

Лабораторная работа №4.1

1.0

Создано системой Doxygen 1.9.1

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	3
4.2.1 Подробное описание	3
4.2.2 Конструктор(ы)	4
4.2.3 Методы	4
5 Файлы	7
5.1 Файл modAlphaCipher.h	7
5.1.1 Подробное описание	8
Предметный указатель	9

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	2
modAlphaCipher	3

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	2
modAlphaCipher	
Класс для шифрования и расшифрования табличным перестановочным шифром. При- мер использования класса для шифрования и расшифрования	3

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

[modAlphaCipher.h](#)

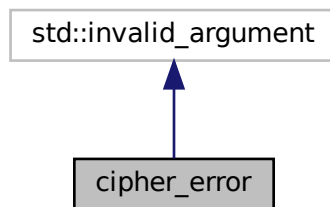
Класс для шифрования перестановочным шифром

[7](#)

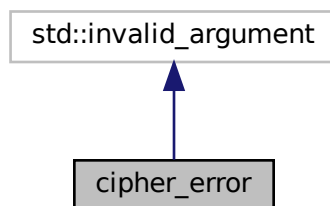
4 Классы

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:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Класс для шифрования и расшифрования табличным перестановочным шифром. Пример использования класса для шифрования и расшифрования.

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

Открытые члены

- `modAlphaCipher ()=delete`
запрет конструктора без параметров
- `modAlphaCipher (const std::wstring &skey)`
Конструктор для установки ключа.
- `std::wstring encrypt (const std::wstring &open_text)`
Зашифрование текста табличным перестановочным шифром.
- `std::wstring decrypt (const std::wstring &cipher_text)`
Расшифрование текста табличным перестановочным шифром.

Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`
Преобразование строки в целочисленный вектор.
- `std::wstring convert (const std::vector< int > &v)`
Преобразование целочисленного вектора в строку.
- `std::wstring getValidKey (const std::wstring &s)`
Проверка ключа на корректность
- `std::wstring getValidOpenText (const std::wstring &s)`
Проверка текста для шифрования на корректность.
- `std::wstring getValidCipherText (const std::wstring &s)`
Проверка шифротекста на корректность.

Закрытые данные

- `std::string numAlpha = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
алфавит шифра.
- `wstring wnumAlpha`
алфавит шифра типа wstring.
- `std::map< wchar_t, int > alphaNum`
ассоциативный массив "номер по символу".
- `std::vector< int > key`
вектор для ключа в виде чисел.

4.2.1 Подробное описание

Класс для шифрования и расшифрования табличным перестановочным шифром. Пример использования класса для шифрования и расшифрования.

```
modAlphaCipher cipher(std::string "КЛЮЧ");
std::string code = cipher.encode(std::string "Сообщение");
std::cout << code << std::endl;
std::string text = cipher.decode(code);
std::cout << text << std::endl;
```

4.2.2 Конструктор(ы)

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

Конструктор для установки ключа.

Аргументы

in	ключ, std::wstring.	
----	---------------------	--

Исключения

cipher_error	в случае некорректного ключа. Параметры исключения: type = invalid_argument, what = "Invalid key"
------------------------------	---

4.2.3 Методы

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

Преобразование целочисленного вектора в строку.

Преобразует целочисленный вектор в строку по ассоциативному массиву.

Аргументы

in	вектор	std::vector<int>
----	--------	------------------

Возвращает

строка теста, std::wstring

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

Преобразование строки в целочисленный вектор.

Преобразует строку в целочисленный вектор по ассоциативному массиву.

Аргументы

in	строка	теста, std::wstring.
----	--------	----------------------

Возвращает

Вектор std::vector<int>

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

Расширение текста табличным перестановочным шифром.

Символы текста заменяются на символы из словаря. по ключу, символы ключа являются величиной смещения символа по алфавиту. Если строка пустая или в ней присутствуют символы кро прописных букв, возбуждает исключение.

Аргументы

in	строка	с шифротекстом, std::wstring.
----	--------	-------------------------------

Возвращает

Строка std::wstring расшифрованный текст.

