

Шифрование методом маршрутной перестановки

1.0

Создано системой Doxygen 1.9.8

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>table_cipher_error</code>	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 <code>table_cipher_error()</code>	8
4.1.3 Методы	8
4.1.3.1 <code>what()</code>	8
4.2 Класс <code>TableCipher</code>	9
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 <code>TableCipher()</code>	9
4.2.3 Методы	10
4.2.3.1 <code>decrypt()</code>	10
4.2.3.2 <code>encrypt()</code>	11
4.2.3.3 <code>validateKey()</code>	12
5 Файлы	15
5.1 Файл <code>main.cpp</code>	15
5.1.1 Подробное описание	16
5.1.2 Функции	16
5.1.2.1 <code>demonstrateCipher()</code>	16
5.1.2.2 <code>demonstrateInputErrors()</code>	16
5.1.2.3 <code>getKeyFromUser()</code>	17
5.1.2.4 <code>getTextFromUser()</code>	17
5.1.2.5 <code>main()</code>	17
5.1.2.6 <code>processCipherOperation()</code>	17
5.1.2.7 <code>showHelp()</code>	18
5.1.2.8 <code>string_to_wstring()</code>	18
5.1.2.9 <code>wstring_to_string()</code>	18
5.2 Файл <code>tableCipher.cpp</code>	19
5.2.1 Подробное описание	19
5.2.2 Функции	20
5.2.2.1 <code>debugPrintTable()</code>	20
5.3 Файл <code>tableCipher.h</code>	20
5.3.1 Подробное описание	21
5.4 <code>tableCipher.h</code>	22

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::exception	
table_cipher_error	7
TableCipher	9

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

table_cipher_error	
Класс для исключений шифрования табличной перестановки	7
TableCipher	
Шифр табличной маршрутной перестановки	9

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный модуль программы шифрования табличной маршрутной перестановки .	15
tableCipher.cpp	Реализация класса TableCipher для шифрования табличной маршрутной перестановки	19
tableCipher.h	Заголовочный файл для модуля шифрования табличной маршрутной перестановки	20

Глава 4

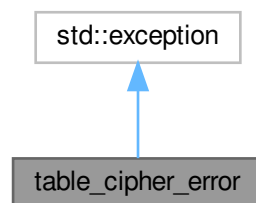
Классы

4.1 Класс `table_cipher_error`

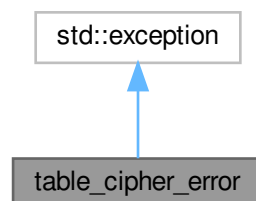
Класс для исключений шифрования табличной перестановки

```
#include <tableCipher.h>
```

Граф наследования: `table_cipher_error`:



Граф связей класса `table_cipher_error`:



Открытые члены

- [table_cipher_error](#) (const std::string &msg)
Конструктор исключения
- const char * [what](#) () const noexcept override
Получить сообщение об ошибке

Закрытые данные

- std::string message
Сообщение об ошибке

4.1.1 Подробное описание

Класс для исключений шифрования табличной перестановки

Наследуется от std::exception для интеграции со стандартной системой исключений

4.1.2 Конструктор(ы)

4.1.2.1 table_cipher_error()

```
table_cipher_error::table_cipher_error (
    const std::string & msg ) [inline], [explicit]
```

Конструктор исключения

Аргументы

msg	Сообщение об ошибке
-----	---------------------

4.1.3 Методы

4.1.3.1 what()

```
const char * table_cipher_error::what ( ) const [inline], [override], [noexcept]
```

Получить сообщение об ошибке

Возвращает

Указатель на строку с сообщением об ошибке

Объявления и описания членов класса находятся в файле:

- [tableCipher.h](#)

4.2 Класс TableCipher

Шифр табличной маршрутной перестановки

```
#include <tableCipher.h>
```

Открытые члены

- [TableCipher](#) (int key)
Конструктор с установкой ключа
- std::wstring [encrypt](#) (const std::wstring &text)
Метод шифрования текста
- std::wstring [decrypt](#) (const std::wstring &cipher_text)
Метод дешифрования текста
- void [validateKey](#) (int key)
Проверка корректности ключа

Закрытые данные

- int numColumns
Количество столбцов таблицы (ключ шифрования)

4.2.1 Подробное описание

Шифр табличной маршрутной перестановки

Реализует шифрование методом табличной перестановки с заданным количеством столбцов. Запись: по горизонтали слева направо, сверху вниз. Чтение: сверху вниз, справа налево.

Предупреждения

Поддерживает только буквы и пробелы

4.2.2 Конструктор(ы)

4.2.2.1 TableCipher()

```
TableCipher::TableCipher (  
    int key )
```

Конструктор с установкой ключа

Конструктор класса [TableCipher](#).

