

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 decrypt() . . . . .	9
4.2.2.2 encrypt() . . . . .	9
4.2.2.3 getValidKey() . . . . .	10
4.2.2.4 getValidKeyText() . . . . .	10
4.2.2.5 getValidOpenText() . . . . .	11
5 Файлы	13
5.1 Файл modAlphaCipher.h . . . . .	13
5.1.1 Подробное описание . . . . .	13
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 Табличной перестановки	8





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

[modAlphaCipher.h](#)

Заголовочный файл для модуля шифра Табличной перестановки . . . . . 13



## 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 Табличной перестановки

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

## Открытые члены

- `int getValidKey (int &s)`  
Метод для проверки ключа  
Если ключ меньше 0, то он считается неправильным  
.
- `int getValidKeyText (int key, std::string &text)`  
Метод для проверки соответствия длинны ключа длинне текста  
Длина ключа должна быть меньше длины сообщения  
.
- `std::string getValidOpenText (std::string &s)`  
Метод для проверки текста для зашифрования/расшифрования  
Все символы не принадлежащие английскому алфавиту удаляются  
.
- `modAlphaCipher ()=delete`  
запретим конструктор без параметров
- `modAlphaCipher (const int &key, std::string)`  
конструктор для установки ключа
- `std::string encrypt (const std::string &user_str)`  
Зашифрование
- `std::string decrypt (const std::string &user_str)`  
Расшифровывание

## Закрытые данные

- `int newkey`  
ключ для конструктора

### 4.2.1 Подробное описание

Шифрование методом Табличной перестановки

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

Предупреждения

Реализация только для английского языка

### 4.2.2 Методы

#### 4.2.2.1 decrypt()

```
std::string modAlphaCipher::decrypt (  
    const std::string & user_str )
```

Расшифровывание

Аргументы

in	user_str	Строка для зашифрования
----	----------	-------------------------

Возвращает

Расшифрованная строка

#### 4.2.2.2 encrypt()

```
std::string modAlphaCipher::encrypt (  
    const std::string & user_str )
```

Зашифрование

Аргументы

in	user_str	Строка для зашифрования
----	----------	-------------------------

Возвращает

Зашифрованная строка

#### 4.2.2.3 getValidKey()

```
int modAlphaCipher::getValidKey (  
    int & s )
```

Метод для проверки ключа

Если ключ меньше 0, то он считается неправильным

.

Аргументы

in	s	число типа int
----	---	----------------

Возвращает

число типа int

Исключения

<a href="#">cipher_error</a> ,если	ключ меньше 0
------------------------------------	---------------

#### 4.2.2.4 getValidKeyText()

```
int modAlphaCipher::getValidKeyText (  
    int key,  
    std::string & text )
```

Метод для проверки соответствия длинны ключа длине текста

Длина ключа должна быть меньше длины сообщения

.

Аргументы

in	text	строка с сообщением типа string
in	key	число типа int

Возвращает

строка типа string

Исключения

<a href="#">cipher_error</a> ,если	строка короче ключа
------------------------------------	---------------------

## 4.2.2.5 getValidOpenText()

```
std::string modAlphaCipher::getValidOpenText (
    std::string & s )
```

Метод для проверки текста для зашифрования/расшифрования  
Все символы не принадлежащие английскому алфавиту удаляются  
.

Аргументы

in	s	строка с сообщением типа string
----	---	---------------------------------

Возвращает

строка типа string

Исключения

<a href="#">cipher_error</a> ,если	строка пуста
------------------------------------	--------------

Объявления и описания членов классов находятся в файлах:

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





## Глава 5

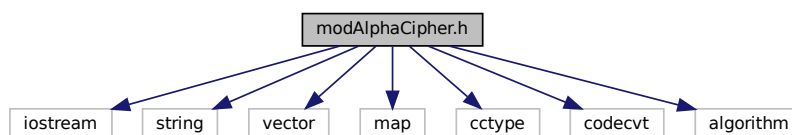
# Файлы

### 5.1 Файл modAlphaCipher.h

Заголовочный файл для модуля шифра Табличной перестановки

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <cctype>
#include <codecvt>
#include <algorithm>
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



## Классы

- class `modAlphaCipher`  
Шифрование методом Табличной перестановки
- class `cipher_error`  
класс-исключение `cipher_error`.

### 5.1.1 Подробное описание

Заголовочный файл для модуля шифра Табличной перестановки

Автор

Авдонин А.С.

Версия

1.0

Дата

5.12.23

Авторство

ИБСТ ПГУ

Предупреждения

Лабораторная работа №4

# Предметный указатель

`cipher_error`, [7](#)

`decrypt`

`modAlphaCipher`, [9](#)

`encrypt`

`modAlphaCipher`, [9](#)

`getValidKey`

`modAlphaCipher`, [9](#)

`getValidKeyText`

`modAlphaCipher`, [10](#)

`getValidOpenText`

`modAlphaCipher`, [10](#)

`modAlphaCipher`, [8](#)

`decrypt`, [9](#)

`encrypt`, [9](#)

`getValidKey`, [9](#)

`getValidKeyText`, [10](#)

`getValidOpenText`, [10](#)

`modAlphaCipher.h`, [13](#)