

My Project

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 modAlphaCipher	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 modAlphaCipher()	9
4.2.3 Методы	9
4.2.3.1 decrypt()	9
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 is_valid_key()	14
5.1.2.2 is_valid_russian_text()	14
5.1.2.3 main()	15
5.1.2.4 remove_spaces()	15
5.1.2.5 to_narrow()	15
5.1.2.6 to_upper_rus()	16
5.1.2.7 to_wide()	16
5.2 Файл modAlphaCipher.cpp	16
5.2.1 Подробное описание	17
5.3 Файл modAlphaCipher.h	17
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.

modAlphaCipher	8
std::runtime_error	
cipher_error	7

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Класс исключения для ошибок шифрования	7
modAlphaCipher	Класс для шифрования методом Гронсфельда	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный файл программы для шифра Гронсфельда	13
modAlphaCipher.cpp	Файл реализации класса modAlphaCipher	16
modAlphaCipher.h	Заголовочный файл для класса шифрования методом Гронсфельда	17

Глава 4

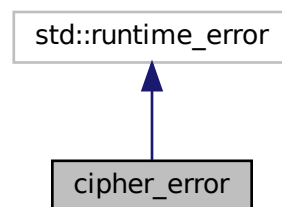
Классы

4.1 Класс `cipher_error`

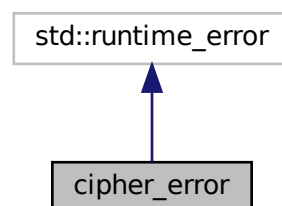
Класс исключения для ошибок шифрования

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

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- [cipher_error](#) (const std::string &what_arg)
Конструктор исключения

4.1.1 Подробное описание

Класс исключения для ошибок шифрования

4.1.2 Конструктор(ы)

4.1.2.1 cipher_error()

```

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

```

Конструктор исключения

Аргументы

what_arg	Описание ошибки
----------	-----------------

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
Запрещённый конструктор без параметров
- [modAlphaCipher](#) (const std::wstring &skey)
Конструктор для установки ключа
- std::wstring [encrypt](#) (const std::wstring &open_text)
Метод для зашифровывания текста
- std::wstring [decrypt](#) (const std::wstring &cipher_text)
Метод для расшифровывания текста

4.2.1 Подробное описание

Класс для шифрования методом Гронсфельда

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 для установки ключа

Аргументы

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

Исключения

cipher_error	Если ключ пустой или содержит недопустимые символы
------------------------------	--

4.2.3 Методы

4.2.3.1 decrypt()

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

Метод для расшифровывания текста

Расшифрование текста

Аргументы

cipher_text	Зашифрованный текст
-------------	---------------------

Возвращает

Расшифрованная строка

Исключения

cipher_error	Если текст пустой или содержит недопустимые символы
------------------------------	---

Аргументы

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

Возвращает

Расшифрованная строка

4.2.3.2 `encrypt()`

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

Метод для зашифровывания текста

Шифрование открытого текста

Аргументы

<code>open_text</code>	Открытый текст для шифрования
------------------------	-------------------------------

Возвращает

Зашифрованная строка

Исключения

cipher_error	Если текст пустой
------------------------------	-------------------

Аргументы

<code>open_text</code>	Текст для шифрования
------------------------	----------------------

Возвращает

Зашифрованная строка

Объявления и описания членов классов находятся в файлах:

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

Глава 5

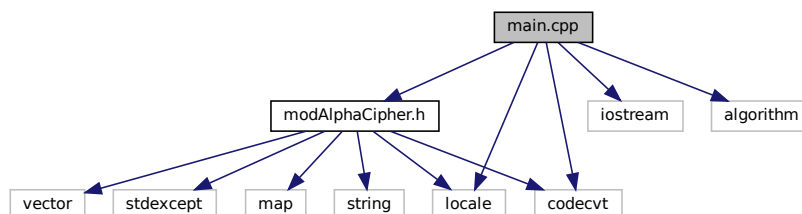
Файлы

5.1 Файл main.cpp

Главный файл программы для шифра Гронсфельда

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- `wstring remove_spaces (const wstring &s)`
Удаление пробелов из строки
- `bool is_valid_russian_text (const wstring &s)`
Проверка валидности русского текста
- `bool is_valid_key (const wstring &s)`
Проверка валидности ключа
- `wstring to_upper_rus (const wstring &s)`
Преобразование строки к верхнему регистру (русский)
- `wstring to_wide (const string &narrow_str)`
Преобразование narrow string в wide string.
- `string to_narrow (const wstring &wide_str)`
Преобразование wide string в narrow string.
- `int main ()`
Главная функция программы

5.1.1 Подробное описание

Главный файл программы для шифра Гронсфельда

Автор

Руслан

Версия

1.0

Дата

04.01.2026

Программа предоставляет интерфейс для шифрования и расшифрования текста методом Гронсфельда на русском языке.

5.1.2 Функции

5.1.2.1 `is_valid_key()`

```
bool is_valid_key (  
    const wstring & s )
```

Проверка валидности ключа

Аргументы

s	Ключ для проверки
---	-------------------

Возвращает

true если ключ содержит только русские буквы

5.1.2.2 `is_valid_russian_text()`

```
bool is_valid_russian_text (  
    const wstring & s )
```

Проверка валидности русского текста

Аргументы

s	Проверяемая строка
---	--------------------

Возвращает

true если текст содержит только русские буквы и пробелы

5.1.2.3 main()

```
int main ( )
```

Главная функция программы

Возвращает

Код завершения программы

5.1.2.4 remove_spaces()

```
wstring remove_spaces (  
    const wstring & s )
```

Удаление пробелов из строки

Аргументы

s	Исходная строка
---	-----------------

Возвращает

Строка без пробелов

5.1.2.5 to_narrow()

```
string to_narrow (  
    const wstring & wide_str )
```

Преобразование wide string в narrow string.

