

Шифрование шифром табличной маршрутной перестановки

1.0

Создано системой Doxygen 1.8.17

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	7
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()	8
4.1.3.3 getValidKey()	9
4.1.3.4 getValidOpenCipherText()	9
4.2 Класс CipherError	10
4.2.1 Подробное описание	10
4.2.2 Конструктор(ы)	11
4.2.2.1 CipherError()	11
5 Файлы	13
5.1 Файл Cipher.h	13
5.1.1 Подробное описание	14
5.2 Файл CipherError.h	14
5.2.1 Подробное описание	15
Предметный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Cipher	7
invalid_argument	
CipherError	10

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Cipher	Класс, который предназначен для зашифрования и расшифрования с помощью шифра табличной маршрутной перестановки	7
CipherError	Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием <code>invalid_argument</code>	10

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Cipher.h	
Описание класса Cipher	13
CipherError.h	
Описание класса CipherError	14

Глава 4

Классы

4.1 Класс Cipher

Класс, который предназначен для зашифрования и расшифрования с помощью шифра табличной маршрутной перестановки

```
#include <Cipher.h>
```

Открытые члены

- `Cipher ()=delete`
Запрещающий конструктор без параметров
- `Cipher (const string k)`
Конструктор с параметром
- `wstring encrypt (wstring str)`
Метод для зашифрования шифром маршрутной табличной перестановки
- `wstring decrypt (wstring str)`
Метод для расшифрования шифра маршрутной табличной перестановки

Закрытые члены

- `void getValidKey (const string &k)`
Метод валидации ключа
- `void getValidOpenCipherText (wstring &ws)`
Метод валидации строки для зашифрования или расшифрования

Закрытые данные

- `int key`
атрибут, который предназначен для хранения ключа. Количество столбцов таблицы

4.1.1 Подробное описание

Класс, который предназначен для зашифрования и расшифрования с помощью шифра табличной маршрутной перестановки

4.1.2 Конструктор(ы)

4.1.2.1 Cipher()

```
Cipher::Cipher (  
    const string k )
```

Конструктор с параметром

Аргументы

k	- ключ в виде строки
---	----------------------

Строка, которая пришла на вход переводится в тип `int`, если она успешно прошла проверку на ошибки. Далее уже целочисленный ключ записывается в атрибут с именем `key`

4.1.3 Методы

4.1.3.1 decrypt()

```
wstring Cipher::decrypt (  
    wstring str )
```

Метод для расшифрования шифра маршрутной табличной перестановки

Аргументы

str	- строка, которую необходимо расшифровать
-----	---

Возвращает

расшифрованная строка типа `wstring`

4.1.3.2 encrypt()

```
wstring Cipher::encrypt (  
    wstring str )
```

Метод для зашифрования шифром маршрутной табличной перестановки

Аргументы

str	- строка, которую необходимо зашифровать
-----	--

Возвращает

зашифрованная строка типа wstring

4.1.3.3 getValidKey()

```
void Cipher::getValidKey (
    const string & k ) [private]
```

Метод валидации ключа

Аргументы

k	- ключ в виде строки, который нужно проверить на наличие ошибок
---	---

Исключения

CipherError ,если	ключ оказался пустым или является некорректным
-----------------------------------	--

4.1.3.4 getValidOpenCipherText()

```
void Cipher::getValidOpenCipherText (
    wstring & ws ) [private]
```

Метод валидации строки для зашифрования или расшифрования

Аргументы

ws	- строка для зашифрования или расшифрования, которую нужно проверить на наличие ошибок
----	--

Исключения

CipherError ,если	строка для шифрования или расшифрования является пустой
-----------------------------------	---

Объявления и описания членов классов находятся в файлах:

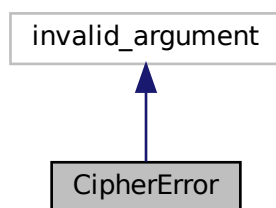
- [Cipher.h](#)
- [Cipher.cpp](#)

4.2 Класс CipherError

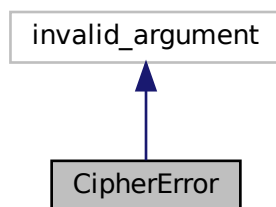
Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

```
#include <CipherError.h>
```

Граф наследования: `CipherError`:



Граф связей класса `CipherError`:



Открытые члены

- `CipherError ()=delete`
Запрещающий конструктор без параметров
- `CipherError (const string &what_arg)`
Конструктор с параметром

4.2.1 Подробное описание

Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

4.2.2 Конструктор(ы)

4.2.2.1 CipherError()

```
CipherError::CipherError (
    const string & what_arg ) [inline], [explicit]
```

Конструктор с параметром

Аргументы

what_arg	- строка, которая хранит описание ошибки
----------	--

Параметр конструктора перегружается конструктором базового класса с названием `invalid_↵`
`argument`

Объявления и описания членов класса находятся в файле:

- [CipherError.h](#)

Глава 5

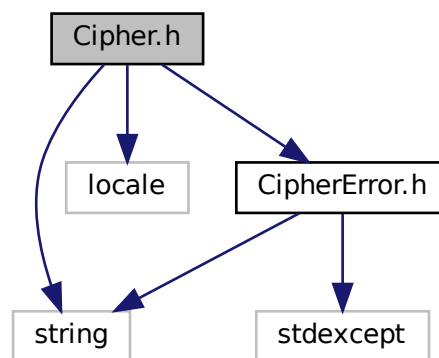
Файлы

5.1 Файл Cipher.h

Описание класса [Cipher](#).

```
#include <string>
#include <locale>
#include "CipherError.h"
```

Граф включаемых заголовочных файлов для Cipher.h:



Классы

- class [Cipher](#)

Класс, который предназначен для зашифрования и расшифрования с помощью шифра табличной маршрутной перестановки

5.1.1 Подробное описание

Описание класса [Cipher](#).

Автор

Мещеряков В.А.

Версия

1.0

Авторство

ИБСТ ПГУ

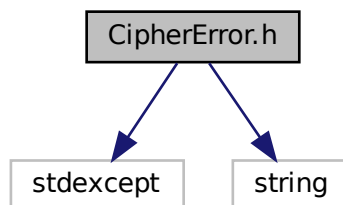
5.2 Файл CipherError.h

Описание класса [CipherError](#).

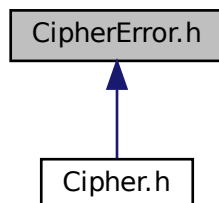
```
#include <stdexcept>
```

```
#include <string>
```

Граф включаемых заголовочных файлов для CipherError.h:



Граф файлов, в которые включается этот файл:



Классы

- class [CipherError](#)

Класс для обработки ошибок. Класс является наследником класса обработки исключений с названием `invalid_argument`.

5.2.1 Подробное описание

Описание класса [CipherError](#).

Автор

Мещеряков В.А.

Версия

1.0

Авторство

ИБСТ ПГУ

Предметный указатель

- Cipher, [7](#)
 - Cipher, [8](#)
 - decrypt, [8](#)
 - encrypt, [8](#)
 - getValidKey, [9](#)
 - getValidOpenCipherText, [9](#)
- Cipher.h, [13](#)
- CipherError, [10](#)
 - CipherError, [11](#)
- CipherError.h, [14](#)
- decrypt
 - Cipher, [8](#)
- encrypt
 - Cipher, [8](#)
- getValidKey
 - Cipher, [9](#)
- getValidOpenCipherText
 - Cipher, [9](#)