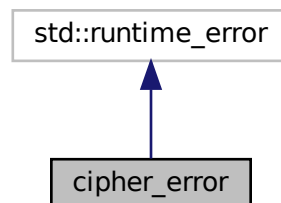
Исключение для ошибок шифрования.

```
#include <modCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- `cipher_error` (`const std::string &what_arg`)  
Конструктор исключения с сообщением.

### 4.1.1 Подробное описание

Исключение для ошибок шифрования.

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 `cipher_error()`

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

Конструктор исключения с сообщением.

Аргументы

<code>what_arg</code>	Сообщение об ошибке.
-----------------------	----------------------

Объявления и описания членов класса находятся в файле:

- `modCipher.h`

## Глава 5

# Файлы

### 5.1 modCipher.h

```
1
8 #pragma once
9 #include <string>
10 #include <stdexcept>
11 #include <vector>
15 class cipher_error : public std::runtime_error {
16 public:
21     explicit cipher_error(const std::string& what_arg) : std::runtime_error(what_arg) {}
22 };
28 std::string prepareText(const std::string& text);
35 std::string coder(int key, const std::string& text);
42 std::string decoder(int key, const std::string& text);
49 int getValidKey(int key, const std::string& text);
```



# Предметный указатель

cipher\_error, [7](#)  
    cipher\_error, [8](#)