

Романов\_ТМР\_ЛАВА4

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() [1/2]	10
4.2.2.2 modAlphaCipher() [2/2]	10
4.2.3 Методы	10
4.2.3.1 convert() [1/2]	10
4.2.3.2 convert() [2/2]	11
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
5 Файлы	15
5.1 Файл modAlphaCipher.h	15
5.1.1 Подробное описание	16
5.2 modAlphaCipher.h	16
Предметный указатель	19



# 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>	Исключение для обработки ошибок шифрования . . . . .	7
<a href="#">modAlphaCipher</a>	Класс для шифрования и расшифровки текста с использованием модифицированного алфавитного шифра . . . . .	9





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

[modAlphaCipher.h](#)

Заголовочный файл для класса [modAlphaCipher](#), реализующего шифрование и  
расшифровку текста . . . . . 15



## Глава 4

# Классы

### 4.1 Класс `cipher_error`

Исключение для обработки 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 с сообщением об ошибке.
- [cipher\\_error](#) (const char \*what\_arg)  
Конструктор исключения с сообщением об ошибке (C-строка).

### 4.1.1 Подробное описание

Исключение для обработки ошибок шифрования.

Данный класс расширяет `std::invalid_argument` и используется для обработки ошибок, возникающих при работе с классом [modAlphaCipher](#).

### 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 исключения с сообщением об ошибке.

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 исключения с сообщением об ошибке (C-строка).

Аргументы

what_arg	Сообщение об ошибке (C-строка).
----------	---------------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

## 4.2 Класс modAlphaCipher

Класс для шифрования и расшифровки текста с использованием модифицированного алфавитного шифра.

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

### Открытые члены

- `modAlphaCipher ()=delete`  
Конструктор класса `modAlphaCipher` (удален).
- `modAlphaCipher (const std::wstring &key)`  
Конструктор класса `modAlphaCipher` с заданным ключом.
- `std::wstring encrypt (const std::wstring &open_text)`  
Шифрует открытый текст.
- `std::wstring decrypt (const std::wstring &cipher_text)`  
Расшифровывает зашифрованный текст.

### Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`  
Преобразует строку в вектор чисел.
- `std::wstring convert (const std::vector< int > &v)`  
Преобразует вектор чисел обратно в строку.
- `std::wstring getValidKey (const std::wstring &s)`  
Проверяет валидность ключа.
- `std::wstring getValidOpenText (const std::wstring &ws)`  
Проверяет валидность открытого текста.
- `std::wstring getValidCipherText (const std::wstring &ws)`  
Проверяет валидность зашифрованного текста.

### Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`  
Русский алфавит в виде строки.
- `std::map< char, int > alphaNum`  
Отображение символов алфавита на их числовые значения.
- `std::vector< int > key`  
Ключ для шифрования/расшифровки.

#### 4.2.1 Подробное описание

Класс для шифрования и расшифровки текста с использованием модифицированного алфавитного шифра.

Данный класс реализует шифр, который использует русский алфавит и ключ для шифрования и расшифровки текстов. Он включает методы для преобразования текста в числовой формат и обратно, а также для проверки валидности ключа и текста.

## 4.2.2 Конструктор(ы)

### 4.2.2.1 modAlphaCipher() [1/2]

`modAlphaCipher::modAlphaCipher ( )` [delete]

Конструктор класса `modAlphaCipher` (удален).

Конструктор не может быть вызван без параметров. Необходимо передать ключ.

### 4.2.2.2 modAlphaCipher() [2/2]

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

Конструктор класса `modAlphaCipher` с заданным ключом.

Аргументы

<code>skey</code>	Ключ для шифрования/расшифровки.
-------------------	----------------------------------

Исключения

<code>cipher_error</code>	Если ключ не валиден.
---------------------------	-----------------------

## 4.2.3 Методы

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

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

Преобразует вектор чисел обратно в строку.

Аргументы

<code>v</code>	Вектор целых чисел для преобразования.
----------------	--

Возвращает

Строка, соответствующая вектору чисел.

## 4.2.3.2 convert() [2/2]

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

Преобразует строку в вектор чисел.

Аргументы

s	Строка для преобразования.
---	----------------------------

Возвращает

Вектор целых чисел, соответствующий символам строки.

## 4.2.3.3 decrypt()

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

Расшифровывает зашифрованный текст.

Аргументы

cipher_text	Зашифрованный текст для расшифровки.
-------------	--------------------------------------

Возвращает

Открытый текст в виде строки.

