

PerestanCipher

1.0

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_error . . . . .	7
4.1.1 Подробное описание . . . . .	8
4.2 Класс PerestanCipher . . . . .	8
4.2.1 Подробное описание . . . . .	8
4.2.2 Конструктор(ы) . . . . .	8
4.2.2.1 PerestanCipher() . . . . .	8
4.2.3 Методы . . . . .	9
4.2.3.1 CoderPerestanCipher() . . . . .	9
4.2.3.2 DecoderPerestanCipher() . . . . .	10
5 Файлы	13
5.1 Файл PerestanCipher.h . . . . .	13
5.1.1 Подробное описание . . . . .	14
Предметный указатель	15



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
PerestanCipher . . . . .	8



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	Класс, предназначенный для обработки исключений . . . . .	<a href="#">7</a>
<a href="#">PerestanCipher</a>	Класс для шифрования и расшифрования методом табличной маршрутной перестановки . . . . .	<a href="#">8</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">PerestanCipher.h</a>	
Описание класса <a href="#">PerestanCipher</a>	13



## Глава 4

# Классы

### 4.1 Класс `cipher_error`

Класс, предназначенный для обработки исключений

```
#include <PerestanCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

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

### 4.1.1 Подробное описание

Класс, предназначенный для обработки исключений

Объявления и описания членов класса находятся в файле:

- [PerestanCipher.h](#)

## 4.2 Класс PerestanCipher

Класс для шифрования и расшифрования методом табличной маршрутной перестановки

`#include <PerestanCipher.h>`

## Открытые члены

- [PerestanCipher](#) ()=delete  
Запрещающий конструктор без параметров
- [PerestanCipher](#) (int w)  
Конструктор для ключа
- `wstring CoderPerestanCipher (PerestanCipher w, wstring &s)`  
Метод , предназначенный для шифрования методом табличной маршрутной перестановки
- `wstring DecoderPerestanCipher (PerestanCipher w, wstring &s)`  
Метод , предназначенный для шифрования методом табличной маршрутной перестановки

## Закрытые члены

- `wstring getValidOpenText (const std::wstring &s)`
- `wstring getValidCipherText (const std::wstring &s)`
- `int getValidKey (const int k, const std::wstring &s)`

## Закрытые данные

- `int k`  
атрибут, хранящий ключ для шифрования и расшифрования

### 4.2.1 Подробное описание

Класс для шифрования и расшифрования методом табличной маршрутной перестановки

### 4.2.2 Конструктор(ы)

#### 4.2.2.1 PerestanCipher()

`PerestanCipher::PerestanCipher (`  
`int w )`

Конструктор для ключа

## Аргументы

Ключ	- это целочисленное значение
------	------------------------------

Число, которое поступило на вход, записывается в "private" атрибут с названием "key"

## 4.2.3 Методы

## 4.2.3.1 CoderPerestanCipher()

```
wstring PerestanCipher::CoderPerestanCipher (
    PerestanCipher w,
    wstring & s )
```

Метод , предназначенный для шифрования методом табличной маршрутной перестановки

## Аргументы

Экземляр	класса "PerestanCipher", в котором установился ключ
wstring	- строка, которую нужно зашифровать

Сначала вычисляется размер таблицы по формуле.

```
int h;
if (s.size()%w.k!=0) {
    h=s.size()/w.k+1;
} else {
    h=s.size()/w.k;
}
```

Затем создаётся двумерный массив типа "wchar\_t", который имеет необходимый размер: размер таблицы в строках вычисляется по формуле, а количество столбцов - это ключ, который устанавливается в экземпляре класса "PerestanCipher".

```
wchar_t a[h][w.k];
```

Далее в созданный массив записываются символы строки, которую нужно зашифровать. Запись символов просходит по следующему маршруту: слева-направо, сверху-вниз. То есть, запись происходит построчно.

