

Метод 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 AlphaCipher	7
4.1.1 Конструктор(ы)	7
4.1.1.1 AlphaCipher()	7
4.1.2 Методы	8
4.1.2.1 decrypt()	8
4.1.2.2 encrypt()	8
4.1.2.3 getValidCipherText()	9
4.1.2.4 getValidKey()	9
4.1.2.5 getValidOpenText()	10
4.2 Класс cipher_error	10
4.2.1 Подробное описание	11
4.2.2 Конструктор(ы)	11
4.2.2.1 cipher_error() [1/2]	12
4.2.2.2 cipher_error() [2/2]	13
4.3 Класс modAlphaCipher	13
4.3.1 Подробное описание	13
5 Файлы	15
5.1 Файл modAlphaCipher.h	15
5.1.1 Подробное описание	15
5.2 modAlphaCipher.h	16
Предметный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

AlphaCipher	7
std::invalid_argument	
cipher_error	10
modAlphaCipher	13

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

AlphaCipher	7
cipher_error	
Исключение для ошибок шифрования	10
modAlphaCipher	
Шифрование методом табличной маршрутной перестановки	13

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.h	
Описание класса modAlphaCipher	15

Глава 4

Классы

4.1 Класс AlphaCipher

Открытые члены

- AlphaCipher ()=delete
Запрет конструктора по умолчанию
- AlphaCipher (const std::wstring &key_str)
Конструктор класса AlphaCipher.
- std::wstring encrypt (const std::wstring &text)
Шифрует открытый текст.
- std::wstring decrypt (const std::wstring &encrypted_text)
Расшифровывает зашифрованный текст.
- int getValidKey (const std::wstring &key_str)
Проверяет и возвращает действительный ключ шифрования.
- std::wstring getValidOpenText (const std::wstring &s)
Проверяет и возвращает действительный открытый текст для шифрования.
- std::wstring getValidCipherText (const std::wstring &s)
Проверяет и возвращает действительный зашифрованный текст для расшифровки.

4.1.1 Конструктор(ы)

4.1.1.1 AlphaCipher()

```
AlphaCipher::AlphaCipher (  
    const std::wstring & key_str )
```

Конструктор класса AlphaCipher.

Инициализирует объект шифрования с заданным ключом.

Аргументы

key_str	Строка, представляющая ключ шифрования.
---------	---

Исключения

cipher_error	Исключение, если ключ недействителен.
------------------------------	---------------------------------------

4.1.2 Методы

4.1.2.1 decrypt()

```
std::wstring AlphaCipher::decrypt (  
    const std::wstring & encrypted_text )
```

Расшифровывает зашифрованный текст.

Дешифрует переданный зашифрованный текст с использованием установленного ключа.

Аргументы

encrypted_text	Зашифрованный текст для расшифровки.
----------------	--------------------------------------

Возвращает

Открытый текст.

4.1.2.2 encrypt()

```
std::wstring AlphaCipher::encrypt (  
    const std::wstring & text )
```

Шифрует открытый текст.

Зашифровывает переданный текст с использованием установленного ключа.

Аргументы

text	Открытый текст для шифрования.
------	--------------------------------

Возвращает

Зашифрованный текст.

Исключения

<code>cipher_error</code>	Исключение, если длина ключа превышает половину длины текста.
---------------------------	---

4.1.2.3 `getValidCipherText()`

```
std::wstring AlphaCipher::getValidCipherText (
    const std::wstring & s )
```

Проверяет и возвращает действительный зашифрованный текст для расшифровки.

Удаляет недопустимые символы и проверяет на наличие звёздочек.

Аргументы

<code>s</code>	Зашифрованный текст для проверки.
----------------	-----------------------------------

Возвращает

Проверенный зашифрованный текст.

Исключения

<code>cipher_error</code>	Исключение, если зашифрованный текст пуст или содержит недопустимые символы.
---------------------------	--

4.1.2.4 `getValidKey()`

```
int AlphaCipher::getValidKey (
    const std::wstring & key_str )
```

Проверяет и возвращает действительный ключ шифрования.

Проверяет, что ключ состоит только из цифр и больше нуля.

Аргументы

<code>key_str</code>	Строка, представляющая ключ шифрования.
----------------------	---

Возвращает

Проверенный ключ.

Исключения

cipher_error	Исключение, если ключ недействителен или пуст.
------------------------------	--

4.1.2.5 getValidOpenText()

```
std::wstring AlphaCipher::getValidOpenText (
    const std::wstring & s )
```

Проверяет и возвращает действительный открытый текст для шифрования.

Удаляет недопустимые символы и преобразует текст в верхний регистр.

