

## Лабораторная работа №4: Документирование модулей шифра Гронсфельда

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() [1/2]	8
4.1.2.2 cipher_error() [2/2]	8
4.2 Класс modAlphaCipher	9
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	10
4.2.2.1 modAlphaCipher()	10
4.2.3 Методы	10
4.2.3.1 convert() [1/2]	10
4.2.3.2 convert() [2/2]	10
4.2.3.3 decrypt()	11
4.2.3.4 encrypt()	11
4.2.3.5 getValidCipherText()	12
4.2.3.6 getValidKey()	12
4.2.3.7 getValidOpenText()	13
4.2.3.8 toUpper()	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 check()	16
5.1.2.2 main()	16
5.1.2.3 testAlphaCipher()	17
5.2 Файл modAlphaCipher.cpp	17
5.2.1 Подробное описание	17
5.3 Файл modAlphaCipher.h	18
5.3.1 Подробное описание	18
5.4 modAlphaCipher.h	19
Предметный указатель	21



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
modAlphaCipher . . . . .	9



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">cipher_error</a>	Класс-исключение для ошибок шифрования . . . . .	7
<a href="#">modAlphaCipher</a>	Класс для шифрования методом Гронсфельда . . . . .	9





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">main.cpp</a>	Главный модуль для тестирования шифра Гронсфельда . . . . .	15
<a href="#">modAlphaCipher.cpp</a>	Реализация модуля шифрования методом Гронсфельда . . . . .	17
<a href="#">modAlphaCipher.h</a>	Заголовочный файл для модуля шифрования методом Гронсфельда . . . . .	18



## Глава 4

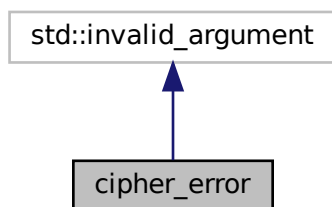
# Классы

### 4.1 Класс `cipher_error`

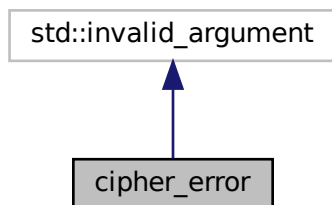
Класс-исключение для ошибок шифрования

```
#include <modAlphaCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

- [cipher\\_error](#) (const std::string &what\_arg)  
Конструктор с параметром типа std::string.
- [cipher\\_error](#) (const char \*what\_arg)  
Конструктор с параметром типа const char\*.

### 4.1.1 Подробное описание

Класс-исключение для ошибок шифрования

Наследуется от std::invalid\_argument для удобства обработки

### 4.1.2 Конструктор(ы)

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

```

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

```

Конструктор с параметром типа std::string.

Аргументы

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

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

```

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

```

Конструктор с параметром типа const char\*.

Аргументы

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

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

## 4.2 Класс modAlphaCipher

Класс для шифрования методом Гронсфельда

```
#include <modAlphaCipher.h>
```

### Открытые члены

- `modAlphaCipher ()=delete`  
Запрет конструктора без параметров
- `modAlphaCipher (const std::wstring &skey)`  
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::vector< int > convert (const std::wstring &s)`  
Преобразование строки в числовой вектор
- `std::wstring convert (const std::vector< int > &v)`  
Преобразование числового вектора в строку
- `std::wstring toUpper (const std::wstring &s)`  
Приведение строки к верхнему регистру с удалением пробелов
- `std::wstring getValidKey (const std::wstring &s)`  
Валидация и нормализация ключа
- `std::wstring getValidOpenText (const std::wstring &s)`  
Валидация открытого текста
- `std::wstring getValidCipherText (const std::wstring &s)`  
Валидация зашифрованного текста

### Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`  
Алфавит по порядку
- `std::map< wchar_t, int > alphaNum`  
Ассоциативный массив "символ-номер".
- `std::vector< int > key`  
Ключ в числовом представлении

#### 4.2.1 Подробное описание

Класс для шифрования методом Гронсфельда

Реализует шифр Гронсфельда для русского алфавита. Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

#### Предупреждения

Реализация только для русского языка

## 4.2.2 Конструктор(ы)

### 4.2.2.1 modAlphaCipher()

```
modAlphaCipher::modAlphaCipher (  
    const std::wstring & skey )
```

Конструктор с установкой ключа

Конструктор класса [modAlphaCipher](#).

Аргументы

skey	Ключ шифрования в виде строки
------	-------------------------------

Исключения

<a href="#">cipher_error</a>	Если ключ невалиден
------------------------------	---------------------

## 4.2.3 Методы

### 4.2.3.1 convert() [1/2]

