

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
4.2 Класс modAlphaCipher . . . . .	8
4.2.1 Конструктор(ы) . . . . .	8
4.2.1.1 modAlphaCipher() . . . . .	8
4.2.2 Методы . . . . .	9
4.2.2.1 encrypt() . . . . .	9
5 Файлы	11
5.1 header.h . . . . .	11
Предметный указатель	13



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error . . . . .	7
modAlphaCipher . . . . .	8



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<code>cipher_error</code>	7
<code>modAlphaCipher</code>	8





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

`/home/stud/Desktop/дз/Laba_timp/1proga/header.h` . . . . . 11

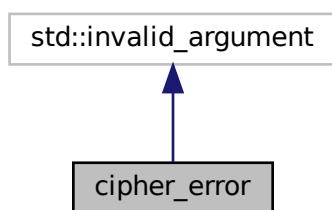


## Глава 4

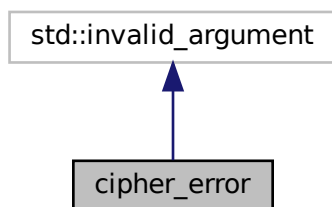
# Классы

### 4.1 Класс cipher\_error

Граф наследования: cipher\_error:



Граф связей класса cipher\_error:



## Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

Объявления и описания членов класса находятся в файле:

- `/home/stud/Desktop/дз/Laba_timp/1proga/header.h`

## 4.2 Класс modAlphaCipher

### Открытые члены

- `modAlphaCipher (const wstring &skey)`  
Класс для шифровки текста
- `wstring encrypt (const wstring &open_text)`  
Метод для шифровки текста
- `wstring decrypt (const wstring &cipher_text)`

### Закрытые члены

- `vector< int > convert (const wstring &s)`
- `wstring convert (const vector< int > &v)`
- `wstring getValidKey (const wstring &s)`
- `wstring getValidOpenText (const wstring &s)`
- `wstring getValidCipherText (const wstring &s)`

### Закрытые данные

- `const wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
- `map< wchar_t, int > alphaNum`
- `vector< int > key`

### 4.2.1 Конструктор(ы)

#### 4.2.1.1 modAlphaCipher()

```
modAlphaCipher::modAlphaCipher (
    const wstring & skey ) [explicit]
```

Класс для шифровки текста

Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

### Предупреждения

Реализация для русского языка

### 4.2.2 Методы

#### 4.2.2.1 encrypt()

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

Метод для шифровки текста

Передаётся текст, который преобразуется в шифр методом прибавления к каждому символу ключа. Для того, чтобы не выходить за границы алфавита, используется деление по модулю.

Объявления и описания членов классов находятся в файлах:

- /home/stud/Desktop/дз/Laba\_timp/1proga/header.h
- /home/stud/Desktop/дз/Laba\_timp/1proga/modAlphaCipher.cpp



## Глава 5

# Файлы

### 5.1 header.h

```
1 #pragma once
2 #include <vector>
3 #include <string>
4 #include <map>
5 #include <stdexcept>
6
7 using namespace std;
8
9 class modAlphaCipher {
10 private:
11 /*
12  * @brief Алфавит из русских букв
13  * @details используется для шифрования/дешифрования
14  */
15     const wstring numAlpha = L"АВВГДЕЁЖЗЙЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"; // Алфавит
16 /*
17  * @brief Массив, содержащий русские буквы и соответствующие им порядковые номера
18  * @details используется для шифрования/дешифрования
19  */
20     map<wchar_t, int> alphaNum; // Соответствие букв и индексов string char
21 /*
22  * @brief Ключ в виде вектора
23  * @details используется для шифрования
24  */
25     vector<int> key; // Ключ шифрования
26
27     vector<int> convert(const wstring& s); // Преобразование строки в вектор индексов
28     wstring convert(const vector<int>& v); // Преобразование вектора индексов в строку
29     wstring getValidKey(const wstring& s); // Проверка и получение валидного ключа
30     wstring getValidOpenText(const wstring& s); // Проверка и получение валидного открытого текста
31     wstring getValidCipherText(const wstring& s); // Проверка и получение валидного зашифрованного текста
32
33 public:
34     modAlphaCipher() = delete; // Удаляем конструктор по умолчанию
35     explicit modAlphaCipher(const wstring& skey); // Конструктор с ключом
36     wstring encrypt(const wstring& open_text); // Шифрование текста
37     wstring decrypt(const wstring& cipher_text); // Дешифрование текста
38 };
39 /*
40  * @brief Класс для обработки ошибок
41  * @details
42  */
43 class cipher_error : public std::invalid_argument {
44 public:
45 /*
46  * @brief Конструктор класса исключений cipher_error.
47  *
48  * @param what_arg Строковое сообщение об ошибке.
49  */
50     explicit cipher_error(const std::string& what_arg) : std::invalid_argument(what_arg) {}
51 /*
52  * @brief Конструктор класса исключений cipher_error.
53  *
54  * @param what_arg Символьная строка об ошибке.
55  */
56     explicit cipher_error(const char* what_arg) : std::invalid_argument(what_arg) {}
57 };
```





# Предметный указатель

`/home/stud/Desktop/дз/Laba_timp/1proga/header.h,`  
[11](#)

`cipher_error,` [7](#)

`encrypt`  
    `modAlphaCipher,` [9](#)

`modAlphaCipher,` [8](#)  
    `encrypt,` [9](#)  
    `modAlphaCipher,` [8](#)