

Лабораторная работа №4.2

1.0

Создано системой Doxygen 1.9.1

1 Иерархический список классов	1
1.1 Иерархия классов . . . . .	1
2 Алфавитный указатель классов	1
2.1 Классы . . . . .	1
3 Список файлов	1
3.1 Файлы . . . . .	1
4 Классы	2
4.1 Класс cipher_error . . . . .	2
4.2 Класс modAlphaCipher . . . . .	3
4.2.1 Конструктор(ы) . . . . .	3
4.2.2 Методы . . . . .	3
5 Файлы	6
5.1 Файл modAlphaCipher.h . . . . .	6
5.1.1 Подробное описание . . . . .	6
Предметный указатель	9

## 1 Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	2
modAlphaCipher	3

## 2 Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

cipher_error	2
modAlphaCipher	3

## 3 Список файлов

### 3.1 Файлы

Полный список документированных файлов.

[modAlphaCipher.h](#)

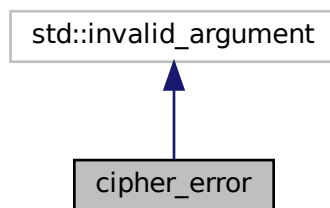
Класс для шифрования перестановочным шифром

[6](#)

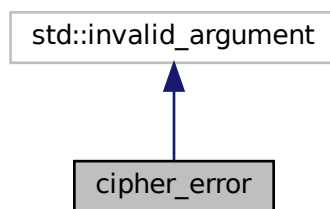
## 4 Классы

### 4.1 Класс cipher\_error

Граф наследования: cipher\_error:



Граф связей класса cipher\_error:



Открытые члены

- cipher\_error (const std::string &what\_arg)
- cipher\_error (const char \*what\_arg)

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

## 4.2 Класс modAlphaCipher

### Открытые члены

- `modAlphaCipher()` = delete  
запрет конструктора без параметров
- `modAlphaCipher(string skey)`  
Конструктор для установки ключа. Ключ является целым положительным числом.
- `string encrypt(string &open_text)`  
Зашифрование текста маршрутным перестановочным шифром.
- `string decrypt(string &cipher_text)`  
Расшифрование текста маршрутным перестановочным шифром.
- `void KeyCheck(string &skey)`  
Проверка ключа на корректность.
- `void LengthCheck(int &skey, string &stext)`  
Проверка длины теста.
- `void TextCheck(string &stext)`  
Проверка теста на пустоту.

### Закрытые данные

- `int key`  
Ключ для шифрования

### 4.2.1 Конструктор(ы)

#### 4.2.1.1 `modAlphaCipher()` `modAlphaCipher::modAlphaCipher (string skey )`

Конструктор для установки ключа. Ключ является целым положительным числом.

#### Аргументы

in	ключ, std::string.	
----	--------------------	--

#### Исключения

<code>cipher_error</code>	в случае некорректного ключа. Параметры исключения: <code>type = invalid_argument, what = "Invalid key"</code>
---------------------------	---

### 4.2.2 Методы

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

Расширение текста маршрутным перестановочным шифром.

Символы текста перставляются по алгоритму маршрутной перестановки: В качестве ключа берётся количество столбцов таблицы. Маршрут записи: сверху вниз, справа налево. Маршрут считывания: по горизонтали слева направо, сверху вниз. Если строка пустая или ключ больше половины шифротекста, возбуждает исключение.

Аргументы

in	строка	с шифротекстом, <code>std::string</code> .
----	--------	--

Возвращает

Строка `std::string` текста.

Исключения

<code>std::cipher_error</code>	в случае некорректного текста. <code>type = invalid_argument, what = "Key is too long", "Empty message"</code>
--------------------------------	---

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

Заширование текста маршрутным перестановочным шифром.

Символы текста перставляются по алгоритму маршрутной перестановки: В качестве ключа берётся количество столбцов таблицы. Маршрут записи: по горизонтали слева направо, сверху вниз. Маршрут считывания: сверху вниз, справа налево. Если строка пустая или ключ больше половины текста для шифрования, возбуждает исключение.

Аргументы

in	строка	с текстом, <code>std::string</code> .
----	--------	---------------------------------------

Возвращает

Строка `std::string` шифротекста.

Исключения

<code>std::cipher_error</code>	в случае некорректного текста. <code>type = invalid_argument, what = "Key is too long", "Empty message"</code>
--------------------------------	---

4.2.2.3 KeyCheck() void modAlphaCipher::KeyCheck (  
string & skey ) [inline]

Проверка ключа на корректность.

Корректный текст: целое положительное число не больше половины длины текста. Если ключ не является целым положительным числом, возбуждает исключение.

Аргументы

in	строка	с ключом, std::string.
----	--------	------------------------

Исключения

std::cipher_error	в случае некорректного ключа. type = invalid_argument, what = "Invalid key"
-------------------	--

4.2.2.4 LengthCheck() void modAlphaCipher::LengthCheck (  
int & skey,  
string & stext ) [inline]

Проверка длины текста.

Корректный текст: строка символов в 2 или более раза длиннее ключа. Если ключ слишком длинный, возбуждает исключение.

Аргументы

in	строка	с текстом, std::string.
----	--------	-------------------------

Исключения

std::cipher_error	в случае некорректного текста. type = invalid_argument, what = "Key is too long"
-------------------	---

4.2.2.5 TextCheck() void modAlphaCipher::TextCheck (  
string & stext ) [inline]

Проверка текста на пустоту.

Если текст пустой, возбуждает исключение.

Аргументы

in	строка	с текстом, std::string.
----	--------	-------------------------

## Исключения

<code>std::cipher_error</code>	в случае некорректного текста. type = invalid_argument, what = "Empty message"
--------------------------------	---

Объявления и описания членов классов находятся в файлах:

- [modAlphaCipher.h](#)
- `modAlphaCipher.cpp`

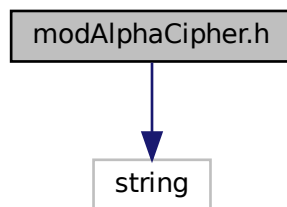
## 5 Файлы

### 5.1 Файл `modAlphaCipher.h`

Класс для шифрования перестановочным шифром

```
#include <string>
```

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



## Классы

- class [modAlphaCipher](#)
- class [cipher\\_error](#)

### 5.1.1 Подробное описание

Класс для шифрования перестановочным шифром

Автор

Соколенко Н.С.

Версия

1.0

Дата

18.12.2022

Авторство

ИБСТ ПГУ





## Предметный указатель

cipher\_error, [2](#)

decrypt

modAlphaCipher, [3](#)

encrypt

modAlphaCipher, [4](#)

KeyCheck

modAlphaCipher, [4](#)

LengthCheck

modAlphaCipher, [5](#)

modAlphaCipher, [3](#)

decrypt, [3](#)

encrypt, [4](#)

KeyCheck, [4](#)

LengthCheck, [5](#)

modAlphaCipher, [3](#)

TextCheck, [5](#)

modAlphaCipher.h, [6](#)

TextCheck

modAlphaCipher, [5](#)