

# Шифрование методом маршрутной перестановки

## 2.0

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 . . . . .	7
4.1.1 Подробное описание . . . . .	8
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() . . . . .	9
4.1.3.3 getValidCipherText() . . . . .	9
4.1.3.4 getValidKey() . . . . .	10
4.1.3.5 getValidOpenText() . . . . .	10
4.1.3.6 set_key() . . . . .	11
4.1.3.7 set_text() . . . . .	11
4.2 Класс cipher_error . . . . .	12
4.2.1 Конструктор(ы) . . . . .	12
4.2.1.1 cipher_error() [1/2] . . . . .	12
4.2.1.2 cipher_error() [2/2] . . . . .	13
5 Файлы	15
5.1 Файл TiMP.cpp . . . . .	15
5.1.1 Подробное описание . . . . .	15
5.2 Файл TiMP.h . . . . .	15
5.2.1 Подробное описание . . . . .	16
Предметный указатель	17

# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Cipher . . . . .	7
std::invalid_argument	
cipher_error . . . . .	12

## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">Cipher</a>	Описание класса <a href="#">Cipher</a> . . . . .	<a href="#">7</a>
<a href="#">cipher_error</a>	. . . . .	<a href="#">12</a>

## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">TiMP.cpp</a>	
Описание класса <a href="#">cipher_error</a> . . . . .	15
<a href="#">TiMP.h</a>	
Шифрование методом табличной перестановки . . . . .	15

## Глава 4

# Классы

### 4.1 Класс Cipher

Описание класса [Cipher](#).

```
#include <TiMP.h>
```

#### Открытые члены

- [Cipher](#) ()=delete  
Запрещающий конструктор без параметров
- [Cipher](#) (std::wstring &ws\_key)  
Конструктор принимает ключ
- std::wstring [encrypt](#) (std::wstring &ws\_open\_text)  
Метод использующийся для зашифрования
- std::wstring [decrypt](#) (const std::wstring &ws\_cipher\_text)  
Метод использующийся для расшифрования
- void [set\\_text](#) (const std::wstring &ws\_text)  
Формирует информацию 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](#)  
для преобразования в широкий формат строки и обратно
- int [pillar](#)  
Ключ, обозначающий количество столбцов в таблице
- int [rad](#)  
Количество строк в таблице
- int [len\\_text](#)  
Количество символов в строке

### 4.1.1 Подробное описание

Описание класса [Cipher](#).

Ключ устанавливается в конструкторе, Для зашифрования и расшифрования предназначены методы `encrypt` и `decrypt`.

Предупреждения

Реализация производится только для русского языка. С использованием `wstring`.

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 Cipher()

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

Конструктор принимает ключ

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

Аргументы

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

Возвращает

Ничего не возвращает

### 4.1.3 Методы

#### 4.1.3.1 decrypt()

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

Метод использующийся для расшифрования

Метод `decrypt` расшифровывает текст.

Аргументы

<code>cipher_text</code>	
--------------------------	--

Возвращает

Возвращает расшифрованный текст.

#### 4.1.3.2 encrypt()

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

Метод использующийся для зашифрования

Метод encrypt зашифровывает принятый текст.

Аргументы

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

Возвращает

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

<a href="#">cipher_error</a> ,если	текст пустой или невалидный
------------------------------------	-----------------------------



#### 4.1.3.4 isValidKey()

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

Проверка на правильность ключа

Метод проверяющий ключ на правильность.

Аргументы

ws_key	
--------	--

Возвращает

Ключ

Исключения

<a href="#">cipher_error</a> ,если	ключ пустой или невалидный
------------------------------------	----------------------------

#### 4.1.3.5 isValidOpenText()

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

Проверка на правильность текста для зашифровки

Данный метод проверяет принятый текст на правильность. В данном методе строчные буквы превращаются в прописные. Когда встречаются знаки, цифры и пробелы они удаляются.

Аргументы

ws_open_text	
--------------	--

Возвращает

Текст для расшифрования

Исключения

<a href="#">cipher_error</a> ,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, который устанавливает кол-во столбцов

Аргументы

ws_key	
--------	--

Возвращает

Ничего не возвращает

## 4.1.3.7 set\_text()

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

Формирует информацию о таблице

Метод, принимающий на вход текст для зашифровки

Данный метод принимает на вход длину текста и формируется кол-во строк в таблице.

