

modAlphaCipher\Солдатенков 22ПТ2

Создано системой Doxygen 1.9.4

1 Иерархический список классов	1
1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	1
2.1 Классы	1
3 Список файлов	2
3.1 Файлы	2
4 Классы	2
4.1 Класс cipher_error	2
4.2 Класс modAlphaCipher	2
4.2.1 Подробное описание	3
4.2.2 Методы	3
5 Файлы	6
5.1 Файл modAlphaCipher.cpp	6
5.1.1 Подробное описание	6
5.2 Файл modAlphaCipher.h	6
5.2.1 Подробное описание	7
5.3 modAlphaCipher.h	7
Предметный указатель	9

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	2
modAlphaCipher	2

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	
Класс исключений для ошибок шифрования	2
modAlphaCipher	
Класс для шифрования и дешифрования текста методом простой замены на основе алфавита	2

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

modAlphaCipher.cpp	
Исполняемый файл класса modAlphaCipher	6
modAlphaCipher.h	
Определение класса modAlphaCipher для шифрования и дешифрования методом простой замены	6

4 Классы

4.1 Класс cipher_error

Класс исключений для ошибок шифрования.

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

Граф наследования: cipher_error:

4.2 Класс modAlphaCipher

Класс для шифрования и дешифрования текста методом простой замены на основе алфавита.

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

Открытые члены

- `std::wstring getValidString (const std::wstring &s)`
Универсальная функция проверки правильности строки. Небуквенные символы пропускаются. Если строка пустая или содержит буквы из нерусского алфавита, то возбуждается исключение
- `std::wstring key_validation (std::wstring key, const std::wstring &text)`
Проверка ключа. Используется метод `getValidString`, затем ключ дополняется до необходимой длины
- `void start (const std::wstring &skey)`
Инициализация объекта класса заданным ключом.
- `modAlphaCipher ()`
Конструктор по умолчанию. Используется для внедрения корректной обработки ошибок ключа
- `std::wstring encrypt (const std::wstring &open_text)`
Шифрует переданный открытый текст с использованием алфавитного шифра.
- `std::wstring decrypt (const std::wstring &cipher_text)`
Дешифрует переданный зашифрованный текст с использованием алфавитного шифра.

Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`
Конвертирует строку в вектор числовых значений.
- `std::wstring convert (const std::vector< int > &v)`
Конвертирует вектор числовых значений обратно в строку.

Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Русский алфавит
- `std::map< wchar_t, int > alphaNum`
Словарь соответствий символов алфавита их числовым значениям
- `std::vector< int > key`
Вектор числовых значений ключа

4.2.1 Подробное описание

Класс для шифрования и дешифрования текста методом простой замены на основе алфавита.

4.2.2 Методы

4.2.2.1 `convert()` [1/2] `std::wstring modAlphaCipher::convert (`
`const std::vector< int > & v)` [inline], [private]

Конвертирует вектор числовых значений обратно в строку.

Аргументы

v	Вектор числовых значений.
---	---------------------------

Возвращает

Строка.

4.2.2.2 `convert()` [2/2] `std::vector< int > modAlphaCipher::convert (`
`const std::wstring & s)` [inline], [private]

Конвертирует строку в вектор числовых значений.

Аргументы

s	Строка для конвертации.
---	-------------------------

Возвращает

Вектор числовых значений.

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

Дешифрует переданный зашифрованный текст с использованием алфавитного шифра.

Функция принимает зашифрованный текст, преобразует его в числовые значения, выполняет поэлементное вычитание ключа по модулю длины алфавита и возвращает расшифрованный текст.

Аргументы

<code>cipher_text</code>	Зашифрованный текст, который необходимо расшифровать.
--------------------------	---

Возвращает

`std::wstring` Расшифрованный текст, полученный в результате дешифрования.

Исключения

<code>cipher_error</code>	Если входной текст содержит недопустимые символы или является пустым.
---------------------------	---

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

Шифрует переданный открытый текст с использованием алфавитного шифра.

Функция принимает открытый текст, преобразует его в числовые значения, выполняет поэлементное сложение с ключом по модулю длины алфавита и возвращает зашифрованный текст.