Аргументы

key	Количество столбцов таблицы
-----	-----------------------------

Исключения

<code>table_cipher_error</code>	Если ключ некорректен
---------------------------------	-----------------------

Аргументы

<code>key</code>	Количество столбцов таблицы (ключ шифрования)
------------------	---

Исключения

<code>table_cipher_error</code>	Если ключ некорректен
---------------------------------	-----------------------

Выполняет валидацию ключа и инициализацию объекта. Ключ должен быть положительным целым числом в диапазоне от 1 до 1000.

Пример использования:

```
TableCipher cipher(5); // Создание шифра с ключом 5
```

4.2.3 Методы

4.2.3.1 `decrypt()`

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

Метод дешифрования текста

Расшифрование текста методом табличной маршрутной перестановки

Аргументы

<code>cipher_text</code>	Зашифрованный текст
--------------------------	---------------------

Возвращает

Расшифрованная строка

Исключения

<code>table_cipher_error</code>	Если текст пустой или содержит недопустимые символы
---------------------------------	---

Аргументы

<code>cipher_text</code>	Зашифрованный текст
--------------------------	---------------------

Возвращает

Расшифрованная строка

Исключения

<code>table_cipher_error</code>	При некорректных входных данных
---------------------------------	---------------------------------

Алгоритм расшифрования (обратный шифрованию):

1. Проверка входных данных: текст не должен быть пустым и содержать только буквы и пробелы
2. Создание таблицы: вычисление количества строк и создание таблицы numRows × numColumns
3. Запись в таблицу: заполнение таблицы по столбцам справа налево, сверху вниз
4. Чтение из таблицы: чтение данных по строкам слева направо, сверху вниз

Особенности алгоритма:

- Учитывается неполнота последней строки таблицы
- Пустые ячейки заполняются пробелами и игнорируются при чтении

Пример работы:

Зашифрованный текст: "ИТРЕРВИПМ", Ключ: 3

Таблица:

П Р И
В Е Т
М И Р

Результат: "ПРИВЕТМИР"

4.2.3.2 encrypt()

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

Метод шифрования текста

Шифрование текста методом табличной маршрутной перестановки

Аргументы

<code>text</code>	Исходный текст для шифрования
-------------------	-------------------------------

Возвращает

Зашифрованная строка

Исключения

<code>table_cipher_error</code>	Если текст пустой или содержит недопустимые символы
---------------------------------	---

Аргументы

text	Исходный текст для шифрования
------	-------------------------------

Возвращает

Зашифрованная строка

Исключения

table_cipher_error	При некорректных входных данных
------------------------------------	---------------------------------

Алгоритм шифрования:

1. Проверка входных данных: текст не должен быть пустым и содержать только буквы и пробелы
2. Создание таблицы: вычисление количества строк и создание таблицы numRows × numColumns
3. Запись в таблицу: заполнение таблицы по горизонтали слева направо, сверху вниз
4. Чтение из таблицы: чтение данных сверху вниз, справа налево

Пример работы:

Текст: "ПРИВЕТМИР", Ключ: 3

Таблица:

П Р И
В Е Т
М И Р

Результат: "ИТРЕРВИПМ"

4.2.3.3 validateKey()

```
void TableCipher::validateKey (
    int key )
```

Проверка корректности ключа

Проверка корректности ключа шифрования

Аргументы

key	Ключ для проверки
-----	-------------------

Исключения

table_cipher_error	Если ключ некорректен
------------------------------------	-----------------------

Аргументы

key	Ключ для проверки
-----	-------------------

Исключения

<code>table_cipher_error</code>	Если ключ ≤ 0 или > 1000
---------------------------------	---------------------------------

Выполняет следующие проверки:

- Ключ должен быть положительным числом
- Ключ не должен превышать максимальное значение 1000

Примеры ошибок:

```
validateKey(-5); // Выбросит исключение: "Ключ должен быть положительным числом"  
validateKey(2000); // Выбросит исключение: "Ключ слишком большой..."
```

Объявления и описания членов классов находятся в файлах:

