

My Project

Создано системой Doxygen 1.9.4

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 Ciper	7
4.1.1 Конструктор(ы)	7
4.1.1.1 Ciper()	8
4.1.2 Методы	8
4.1.2.1 check()	8
4.1.2.2 rasshifr()	8
4.1.2.3 shifr()	8
4.1.3 Данные класса	8
4.1.3.1 ish	9
4.1.3.2 key	9
4.1.3.3 matrix	9
4.1.3.4 result	9
4.1.3.5 stroki	9
4.2 Класс size_error	10
5 Файлы	11
5.1 header.h	11
Предметный указатель	13

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Ciper	7
std::logic_error	
size_error	10

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

<code>Ciper</code>	7
<code>size_error</code>	10

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

/home/stud/Desktop/дз/Laba_timp/2proga/[header.h](#) 11

Глава 4

Классы

4.1 Класс Ciper

Открытые члены

- `Ciper` (`const wstring &text, const int keys`)
Шифрование методом маршрутной перестановки
- `wstring shifr` ()
функция шифрования
- `wstring rasshifr` ()
функция дешифрования
- `wstring check` ()
функция проверки

Закрытые данные

- `vector< vector< wchar_t > > matrix`
матрица для заполнения
- `int key`
Ключ (количество столбцов матрицы)
- `int stroki`
количество строк матрицы
- `wstring result`
- `wstring ish`
Исходный текст

4.1.1 Конструктор(ы)

4.1.1.1 Ciper()

```
Ciper::Ciper (
    const wstring & text,
    const int keys )
```

Шифрование методом маршрутной перестановки

заполняется пустая матрица, которая добавляет к тексту точки. Для зашифровывания и расшифровывания предназначены методы `shifr` и `rasshifr`.

Предупреждения

Реализация для русского языка

4.1.2 Методы

4.1.2.1 check()

```
wstring Ciper::check ( )
```

функция проверки

создается 3 алфавита: 1 - русский, 2 - латинский, 3 - цифровой. Переданный текст и ключ проверяются на содержание в них запрещённых символов (латинский, цифра).

4.1.2.2 rasshifr()

```
wstring Ciper::rasshifr ( )
```

функция дешифрования

матрица заполняется справа налево и сверху вниз. Пропуск дополняется точками.

4.1.2.3 shifr()

```
wstring Ciper::shifr ( )
```

функция шифрования

создается матрица необходимого размера и заполняется справа налево. Пропуск дополняется точками.

4.1.3 Данные класса

4.1.3.1 ish

```
wstring Ciper::ish [private]
```

Исходный текст

передаётся в качестве аргумента. Текст, который подвергнется шифрованию/дешифрованию

4.1.3.2 key

```
int Ciper::key [private]
```

Ключ (количество столбцов матрицы)

передаётся в качестве аргумента

4.1.3.3 matrix

```
vector<vector<wchar_t>> Ciper::matrix [private]
```

матрица для заполнения

С помощью данного атрибута совершается маршрутная перестановка Размер матрицы определяется длиной введённого текста

4.1.3.4 result

```
wstring Ciper::result [private]
```

результатирующая строка, которая должна быть возвращена после шифрования/дешифрования

возвращается шифратором/дешифратором

4.1.3.5 stroki

```
int Ciper::stroki [private]
```

количество строк матрицы

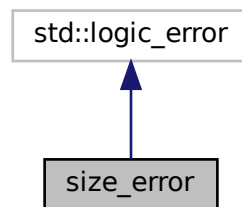
передаётся в качестве аргумента

Объявления и описания членов классов находятся в файлах:

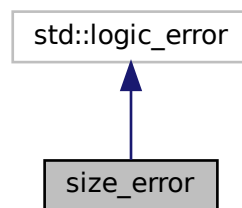
- /home/stud/Desktop/дз/Laba_timp/2proga/header.h
- /home/stud/Desktop/дз/Laba_timp/2proga/modAlphaCipher.cpp

4.2 Класс `size_error`

Граф наследования:`size_error`:



Граф связей класса `size_error`:



Открытые члены

- `size_error (const std::string &message)`

Объявления и описания членов класса находятся в файле:

- `/home/stud/Desktop/дз/Laba_timp/2proga/header.h`

Глава 5

Файлы

5.1 header.h

```
1 #include <stdexcept>
2 #include <iostream>
3 #include <cctype>
4 #include <locale>
5 #include <string>
6 #include <cmath>
7 #include <vector>
8 #pragma once
9 using namespace std;
10
11 class Ciper{
12 private:
13     vector<vector<wchar_t>> matrix;
14     int key;
15     int stroki;
16     wstring result;
17     wstring ish;
18 public:
19     Ciper() = delete;
20     Ciper(const wstring& text, const int keys);
21     wstring shifr();
22     wstring rasshifr();
23     wstring check();
24 };
25
26 class size_error: public std::logic_error {
27 public:
28     explicit size_error(const std::string& message) : std::logic_error(message) {}; //конструктор исключения, после
29     двоеточия происх. ИНИЦИАЛИЗАЦИЯ родительского класса!! сообщением переданным
30 };
```


Предметный указатель

/home/stud/Desktop/дз/Laba_timp/2proga/header.h,
[11](#)

check

Ciper, [8](#)

Ciper, [7](#)

check, [8](#)

Ciper, [7](#)

ish, [8](#)

key, [9](#)

matrix, [9](#)

rasshifr, [8](#)

result, [9](#)

shifr, [8](#)

stroki, [9](#)

ish

Ciper, [8](#)

key

Ciper, [9](#)

matrix

Ciper, [9](#)

rasshifr

Ciper, [8](#)

result

Ciper, [9](#)

shifr

Ciper, [8](#)

size_error, [10](#)

stroki

Ciper, [9](#)