

Permutation cipher

1.0.0

Создано системой Doxygen 1.8.13

Оглавление

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 <code>Cipher</code>	7
4.1.1	Подробное описание	8
4.1.2	Конструктор(ы)	8
4.1.2.1	<code>Cipher()</code>	8
4.1.3	Методы	8
4.1.3.1	<code>decrypt()</code>	8
4.1.3.2	<code>encrypt()</code>	9
4.1.3.3	<code>getValidCipherText()</code>	9
4.1.3.4	<code>getValidKey()</code>	10
4.1.3.5	<code>getValidOpenText()</code>	10
4.1.3.6	<code>set_key()</code>	11
4.1.3.7	<code>set_tableform()</code>	11
4.2	Класс <code>cipher_error</code>	11
4.2.1	Подробное описание	12
4.2.2	Конструктор(ы)	12
4.2.2.1	<code>cipher_error()</code> [1/2]	12
4.2.2.2	<code>cipher_error()</code> [2/2]	13
5	Файлы	15
5.1	Файл <code>Cipher.h</code>	15
5.1.1	Подробное описание	15
	Алфавитный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Cipher	7
invalid_argument	
cipher_error	11

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Cipher	Шифрование методом табличной перестановки	7
cipher_error	Класс-исключение	11

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Cipher.h	
Класс-исключение	15

Глава 4

Классы

4.1 Класс Cipher

Шифрование методом табличной перестановки

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

Открытые члены

- `Cipher` ()=delete
Конструктор по умолчанию запрещён
- `Cipher` (std::wstring &ws_key)
Конструктор принимает ключ (количество столбцов в таблице)
- std::wstring `encrypt` (std::wstring &ws_open_text)
Метод для зашифрования
- std::wstring `decrypt` (const std::wstring &ws_cipher_text)
Метод для расшифрования
- void `set_tableform` (const std::wstring &ws_text)
Формирование информации о таблице
- void `set_key` (std::wstring &ws_key)
Установка нового ключа
- int `getValidKey` (std::wstring &ws_key)
Проверка на правильность ключа
- std::wstring `getValidOpenText` (const std::wstring &ws_open_text)
Проверка на правильность текста для зашифровки
- std::wstring `getValidCipherText` (const std::wstring &ws_cipher_text)
Проверка на правильность текста для расшифровки

Закрытые данные

- std::wstring_convert< std::codecvt_utf8< wchar_t >, wchar_t > `codec`
codec для преобразования в широкий формат строки и обратно
- int `columns`
Количество столбцов в таблице (ключ)
- int `rows`
Количество строк в таблице
- int `len_text`
Количество символов в слове

4.1.1 Подробное описание

Шифрование методом табличной перестановки

Ключ устанавливается в конструкторе, а также с помощью метода `set_key`. Для зашифрования и расшифрования предназначены методы `encrypt` и `decrypt`. Метод `set_tableform` - вспомогательный. Методы: `getValidKey`, `getValidOpenText`, `getValidCipherText` - специализируются на проверке входных данных.

Предупреждения

Реализация только для русского языка! С использованием `wstring`. Шифрование методом табличной перестановки

4.1.2 Конструктор(ы)

4.1.2.1 Cipher()

```
Cipher::Cipher (
    std::wstring & ws_key )
```

Конструктор принимает ключ (количество столбцов в таблице)

Конструктор, принимающий на вход ключ, устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

4.1.3 Методы

4.1.3.1 decrypt()

```
std::wstring Cipher::decrypt (
    const std::wstring & cipher_text )
```

Метод для расшифрования

Метод `decrypt` расшифровывает текст.

Аргументы

cipher_text	
-------------	--

Возвращает

Зашифрованный текст.

4.1.3.2 encrypt()

```
std::wstring Cipher::encrypt (
    std::wstring & open_text )
```

Метод для зашифрования

Метод encrypt зашифровывает текст.

Аргументы

open_text	
-----------	--

Возвращает

Зашифрованный текст

4.1.3.3 getValidCipherText()

```
std::wstring Cipher::getValidCipherText (
    const std::wstring & ws_cipher_text ) [inline]
```

Проверка на правильность текста для расшифровки

Данный метод проверяет зашифрованный текст на правильность.

