

ShifrPerestan by PT1

1.3

Создано системой Doxygen 1.8.13

Оглавление

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 <code>cipher_error</code>	7
4.1.1	Подробное описание	8
4.2	Класс <code>Cs</code>	8
4.2.1	Подробное описание	8
4.2.2	Конструктор(ы)	9
4.2.2.1	<code>Cs()</code> [1/2]	9
4.2.2.2	<code>Cs()</code> [2/2]	9
4.2.3	Методы	9
4.2.3.1	<code>Coder()</code>	10
4.2.3.2	<code>Decoder()</code>	10
4.2.3.3	<code>getValidCipherText()</code>	11
4.2.3.4	<code>getValidKey()</code>	11
4.2.3.5	<code>getValidOpenText()</code>	12
5	Файлы	13
5.1	Файл <code>CesarDop.cpp</code>	13
5.1.1	Подробное описание	13
5.2	Файл <code>CesarDop.h</code>	14
5.2.1	Подробное описание	15
5.3	Файл <code>main.cpp</code>	15
5.3.1	Подробное описание	16
5.3.2	Функции	16
5.3.2.1	<code>check()</code>	16
5.3.2.2	<code>main()</code>	17
	Алфавитный указатель	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Cs	8
invalid_argument	
cipher_error	7

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Класс для вызова исключений	7
Cs	Класс выполняющий зашифровку и расшифровку методом маршрутной перестановки	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

CesarDop.cpp	
Файл с описаниями методов для модуля Cs	13
CesarDop.h	
Файл с описаниями методов для класса Cs	14
main.cpp	
Файл с проверками методов для модуля Cs(main.cpp)	15

Глава 4

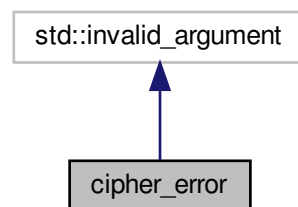
Классы

4.1 Класс `cipher_error`

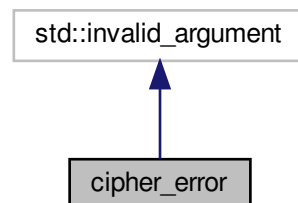
Класс для вызова исключений

```
#include <CesarDop.h>
```

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- `cipher_error (const std::string &what_arg)`
- `cipher_error (const char *what_arg)`

4.1.1 Подробное описание

Класс для вызова исключений

Возвращает

исключение с аргументом

Объявления и описания членов класса находятся в файле:

- [CesarDop.h](#)

4.2 Класс Cs

Класс выполняющий зашифровку и расшифровку методом маршрутной перестановки

```
#include <CesarDop.h>
```

Открытые члены

- [Cs \(int k\)](#)
Конструктор с параметром
- [Cs \(\)=delete](#)
Конструктор без параметра
- `wstring Coder (Cs w, wstring &s)`
Метод зашифровки
- `wstring Decoder (int w, wstring &s)`
Метод расшифровки

Закрытые члены

- `wstring getValidOpenText (const std::wstring &s)`
Метод вызывающий исключение при не верном введенном для шифрования тексте
- `wstring getValidCipherText (const std::wstring &s)`
Метод вызывающий исключение при не верном зашифрованном тексте
- `int getValidKey (const int k, const std::wstring &s)`
Метод вызывающий исключение при не верном ключе

Закрытые данные

- `int k`
Ключ для использования методов

4.2.1 Подробное описание

Класс выполняющий зашифровку и расшифровку методом маршрутной перестановки

Аргументы

in	л	Ключ для использования методов класса. должен быть цифрой, меньшей чем половина (рас)шифруемого текста
----	---	--

Возвращает

(за)расшифрованный текст

Исключения

<code>cipher_error</code> ,если	входной или выходной текст не правильного типа
---------------------------------	--

4.2.2 Конструктор(ы)

4.2.2.1 Cs() [1/2]

Cs::Cs (
 int k)

Конструктор с параметром

Аргументы

in	k	Ключ для шифрования или раасшифрования. Не должен быть строкой (число).
----	---	---

Возвращает

создание элемента класса с ключом

4.2.2.2 Cs() [2/2]

Cs::Cs () [delete]

Конструктор без параметра

Возвращает

Удаление элемента класса

4.2.3 Методы

4.2.3.1 `Coder()`

```
wstring Cs::Coder (
    Cs w,
    wstring & s )
```

Метод зашифровки

Аргументы

in	s	Текст для шифрования. Не должен быть пустой строкой.
in	w	Ключ для шифрования. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Возвращает

зашифрованная строка

Исключения

<code>cipher_error</code> , если	ключ или текст для шифровки не подходит для использования метода
----------------------------------	--

4.2.3.2 `Decoder()`

```
wstring Cs::Decoder (
    int w,
    wstring & s )
```

Метод расшифровки

Аргументы

in	s	Текст расшифрования. Не должен быть пустой строкой.
in	w	Ключ расшифрования. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются

Возвращает

расшифрованная строка

Исключения

<code>cipher_error</code> , если	ключ или текст для расшифровки не подходит для использования метода.
----------------------------------	--

4.2.3.3 `getValidCipherText()`

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

Метод вызывающий исключение при не верном зашифрованном тексте

Аргументы

in	s	Текст для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	---

Возвращает

зашифрованный текст

Исключения

cipher_error ,если	текст для расшифровки не правильного типа
------------------------------------	---

