

Шифр табличной маршрутной перестановки.

1.0

Создано системой Doxygen 1.8.17

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 MarshCipher	7
4.1.1 Подробное описание	7
4.1.2 Конструктор(ы)	8
4.1.2.1 MarshCipher()	8
4.1.3 Методы	8
4.1.3.1 Decrypt()	8
4.1.3.2 Encrypt()	9
4.1.3.3 SpaceNum()	10
4.1.3.4 StrNum()	10
4.1.4 Данные класса	10
4.1.4.1 key	10
4.2 Класс MyException	11
4.2.1 Подробное описание	12
4.2.2 Конструктор(ы)	12
4.2.2.1 MyException()	12
4.2.3 Методы	12
4.2.3.1 CheckAlpha()	12
4.2.3.2 CheckKey()	13
4.2.3.3 CheckText()	13
5 Файлы	15
5.1 Файл Exception.h	15
5.1.1 Подробное описание	15
5.2 Файл MyClass.h	16
5.2.1 Подробное описание	17
Предметный указатель	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

exception	
MyException	11
MarshCipher	7

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

MarshCipher

Класс зашифрования и расшифрования текста методом табличной маршрутной перестановки 7

MyException

Собственный класс исключений 11

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Exception.h	
Описание класса MyException	15
MyClass.h	
Описание класса MarshCipher	16

Глава 4

Классы

4.1 Класс MarshCipher

Класс зашифрования и расшифрования текста методом табличной маршрутной перестановки.

```
#include <MyClass.h>
```

Открытые члены

- `MarshCipher` (const wstring &`key`)
Конструктор ключа строкой.
- wstring `Encrypt` (const wstring &text)
Функция зашифрования методом табличной маршрутной перестановки.
- wstring `Decrypt` (const wstring &text)
Функция расшифрования методом табличной маршрутной перестановки.

Закрытые члены

- int `StrNum` (const wstring &text)
Метод, вычисляющий число строк таблицы.
- int `SpaceNum` (const wstring &text)
Метод, вычисляющий число пробелов в таблице.

Закрытые данные

- int `key`
Ключ шифрования методом табличной маршрутной перестановки.

4.1.1 Подробное описание

Класс зашифрования и расшифрования текста методом табличной маршрутной перестановки.

Предупреждения

Реализация только для русского и английского языка.

4.1.2 Конструктор(ы)

4.1.2.1 MarshCipher()

```
MarshCipher::MarshCipher (
    const wstring & key )
```

Конструктор ключа строкой.

Аргументы

key	- ключ, вводимый в виде строки C++.
-----	-------------------------------------

Полученный ключ в виде строки проверяется методом CheckKey на корректность ввода. Далее с помощью функции stoi строка переводится в целое число и присваивается полю key.

4.1.3 Методы

4.1.3.1 Decrypt()

```
wstring MarshCipher::Decrypt (
    const wstring & text )
```

Функция расшифрования методом табличной маршрутной перестановки.

Аргументы

text	- текст, введенный пользователем для выполнения операции расшифрования.
------	---

Первоначально осуществляется проверка текста на корректность ввода. Формируется двумерный массив (таблица) с помощью методов StrNum и SpaceNum. Далее текст записывается в массив по маршруту считывания: справа-налево, сверху-вниз. Spacenum - счетчик пробелов в цикле. Пока он не равен 0 в нижнем элементе столбца будет ставиться пробел.

```
int l=0;
for (int k=key-1; k>-1; k--) {
    if (spacenum == 0) {
        for (int i=0; i<str; i++,l++) {
            arr[i][k]=text[l];
        }
    } else {
        for (int i=0; i<str-1; i++,l++) {
            arr[i][k]=text[l];
            arr[str-1][k] = '\\0';
        }
        spacenum--;
    }
}
```

Далее из заполненного массива текст выписывается в строку по маршруту записи: слева-направо, сверху-вниз. Это и есть расшифрованная строка.

```

int t=0;
for (int i=0; i<str; i++) {
    for (int k=0; k<key; k++,t++) {
        if (arr[i][k]!='\0') {
            newtext[t] = arr[i][k];
        } else {
            break;
        }
    }
}
}

```

Возвращает

Рашифрованная методом табличной маршрутной перестановки строка.

