

# Route Cipher

Создано системой Doxygen 1.9.4



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
4.1.2 Конструктор(ы)	8
4.1.2.1 cipher_error()	8
4.2 Класс RouteCipher	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 RouteCipher()	9
4.2.3 Методы	10
4.2.3.1 decrypt()	10
4.2.3.2 encrypt()	10
5 Файлы	13
5.1 Файл main.cpp	13
5.1.1 Подробное описание	14
5.1.2 Функции	14
5.1.2.1 main()	14
5.2 Файл route_cipher.cpp	15
5.2.1 Подробное описание	15
5.2.2 Функции	16
5.2.2.1 isRussianLetter()	16
5.2.2.2 toUpperRussian()	16
5.3 Файл route_cipher.h	16
5.3.1 Подробное описание	17
5.4 route_cipher.h	18
Предметный указатель	19



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
RouteCipher . . . . .	8



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	Класс исключения для ошибок шифрования . . . . .	<a href="#">7</a>
<a href="#">RouteCipher</a>	Класс для шифрования методом табличной маршрутной перестановки . . . . .	<a href="#">8</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">main.cpp</a>	Главный модуль программы для шифрования методом табличной маршрутной перестановки . . . . .	13
<a href="#">route_cipher.cpp</a>	Файл реализации модуля шифрования методом табличной маршрутной перестановки . . . . .	15
<a href="#">route_cipher.h</a>	Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки . . . . .	16



## Глава 4

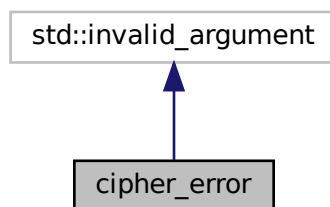
# Классы

### 4.1 Класс `cipher_error`

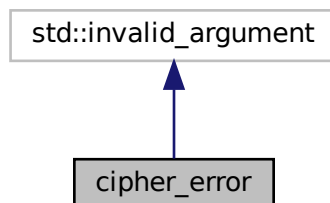
Класс исключения для ошибок шифрования

```
#include <route_cipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- [cipher\\_error](#) (const std::string &message)  
Конструктор класса [cipher\\_error](#).

### 4.1.1 Подробное описание

Класс исключения для ошибок шифрования

Производный класс от std::invalid\_argument для обработки ошибок, возникающих при шифровании и расшифровании

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 cipher\_error()

```

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

```

Конструктор класса [cipher\\_error](#).

Аргументы

in	message	Сообщение об ошибке
----	---------	---------------------

Объявления и описания членов класса находятся в файле:

- [route\\_cipher.h](#)

## 4.2 Класс RouteCipher

Класс для шифрования методом табличной маршрутной перестановки

```
#include <route_cipher.h>
```

## Открытые члены

- [RouteCipher](#) (int cols)  
Конструктор класса [RouteCipher](#).
- std::wstring [encrypt](#) (const std::wstring &text)  
Метод для зашифрования текста
- std::wstring [decrypt](#) (const std::wstring &cipherText)  
Метод для расшифрования текста

## Закрытые данные

- `int columns`  
Количество столбцов таблицы (ключ шифрования)

### 4.2.1 Подробное описание

Класс для шифрования методом табличной маршрутной перестановки

Реализует шифр табличной маршрутной перестановки для русского текста. Ключом является количество столбцов таблицы. Маршрут записи: по горизонтали слева направо, сверху вниз. Маршрут считывания: сверху вниз, справа налево.

#### Предупреждения

Реализация поддерживает только русские буквы и пробелы

### 4.2.2 Конструктор(ы)

#### 4.2.2.1 RouteCipher()

```
RouteCipher::RouteCipher (
    int cols )
```

Конструктор класса [RouteCipher](#).

#### Аргументы

in	cols	Количество столбцов таблицы (должно быть положительным числом)
----	------	--

#### Исключения

<a href="#">cipher_error</a>	Если количество столбцов меньше или равно 0
------------------------------	---

Инициализирует количество столбцов таблицы и проверяет корректность ключа

#### Аргументы

in	cols	Количество столбцов таблицы
----	------	-----------------------------

#### Исключения

<a href="#">cipher_error</a>	Если количество столбцов меньше или равно 0
------------------------------	---

### 4.2.3 Методы

#### 4.2.3.1 decrypt()

