

modAlphaCipher

1.0

Создано системой Doxygen 1.9.1



---

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 . . . . .	8
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() . . . . .	9
4.1.3.3 getValidCipherText() . . . . .	9
4.1.3.4 getValidKey() . . . . .	9
4.1.3.5 getValidOpenText() . . . . .	11
4.2 Класс cipher_error . . . . .	11
4.2.1 Подробное описание . . . . .	12
5 Файлы	13
5.1 Файл Table.h . . . . .	13
5.1.1 Подробное описание . . . . .	14
Предметный указатель	15



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Cipher . . . . .	7
std::invalid_argument	
cipher_error . . . . .	11



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">Cipher</a>	Класс, реализующий шифрование на основе таблицы . . . . .	<a href="#">7</a>
<a href="#">cipher_error</a>	Пользовательское исключение для ошибок, связанных с шифром . . . . .	<a href="#">11</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">Table.h</a>	Объявление класса <a href="#">Cipher</a> для шифрования на основе таблицы . . . . .	13
-------------------------	---	----



## Глава 4

# Классы

### 4.1 Класс Cipher

Класс, реализующий шифрование на основе таблицы

```
#include <Table.h>
```

Открытые члены

- `Cipher ()=delete`  
Удаленный конструктор по умолчанию
- `Cipher (const std::string &skey)`  
Параметризованный конструктор для инициализации ключа
- `std::string encrypt (const std::string &open_text)`  
Шифрует открытый текст на основе таблицы
- `std::string decrypt (const std::string &cipher_text)`  
Дешифрует зашифрованный текст на основе таблицы

Закрытые члены

- `std::string getValidOpenText (const std::string &s)`  
Проверяет и возвращает валидный открытый текст
- `std::string getValidCipherText (const std::string &s)`  
Проверяет и возвращает валидный зашифрованный текст
- `int getValidKey (const int key, const std::string &s)`  
Проверяет и возвращает валидный ключ

Закрытые данные

- `int key`  
Ключ для шифрования на основе таблицы

### 4.1.1 Подробное описание

Класс, реализующий шифрование на основе таблицы

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 Cipher()

```
Cipher::Cipher (  
    const std::string & skey )
```

Параметризованный конструктор для инициализации ключа

Аргументы

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

### 4.1.3 Методы

#### 4.1.3.1 decrypt()

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

Дешифрует зашифрованный текст на основе таблицы

Аргументы

cipher_text	Входной зашифрованный текст
-------------	-----------------------------

Исключения

<a href="#">cipher_error</a>	Если зашифрованный текст пуст или содержит недопустимые символы
------------------------------	---

Возвращает

std::string Расшифрованный текст

## 4.1.3.2 encrypt()

```
std::string Cipher::encrypt (
    const std::string & open_text )
```

Шифрует открытый текст на основе таблицы

Аргументы

open_text	Входной открытый текст
-----------	------------------------

Исключения

<a href="#">cipher_error</a>	Если открытый текст пуст или содержит недопустимые символы
------------------------------	--

Возвращает

std::string Зашифрованный текст

## 4.1.3.3 getValidCipherText()

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

Проверяет и возвращает валидный зашифрованный текст

Аргументы

s	Входной зашифрованный текст
---	-----------------------------

Возвращает

std::string Валидный зашифрованный текст

Исключения

<a href="#">cipher_error</a>	Если зашифрованный текст пуст или содержит недопустимые символы
------------------------------	---

## 4.1.3.4 getValidKey()

```
int Cipher::getValidKey (
    const int key,
    const std::string & s ) [private]
```

Проверяет и возвращает валидный ключ

## Аргументы

key	Входной ключ
s	Исходная строка ключа (для сообщений об ошибках)

## Возвращает

int Валидный ключ

## Исключения

<a href="#">cipher_error</a>	Если ключ отрицательный или не является числом
------------------------------	--

## 4.1.3.5 getValidOpenText()

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

Проверяет и возвращает валидный открытый текст

## Аргументы

s	Входной открытый текст
---	------------------------

## Возвращает

std::string Валидный открытый текст

## Исключения

<a href="#">cipher_error</a>	Если открытый текст пуст или содержит недопустимые символы
------------------------------	--

Объявления и описания членов класса находятся в файле:

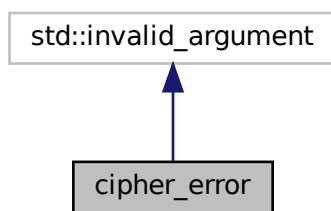
- [Table.h](#)

## 4.2 Класс cipher\_error

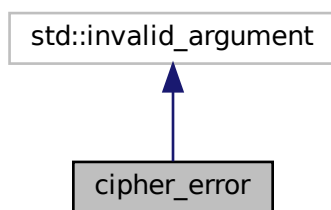
Пользовательское исключение для ошибок, связанных с шифром

```
#include <Table.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)`

#### 4.2.1 Подробное описание

Пользовательское исключение для ошибок, связанных с шифром

Объявления и описания членов класса находятся в файле:

- [Table.h](#)



## Глава 5

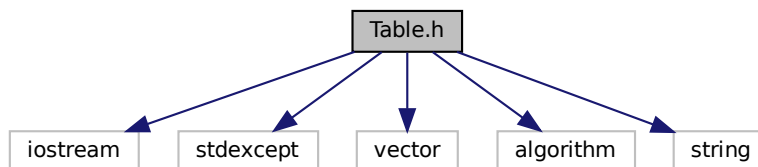
# Файлы

### 5.1 Файл Table.h

Объявление класса [Cipher](#) для шифрования на основе таблицы

```
#include <iostream>
#include <stdexcept>
#include <vector>
#include <algorithm>
#include <string>
```

Граф включаемых заголовочных файлов для Table.h:



### Классы

- class [Cipher](#)  
Класс, реализующий шифрование на основе таблицы
- class [cipher\\_error](#)  
Пользовательское исключение для ошибок, связанных с шифром

### 5.1.1 Подробное описание

Объявление класса [Cipher](#) для шифрования на основе таблицы

Автор

Коновалов К.А

Версия

1.0

Дата

20.11.2023

# Предметный указатель

- Cipher, [7](#)
  - Cipher, [8](#)
  - decrypt, [8](#)
  - encrypt, [8](#)
  - getValidCipherText, [9](#)
  - getValidKey, [9](#)
  - getValidOpenText, [11](#)
- cipher\_error, [11](#)
- decrypt
  - Cipher, [8](#)
- encrypt
  - Cipher, [8](#)
- getValidCipherText
  - Cipher, [9](#)
- getValidKey
  - Cipher, [9](#)
- getValidOpenText
  - Cipher, [11](#)
- Table.h, [13](#)