

Shifry 4 labwork

1

Создано системой Doxygen 1.9.1



---

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
4.2 Класс modAlphaCipher . . . . .	8
4.2.1 Подробное описание . . . . .	9
4.2.2 Методы . . . . .	9
4.2.2.1 convert() [1/2] . . . . .	9
4.2.2.2 convert() [2/2] . . . . .	9
4.2.2.3 decrypt() . . . . .	10
4.2.2.4 encrypt() . . . . .	10
4.2.2.5 getValidCipherText() . . . . .	11
4.2.2.6 getValidKey() . . . . .	11
4.2.2.7 getValidOpenText() . . . . .	12
5 Файлы	13
5.1 Файл modAlphaCipher.h . . . . .	13
5.1.1 Подробное описание . . . . .	14
Предметный указатель	15



# 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.

<a href="#">cipher_error</a>	
класс-исключение <a href="#">cipher_error</a> . . . . .	7
<a href="#">modAlphaCipher</a>	
Шифрование методом Гронсфельда . . . . .	8





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">modAlphaCipher.h</a>	
Заголовочный файл для модуля Gronsfeld . . . . .	<a href="#">13</a>



## Глава 4

# Классы

### 4.1 Класс `cipher_error`

класс-исключение `cipher_error`.

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

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



## Открытые члены

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

### 4.1.1 Подробное описание

класс-исключение `cipher_error`.

производный от класса `std::invalid_argument` В данном классе перегружены конструкторы с параметрами.

При перегрузке явно указан вызов конструктора базового класса с параметром

Объявления и описания членов класса находятся в файле:

- `modAlphaCipher.h`

## 4.2 Класс `modAlphaCipher`

Шифрование методом Гронсфелда

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

## Открытые члены

- `modAlphaCipher ()=delete`  
запретим конструктор без параметров
- `modAlphaCipher (const std::wstring &skey)`  
конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`  
Зашифровывание
- `std::wstring decrypt (const std::wstring &cipher_text)`  
Расшифровывание

## Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`  
Преобразование строка-вектор
- `std::wstring convert (const std::vector< int > &v)`  
Преобразование вектор-строка
- `std::wstring getValidKey (const std::wstring &s)`  
Проверка на хороший ключ
- `std::wstring getValidOpenText (const std::wstring &s)`  
Метод для валидации открытого текста  
Все символы не принадлежащие алфавиту игнорируются  
Все строчные символы преобразуются к прописным
- `std::wstring getValidCipherText (const std::wstring &s)`  
Метод для валидации текста зашифрования

## Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`  
алфавит по порядку
- `std::map< char, int > alphaNum`  
ассоциативный массив "номер по символу".
- `std::vector< int > key`  
ключ

### 4.2.1 Подробное описание

#### Шифрование методом Гронсфеляда

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

#### Предупреждения

Реализация только для русского языка

### 4.2.2 Методы

#### 4.2.2.1 `convert()` [1/2]

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

#### Преобразование вектор-строка

##### Аргументы

in	v	Вектор типа int
----	---	-----------------

#### Возвращает

string

#### 4.2.2.2 `convert()` [2/2]

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

#### Преобразование строка-вектор

## Аргументы

in	s	Строка
----	---	--------

## Возвращает

Вектор типа int

## 4.2.2.3 decrypt()

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

## Расшифровывание

## Аргументы

in	cipher_text	Строка для зашифрования
----	-------------	-------------------------

## Возвращает

Расшифрованная строка

## Исключения

<a href="#">cipher_error</a> , если	текст пустой
-------------------------------------	--------------

## 4.2.2.4 encrypt()

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

## Зашифровывание

## Аргументы

in	open_text	Открытый текст. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	-----------	---

## Возвращает

Зашифрованная строка

## Исключения

<code>cipher_error</code> ,если	текст пустой
---------------------------------	--------------

## 4.2.2.5 getValidCipherText()

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

Метод для валидации текста зашифрования

.

## Аргументы

in	s	Строка wstring
----	---	----------------

Возвращает

Строка wstring

## Исключения

<code>cipher_error</code> ,если	текст пустой и в тексте содержатся строчные буквы
---------------------------------	---

## 4.2.2.6 getValidKey()

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

Проверка на хороший ключ

## Аргументы

in	s	Строка wstring
----	---	----------------

Возвращает

Строка wstring

## Исключения

<code>cipher_error</code> ,если	ключ пустой или в ключе находится символ не принадлежащий алфавиту
---------------------------------	--

#### 4.2.2.7 getValidOpenText()

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

Метод для валидации открытого текста

Все символы не принадлежащие алфавиту игнорируются

Все строчные символы преобразуются к прописным

Аргументы

in	s	Строка wstring
----	---	----------------

Возвращает

Строка wstring

Исключения

<a href="#">cipher_error</a> , если	текст пустой
-------------------------------------	--------------

Объявления и описания членов классов находятся в файлах:

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



## Глава 5

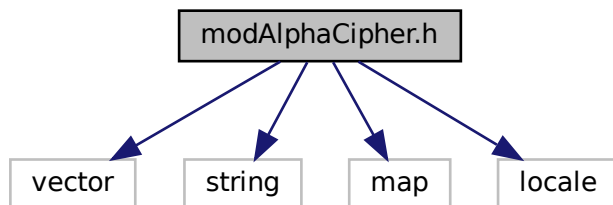
# Файлы

### 5.1 Файл modAlphaCipher.h

Заголовочный файл для модуля Gronsfeld.

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



## Классы

- class `modAlphaCipher`  
Шифрование методом Гронсфельда
- class `cipher_error`  
класс-исключение `cipher_error`.

### 5.1.1 Подробное описание

Заголовочный файл для модуля Gronsfeld.

Автор

Авдонин А.С.

Версия

1.0

Дата

5.12.23

Авторство

ИБСТ ПГУ

Предупреждения

Лабораторная работа №4

# Предметный указатель

- cipher\_error, [7](#)
- convert
  - modAlphaCipher, [9](#)
- decrypt
  - modAlphaCipher, [10](#)
- encrypt
  - modAlphaCipher, [10](#)
- getValidCipherText
  - modAlphaCipher, [11](#)
- getValidKey
  - modAlphaCipher, [11](#)
- getValidOpenText
  - modAlphaCipher, [12](#)
- modAlphaCipher, [8](#)
  - convert, [9](#)
  - decrypt, [10](#)
  - encrypt, [10](#)
  - getValidCipherText, [11](#)
  - getValidKey, [11](#)
  - getValidOpenText, [12](#)
- modAlphaCipher.h, [13](#)