```
std::wstring RouteCipher::decrypt (
    const std::wstring & cipherText )
```

Метод для расшифрования текста

Аргументы

in	cipherText	Зашифрованный текст (только русские прописные буквы)
----	------------	--

Возвращает

Расшифрованная строка (русские прописные буквы)

Исключения

<a href="#">cipher_error</a>	Если зашифрованный текст пустой, содержит недопустимые символы или возникла ошибка при расшифровании
------------------------------	--

Реализует обратный алгоритм табличной маршрутной перестановки:

1. Заполнение таблицы зашифрованным текстом по маршруту чтения
2. Чтение таблицы по строкам слева направо

Аргументы

in	cipherText	Зашифрованный текст
----	------------	---------------------

Возвращает

Расшифрованная строка

Исключения

<a href="#">cipher_error</a>	Если зашифрованный текст пустой, содержит недопустимые символы или возникла ошибка при расшифровании
------------------------------	--

#### 4.2.3.2 encrypt()

```
std::wstring RouteCipher::encrypt (
    const std::wstring & text )
```

Метод для зашифрования текста

Аргументы

in	text	Текст для зашифрования. Может содержать русские буквы и пробелы. Строчные буквы автоматически преобразуются в прописные.
----	------	--

Возвращает

Зашифрованная строка (только русские прописные буквы)

Исключения

<a href="#">cipher_error</a>	Если текст пустой, не содержит русских букв или содержит недопустимые символы
------------------------------	---

Реализует алгоритм табличной маршрутной перестановки:

1. Удаление пробелов и преобразование к прописным буквам
2. Заполнение таблицы по горизонтали слева направо, сверху вниз
3. Чтение таблицы по маршруту: сверху вниз, справа налево

Аргументы

in	text	Текст для зашифрования
----	------	------------------------

Возвращает

Зашифрованная строка

Исключения

<a href="#">cipher_error</a>	Если текст пустой, не содержит русских букв или содержит недопустимые символы
------------------------------	---

Объявления и описания членов классов находятся в файлах:

- [route\\_cipher.h](#)
- [route\\_cipher.cpp](#)





## Глава 5

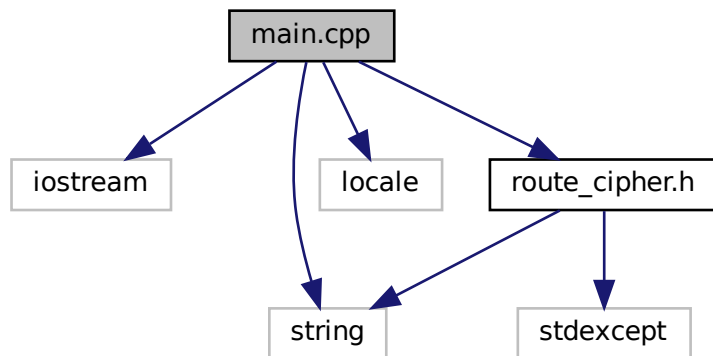
# Файлы

### 5.1 Файл main.cpp

Главный модуль программы для шифрования методом табличной маршрутной перестановки

```
#include <iostream>
#include <string>
#include <locale>
#include "route_cipher.h"
```

Граф включаемых заголовочных файлов для main.cpp:



### Функции

- `int main ()`

Главная функция программы

### 5.1.1 Подробное описание

Главный модуль программы для шифрования методом табличной маршрутной перестановки

Автор

Рябов Кирилл

Версия

1.0

Дата

02.12.2025г.

Авторство

ИБСТ ПГУ

Программа предоставляет пользовательский интерфейс для работы с шифром табличной маршрутной перестановки.

Программа позволяет:

1. Зашифровать текст с использованием заданного ключа (количество столбцов)
2. Расшифровать ранее зашифрованный текст
3. Выйти из программы

### 5.1.2 Функции

#### 5.1.2.1 main()

```
int main ( )
```

Главная функция программы

Предоставляет пользовательский интерфейс в виде меню для работы с шифром. Обработывает пользовательский ввод, вызывает соответствующие методы класса [RouteCipher](#) и обрабатывает исключения.

Возвращает

Код завершения программы (0 - успешное завершение)

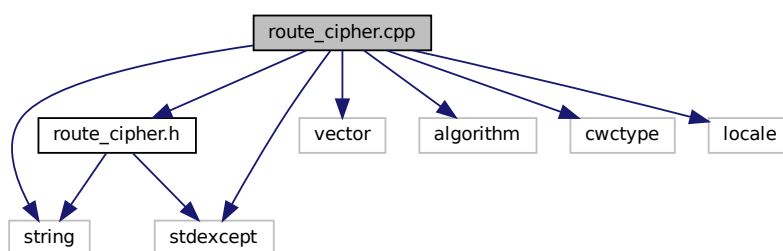
< Игнорируем оставшийся символ новой строки

## 5.2 Файл route\_cipher.cpp

Файл реализации модуля шифрования методом табличной маршрутной перестановки