```
for (int i=0; i<h; i++) {
    for (int j=0; j<w.k; j++) {
        if (k<s.size()) {
            a[i][j]=s[k];
            k++;
        } else a[i][j]=' ';
    }
}
```

Внимание!: Если в таблице остаются незаполненные "ячейки", то в них записывается символ "пробела". Такая ситуации может произойти, если длина строки, которую нужно зашифровать, не кратна значению ключа.

После записи поступившей на вход строки в таблицу, происходит процесс её шифрования. Он заключается в том, что символы, которые находятся в двумерном массиве записываются в строку типа wstring с именем "code" по следующему маршруту: сверху-вниз, слева-направо. То есть, чтение происходит по столбцам.

```
for (int i=0; i<w.k; i++) {
```

```

    for (int j=0; j<h; j++) {
        code+=a[j][i];
    }
}
return code;

```

Возвращает

зашифрованная строка типа "wstring"

#### 4.2.3.2 DecoderPerestanCipher()

```

wstring PerestanCipher::DecoderPerestanCipher (
    PerestanCipher w,
    wstring & s )

```

Метод , предназначенный для шифрования методом табличной маршрутной перестановки

Аргументы

Экземляр	класса "PerestanCipher", в котором установился ключ
wstring	- строка, которую нужно расшифровать

Сначала вычисляется размер таблицы по формуле.

```

int h;
if (s.size()%w.k!=0) {
    h=s.size()/w.k+1;
} else {
    h=s.size()/w.k;
}

```

Затем создаётся двумерный массив типа "wchar\_t", который имеет необходимый размер: размер таблицы в строках вычисляется по формуле, а количество столбцов - это ключ, который устанавливается в экземпляре класса "PerestanCipher".

```

wchar_t a[h][w.k];

```

Далее в созданный массив записываются символы строки, которую нужно расшифровать. Запись символов просходит по следующему маршруту: сверху-вниз, слева-направо. То есть, запись происходит по столбцам.

```

for (int i=0; i<w.k; i++) {
    for (int j=0; j<h; j++) {
        a[j][i]=s[k];
        k++;
    }
}

```

Внимание!: Если в таблице остаются незаполненные "ячейки", то в них записывается символ "пробела". Такая ситуация может произойти, если длина строки, которую необходимо расшифровать, не кратна значению ключа.

Аргументы

wstring	- строка, которую нужно расшифровать
---------	--------------------------------------

В конечном итоге происходит процесс расшифрования. Он заключается в том, что символы, которые находятся в двумерном массиве записываются в строку типа wstring с именем "decode" по

следующему маршруту: сверху-вниз, слева-направо. То есть, чтение происходит построчно.

```
for (int i=0; i<h; i++) {  
    for (int j=0; j<w.k; j++) {  
        decode+=a[i][j];  
    }  
}  
return decode;
```

Возвращает

расшифрованная строка типа "wstring"

Объявления и описания членов классов находятся в файлах:

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





## Глава 5

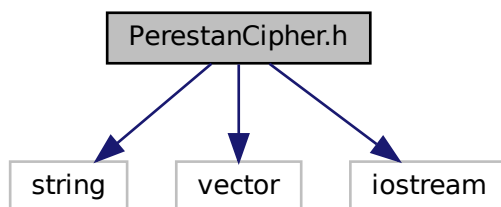
# Файлы

### 5.1 Файл PerestanCipher.h

Описание класса [PerestanCipher](#).

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

Граф включаемых заголовочных файлов для PerestanCipher.h:



### Классы

- class [PerestanCipher](#)  
Класс для шифрования и расшифрования методом табличной маршрутной перестановки
- class [cipher\\_error](#)  
Класс, предназначенный для обработки исключений

### 5.1.1 Подробное описание

Описание класса [PerestanCipher](#).

Автор

Губин А.А.

Версия

1.0

Дата

05.11.2022

Авторство

ИБСТ ПГУ

# Предметный указатель

cipher\_error, [7](#)

CoderPerestanCipher  
PerestanCipher, [9](#)

DecoderPerestanCipher  
PerestanCipher, [10](#)

PerestanCipher, [8](#)  
CoderPerestanCipher, [9](#)  
DecoderPerestanCipher, [10](#)  
PerestanCipher, [8](#)  
PerestanCipher.h, [13](#)