

Удалов 23ПТ1

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 Конструктор(ы)	8
4.1.1.1 cipher_error() [1/2]	8
4.1.1.2 cipher_error() [2/2]	8
4.2 Класс modAlphaCipher	9
4.2.1 Подробное описание	9
4.2.2 Методы	9
4.2.2.1 decrypt()	9
4.2.2.2 encrypt()	10
5 Файлы	11
5.1 Файл modAlphaCipher.h	11
5.1.1 Подробное описание	11
5.2 modAlphaCipher.h	12
Предметный указатель	13

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	7
modAlphaCipher	9

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

<code>cipher_error</code>	7
<code>modAlphaCipher</code>	9

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

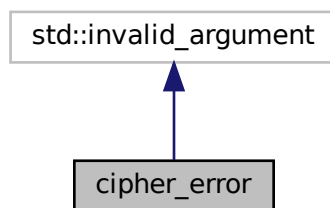
modAlphaCipher.h	
Заголовочный файл для модуля modAlphaCipher	11

Глава 4

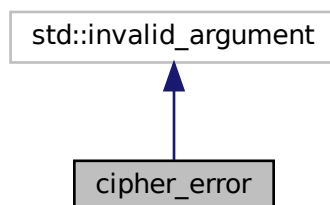
Классы

4.1 Класс cipher_error

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- [cipher_error](#) (const std::string &what_arg)
Валидация текста
- [cipher_error](#) (const char *what_arg)
Валидация текста

4.1.1 Конструктор(ы)

4.1.1.1 cipher_error() [1/2]

```

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

```

Валидация текста

Аргументы

in	what_arg	
		После проверки текста при помощи <code>getValidText</code> , если возвращенный параметр вызывает исключения, причина вызова исключений передается в качестве параметра функции. @warnings Текс не должен быть пустой строкой и не должен содержать пробелы. В тексте могут содержаться знаки перпинания и цифры.

Возвращает

result Расшифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	----------------------------------------------

4.1.1.2 cipher_error() [2/2]

```

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

```

Валидация текста

Аргументы

in	what_arg	После проверки номера операции, если возвращенный параметр вызывает исключения, причина вызова исключений передается в качестве параметра функции. @warnings Текс не должен быть пустой строкой и не должен содержать пробелы. В тексте могут содержаться знаки перпинания и цифры.
----	----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Возвращает

result Расшифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	----------------------------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

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

Открытые члены

- modAlphaCipher (const string &key1)
- std::string [encrypt](#) (const std::string &open_text)
Зашифрование
- std::string [decrypt](#) (const std::string &cipher_text)
Расшифрование

4.2.1 Подробное описание

Автор

Удалов В.В.

Дата

20.11.2024

4.2.2 Методы

4.2.2.1 decrypt()

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

Расшифрование

Аргументы

in	text	Зашифрованный текст на английском языке @warnings Текс не должен быть пустой строкой и не должен содержать пробелы. В тексте могут содержаться знаки перпинания и цифры.
----	------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Возвращает

result Расшифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	----------------------------------------------

4.2.2.2 encrypt()

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

Зашифрование

Аргументы

in	text	Открытый текст на английском языке @warnings Текс не должен быть пустой строкой и не должен содержать пробелы. В тексте могут содержаться знаки перпинания и цифры.
----	------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

Возвращает

result Зашифрованный текст

Исключения

cipher_error	Если в качестве текста введена пустая строка
------------------------------	----------------------------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

Глава 5

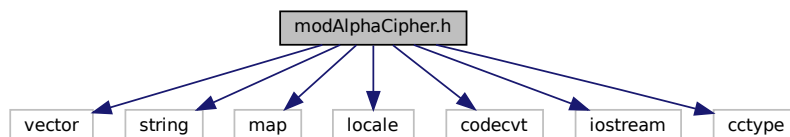
Файлы

5.1 Файл modAlphaCipher.h

Заголовочный файл для модуля `modAlphaCipher`.

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

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Классы

- class `modAlphaCipher`
- class `cipher_error`

5.1.1 Подробное описание

Заголовочный файл для модуля `modAlphaCipher`.

Класс вызова исключений

Шифрование методом Маршрутной перестановки @detalies Для зашифрования и расшифрования сообщения предназначены методы `encrypt` и `decrypt`. Текст на обработку передается до выбора операции. Доступна отмена выполнения программы. @warnings Реализация только для английского языка!

Автор

- Удалов В.В.

Версия

- 1.0.0

Дата

- 20.11.2024

-

5.2 modAlphaCipher.h

[См. документацию.](#)

```
1
7 #pragma once
8 #include <vector>
9 #include <string>
10 #include <map>
11 #include <locale>
12 #include <codecvt>
13 #include <iostream>
14 #include <cctype>
15 using namespace std;
16
26 class modAlphaCipher
27 {
28 private:
29     int key;
34     std::string getValidOpenText(const std::string & s);
39     std::string getValidCipherText(const std::string & s);
40 public:
41     modAlphaCipher()=delete;
42     modAlphaCipher(const string& key1);
49     std::string encrypt(const std::string& open_text);
56     std::string decrypt(const std::string& cipher_text);
57 };
62 class cipher_error: public std::invalid_argument
63 {
64 public:
71     explicit cipher_error (const std::string& what_arg):
72         std::invalid_argument(what_arg) {}
79     explicit cipher_error (const char* what_arg):
80         std::invalid_argument(what_arg) {}
81 };
```


Предметный указатель

- `cipher_error`, [7](#)
 - `cipher_error`, [8](#)
- `decrypt`
 - `modAlphaCipher`, [9](#)
- `encrypt`
 - `modAlphaCipher`, [10](#)
- `modAlphaCipher`, [9](#)
 - `decrypt`, [9](#)
 - `encrypt`, [10](#)
- `modAlphaCipher.h`, [11](#)