- `tableCipher.h`
- `tableCipher.cpp`

Глава 5

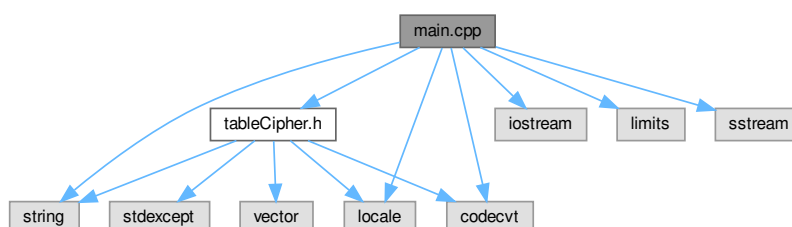
Файлы

5.1 Файл main.cpp

Главный модуль программы шифрования табличной маршрутной перестановки

```
#include "tableCipher.h"
#include <iostream>
#include <string>
#include <limits>
#include <locale>
#include <codecvt>
#include <sstream>
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- `std::wstring string_to_wstring (const std::string &str)`
Преобразование строки в широкую строку
- `std::string wstring_to_string (const std::wstring &wstr)`
Преобразование широкой строки в обычную строку
- `void displayMenu ()`
Отображение главного меню программы
- `int getKeyFromUser ()`
Получение ключа от пользователя с валидацией
- `std::wstring getTextFromUser (const std::wstring &prompt)`

- `void processCipherOperation (bool isEncryption)`
Получение текста от пользователя
- `void demonstrateCipher ()`
Обработка операции шифрования или дешифрования
- `void demonstrateInputErrors ()`
Демонстрация работы шифра на тестовых примерах
- `void showHelp ()`
Демонстрация обработки ошибок ввода
- `int main ()`
Отображение справки по программе
- `int main ()`
Главная функция программы

5.1.1 Подробное описание

Главный модуль программы шифрования табличной маршрутной перестановки

Автор

Ганьшин В.А.

Версия

1.0

Дата

17.12.2025

Содержит пользовательский интерфейс для работы с шифром табличной перестановки

5.1.2 Функции

5.1.2.1 demonstrateCipher()

`void demonstrateCipher ()`

Демонстрация работы шифра на тестовых примерах

Показывает корректную работу шифра и обработку различных ошибок

5.1.2.2 demonstrateInputErrors()

`void demonstrateInputErrors ()`

Демонстрация обработки ошибок ввода

Показывает, какие типы ошибок ввода обрабатывает программа

5.1.2.3 getKeyFromUser()

```
int getKeyFromUser ( )
```

Получение ключа от пользователя с валидацией

Возвращает

Корректный ключ (количество столбцов)

Обрабатывает различные ошибки ввода: текст вместо чисел, символы, пустой ввод и т.д.

5.1.2.4 getTextFromUser()

```
std::wstring getTextFromUser (
    const std::wstring & prompt )
```

Получение текста от пользователя

Аргументы

prompt	Приглашение для ввода
--------	-----------------------

Возвращает

Введенный текст

5.1.2.5 main()

```
int main ( )
```

Главная функция программы

Возвращает

Код завершения программы

Реализует основной цикл программы с меню и обработкой пользовательского ввода

5.1.2.6 processCipherOperation()

```
void processCipherOperation (
    bool isEncryption )
```

Обработка операции шифрования или дешифрования

Аргументы

isEncryption	Флаг операции: true - шифрование, false - дешифрование
--------------	--

Выполняет всю последовательность операций для шифрования/дешифрования с обработкой исключений

5.1.2.7 showHelp()

```
void showHelp ( )
```

Отображение справки по программе

Содержит описание алгоритма, ограничения и демонстрационные примеры

5.1.2.8 string_to_wstring()

```
std::wstring string_to_wstring (
    const std::string & str )
```

Преобразование строки в широкую строку

Аргументы

str	Обычная строка
-----	----------------

Возвращает

Широкая строка

5.1.2.9 wstring_to_string()

```
std::string wstring_to_string (
    const std::wstring & wstr )
```

Преобразование широкой строки в обычную строку

Аргументы

wstr	Широкая строка
------	----------------

Возвращает

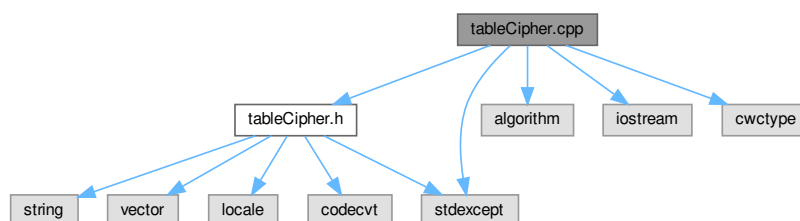
Обычная строка

5.2 Файл tableCipher.cpp

Реализация класса [TableCipher](#) для шифрования табличной маршрутной перестановки

```
#include "tableCipher.h"  
#include <algorithm>  
#include <iostream>  
#include <cwctype>  
#include <stdexcept>
```

Граф включаемых заголовочных файлов для tableCipher.cpp:



Функции

- void [debugPrintTable](#) (const std::vector< std::vector< wchar_t > > &table, int numRows, int numColumns)

Вспомогательная функция для отладки - вывод таблицы в консоль

5.2.1 Подробное описание

Реализация класса [TableCipher](#) для шифрования табличной маршрутной перестановки

Автор

Ганьшин В.А.

Версия

1.0

Дата

17.12.2025

Содержит полную реализацию всех методов класса [TableCipher](#):

- Конструктор с валидацией ключа
- Метод `encrypt()` для шифрования текста
- Метод `decrypt()` для расшифрования текста
- Метод `validateKey()` для проверки корректности ключа

Авторство

Учебный проект

Предупреждения

Реализация поддерживает только буквы и пробелы

5.2.2 Функции

5.2.2.1 `debugPrintTable()`