4.1.3.2 Encrypt()

```

wstring MarshCipher::Encrypt (
    const wstring & text )

```

Функция зашифрования методом табличной маршрутной перестановки.

Аргументы

text	- текст, введенный пользователем для выполнения операции зашифрования.
------	--

Первоначально осуществляется проверка текста на корректность ввода. Формируется двумерный массив (таблица) с помощью метода StrNum. Далее текст записывается в массив по маршруту записи: слева-направо, сверху-вниз.

```

int t=0;
int size = text.size();
for (int i=0; i<str; i++) {
    for (int k=0; k<key; k++,t++) {
        if (i*key+k<size+1) {
            arr[i][k] = text [t];
        } else {
            arr[i][k] = '\0';
        }
    }
}
}

```

Далее из заполненного массива текст выписывается в строку по маршруту считывания: справа-налево, сверху-вниз. Это и есть зашифрованная строка.

```

int l=0;
for (int k=key-1; k>=0; k--) {
    for (int i=0; i<str; i++,l++) {
        if (arr[i][k]!='\0') {
            newtext[l]=arr[i][k];
        } else {
            break;
        }
    }
}
}

```

Возвращает

Зашифрованная методом табличной маршрутной перестановки строка.

4.1.3.3 SpaceNum()

```
int MarshCipher::SpaceNum (
    const wstring & text ) [private]
```

Метод, вычисляющий число пробелов в таблице.

Аргументы

text	- текст, введенный пользователем для выполнения операций зашифрования/расшифрования.
------	--

Это приватный метод, включенный в метод Decrypt.

Возвращает

Целое число пробелов.

4.1.3.4 StrNum()

```
int MarshCipher::StrNum (
    const wstring & text ) [private]
```

Метод, вычисляющий число строк таблицы.

Аргументы

text	- текст, введенный пользователем для выполнения операций зашифрования/расшифрования.
------	--

Это приватный метод, включенный в методы Encrypt и Decrypt.

Возвращает

Целое число строк.

4.1.4 Данные класса

4.1.4.1 key

```
int MarshCipher::key [private]
```

Ключ шифрования методом табличной маршрутной перестановки.

Ключ определяет количество столбцов таблицы, по которой будет производиться работа с текстом.

Объявления и описания членов классов находятся в файлах:

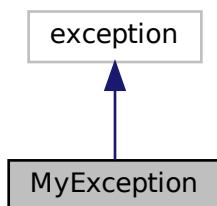
- [MyClass.h](#)
- [description.cpp](#)

4.2 Класс MyException

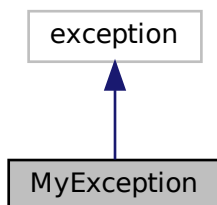
Собственный класс исключений

```
#include <Exception.h>
```

Граф наследования: MyException:



Граф связей класса MyException:



Открытые члены

- `MyException` (`const wstring &err, const wstring &desc`)
Конструктор инициализации ошибки строками.
- `wstring what () noexcept`
Метод, возвращающий информацию об ошибке.

Открытые статические члены

- `static bool CheckKey` (`const wstring &key`)
Метод проверки ключа.
- `static bool CheckText` (`const wstring &text, const int &key`)
Метод проверки текста.

Закрытые статические члены

- static bool [CheckAlpha](#) (const wstring &text, const wstring &EnAlpha, const wstring &RusAlpha)
Функция определения алфавита текста.

Закрытые данные

- wstring [error](#)
поле ошибки, в котором указывается ее вид.
- wstring [description](#)
поле ошибки, в котором указывается ее описание.

4.2.1 Подробное описание

Собственный класс исключений

4.2.2 Конструктор(ы)

4.2.2.1 MyException()

```
MyException::MyException (
    const wstring & err,
    const wstring & desc )
```

Конструктор инициализации ошибки строками.

Аргументы

err	- вид ошибки.
desc	- описание ошибки.

err присваивается приватному полю error, а desc присваивается приватному полю description.

4.2.3 Методы

4.2.3.1 CheckAlpha()

```
bool MyException::CheckAlpha (
    const wstring & text,
    const wstring & EnAlpha,
    const wstring & RusAlpha ) [static], [private]
```

Функция определения алфавита текста.