Аргументы

<code>open_text</code>	Открытый текст, который необходимо зашифровать.
------------------------	---

Возвращает

`std::wstring` Зашифрованный текст, полученный в результате шифрования.

Исключения

<code>cipher_error</code>	Если входной текст содержит недопустимые символы или является пустым.
---------------------------	---

4.2.2.5 `getValidString()` `std::wstring modAlphaCipher::getValidString (`
`const std::wstring & s) [inline]`

Универсальная функция проверки правильности строки. Небуквенные символы пропускаются. Если строка пустая или содержит буквы из нерусского алфавита, то возбуждается исключение

Аргументы

s	Входная строка.
---	-----------------

Возвращает

Очищенная и валидная строка.

4.2.2.6 `key_validation()` `std::wstring modAlphaCipher::key_validation (`
`std::wstring key,`
`const std::wstring & text)`

Проверка ключа. Используется метод `getValidString`, затем ключ дополняется до необходимой длины

Аргументы

key	Ключ для проверки.
text	Исходный текст.

Возвращает

Скорректированный ключ.

4.2.2.7 `start()` `void modAlphaCipher::start (`
`const std::wstring & skey)`

Инициализация объекта класса заданным ключом.

Аргументы

skey	Ключ шифрования.
------	------------------

Объявления и описания членов классов находятся в файлах:

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

5 Файлы

5.1 Файл modAlphaCipher.cpp

Исполняемый файл класса [modAlphaCipher](#).

```
#include "modAlphaCipher.h"
```

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:

5.1.1 Подробное описание

Исполняемый файл класса [modAlphaCipher](#).

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.2 Файл modAlphaCipher.h

Определение класса [modAlphaCipher](#) для шифрования и дешифрования методом простой замены.

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

Граф включаемых заголовочных файлов для modAlphaCipher.h: Граф файлов, в которые включается этот файл:

Классы

- class [modAlphaCipher](#)

Класс для шифрования и дешифрования текста методом простой замены на основе алфавита.

- class [cipher_error](#)

Класс исключений для ошибок шифрования.

5.2.1 Подробное описание

Определение класса `modAlphaCipher` для шифрования и дешифрования методом простой замены.

Автор

Солдатенков А.Д.

Версия

1.0

Дата

08.01.2025

Авторство

ИБСТ ПГУ

5.3 modAlphaCipher.h

[См. документацию.](#)

```
1
8 #pragma once
9 #include <vector>
10 #include <iostream>
11 #include <string>
12 #include <map>
13 #include <cctype>
14 #include <locale>
15 #include <algorithm>
16
21 class modAlphaCipher
22 {
23 private:
24     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
25     std::map<wchar_t, int> alphaNum;
26     std::vector<int> key;
27
33     std::vector<int> convert(const std::wstring& s);
34
40     std::wstring convert(const std::vector<int>& v);
41
42 public:
49     std::wstring getValidString(const std::wstring& s);
50
57     std::wstring key_validation(std::wstring key, const std::wstring& text);
58
63     void start(const std::wstring& skey);
64
69     modAlphaCipher();
70
82     std::wstring encrypt(const std::wstring& open_text);
83
95     std::wstring decrypt(const std::wstring& cipher_text);
96
97 };
98
103 class cipher_error: public std::invalid_argument
104 {
105 public:
110     explicit cipher_error(const std::string& what_arg) : std::invalid_argument(what_arg) {}
111
116     explicit cipher_error(const char* what_arg) : std::invalid_argument(what_arg) {}
117 };
```


Предметный указатель

cipher_error, [2](#)

convert

modAlphaCipher, [3](#)

decrypt

modAlphaCipher, [4](#)

encrypt

modAlphaCipher, [4](#)

getValidString

modAlphaCipher, [5](#)

key_validation

modAlphaCipher, [5](#)

modAlphaCipher, [2](#)

convert, [3](#)

decrypt, [4](#)

encrypt, [4](#)

getValidString, [5](#)

key_validation, [5](#)

start, [5](#)

modAlphaCipher.cpp, [6](#)

modAlphaCipher.h, [6](#)

start

modAlphaCipher, [5](#)