

Project C++

Создано системой Doxygen 1.9.1

1	Алфавитный указатель классов	1
1.1	Классы	1
2	Классы	3
2.1	Класс <code>modPermCipher</code>	3
2.1.1	Подробное описание	3
2.1.2	Конструктор(ы)	4
2.1.2.1	<code>modPermCipher()</code>	4
2.1.3	Методы	4
2.1.3.1	<code>decrypt()</code>	4
2.1.3.2	<code>encrypt()</code>	5
2.1.3.3	<code>validateKey()</code>	5
2.1.3.4	<code>validateText()</code>	6
	Предметный указатель	7

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

<code>modPermCipher</code>	Класс для шифрования и расшифрования текста с использованием шифра перестановок	3
----------------------------	---	---

Глава 2

Классы

2.1 Класс modPermCipher

Класс для шифрования и расшифрования текста с использованием шифра перестановок.

```
#include <modPerm.h>
```

Открытые члены

- `modPermCipher` (const std::wstring &skey)
Конструктор, инициализирующий ключ шифрования.
- std::wstring `encrypt` (const std::wstring &open_text)
Шифрование текста.
- std::wstring `decrypt` (const std::wstring &cipher_text)
Расшифрование текста.
- void `validateKey` (const std::wstring &key)
Валидация ключа.
- void `validateText` (const std::wstring &text)
Валидация текста перед шифрованием или расшифрованием.

Закрытые данные

- std::wstring `alphabet`
Строка, представляющая алфавит для шифрования/расшифрования.
- std::vector< int > `key`
Вектор, представляющий ключ шифрования.

2.1.1 Подробное описание

Класс для шифрования и расшифрования текста с использованием шифра перестановок.

Этот класс поддерживает шифрование и расшифрование текста на основе заданного ключа, который определяет смещения символов в алфавите.

2.1.2 Конструктор(ы)

2.1.2.1 modPermCipher()

```
modPermCipher::modPermCipher (
    const std::wstring & skey )
```

Конструктор, инициализирующий ключ шифрования.

Этот метод принимает ключ в виде строки и преобразует его в вектор целых чисел.

Аргументы

skey	Строка, представляющая ключ для шифрования.
------	---

Исключения

std::invalid_argument	Если ключ пустой или содержит недопустимые символы.
-----------------------	---

Аргументы

skey	Строка, представляющая ключ для шифрования.
------	---

Исключения

std::invalid_argument	Если ключ пуст или содержит недопустимые символы.
-----------------------	---

2.1.3 Методы

2.1.3.1 decrypt()

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

Расшифрование текста.

Применяет обратные смещения, заданные ключом, к зашифрованному тексту.

Аргументы

cipher_text	Зашифрованный текст для расшифрования.
-------------	--

Возвращает

Расшифрованный текст.

Исключения

std::invalid_argument	Если текст не валиден.
-----------------------	------------------------

2.1.3.2 encrypt()

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

Шифрование текста.

Применяет смещения, заданные ключом, к открытому тексту.

Аргументы

open_text	Открытый текст для шифрования.
-----------	--------------------------------

Возвращает

Зашифрованный текст.

Исключения

std::invalid_argument	Если текст не валиден.
-----------------------	------------------------

2.1.3.3 validateKey()

```
void modPermCipher::validateKey (  
    const std::wstring & skey )
```

Валидация ключа.

Проверяет, что ключ состоит только из цифр и является положительным целым числом.

Аргументы

key	Ключ для валидации.
-----	---------------------

Исключения

<code>std::invalid_argument</code>	Если ключ содержит недопустимые символы или является неположительным.
------------------------------------	---

Проверяет, что ключ состоит только из цифр и является положительным целым числом.

Аргументы

<code>skey</code>	Ключ для валидации.
-------------------	---------------------

Исключения

<code>std::invalid_argument</code>	Если ключ содержит недопустимые символы или является неположительным.
------------------------------------	---

2.1.3.4 validateText()

```
void modPermCipher::validateText (
    const std::wstring & text )
```

Валидация текста перед шифрованием или расшифрованием.

Проверяет, что текст не пустой и содержит только допустимые символы.

Аргументы

<code>text</code>	Текст для валидации.
-------------------	----------------------

Исключения

<code>std::invalid_argument</code>	Если текст пустой или содержит недопустимые символы.
------------------------------------	--

Объявления и описания членов классов находятся в файлах:

- `modPerm.h`
- `modPerm.cpp`

Предметный указатель

decrypt
 modPermCipher, [4](#)

encrypt
 modPermCipher, [5](#)

modPermCipher, [3](#)
 decrypt, [4](#)
 encrypt, [5](#)
 modPermCipher, [4](#)
 validateKey, [5](#)
 validateText, [6](#)

validateKey
 modPermCipher, [5](#)

validateText
 modPermCipher, [6](#)