Аргументы

s	Открытый текст для проверки.
---	------------------------------

Возвращает

Проверенный открытый текст.

Исключения

cipher_error	Исключение, если открытый текст пуст или содержит недопустимые символы.
------------------------------	---

Объявления и описания членов класса находятся в файле:

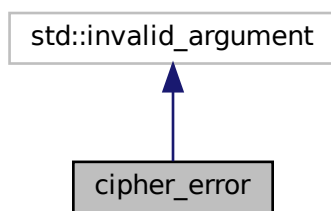
- [modAlphaCipher.h](#)

4.2 Класс cipher_error

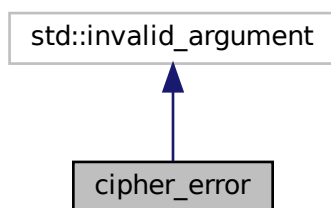
Исключение для ошибок шифрования.

```
#include <modAlphaCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

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

4.2.1 Подробное описание

Исключение для ошибок шифрования.

Наследуется от `std::invalid_argument` и используется для обработки ошибок, связанных с шифрованием.

4.2.2 Конструктор(ы)

4.2.2.1 `cipher_error()` [1/2]

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

Конструктор исключения `cipher_error` с сообщением об ошибке.

Аргументы

what_arg	Сообщение об ошибке в виде строки.
----------	------------------------------------

4.2.2.2 cipher_error() [2/2]

```

cipher_error::cipher_error (
    const char * what_arg ) [explicit]

```

Конструктор исключения [cipher_error](#) с сообщением об ошибке.

Аргументы

what_arg	Сообщение об ошибке в виде C-строки.
----------	--------------------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.3 Класс modAlphaCipher

Шифрование методом табличной маршрутной перестановки

```
#include <modAlphaCipher.h>
```

4.3.1 Подробное описание

Шифрование методом табличной маршрутной перестановки

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы encrypt и decrypt.

Предупреждения

Реализация только для русского языка

Класс, реализующий табличный шифр. Класс реализует шифр табличной маршрутной перестановки.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

Глава 5

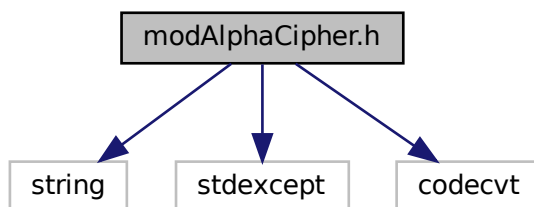
Файлы

5.1 Файл modAlphaCipher.h

Описание класса `modAlphaCipher`.

```
#include <string>
#include <stdexcept>
#include <codecvt>
```

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Классы

- class `AlphaCipher`
- class `cipher_error`

Исключение для ошибок шифрования.

5.1.1 Подробное описание

Описание класса `modAlphaCipher`.

Автор

Дьякова С.М.

Версия

1.0

Дата

03.12.2024

Авторство

ИБСТ ПГУ

5.2 modAlphaCipher.h

[См. документацию.](#)

```
1
10 #pragma once
11 #include <string>
12 #include <stdexcept>
13 #include <codecvt>
14
26 class AlphaCipher
27 {
28 private:
29     int key;
30
31 public:
32     AlphaCipher() = delete;
33
34     AlphaCipher(const std::wstring& key_str);
35
36     std::wstring encrypt(const std::wstring& text);
37
38     std::wstring decrypt(const std::wstring& encrypted_text);
39
40     int getValidKey(const std::wstring& key_str);
41
42     std::wstring getValidOpenText(const std::wstring & s);
43
44     std::wstring getValidCipherText(const std::wstring & s);
45 };
46
104 class cipher_error : public std::invalid_argument
105 {
106 public:
107     explicit cipher_error(const std::string& what_arg);
108
109     explicit cipher_error(const char* what_arg);
110 };
120
```

Предметный указатель

- AlphaCipher, [7](#)
 - AlphaCipher, [7](#)
 - decrypt, [8](#)
 - encrypt, [8](#)
 - getValidCipherText, [9](#)
 - getValidKey, [9](#)
 - getValidOpenText, [10](#)
- cipher_error, [10](#)
 - cipher_error, [11](#), [13](#)
- decrypt
 - AlphaCipher, [8](#)
- encrypt
 - AlphaCipher, [8](#)
- getValidCipherText
 - AlphaCipher, [9](#)
- getValidKey
 - AlphaCipher, [9](#)
- getValidOpenText
 - AlphaCipher, [10](#)
- modAlphaCipher, [13](#)
- modAlphaCipher.h, [15](#)