

TableTranspositionCipher

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 modBetaCipher	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 modBetaCipher()	9
4.2.3 Методы	9
4.2.3.1 convert() [1/2]	9
4.2.3.2 convert() [2/2]	10
4.2.3.3 decrypt()	10
4.2.3.4 encrypt()	11
4.2.3.5 getValidCipherText()	11
4.2.3.6 getValidKey()	11
4.2.3.7 getValidOpenText()	12
5 Файлы	13
5.1 Файл modBetaCipher.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
modBetaCipher	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error		
Класс исключений	7
modBetaCipher		
Шифрование методом Гронсфельда	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

modBetaCipher.h	
Заголовочный файл для модуля Гронсфельда	13

Глава 4

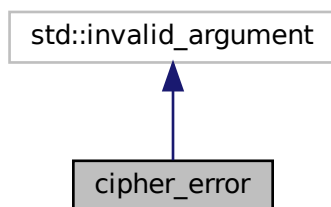
Классы

4.1 Класс `cipher_error`

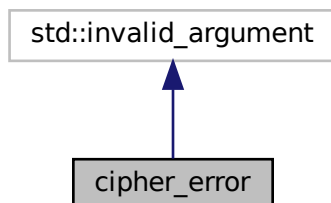
Класс исключений

```
#include <modBetaCipher.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- [cipher_error](#) (const std::string &what_arg)
Вызов конструктора базового класса с передачей параметра
- [cipher_error](#) (const char *what_arg)
Вызов конструктора базового класса с передачей параметра

4.1.1 Подробное описание

Класс исключений

Класс исключений для того чтобы отслеживать исключения. Когда возникает ошибка в процессе шифрования или дешифрования, объект этого класса может быть создан с сообщением об ошибке, которое затем может быть обработано в программе

Объявления и описания членов класса находятся в файле:

- [modBetaCipher.h](#)

4.2 Класс modBetaCipher

Шифрование методом Гронсфельда

```
#include <modBetaCipher.h>
```

Открытые члены

- [modBetaCipher](#) ()=delete
Удаляем конструктор по умолчанию
- [modBetaCipher](#) (const std::wstring &wskey)
Конструктор для установки ключа
- std::wstring [encrypt](#) (const std::wstring &open_text)
Функция зашифрования
- std::wstring [decrypt](#) (const std::wstring &cipher_text)
Функция расшифрования

Закрытые члены

- std::vector< int > [convert](#) (const std::wstring &ws)
Метод преобразования вектора в строку
- std::wstring [convert](#) (const std::vector< int > &v)
Метод преобразования строки в вектор
- std::wstring [getValidKey](#) (const std::wstring &ws)
Метод проверки ключа
- std::wstring [getValidOpenText](#) (const std::wstring &ws)
Метод проверки текста
- std::wstring [getValidCipherText](#) (const std::wstring &ws)
Метод проверки зашифрованного текста

Закрытые данные

- `std::wstring_convert< std::codecvt_utf8< wchar_t >, wchar_t > codec`
Используется для конвертации между `std::wstring` и UTF-8.
- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Алфавит
- `std::map< wchar_t, int > alphaNum`
Сопоставление символов алфавита и их числовых значений
- `std::vector< int > key`
Ключ шифрования/дешифрования

4.2.1 Подробное описание

Шифрование методом Гронсфельда

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

Предупреждения

Реализация только для русского языка

4.2.2 Конструктор(ы)

4.2.2.1 modBetaCipher()

```
modBetaCipher::modBetaCipher (
    const std::wstring & wskey )
```

Конструктор для установки ключа

Записывает в переменную класса вектор позиции букв полученные из строки с помощью `convert`

Аргументы

in	wskey	Строка-ключ @throw cipher_error , если ключ не соответствует требованиям Конструктор с параметром - устанавливает ключ шифрования/дешифрования
----	-------	---

4.2.3 Методы

4.2.3.1 convert() [1/2]

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

Метод преобразования строки в вектор

Каждой букве присваивается номер согласно алфавиту по порядку

Аргументы

in	v	Вектор позиций букв
----	---	---------------------

Возвращает

Строка Приватный метод для конвертации вектора чисел в строку

4.2.3.2 convert() [2/2]