```
std::wstring modAlphaCipher::convert (  
    const std::vector< int > & v ) [private]
```

Преобразование числового вектора в строку

Аргументы

v	Входной вектор чисел
---	----------------------

Возвращает

Строка, составленная из символов алфавита

### 4.2.3.2 convert() [2/2]

```
std::vector< int > modAlphaCipher::convert (  
    const std::wstring & s ) [private]
```

Преобразование строки в числовой вектор

Аргументы

s	Входная строка
---	----------------

Возвращает

Вектор чисел, соответствующих символам строки

#### 4.2.3.3 decrypt()

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

Метод расшифровывания

Метод расшифровывания зашифрованного текста

Аргументы

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

Возвращает

Расшифрованная строка

Исключения

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

#### 4.2.3.4 encrypt()

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

Метод зашифровывания

Метод зашифровывания открытого текста

Аргументы

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

Возвращает

Зашифрованная строка

Исключения

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

#### 4.2.3.5 getValidCipherText()

```
std::wstring modAlphaCipher::getValidCipherText (  
    const std::wstring & s ) [private]
```

Валидация зашифрованного текста

Аргументы

s	Зашифрованный текст
---	---------------------

Возвращает

Валидированный зашифрованный текст

Исключения

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

#### 4.2.3.6 getValidKey()

```
std::wstring modAlphaCipher::getValidKey (  
    const std::wstring & s ) [private]
```

Валидация и нормализация ключа

Аргументы

s	Ключ в виде строки
---	--------------------

Возвращает

Валидированный ключ в верхнем регистре



## Исключения

<a href="#">cipher_error</a>	Если ключ пустой, содержит недопустимые символы или слишком слабый
------------------------------	--

## 4.2.3.7 getValidOpenText()

```
std::wstring modAlphaCipher::getValidOpenText (
    const std::wstring & s ) [private]
```

## Валидация открытого текста

## Аргументы

s	Открытый текст
---	----------------

## Возвращает

Валидированный текст в верхнем регистре

## Исключения

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

## 4.2.3.8 toUpper()

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

## Приведение строки к верхнему регистру с удалением пробелов

## Аргументы

s	Входная строка
---	----------------

## Возвращает

Строка в верхнем регистре без пробелов

Объявления и описания членов классов находятся в файлах:

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



## Глава 5

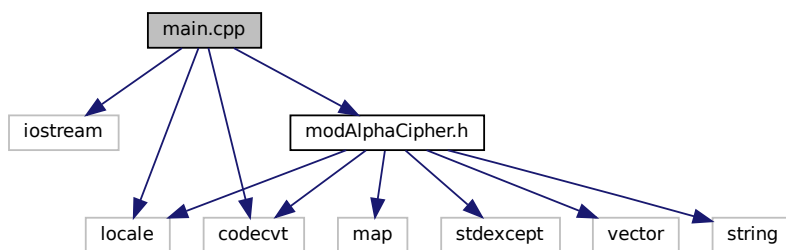
# Файлы

### 5.1 Файл main.cpp

Главный модуль для тестирования шифра Гронсфельда

```
#include <iostream>
#include <locale>
#include <codecvt>
#include "modAlphaCipher.h"
```

Граф включаемых заголовочных файлов для main.cpp:



### Функции

- void `check` (const wstring &text, const wstring &key, bool destructCipherText=false)  
Функция для проверки работы шифра
- void `testAlphaCipher` ()  
Функция тестирования шифра Гронсфельда
- int `main` ()  
Главная функция программы

### 5.1.1 Подробное описание

Главный модуль для тестирования шифра Гронсфельда

Автор

Гришин Н.С.

Версия

1.0

Дата

03.12.2025

Авторство

ИБСТ ПГУ

### 5.1.2 Функции

#### 5.1.2.1 check()

```
void check (
    const wstring & text,
    const wstring & key,
    bool destructCipherText = false )
```

Функция для проверки работы шифра

Аргументы

text	Текст для обработки
key	Ключ шифрования
destructCipherText	Флаг для порчи зашифрованного текста (для тестирования обработки ошибок)

#### 5.1.2.2 main()

```
int main ( )
```

Главная функция программы

Возвращает

0 при успешном выполнении

### 5.1.2.3 testAlphaCipher()