4.2.3.4 `getValidKey()`

```
int Cs::getValidKey (
    const int k,
    const std::wstring & s ) [inline], [private]
```

Метод вызывающий исключение при не верном ключе

Аргументы

in	s	Текст для шифрования или расшифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	---

Возвращает

правильный ключ

Исключения

cipher_error ,если	ключ для расшифровки или шифровки не подходит для использования метода
------------------------------------	--

4.2.3.5 `getValidOpenText()`

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

Метод вызывающий исключение при не верном введенном для шифрования тексте

Аргументы

in	s	Ключ для шифрования. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	---	--

Возвращает

правильный текст

Исключения

cipher_error ,если	текст не верный
------------------------------------	-----------------

Объявления и описания членов классов находятся в файлах:

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

Глава 5

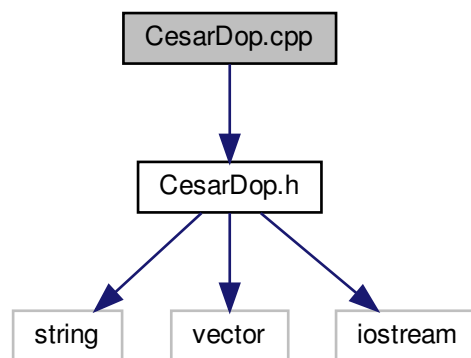
Файлы

5.1 Файл CesarDop.cpp

Файл с описаниями методов для модуля [Cs](#).

```
#include "CesarDop.h"
```

Граф включаемых заголовочных файлов для CesarDop.cpp:



5.1.1 Подробное описание

Файл с описаниями методов для модуля [Cs](#).

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

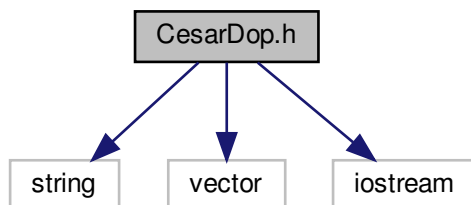
Это учебный пример

5.2 Файл CesarDop.h

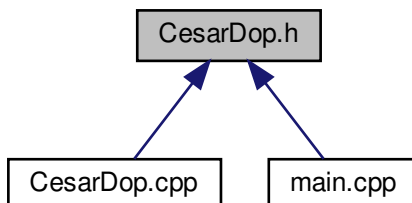
Файл с описаниями методов для класса [Cs](#).

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

Граф включаемых заголовочных файлов для CesarDop.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Cs](#)
Класс выполняющий зашифровку и расшифровку методом маршрутной перестановки
- class [cipher_error](#)
Класс для вызова исключений

5.2.1 Подробное описание

Файл с описаниями методов для класса [Cs](#).

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

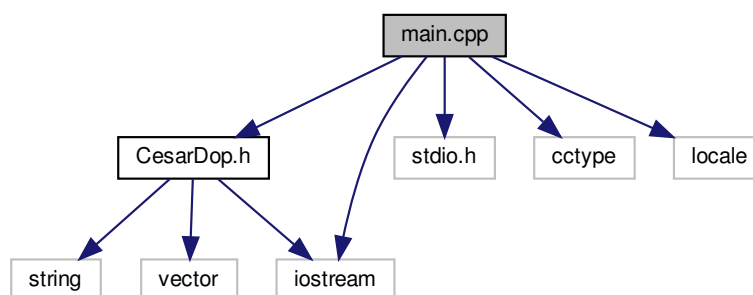
Это учебный пример

5.3 Файл main.cpp

Файл с проверками методов для модуля [Cs](#)([main.cpp](#))

```
#include "CesarDop.h"  
#include <stdio.h>  
#include <iostream>  
#include <cctype>  
#include <locale>
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- void `check` (const wstring &Text, const int &key)
Функция для проверки с помощью UnitTest++.
- int `main` ()
Главная функция запуска метода Гринфильда

5.3.1 Подробное описание

Файл с проверками методов для модуля `Cs(main.cpp)`

Автор

Елдин И.М.

Версия

1.3

Дата

02.06.2021

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.3.2 Функции

5.3.2.1 `check()`

```
void check (  
    const wstring & Text,  
    const int & key )
```

Функция для проверки с помощью UnitTest++.

Аргументы

in	Text	текст для использования методов шифра.
in	key	ключ для использования методов шифра.

Возвращает

Зашифрованную и расшифрованную строку

Исключения

<code>cipher_error</code> ,если	в используемых методах был не верный формат параметров
---------------------------------	--

5.3.2.2 main()

int main ()

Главная функция запуска метода Гринфильда

Возвращает

Количество пройденных программой тестирований

Предметный указатель

CesarDop.cpp, [13](#)

CesarDop.h, [14](#)

check

 main.cpp, [16](#)

cipher_error, [7](#)

Coder

 Cs, [9](#)

Cs, [8](#)

 Coder, [9](#)

 Cs, [9](#)

 Decoder, [10](#)

 getValidCipherText, [10](#)

 getValidKey, [11](#)

 getValidOpenText, [11](#)

Decoder

 Cs, [10](#)

getValidCipherText

 Cs, [10](#)

getValidKey

 Cs, [11](#)

getValidOpenText

 Cs, [11](#)

main

 main.cpp, [17](#)

main.cpp, [15](#)

 check, [16](#)

 main, [17](#)