

Лабораторная работа №4: Документирование модулей шифра табличной перестановки

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 tableCipher	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 tableCipher()	8
4.1.3 Методы	8
4.1.3.1 decrypt()	8
4.1.3.2 encrypt()	9
4.1.3.3 isValidRussianText()	9
4.1.3.4 prepareText()	10
4.1.3.5 toUpper()	10
4.1.3.6 validateKey()	10
4.1.3.7 validateTextLength()	11
4.2 Класс tableCipher_error	11
4.2.1 Подробное описание	12
4.2.2 Конструктор(ы)	12
4.2.2.1 tableCipher_error() [1/2]	12
4.2.2.2 tableCipher_error() [2/2]	13
4.2.3 Методы	13
4.2.3.1 what()	13
5 Файлы	15
5.1 Файл main.cpp	15
5.1.1 Подробное описание	16
5.1.2 Функции	16
5.1.2.1 getKey()	16
5.1.2.2 getText()	16
5.1.2.3 main()	17
5.1.2.4 string_to_wstring()	17
5.2 Файл tableCipher.cpp	17
5.2.1 Подробное описание	18
5.3 Файл tableCipher.h	18
5.3.1 Подробное описание	19
5.4 tableCipher.h	20
Предметный указатель	21

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::runtime_error	
tableCipher_error	11
tableCipher	7

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

tableCipher	Класс для шифрования методом табличной маршрутной перестановки	7
tableCipher_error	Класс-исключение для ошибок шифра табличной перестановки	11

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный модуль для программы шифрования табличной маршрутной перестановкой	15
tableCipher.cpp	Реализация модуля шифрования методом табличной маршрутной перестановки .	17
tableCipher.h	Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки	18

Глава 4

Классы

4.1 Класс tableCipher

Класс для шифрования методом табличной маршрутной перестановки

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

Открытые члены

- `tableCipher ()=delete`
Запрет конструктора без параметров
- `tableCipher (int k)`
Конструктор с установкой ключа
- `std::wstring encrypt (const std::wstring &open_text)`
Метод зашифровывания
- `std::wstring decrypt (const std::wstring &cipher_text)`
Метод расшифровывания

Закрытые члены

- `std::wstring toUpper (const std::wstring &s)`
Приведение строки к верхнему регистру
- `bool isValidRussianText (const std::wstring &text)`
Проверка текста на соответствие русскому алфавиту
- `std::wstring prepareText (const std::wstring &s)`
Подготовка текста к шифрованию
- `void validateKey (int k)`
Валидация ключа
- `void validateTextLength (const std::wstring &text, const std::string &operation)`
Валидация длины текста относительно ключа

Закрытые данные

- `int key`
Ключ шифрования (количество столбцов)

4.1.1 Подробное описание

Класс для шифрования методом табличной маршрутной перестановки

Реализует табличную маршрутную перестановку для русского алфавита. Маршрут записи: по горизонтали слева направо, сверху вниз. Маршрут считывания: сверху вниз, справа налево. Ключ - количество столбцов таблицы.

4.1.2 Конструктор(ы)

4.1.2.1 tableCipher()

```
tableCipher::tableCipher (
    int k )
```

Конструктор с установкой ключа

Конструктор класса [tableCipher](#).

Аргументы

k	Ключ шифрования (количество столбцов)
---	---------------------------------------

Исключения

tableCipher_error	Если ключ невалиден
-----------------------------------	---------------------

4.1.3 Методы

4.1.3.1 decrypt()

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

Метод расшифровывания

Аргументы

cipher_text	Зашифрованный текст для расшифрования
-------------	---------------------------------------

Возвращает

Расшифрованная строка

Исключения

tableCipher_error	Если текст пустой или недостаточной длины
-----------------------------------	---

4.1.3.2 encrypt()

```
std::wstring tableCipher::encrypt (
    const std::wstring & open_text )
```

Метод зашифровывания

Аргументы

open_text	Открытый текст для шифрования
-----------	-------------------------------

Возвращает

Зашифрованная строка

Исключения

tableCipher_error	Если текст пустой или недостаточной длины
-----------------------------------	---

4.1.3.3 isValidRussianText()

```
bool tableCipher::isValidRussianText (
    const std::wstring & text ) [private]
```

Проверка текста на соответствие русскому алфавиту

Аргументы

text	Проверяемый текст
------	-------------------

Возвращает

true если текст содержит только русские буквы и пробелы, иначе false

4.1.3.4 prepareText()

```
std::wstring tableCipher::prepareText (
    const std::wstring & s ) [private]
```

Подготовка текста к шифрованию

Аргументы

s	Исходный текст
---	----------------

Возвращает

Текст в верхнем регистре без пробелов

Исключения

tableCipher_error	Если текст пустой, содержит недопустимые символы или только пробелы
-----------------------------------	---

4.1.3.5 toUpper()

```
std::wstring tableCipher::toUpper (
    const std::wstring & s ) [private]
```

