

modBetaCipher

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_error	7
4.2 Класс Shifrovshik	8
4.2.1 Конструктор(ы)	8
4.2.1.1 Shifrovshik()	8
4.2.2 Методы	8
4.2.2.1 getValidCipherText()	9
4.2.2.2 getValidKey()	9
4.2.2.3 getValidOpenText()	9
4.2.2.4 Rasshifr()	10
4.2.2.5 Shifr()	10
5 Файлы	11
5.1 Файл modBetaCipher.h	11
5.1.1 Подробное описание	11
Предметный указатель	13

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

invalid_argument	
cipher_error	7
Shifrovshik	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	7
Shifrovshik	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

[modBetaCipher.h](#)

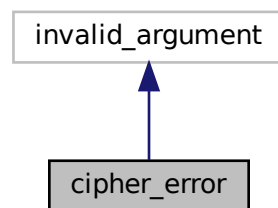
Класс для шифрования методом табличной маршрутной перестановки 11

Глава 4

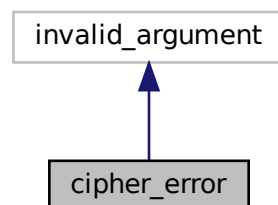
Классы

4.1 Класс cipher_error

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

Объявления и описания членов класса находятся в файле:

- [modBetaCipher.h](#)

4.2 Класс Shifrovshik

Открытые члены

- [Shifrovshik](#) ()=delete
Ключ для шифрования и расшифрования
- `Shifrovshik (int w)`
- `string Rasshifr (Shifrovshik w, string &s)`
Метод для установления шифрования и расшифрования
- `string Shifr (Shifrovshik w, string &s)`
Расшифрование
- `int getValidKey (const int k)`
Зашифрование
- `string getValidOpenText (const string &s)`
Метод для проверки ключа на обработку исключений
- `string getValidCipherText (const string &s)`
Метод, проверяющий текст для зашифровки на валидность

4.2.1 Конструктор(ы)

4.2.1.1 Shifrovshik()

`Shifrovshik::Shifrovshik ()` [delete]

Ключ для шифрования и расшифрования

Аргументы

Только	целые числа
--------	-------------

4.2.2 Методы

4.2.2.1 getValidCipherText()

```
string Shifrovshik::getValidCipherText (
    const string & s )
```

Метод, проверяющий текст для зашифровки на валидность

Аргументы

in	s	Текст на английском языке
----	---	---------------------------

4.2.2.2 getValidKey()

```
int Shifrovshik::getValidKey (
    const int k )
```

Зашифрование

Аргументы

in	s	Открытый текст на английском языке @warnings Текст не должен быть пустой строкой, содержать пробелы.
----	---	--

Возвращает

Расшифрованный текст

Исключения

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

4.2.2.3 getValidOpenText()

```
string Shifrovshik::getValidOpenText (
    const string & s )
```

Метод для проверки ключа на обработку исключений

Аргументы

Только	целые числа
--------	-------------

4.2.2.4 Rasshifr()

```
string Shifrovshik::Rasshifr (
    Shifrovshik w,
    string & s )
```

Метод для установленич шифрования и расшифрования

Аргументы

Только	целые числа
--------	-------------

Исключения

cipher_error	Если ключ не число или пустая строка
------------------------------	--------------------------------------

4.2.2.5 Shifr()

```
string Shifrovshik::Shifr (
    Shifrovshik w,
    string & s )
```

Расшифрование

Аргументы

Открытый	текст на английском языке @warnings Текст не должен быть пустой строкой, содержать пробелы.
----------	---

Возвращает

Расшифрованный текст

Исключения

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

Объявления и описания членов класса находятся в файле:

- [modBetaCipher.h](#)

Глава 5

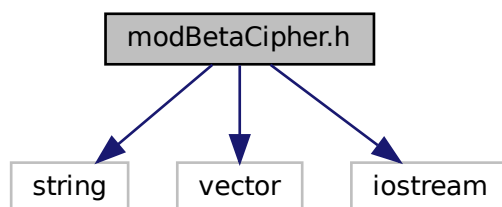
Файлы

5.1 Файл modBetaCipher.h

Класс для шифрования методом табличной маршрутной перестановки

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

Граф включаемых заголовочных файлов для modBetaCipher.h:



Классы

- class [Shifrovshik](#)
- class [cipher_error](#)

5.1.1 Подробное описание

Класс для шифрования методом табличной маршрутной перестановки

Автор

Фейгина Е.А.

Версия

1.0

Дата

26.01.2023

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

Предметный указатель

`cipher_error`, [7](#)

`getValidCipherText`
 [Shifrovshik](#), [8](#)

`getValidKey`
 [Shifrovshik](#), [9](#)

`getValidOpenText`
 [Shifrovshik](#), [9](#)

`modBetaCipher.h`, [11](#)

`Rasshifr`
 [Shifrovshik](#), [9](#)

`Shifr`
 [Shifrovshik](#), [10](#)

`Shifrovshik`, [8](#)
 [getValidCipherText](#), [8](#)
 [getValidKey](#), [9](#)
 [getValidOpenText](#), [9](#)
 [Rasshifr](#), [9](#)
 [Shifr](#), [10](#)
 [Shifrovshik](#), [8](#)