```
void debugPrintTable (
    const std::vector< std::vector< wchar_t > > & table,
    int numRows,
    int numColumns )
```

Вспомогательная функция для отладки - вывод таблицы в консоль

Аргументы

<code>table</code>	Таблица для вывода
<code>numRows</code>	Количество строк
<code>numColumns</code>	Количество столбцов

Заметки

Эта функция используется только для отладки и не включена в публичный интерфейс

5.3 Файл `tableCipher.h`

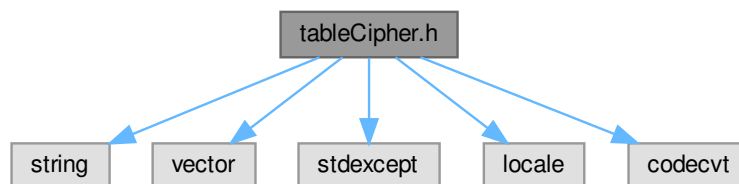
Заголовочный файл для модуля шифрования табличной маршрутной перестановки

```
#include <string>
#include <vector>
#include <stdexcept>
#include <locale>
```

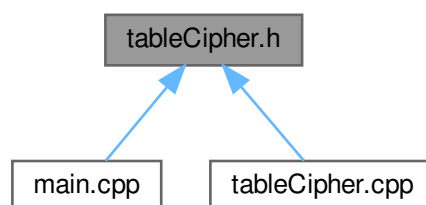


```
#include <codecvt>
```

Граф включаемых заголовочных файлов для tableCipher.h:



Граф файлов, в которые включается этот файл:



Классы

- class `table_cipher_error`
Класс для исключений шифрования табличной перестановки
- class `TableCipher`
Шифр табличной маршрутной перестановки

5.3.1 Подробное описание

Заголовочный файл для модуля шифрования табличной маршрутной перестановки

Автор

Ганьшин В.А.

Версия

1.0

Дата

17.12.2025

Предупреждения

Это документирование программы шифрования методом маршрутной перестановки

5.4 tableCipher.h

[См. документацию.](#)

```
00001 #pragma once
00002 #include <string>
00003 #include <vector>
00004 #include <stdexcept>
00005 #include <locale>
00006 #include <codecvt>
00007
00021 class table_cipher_error : public std::exception {
00022 private:
00023     std::string message;
00024 public:
00029     explicit table_cipher_error(const std::string& msg) : message(msg) {}
00030
00035     const char* what() const noexcept override {
00036         return message.c_str();
00037     }
00038 };
00039
00047 class TableCipher {
00048 private:
00049     int numColumns;
00050
00051 public:
00057     TableCipher(int key);
00058
00065     std::wstring encrypt(const std::wstring& text);
00066
00073     std::wstring decrypt(const std::wstring& cipher_text);
00074
00080     void validateKey(int key);
00081 };
```

Предметный указатель

debugPrintTable
 tableCipher.cpp, [20](#)

decrypt
 TableCipher, [10](#)

demonstrateCipher
 main.cpp, [16](#)

demonstrateInputErrors
 main.cpp, [16](#)

encrypt
 TableCipher, [11](#)

getKeyFromUser
 main.cpp, [16](#)

getTextFromUser
 main.cpp, [17](#)

main
 main.cpp, [17](#)

main.cpp, [15](#)
 demonstrateCipher, [16](#)
 demonstrateInputErrors, [16](#)
 getKeyFromUser, [16](#)
 getTextFromUser, [17](#)
 main, [17](#)
 processCipherOperation, [17](#)
 showHelp, [18](#)
 string_to_wstring, [18](#)
 wstring_to_string, [18](#)

processCipherOperation
 main.cpp, [17](#)

showHelp
 main.cpp, [18](#)

string_to_wstring
 main.cpp, [18](#)

table_cipher_error, [7](#)
 table_cipher_error, [8](#)
 what, [8](#)

TableCipher, [9](#)
 decrypt, [10](#)
 encrypt, [11](#)
 TableCipher, [9](#)
 validateKey, [12](#)

tableCipher.cpp, [19](#)
 debugPrintTable, [20](#)

tableCipher.h, [20](#)

validateKey
 TableCipher, [12](#)

what
 table_cipher_error, [8](#)

wstring_to_string
 main.cpp, [18](#)