Приведение строки к верхнему регистру

Аргументы

s	Входная строка
---	----------------

Возвращает

Строка в верхнем регистре

4.1.3.6 validateKey()

```
void tableCipher::validateKey (
    int k ) [private]
```

Валидация ключа

Аргументы

k	Проверяемый ключ
---	------------------

Исключения

tableCipher_error	Если ключ невалиден
-----------------------------------	---------------------

4.1.3.7 validateTextLength()

```
void tableCipher::validateTextLength (
    const std::wstring & text,
    const std::string & operation ) [private]
```

Валидация длины текста относительно ключа

Аргументы

text	Проверяемый текст
operation	Название операции (для сообщения об ошибке)

Исключения

tableCipher_error	Если длина текста недостаточна для операции
-----------------------------------	---

Объявления и описания членов классов находятся в файлах:

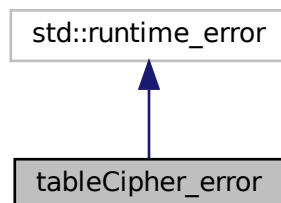
- [tableCipher.h](#)
- [tableCipher.cpp](#)

4.2 Класс tableCipher_error

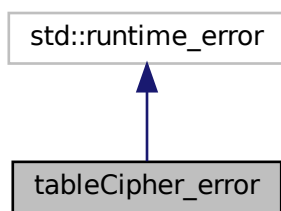
Класс-исключение для ошибок шифра табличной перестановки

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

Граф наследования:tableCipher_error:



Граф связей класса tableCipher_error:



Открытые члены

- `tableCipher_error` (const std::string &what_arg)
Конструктор с параметром типа std::string.
- `tableCipher_error` (const char *what_arg)
Конструктор с параметром типа const char*.
- const char * `what` () const noexcept override
Получение сообщения об ошибке

Закрытые данные

- std::string message
Сообщение об ошибке

4.2.1 Подробное описание

Класс-исключение для ошибок шифра табличной перестановки

4.2.2 Конструктор(ы)

4.2.2.1 tableCipher_error() [1/2]

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

Конструктор с параметром типа std::string.

Аргументы

what_arg	Сообщение об ошибке
----------	---------------------

4.2.2.2 tableCipher_error() [2/2]

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

Конструктор с параметром типа const char*.

Аргументы

what_arg	Сообщение об ошибке
----------	---------------------

4.2.3 Методы

4.2.3.1 what()

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

Получение сообщения об ошибке

Возвращает

Сообщение об ошибке в виде C-строки

Объявления и описания членов класса находятся в файле:

- [tableCipher.h](#)

Глава 5

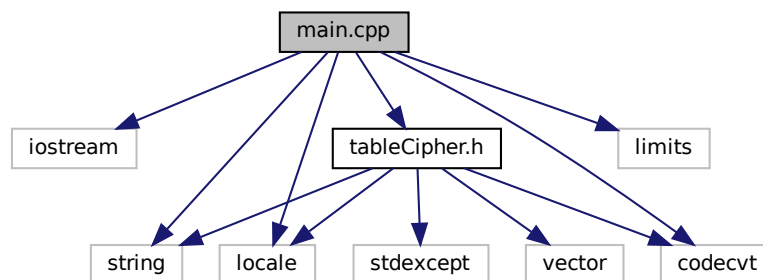
Файлы

5.1 Файл main.cpp

Главный модуль для программы шифрования табличной маршрутной перестановкой

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- std::wstring [string_to_wstring](#) (const std::string &str)
Конвертация string в wstring.
- int [getKey](#) ()
Получение ключа от пользователя
- wstring [getText](#) (const wstring &prompt)
Получение текста от пользователя
- int [main](#) ()
Главная функция программы

5.1.1 Подробное описание

Главный модуль для программы шифрования табличной маршрутной перестановкой

Автор

Гришин Н.С.

Версия

1.0

Дата

03.12.2025

Авторство

ИБСТ ПГУ

5.1.2 Функции

5.1.2.1 getKey()

int getKey ()

Получение ключа от пользователя

Возвращает

Валидный ключ (количество столбцов)

5.1.2.2 getText()

wstring getText (
 const wstring & prompt)

Получение текста от пользователя

Аргументы

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

Возвращает

Введенный текст

5.1.2.3 main()

```
int main ( )
```

Главная функция программы

Возвращает

0 при успешном выполнении, 1 при ошибке

5.1.2.4 string_to_wstring()

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

Конвертация string в wstring.