Аргументы

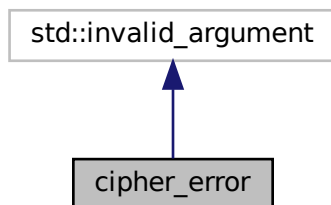
open_text	
-----------	--

Объявления и описания членов классов находятся в файлах:

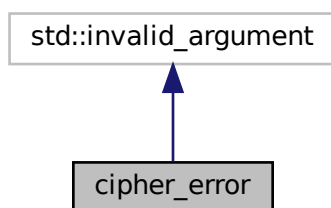
- [TiMP.h](#)
- [TiMP.cpp](#)

## 4.2 Класс cipher\_error

Граф наследования: cipher\_error:



Граф связей класса cipher\_error:



### Открытые члены

- [cipher\\_error](#) (const std::string &what\_arg)  
Принимает на вход строку, вызывает исключение
- [cipher\\_error](#) (const char \*what\_arg)  
Принимает на вход си строку, вызывает исключение

### 4.2.1 Конструктор(ы)

#### 4.2.1.1 cipher\_error() [1/2]

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

Принимает на вход строку, вызывает исключение

Аргументы

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

#### 4.2.1.2 cipher\_error() [2/2]

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

Принимает на вход строку, вызывает исключение

Аргументы

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

Объявления и описания членов класса находятся в файле:

- [TiMP.h](#)

## Глава 5

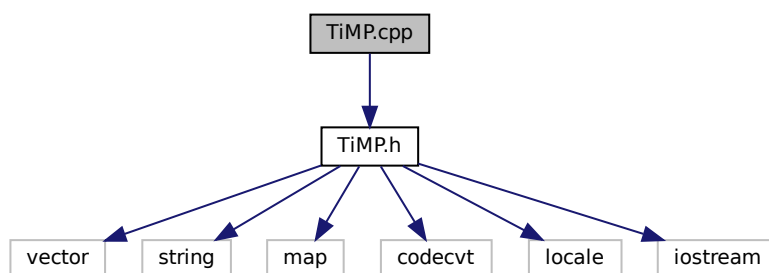
# Файлы

### 5.1 Файл TiMP.cpp

Описание класса [cipher\\_error](#).

```
#include "TiMP.h"
```

Граф включаемых заголовочных файлов для TiMP.cpp:



#### 5.1.1 Подробное описание

Описание класса [cipher\\_error](#).

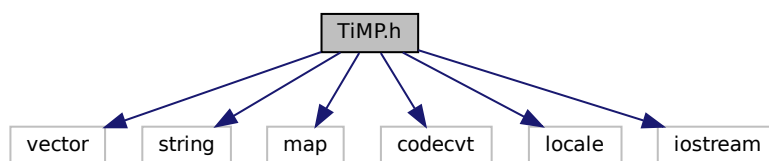
### 5.2 Файл TiMP.h

Шифрование методом табличной перестановки

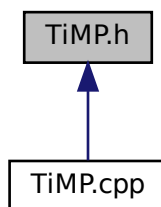
```
#include <vector>
#include <string>
#include <map>
#include <codecvt>
#include <locale>
```

```
#include <iostream>
```

Граф включаемых заголовочных файлов для TiMP.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Cipher](#)  
Описание класса [Cipher](#).
- class [cipher\\_error](#)

### 5.2.1 Подробное описание

Шифрование методом табличной перестановки

Автор

Пицан А. П.

Версия

1.0

Дата

7.11.2022

Авторство

ИБСТ ПГУ

# Предметный указатель

- Cipher, [7](#)
  - Cipher, [8](#)
  - decrypt, [8](#)
  - encrypt, [9](#)
  - getValidCipherText, [9](#)
  - getValidKey, [9](#)
  - getValidOpenText, [10](#)
  - set\_key, [10](#)
  - set\_text, [11](#)
- cipher\_error, [12](#)
  - cipher\_error, [12](#), [13](#)
- decrypt
  - Cipher, [8](#)
- encrypt
  - Cipher, [9](#)
- getValidCipherText
  - Cipher, [9](#)
- getValidKey
  - Cipher, [9](#)
- getValidOpenText
  - Cipher, [10](#)
- set\_key
  - Cipher, [10](#)
- set\_text
  - Cipher, [11](#)
- TiMP.cpp, [15](#)
- TiMP.h, [15](#)