





---

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 . . . . .	7
4.1.1 Подробное описание . . . . .	7
4.1.2 Конструктор(ы) . . . . .	8
4.1.2.1 Cipher() . . . . .	8
4.1.3 Методы . . . . .	8
4.1.3.1 decrypt() . . . . .	8
4.1.3.2 encrypt() . . . . .	8
4.1.3.3 getValidCipherText() . . . . .	9
4.1.3.4 getValidKey() . . . . .	9
4.1.3.5 getValidOpenText() . . . . .	10
4.1.3.6 set_key() . . . . .	10
4.1.3.7 set_tableform() . . . . .	11
4.2 Класс cipher_error . . . . .	11
4.2.1 Подробное описание . . . . .	12
4.2.2 Конструктор(ы) . . . . .	12
4.2.2.1 cipher_error() [1/2] . . . . .	12
4.2.2.2 cipher_error() [2/2] . . . . .	12
5 Файлы	15
5.1 Файл Cipher.h . . . . .	15
5.1.1 Подробное описание . . . . .	15
Предметный указатель	17



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Cipher . . . . .	7
invalid_argument	
cipher_error . . . . .	11



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">Cipher</a>	Шифрование методом табличной перестановки . . . . .	<a href="#">7</a>
<a href="#">cipher_error</a>	Класс-исключение . . . . .	<a href="#">11</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">Cipher.h</a>	
Класс-исключение . . . . .	15



## Глава 4

# Классы

### 4.1 Класс Cipher

Шифрование методом табличной перестановки

```
#include <Cipher.h>
```

Открытые члены

- `Cipher ()=delete`  
Конструктор по умолчанию запрещён
- `Cipher (std::wstring &ws_key)`  
Конструктор принимает ключ (количество столбцов в таблице)
- `std::wstring encrypt (std::wstring &ws_open_text)`  
Метод для зашифрования
- `std::wstring decrypt (const std::wstring &ws_cipher_text)`  
Метод для расшифрования
- `void set_tableform (const std::wstring &ws_text)`  
Формирование информации о таблице
- `void set_key (std::wstring &ws_key)`  
Установка нового ключа
- `int getValidKey (std::wstring &ws_key)`  
Проверка на правильность ключа
- `std::wstring getValidOpenText (const std::wstring &ws_open_text)`  
Проверка на правильность текста для зашифровки
- `std::wstring getValidCipherText (const std::wstring &ws_cipher_text)`  
Проверка на правильность текста для расшифровки

#### 4.1.1 Подробное описание

Шифрование методом табличной перестановки

Ключ устанавливается в конструкторе, а также с помощью метода `set_key`. Для зашифрования и расшифрования используются методы `encrypt` и `decrypt`. Метод `set_tableform` - вспомогательный. Методы: `getValidKey`, `getValidOpenText`, `getValidCipherText` - специализируются на проверке входных данных.

Предупреждения

Реализация только для русского языка. С использованием `wstring`. Шифрование табличной перестановкой

## 4.1.2 Конструктор(ы)

### 4.1.2.1 Cipher()

```
Cipher::Cipher (
    std::wstring & ws_key )
```

Конструктор принимает ключ (количество столбцов в таблице)

Конструктор, который принимает на вход ключ, устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

## 4.1.3 Методы

### 4.1.3.1 decrypt()

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

Метод для расшифрования

Метод decrypt расшифровывает текст.

Аргументы

cipher_text	
-------------	--

Возвращает

Зашифрованный текст.

### 4.1.3.2 encrypt()

```
std::wstring Cipher::encrypt (
    std::wstring & open_text )
```

Метод для зашифрования

Метод `encrypt` зашифровывает текст.

Аргументы

<code>open_text</code>	
------------------------	--

Возвращает

Зашифрованный текст

#### 4.1.3.3 `getValidCipherText()`

```
std::wstring Cipher::getValidCipherText (
    const std::wstring & ws_cipher_text ) [inline]
```

Проверка на правильность текста для расшифровки

метод проверяет зашифрованный текст на правильность.

Аргументы

<code>ws_cipher_text</code>	
-----------------------------	--

Возвращает

Зашифрованный текст

Исключения

<a href="#"><code>cipher_error</code></a> ,если	текст пустой или невалидный
---	-----------------------------

#### 4.1.3.4 `getValidKey()`

```
int Cipher::getValidKey (
    std::wstring & ws_key ) [inline]
```

Проверка на правильность ключа

Метод проверяет ключ на правильность.