Аргументы

str	Входная строка
-----	----------------

Возвращает

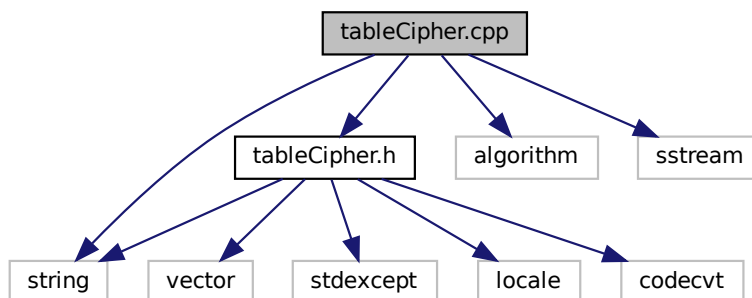
Строка в формате wstring

5.2 Файл tableCipher.cpp

Реализация модуля шифрования методом табличной маршрутной перестановки

```
#include "tableCipher.h"
#include <algorithm>
#include <sstream>
#include <string>
```

Граф включаемых заголовочных файлов для tableCipher.cpp:



5.2.1 Подробное описание

Реализация модуля шифрования методом табличной маршрутной перестановки

Автор

Гришин Н.С.

Версия

1.0

Дата

03.12.2025

Авторство

ИБСТ ПГУ

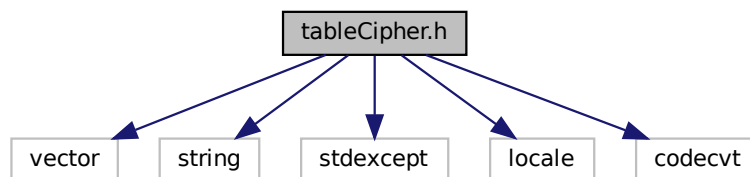
5.3 Файл tableCipher.h

Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки

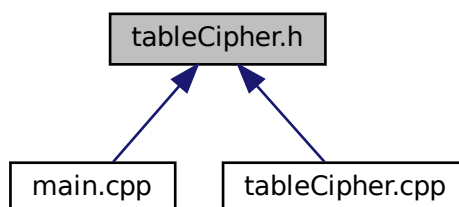
```
#include <vector>
#include <string>
#include <stdexcept>
#include <locale>
```

```
#include <codecvt>
```

Граф включаемых заголовочных файлов для tableCipher.h:



Граф файлов, в которые включается этот файл:



Классы

- class `tableCipher_error`
Класс-исключение для ошибок шифра табличной перестановки
- class `tableCipher`
Класс для шифрования методом табличной маршрутной перестановки

5.3.1 Подробное описание

Заголовочный файл для модуля шифрования методом табличной маршрутной перестановки

Автор

Гришин Н.С,

Версия

1.0

Дата

03.12.2025

Авторство

ИБСТ ПГУ

5.4 tableCipher.h

[См. документацию.](#)

```
1
10 #pragma once
11 #include <vector>
12 #include <string>
13 #include <stdexcept>
14 #include <locale>
15 #include <codecvt>
16
20 class tableCipher_error : public std::runtime_error {
21 private:
22     std::string message;
23 public:
28     explicit tableCipher_error(const std::string& what_arg) : std::runtime_error(what_arg), message(what_arg) {}
29
34     explicit tableCipher_error(const char* what_arg) : std::runtime_error(what_arg), message(what_arg) {}
35
40     const char* what() const noexcept override { return message.c_str(); }
41 };
42
50 class tableCipher
51 {
52 private:
53     int key;
54
60     std::wstring toUpper(const std::wstring& s);
61
67     bool isValidRussianText(const std::wstring& text);
68
75     std::wstring prepareText(const std::wstring& s);
76
82     void validateKey(int k);
83
90     void validateTextLength(const std::wstring& text, const std::string& operation);
91
92 public:
96     tableCipher() = delete;
97
103     tableCipher(int k);
104
111     std::wstring encrypt(const std::wstring& open_text);
112
119     std::wstring decrypt(const std::wstring& cipher_text);
120 };
```


Предметный указатель

- decrypt
 - tableCipher, [8](#)
- encrypt
 - tableCipher, [9](#)
- getKey
 - main.cpp, [16](#)
- getText
 - main.cpp, [16](#)
- isValidRussianText
 - tableCipher, [9](#)
- main
 - main.cpp, [17](#)
- main.cpp, [15](#)
 - getKey, [16](#)
 - getText, [16](#)
 - main, [17](#)
 - string_to_wstring, [17](#)
- prepareText
 - tableCipher, [9](#)
- string_to_wstring
 - main.cpp, [17](#)
- tableCipher, [7](#)
 - decrypt, [8](#)
 - encrypt, [9](#)
 - isValidRussianText, [9](#)
 - prepareText, [9](#)
 - tableCipher, [8](#)
 - toUpper, [10](#)
 - validateKey, [10](#)
 - validateTextLength, [11](#)
- tableCipher.cpp, [17](#)
- tableCipher.h, [18](#)
- tableCipher_error, [11](#)
 - tableCipher_error, [12](#), [13](#)
 - what, [13](#)
- toUpper
 - tableCipher, [10](#)
- validateKey
 - tableCipher, [10](#)
- validateTextLength
 - tableCipher, [11](#)
- what
 - tableCipher_error, [13](#)