Аргументы

text	- текст, введенный пользователем для выполнения операций зашифрования/расшифрования.
EnAlpha	- английский алфавит в виде строки.
RusAlpha	- русский алфавит в виде строки.

Функция проверяет только первый элемент текста и выбрасывает исключение если он не является символом ни русского, ни английского алфавита. Это приватный метод, использующийся в методе CheckText.

Возвращает

1 - если, первый символ текста принадлежит к английскому алфавиту. 0 - если к русскому.

4.2.3.2 CheckKey()

```
bool MyException::CheckKey (  
    const wstring & key ) [static]
```

Метод проверки ключа.

Аргументы

key	- ключ, введенный пользователем.
-----	----------------------------------

Бросает исключение в случае если: ключ пустой, ключ содержит символы, отличные от цифр, ключ равен 0.

4.2.3.3 CheckText()

```
bool MyException::CheckText (  
    const wstring & text,  
    const int & key ) [static]
```

Метод проверки текста.

Аргументы

text	- текст, введенный пользователем для выполнения операций зашифрования/расшифрования.
key	- ключ, обработанный программой.

Бросает исключение в случае если: текст пустой, текст содержит символы, не принадлежащие к русскому или английскому алфавиту, текст содержит одновременно символы русского и английского алфавитов, длина текста меньше величины ключа.

Объявления и описания членов классов находятся в файлах:

- [Exception.h](#)
- ExceptionDescr.cpp

Глава 5

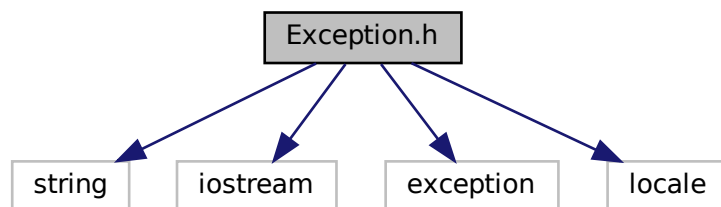
Файлы

5.1 Файл Exception.h

Описание класса [MyException](#).

```
#include <string>
#include <iostream>
#include <exception>
#include <locale>
```

Граф включаемых заголовочных файлов для Exception.h:



Классы

- class [MyException](#)

Собственный класс исключений

5.1.1 Подробное описание

Описание класса [MyException](#).

Автор

Самборский И.С.

Версия

1.0

Дата

21.05.2021

Авторство

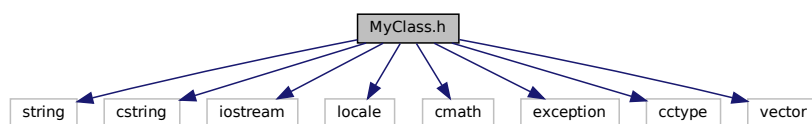
ИБСТ ПГУ

5.2 Файл MyClass.h

Описание класса [MarshCipher](#).

```
#include <string>
#include <cstring>
#include <iostream>
#include <locale>
#include <cmath>
#include <exception>
#include <cctype>
#include <vector>
```

Граф включаемых заголовочных файлов для MyClass.h:



Классы

- class [MarshCipher](#)

Класс зашифрования и расшифрования текста методом табличной маршрутной перестановки.

5.2.1 Подробное описание

Описание класса [MarshCipher](#).

Автор

Самборский И.С.

Версия

1.0

Дата

21.05.2021

Авторство

ИБСТ ПГУ

Предметный указатель

- CheckAlpha
 - MyException, [12](#)
- CheckKey
 - MyException, [13](#)
- CheckText
 - MyException, [13](#)
- Decryct
 - MarshCipher, [8](#)
- Encryct
 - MarshCipher, [9](#)
- Exception.h, [15](#)
- key
 - MarshCipher, [10](#)
- MarshCipher, [7](#)
 - Decryct, [8](#)
 - Encryct, [9](#)
 - key, [10](#)
 - MarshCipher, [8](#)
 - SpaceNum, [9](#)
 - StrNum, [10](#)
- MyClass.h, [16](#)
- MyException, [11](#)
 - CheckAlpha, [12](#)
 - CheckKey, [13](#)
 - CheckText, [13](#)
 - MyException, [12](#)
- SpaceNum
 - MarshCipher, [9](#)
- StrNum
 - MarshCipher, [10](#)