Аргументы

ws_key	
--------	--

Возвращает

Ключ

Исключения

cipher_error,если	ключ пустой или невалидный
-------------------	----------------------------

#### 4.1.3.5 getValidOpenText()

```
std::wstring Cipher::getValidOpenText (
    const std::wstring & ws_open_text ) [inline]
```

Проверка на правильность текста для зашифровки

Метод проверяет открытый текст на правильность. Строчные буквы становятся прописными. Все не-буквы удаляются.

Аргументы

ws_open_text	
--------------	--

Возвращает

Текст для расшифровки

Исключения

cipher_error,если	текст пустой
-------------------	--------------

#### 4.1.3.6 set\_key()

```
void Cipher::set_key (
    std::wstring & ws_key )
```

Установка нового ключа

Метод, который принимает на вход ключ, устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

#### 4.1.3.7 set\_tableform()

```
void Cipher::set_tableform (
    const std::wstring & open_text )
```

Формирование информации о таблице

Принимает текст для зашифровки, далее по нему формирует кол-во строк в таблице, а также получает длину текста.

Аргументы

open_text	
-----------	--

Объявления и описания членов классов находятся в файлах:

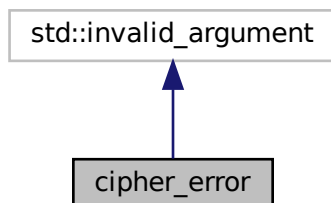
- [Cipher.h](#)
- [Cipher.cpp](#)

## 4.2 Класс cipher\_error

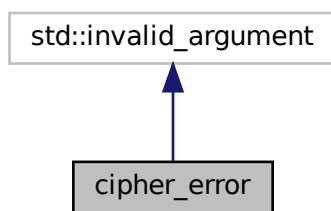
Класс-исключение

```
#include <Cipher.h>
```

Граф наследования: cipher\_error:



Граф связей класса `cipher_error`:



## Открытые члены

- `cipher_error` (`const std::string &what_arg`)  
Принимает строку, поднимает исключение
- `cipher_error` (`const char *what_arg`)  
Принимает си строку, поднимает исключение

## 4.2.1 Подробное описание

Класс-исключение

## 4.2.2 Конструктор(ы)

### 4.2.2.1 `cipher_error()` [1/2]

```

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

Принимает строку, поднимает исключение

Аргументы

what_arg	
----------	--

### 4.2.2.2 `cipher_error()` [2/2]

```

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



Принимает си строку, поднимает исключение

Аргументы

what_arg	
----------	--

Объявления и описания членов класса находятся в файле:

- [Cipher.h](#)



## Глава 5

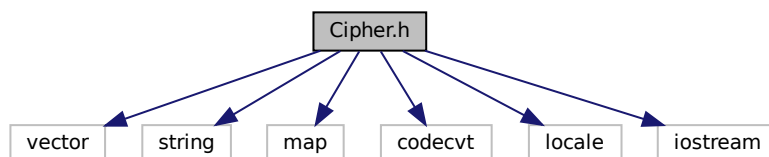
# Файлы

### 5.1 Файл Cipher.h

Класс-исключение

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

Граф включаемых заголовочных файлов для Cipher.h:



Классы

- class [Cipher](#)  
Шифрование методом табличной перестановки
- class [cipher\\_error](#)  
Класс-исключение

#### 5.1.1 Подробное описание

Класс-исключение



# Предметный указатель

- Cipher, [7](#)
  - Cipher, [8](#)
  - decrypt, [8](#)
  - encrypt, [8](#)
  - getValidCipherText, [9](#)
  - getValidKey, [9](#)
  - getValidOpenText, [10](#)
  - set\_key, [10](#)
  - set\_tableform, [11](#)
- Cipher.h, [15](#)
- cipher\_error, [11](#)
  - cipher\_error, [12](#)
- decrypt
  - Cipher, [8](#)
- encrypt
  - Cipher, [8](#)
- getValidCipherText
  - Cipher, [9](#)
- getValidKey
  - Cipher, [9](#)
- getValidOpenText
  - Cipher, [10](#)
- set\_key
  - Cipher, [10](#)
- set\_tableform
  - Cipher, [11](#)