Аргументы

ws_cipher_text	
----------------	--

Возвращает

Зашифрованный текст

Исключения

<code>cipher_error</code> ,если	текст пустой или невалидный
---------------------------------	-----------------------------

4.1.3.4 `getValidKey()`

```
int Cipher::getValidKey (
    std::wstring & ws_key ) [inline]
```

Проверка на правильность ключа

Данный метод проверяет ключ на правильность.

Аргументы

<code>ws_key</code>	
---------------------	--

Возвращает

Ключ

Исключения

<code>cipher_error</code> ,если	ключ пустой или невалидный
---------------------------------	----------------------------

4.1.3.5 `getValidOpenText()`

```
std::wstring Cipher::getValidOpenText (
    const std::wstring & ws_open_text ) [inline]
```

Проверка на правильность текста для зашифровки

Данный метод проверяет открытый текст на правильность. Строчные буквы превращаются в прописные. Все не-буквы удаляются.

Аргументы

<code>ws_open_text</code>	
---------------------------	--

Возвращает

Текст для расшифровки

Исключения

cipher_error ,если	текст пустой
------------------------------------	--------------

4.1.3.6 set_key()

```
void Cipher::set_key (
    std::wstring & ws_key )
```

Установка нового ключа

Метод, принимающий на вход ключ, устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

4.1.3.7 set_tableform()

```
void Cipher::set_tableform (
    const std::wstring & open_text )
```

Формирование информации о таблице

Принимает текст для зашифровки, далее по нему формирует кол-во строк в таблице, а также получает длину текста.

Аргументы

open_text	
-----------	--

Объявления и описания членов классов находятся в файлах:

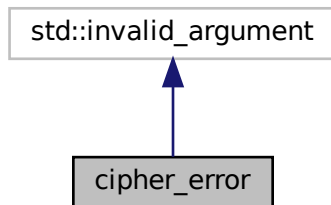
- [Cipher.h](#)
- [Cipher.cpp](#)

4.2 Класс cipher_error

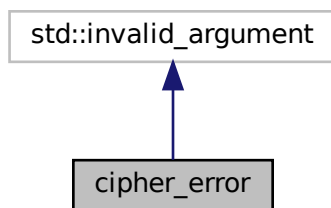
Класс-исключение

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

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- `cipher_error` (`const std::string &what_arg`)
Принимает строку, поднимает исключение
- `cipher_error` (`const char *what_arg`)
Принимает строку, поднимает исключение

4.2.1 Подробное описание

Класс-исключение

4.2.2 Конструктор(ы)

4.2.2.1 `cipher_error()` [1/2]

```

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

Принимает строку, поднимает исключение

Аргументы

what_arg	
----------	--

4.2.2.2 cipher_error() [2/2]

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

Принимает строку, поднимает исключение

Аргументы

what_arg	
----------	--

Объявления и описания членов класса находятся в файле:

- [Cipher.h](#)

Глава 5

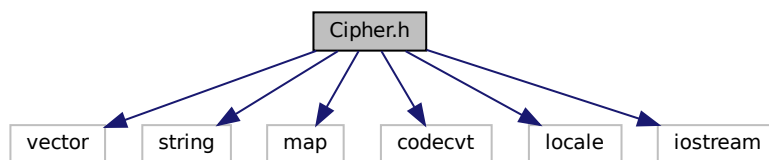
Файлы

5.1 Файл Cipher.h

Класс-исключение

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

Граф включаемых заголовочных файлов для Cipher.h:



Классы

- class [Cipher](#)
Шифрование методом табличной перестановки
- class [cipher_error](#)
Класс-исключение

5.1.1 Подробное описание

Класс-исключение

Предметный указатель

- Cipher, [7](#)
 - Cipher, [8](#)
 - decrypt, [8](#)
 - encrypt, [9](#)
 - getValidCipherText, [9](#)
 - getValidKey, [10](#)
 - getValidOpenText, [10](#)
 - set_key, [11](#)
 - set_tableform, [11](#)
- Cipher.h, [15](#)
- cipher_error, [11](#)
 - cipher_error, [12](#), [13](#)
- decrypt
 - Cipher, [8](#)
- encrypt
 - Cipher, [9](#)
- getValidCipherText
 - Cipher, [9](#)
- getValidKey
 - Cipher, [10](#)
- getValidOpenText
 - Cipher, [10](#)
- set_key
 - Cipher, [11](#)
- set_tableform
 - Cipher, [11](#)