```
void testAlphaCipher ( )
```

Функция тестирования шифра Гронсфеляда

Выполняет серию тестов с различными входными данными

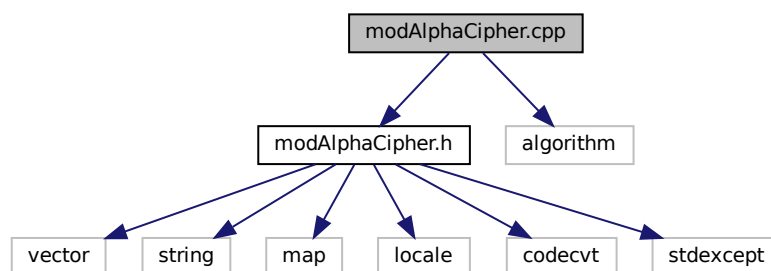
## 5.2 Файл modAlphaCipher.cpp

Реализация модуля шифрования методом Гронсфеляда

```
#include "modAlphaCipher.h"
```

```
#include <algorithm>
```

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



### 5.2.1 Подробное описание

Реализация модуля шифрования методом Гронсфеляда

Автор

Гришин Н.С.

Версия

1.0

Дата

03.12.2025

Авторство

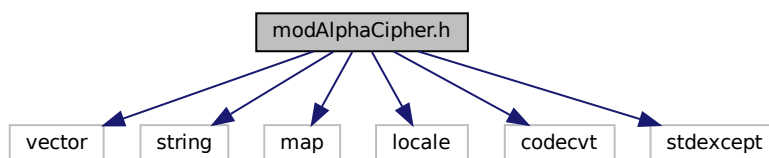
ИБСТ ПГУ

## 5.3 Файл modAlphaCipher.h

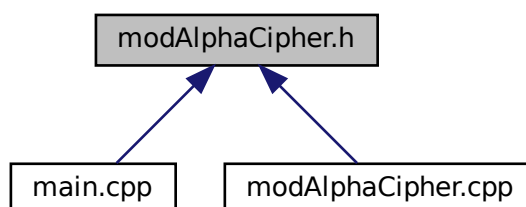
Заголовочный файл для модуля шифрования методом Гронсфельда

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



### Классы

- class [cipher\\_error](#)  
Класс-исключение для ошибок шифрования
- class [modAlphaCipher](#)  
Класс для шифрования методом Гронсфельда

#### 5.3.1 Подробное описание

Заголовочный файл для модуля шифрования методом Гронсфельда

Автор

Гришин Н.С.

Версия

1.0

Дата

03.12.2025

Авторство

ИБСТ ПГУ

## 5.4 modAlphaCipher.h

[См. документацию.](#)

```
1
10 #pragma once
11 #include <vector>
12 #include <string>
13 #include <map>
14 #include <locale>
15 #include <codecvt>
16 #include <stdexcept>
17
22 class cipher_error : public std::invalid_argument {
23 public:
28     explicit cipher_error(const std::string& what_arg) : std::invalid_argument(what_arg) {}
29
34     explicit cipher_error(const char* what_arg) : std::invalid_argument(what_arg) {}
35 };
36
44 class modAlphaCipher
45 {
46 private:
47     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
48     std::map<wchar_t, int> alphaNum;
49     std::vector<int> key;
50
56     std::vector<int> convert(const std::wstring& s);
57
63     std::wstring convert(const std::vector<int>& v);
64
70     std::wstring toUpper(const std::wstring& s);
71
78     std::wstring getValidKey(const std::wstring& s);
79
86     std::wstring getValidOpenText(const std::wstring& s);
87
94     std::wstring getValidCipherText(const std::wstring& s);
95
96 public:
100     modAlphaCipher() = delete;
101
107     modAlphaCipher(const std::wstring& skey);
108
115     std::wstring encrypt(const std::wstring& open_text);
116
123     std::wstring decrypt(const std::wstring& cipher_text);
124 };
```





# Предметный указатель

- check
  - main.cpp, [16](#)
- cipher\_error, [7](#)
  - cipher\_error, [8](#)
- convert
  - modAlphaCipher, [10](#)
- decrypt
  - modAlphaCipher, [11](#)
- encrypt
  - modAlphaCipher, [11](#)
- getValidCipherText
  - modAlphaCipher, [12](#)
- getValidKey
  - modAlphaCipher, [12](#)
- getValidOpenText
  - modAlphaCipher, [13](#)
- main
  - main.cpp, [16](#)
- main.cpp, [15](#)
  - check, [16](#)
  - main, [16](#)
  - testAlphaCipher, [17](#)
- modAlphaCipher, [9](#)
  - convert, [10](#)
  - decrypt, [11](#)
  - encrypt, [11](#)
  - getValidCipherText, [12](#)
  - getValidKey, [12](#)
  - getValidOpenText, [13](#)
  - modAlphaCipher, [10](#)
  - toUpper, [13](#)
- modAlphaCipher.cpp, [17](#)
- modAlphaCipher.h, [18](#)
- testAlphaCipher
  - main.cpp, [17](#)
- toUpper
  - modAlphaCipher, [13](#)