## 4.2.3.4 encrypt()

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

Шифрует открытый текст.

Аргументы

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

Возвращает

Зашифрованный текст в виде строки.

#### 4.2.3.5 `getValidCipherText()`

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

Проверяет валидность зашифрованного текста.

Аргументы

ws	Зашифрованный текст для проверки.
----	-----------------------------------

Возвращает

Валидный зашифрованный текст в виде строки.

Исключения

<a href="#">cipher_error</a>	Если текст не валиден.
------------------------------	------------------------

#### 4.2.3.6 `getValidKey()`

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

Проверяет валидность ключа.

Аргументы

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

Возвращает

Валидный ключ в виде строки.

Исключения

<a href="#">cipher_error</a>	Если ключ не валиден.
------------------------------	-----------------------



## 4.2.3.7 getValidOpenText()

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

Проверяет валидность открытого текста.

Аргументы

ws	Открытый текст для проверки.
----	------------------------------

Возвращает

Валидный открытый текст в виде строки.

Исключения

<a href="#">cipher_error</a>	Если текст не валиден.
------------------------------	------------------------

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)



## Глава 5

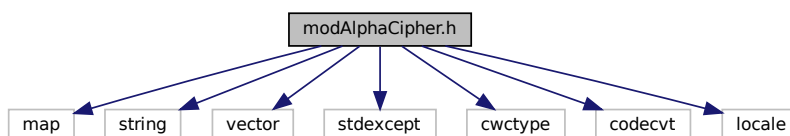
# Файлы

### 5.1 Файл modAlphaCipher.h

Заголовочный файл для класса `modAlphaCipher`, реализующего шифрование и расшифровку текста.

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

Граф включаемых заголовочных файлов для `modAlphaCipher.h`:



## Классы

- class `modAlphaCipher`  
Класс для шифрования и расшифровки текста с использованием модифицированного алфавитного шифра.
- class `cipher_error`  
Исключение для обработки ошибок шифрования.

### 5.1.1 Подробное описание

Заголовочный файл для класса `modAlphaCipher`, реализующего шифрование и расшифровку текста.

Автор

Пономарев А.А

Версия

1.0

Дата

04.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Данный модуль реализует шифр модифицированного алфавита.

## 5.2 modAlphaCipher.h

[См. документацию.](#)

```
1
10 #pragma once
11 #include <map>
12 #include <string>
13 #include <vector>
14 #include <stdexcept>
15 #include <cwctype>
16 #include <codecvt>
17 #include <locale>
18
27 class modAlphaCipher
28 {
29 private:
30     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
31     std::map<char, int> alphaNum;
32     std::vector<int> key;
33
39     std::vector<int> convert(const std::wstring& s);
40
46     std::wstring convert(const std::vector<int>& v);
47
54     std::wstring getValidKey(const std::wstring& s);
55
62     std::wstring getValidOpenText(const std::wstring & ws);
63
70     std::wstring getValidCipherText(const std::wstring & ws);
71
72 public:
77     modAlphaCipher() = delete;
78
84     modAlphaCipher(const std::wstring& skey);
85
91     std::wstring encrypt(const std::wstring& open_text);
92
98     std::wstring decrypt(const std::wstring& cipher_text);
99 };
```

```
100
108 class cipher_error : public std::invalid_argument
109 {
110 public:
115     explicit cipher_error(const std::string& what_arg)
116         : std::invalid_argument(what_arg)
117     {
118     }
119
124     explicit cipher_error(const char* what_arg)
125         : std::invalid_argument(what_arg)
126     {
127     }
128 };
```



# Предметный указатель

- cipher\_error, [7](#)
  - cipher\_error, [8](#)
- convert
  - modAlphaCipher, [10](#)
- decrypt
  - modAlphaCipher, [11](#)
- encrypt
  - modAlphaCipher, [11](#)
- getValidCipherText
  - modAlphaCipher, [12](#)
- getValidKey
  - modAlphaCipher, [12](#)
- getValidOpenText
  - modAlphaCipher, [12](#)
- modAlphaCipher, [9](#)
  - convert, [10](#)
  - decrypt, [11](#)
  - encrypt, [11](#)
  - getValidCipherText, [12](#)
  - getValidKey, [12](#)
  - getValidOpenText, [12](#)
  - modAlphaCipher, [10](#)
- modAlphaCipher.h, [15](#)