```
#include "route_cipher.h"
#include <string>
#include <vector>
#include <algorithm>
#include <cwctype>
#include <stdexcept>
#include <locale>
```

Граф включаемых заголовочных файлов для route\_cipher.cpp:



### Функции

- `wchar_t toUpperRussian (wchar_t c)`  
Преобразует русскую строчную букву в прописную
- `bool isRussianLetter (wchar_t c)`  
Проверяет, является ли символ русской буквой

#### 5.2.1 Подробное описание

Файл реализации модуля шифрования методом табличной маршрутной перестановки

Автор

Рябов Кирилл

Версия

1.0

Дата

02.12.2025г.

Авторство

ИБСТ ПГУ

## 5.2.2 Функции

### 5.2.2.1 isRussianLetter()

```
bool isRussianLetter (  
    wchar_t c )
```

Проверяет, является ли символ русской буквой

Аргументы

in	c	Символ для проверки
----	---	---------------------

Возвращает

true, если символ является русской буквой (прописной или строчной, включая Ё/ё), false в противном случае

### 5.2.2.2 toUpperRussian()

```
wchar_t toUpperRussian (  
    wchar_t c )
```

Преобразует русскую строчную букву в прописную

Аргументы

in	c	Символ для преобразования
----	---	---------------------------

Возвращает

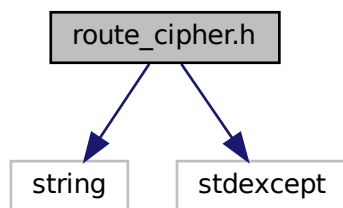
Прописной символ или исходный символ, если он не является русской строчной буквой

## 5.3 Файл route\_cipher.h

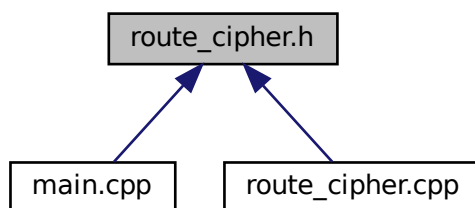
Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки

```
#include <string>  
#include <stdexcept>
```

Граф включаемых заголовочных файлов для route\_cipher.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [RouteCipher](#)  
Класс для шифрования методом табличной маршрутной перестановки
- class [cipher\\_error](#)  
Класс исключения для ошибок шифрования

### 5.3.1 Подробное описание

Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки

Автор

Рябов Кирилл

Версия

1.0

Дата

02.12.2025г.

Авторство

ИБСТ ПГУ

Предупреждения

Это реализация шифра табличной маршрутной перестановки

Данный файл содержит объявление класса [RouteCipher](#), реализующего шифрование методом табличной маршрутной перестановки, а также класса исключения [cipher\\_error](#).

## 5.4 route\_cipher.h

[См. документацию.](#)

```
1
14 #pragma once
15 #include <string>
16 #include <stdexcept>
17
25 class RouteCipher {
26 private:
27     int columns;
28 public:
34     RouteCipher(int cols);
43     std::wstring encrypt(const std::wstring& text);
51     std::wstring decrypt(const std::wstring& cipherText);
52 };
53
59 class cipher_error : public std::invalid_argument {
60 public:
65     explicit cipher_error(const std::string& message)
66         : std::invalid_argument(message) {}
67 };
```

# Предметный указатель

- cipher\_error, [7](#)
  - cipher\_error, [8](#)
- decrypt
  - RouteCipher, [10](#)
- encrypt
  - RouteCipher, [10](#)
- isRussianLetter
  - route\_cipher.cpp, [16](#)
- main
  - main.cpp, [14](#)
- main.cpp, [13](#)
  - main, [14](#)
- route\_cipher.cpp, [15](#)
  - isRussianLetter, [16](#)
  - toUpperRussian, [16](#)
- route\_cipher.h, [16](#)
- RouteCipher, [8](#)
  - decrypt, [10](#)
  - encrypt, [10](#)
  - RouteCipher, [9](#)
- toUpperRussian
  - route\_cipher.cpp, [16](#)