Исключения

std::cipher_error	в случае некорректного шифротекста. type = invalid_argument, what ="Invalid cipher text"
-------------------	---

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

Заширование текста табличным перестановочным шифром.

Символы текста заменяются на символы из словаря по ключу, символы ключа являются величиной смещения символа по алфавиту. Если строка пустая или в ней отсутствуют буквы, возбуждает исключение.

Аргументы

in	строка	с текстом, std::wstring.
----	--------	--------------------------

Возвращает

Строка `std::wstring` шифротекста.

Исключения

<code>std::cipher_error</code>	в случае некорректного текста. <code>type = invalid_argument, what = "Invalid open text"</code>
--------------------------------	--

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

Проверка шифротекста на корректность.

Корректный текст: строка букв в верхнем регистре. Если регистр нижний, присутствуют символы кроме букв или строка пустая, возбуждает исключение.

Аргументы

in	строка	с шифротекстом, <code>std::wstring</code> .
----	--------	---

Возвращает

Строка `std::wstring` с шифротекстом без изменений.

Исключения

<code>std::cipher_error</code>	в случае некорректного шифротекста. <code>type = invalid_argument, what = "Invalid cipher text"</code>
--------------------------------	---

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

Проверка ключа на корректность

Корректный ключ: строка букв в верхнем регистре, если регистр нижний, возвращает строку. в верхнем регистре. Если присутствуют символы кроме букв, возбуждает исключение.

Аргументы

in	строка	с ключом, <code>std::wstring</code> .
----	--------	---------------------------------------

Возвращает

Строка `std::wstring` с исправленным ключом.

Исключения

<code>std::cipher_error</code>	в случае некорректного ключа. <code>type = invalid_argument, what = "Invalid key"</code>
--------------------------------	---

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

Проверка текста для шифрования на корректность.

Корректный текст: строка букв в верхнем регистре, если регистр нижний, возвращает строку. в верхнем регистре. Если присутствуют символы кроме букв, возвращает строку без этих символов. Если строка пустая или в ней отсутствуют буквы, возбуждает исключение.

Аргументы

in	строка	с текстом, <code>std::wstring</code> .
----	--------	--

Возвращает

Строка `std::wstring` с исправленным ключом.

Исключения

<code>std::cipher_error</code>	в случае некорректного текста. <code>type = invalid_argument, what = "Invalid open text"</code>
--------------------------------	--

Объявления и описания членов классов находятся в файлах:

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

5 Файлы

5.1 Файл `modAlphaCipher.h`

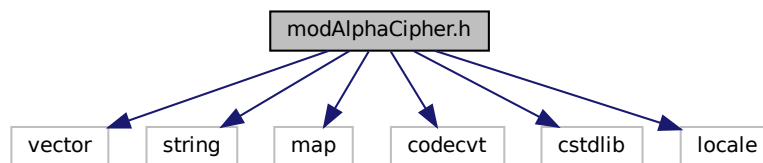
Класс для шифрования перестановочным шифром

```
#include <vector>
#include <string>
```



```
#include <map>
#include <codecvt>
#include <cstdlib>
#include <locale>
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class [modAlphaCipher](#)

Класс для шифрования и расшифрования табличным перестановочным шифром. Пример использования класса для шифрования и расшифрования.

- class [cipher_error](#)

5.1.1 Подробное описание

Класс для шифрования перестановочным шифром

Автор

Соколенко Н.С.

Версия

1.0

Дата

18.12.2022

Авторство

ИБСТ ПГУ

Предметный указатель

- cipher_error, [2](#)
- convert
 - modAlphaCipher, [4](#)
- decrypt
 - modAlphaCipher, [5](#)
- encrypt
 - modAlphaCipher, [5](#)
- getValidCipherText
 - modAlphaCipher, [6](#)
- getValidKey
 - modAlphaCipher, [6](#)
- getValidOpenText
 - modAlphaCipher, [7](#)
- modAlphaCipher, [3](#)
 - convert, [4](#)
 - decrypt, [5](#)
 - encrypt, [5](#)
 - getValidCipherText, [6](#)
 - getValidKey, [6](#)
 - getValidOpenText, [7](#)
 - modAlphaCipher, [4](#)
- modAlphaCipher.h, [7](#)