```
std::vector<int> modBetaCipher::convert (
    const std::wstring & ws ) [private]
```

Метод преобразования вектора в строку

Каждой букве присваивается номер согласно алфавиту по порядку

Аргументы

in	ws	Строка
----	----	--------

Возвращает

Вектор букв со своими позициями Приватный метод для конвертации строки в вектор чисел

4.2.3.3 decrypt()

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

Функция расшифрования

Функция расшифровывает строку методом Гронсфельда

Аргументы

in	cipher_text	cipher_text @throw cipher_error , если строка не соответствует требованиям Метод для дешифрования зашифрованного текста
----	-------------	--

4.2.3.4 encrypt()

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

Функция зашифрования

Функция шифрует строку методом Гронсфельда

Аргументы

in	open_text	Строка с текстом @throw cipher_error, если строка не соответствует требованиям Метод для шифрования открытого текста
----	-----------	--

4.2.3.5 getValidCipherText()

```
std::wstring modBetaCipher::getValidCipherText (
    const std::wstring & ws ) [private]
```

Метод проверки зашифрованного текста

Преобразовывает строчные буквы в прописные

Аргументы

in	ws	Строка ключа
----	----	--------------

Возвращает

Строка @throw cipher_error, если текст не соответствует требованиям Приватный метод для проверки зашифрованного текста на валидность

4.2.3.6 getValidKey()

```
std::wstring modBetaCipher::getValidKey (
    const std::wstring & ws ) [private]
```

Метод проверки ключа

Преобразовывает строчные буквы в прописные

Аргументы

in	s	Строка ключа
----	---	--------------

Возвращает

Строка @throw [cipher_error](#), если ключ не соответствует требованиям Приватный метод для проверки ключа на валидность

4.2.3.7 getValidOpenText()

```
std::wstring modBetaCipher::getValidOpenText (  
    const std::wstring & ws ) [private]
```

Метод проверки текста

Преобразовывает строчные буквы в прописные

Аргументы

in	ws	Строка ключа
----	----	--------------

Возвращает

Строка @throw [cipher_error](#), если текст не соответствует требованиям Приватный метод для проверки открытого текста на валидность

Объявления и описания членов класса находятся в файле:

- [modBetaCipher.h](#)

Глава 5

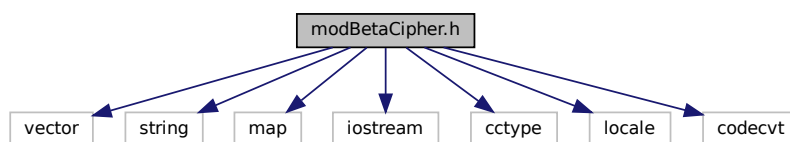
Файлы

5.1 Файл modBetaCipher.h

Заголовочный файл для модуля Гронсфельда

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

Граф включаемых заголовочных файлов для modBetaCipher.h:



Классы

- class `modBetaCipher`
Шифрование методом Гронсфельда
- class `cipher_error`
Класс исключений

5.1.1 Подробное описание

Заголовочный файл для модуля Гронсфельда

Автор

Кучеренков А.С.

Версия

1.0

Дата

23.01.2024

Авторство

ИБСТ ПГУ

Предупреждения

Тестовый

Предметный указатель

- cipher_error, [7](#)
- convert
 - modBetaCipher, [9](#), [10](#)
- decrypt
 - modBetaCipher, [10](#)
- encrypt
 - modBetaCipher, [10](#)
- getValidCipherText
 - modBetaCipher, [11](#)
- getValidKey
 - modBetaCipher, [11](#)
- getValidOpenText
 - modBetaCipher, [12](#)
- modBetaCipher, [8](#)
 - convert, [9](#), [10](#)
 - decrypt, [10](#)
 - encrypt, [10](#)
 - getValidCipherText, [11](#)
 - getValidKey, [11](#)
 - getValidOpenText, [12](#)
 - modBetaCipher, [9](#)
- modBetaCipher.h, [13](#)