Аргументы

<code>wide_str</code>	Строка в формате <code>wstring</code>
-----------------------	---------------------------------------

Возвращает

Строка в кодировке UTF-8

5.1.2.6 `to_upper_rus()`

```
wstring to_upper_rus (  
    const wstring & s )
```

Преобразование строки к верхнему регистру (русский)

Аргументы

<code>s</code>	Исходная строка
----------------	-----------------

Возвращает

Строка в верхнем регистре

5.1.2.7 `to_wide()`

```
wstring to_wide (  
    const string & narrow_str )
```

Преобразование `narrow string` в `wide string`.

Аргументы

<code>narrow_str</code>	Строка в кодировке UTF-8
-------------------------	--------------------------

Возвращает

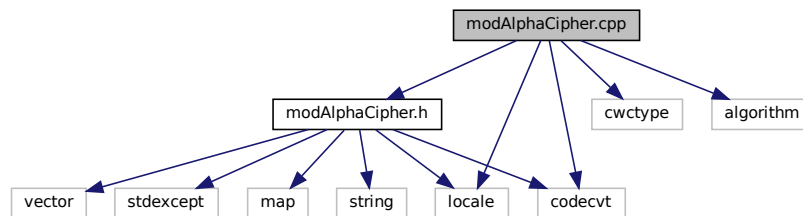
Строка в формате `wstring`

5.2 Файл `modAlphaCipher.cpp`

Файл реализации класса [modAlphaCipher](#).

```
#include "modAlphaCipher.h"
#include <locale>
#include <codecvt>
#include <cwctype>
#include <algorithm>
```

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



5.2.1 Подробное описание

Файл реализации класса [modAlphaCipher](#).

Автор

Руслан Жидков

Версия

1.0

Дата

04.01.2026

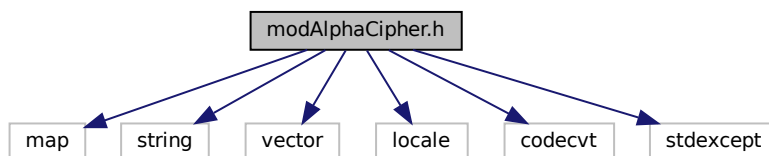
5.3 Файл modAlphaCipher.h

Заголовочный файл для класса шифрования методом Гронсфельда

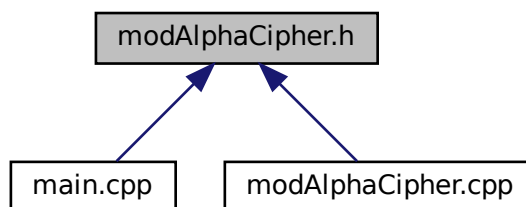
```
#include <map>
#include <string>
#include <vector>
#include <locale>
#include <codecvt>
```

```
#include <stdexcept>
```

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



Граф файлов, в которые включается этот файл:



Классы

- class `cipher_error`
Класс исключения для ошибок шифрования
- class `modAlphaCipher`
Класс для шифрования методом Гронсфельда

5.3.1 Подробное описание

Заголовочный файл для класса шифрования методом Гронсфельда

Автор

Руслан Жидков

Версия

1.0

Дата

04.01.2026

Авторство

ПГУ

Предупреждения

Реализация только для русского языка

5.4 modAlphaCipher.h

[См. документацию.](#)

```
1
11 #pragma once
12 #include <map>
13 #include <string>
14 #include <vector>
15 #include <locale>
16 #include <codecvt>
17 #include <stdexcept>
18
22 class cipher_error : public std::runtime_error {
23 public:
28     explicit cipher_error(const std::string& what_arg) : std::runtime_error(what_arg) {}
29 };
30
37 class modAlphaCipher
38 {
39 private:
40     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
41     std::map<wchar_t, int> alphaNum;
42     std::vector<int> key;
43
49     std::vector<int> convert(const std::wstring& s);
50
56     std::wstring convert(const std::vector<int>& v);
57
58 public:
62     modAlphaCipher() = delete;
63
69     modAlphaCipher(const std::wstring& skey);
70
77     std::wstring encrypt(const std::wstring& open_text);
78
85     std::wstring decrypt(const std::wstring& cipher_text);
86 };
```


Предметный указатель

- cipher_error, [7](#)
 - cipher_error, [8](#)
- decrypt
 - modAlphaCipher, [9](#)
- encrypt
 - modAlphaCipher, [10](#)
- is_valid_key
 - main.cpp, [14](#)
- is_valid_russian_text
 - main.cpp, [14](#)
- main
 - main.cpp, [15](#)
- main.cpp, [13](#)
 - is_valid_key, [14](#)
 - is_valid_russian_text, [14](#)
 - main, [15](#)
 - remove_spaces, [15](#)
 - to_narrow, [15](#)
 - to_upper_rus, [16](#)
 - to_wide, [16](#)
- modAlphaCipher, [8](#)
 - decrypt, [9](#)
 - encrypt, [10](#)
 - modAlphaCipher, [9](#)
- modAlphaCipher.cpp, [16](#)
- modAlphaCipher.h, [17](#)
- remove_spaces
 - main.cpp, [15](#)
- to_narrow
 - main.cpp, [15](#)
- to_upper_rus
 - main.cpp, [16](#)
- to_wide
 - main.cpp, [16](#)