

Crypt

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 Crypt	7
4.1.1 Подробное описание	7
4.1.2 Конструктор(ы)	8
4.1.2.1 Crypt()	8
4.1.3 Методы	8
4.1.3.1 decrypt()	8
4.1.3.2 encrypt()	8
4.1.3.3 getValidCipherText()	9
4.1.3.4 getValidKey()	9
4.1.3.5 getValidOpenText()	10
4.2 Класс CryptException	10
4.2.1 Подробное описание	11
4.2.2 Конструктор(ы)	11
4.2.2.1 CryptException() [1/2]	12
4.2.2.2 CryptException() [2/2]	12
5 Файлы	13
5.1 Файл crypt.h	13
5.1.1 Подробное описание	13
5.2 crypt.h	14
Предметный указатель	15

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Crypt	7
std::invalid_argument	
CryptException	10

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Crypt	Шифрование методом табличной маршрутной перестановки	7
CryptException	Исключение для ошибок в классе Crypt	10

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

crypt.h	Описание класса Crypt	13
-------------------------	---	----

Глава 4

Классы

4.1 Класс Crypt

Шифрование методом табличной маршрутной перестановки

```
#include <crypt.h>
```

Открытые члены

- `Crypt ()=delete`
Запрет на использование конструктора по умолчанию
- `Crypt (const std::string &keyStr)`
Конструктор класса `Crypt`.
- `std::wstring encrypt (const std::wstring &text)`
Зашифрование текста
- `std::wstring decrypt (const std::wstring &encrypted_text)`
Расшифрование текста
- `int getValidKey (const int &key, const std::wstring &text)`
Валидация ключа
- `std::wstring getValidOpenText (const std::wstring &s)`
Валидация открытого текста
- `std::wstring getValidCipherText (const std::wstring &s)`
Валидация зашифрованного текста

4.1.1 Подробное описание

Шифрование методом табличной маршрутной перестановки

Класс `Crypt` реализует шифрование и расшифрование текста с использованием метода табличной маршрутной перестановки. Ключ задается в конструкторе, а методы `encrypt` и `decrypt` используются для шифрования и расшифрования текста соответственно.

Предупреждения

Реализация предназначена только для русского языка.

4.1.2 Конструктор(ы)

4.1.2.1 Crypt()

```
Crypt::Crypt (  
    const std::string & keyStr )
```

Конструктор класса [Crypt](#).

Аргументы

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

Исключения

CryptException	Если ключ недействителен.
--------------------------------	---------------------------

4.1.3 Методы

4.1.3.1 decrypt()

```
std::wstring Crypt::decrypt (  
    const std::wstring & encrypted_text )
```

Расшифрование текста

Метод расшифровывает переданный зашифрованный текст с использованием установленного ключа.

Аргументы

encrypted_text	Зашифрованный текст для расшифрования.
----------------	--

Возвращает

Открытый текст в виде строки.

4.1.3.2 encrypt()

```
std::wstring Crypt::encrypt (  
    const std::wstring & text )
```

Зашифрование текста

Метод шифрует переданный текст с использованием установленного ключа.

Аргументы

text	Открытый текст для шифрования.
------	--------------------------------

Возвращает

Зашифрованный текст в виде строки.

4.1.3.3 getValidCipherText()

```
std::wstring Crypt::getValidCipherText (  
    const std::wstring & s ) [inline]
```

Валидация зашифрованного текста

Проверяет, что зашифрованный текст содержит только допустимые символы.

Аргументы

s	Зашифрованный текст для проверки.
---	-----------------------------------

Возвращает

Проверенный зашифрованный текст.

Исключения

CryptException	Если зашифрованный текст содержит недопустимые символы.
--------------------------------	---

4.1.3.4 getValidKey()

```
int Crypt::getValidKey (  
    const int & key,  
    const std::wstring & text ) [inline]
```

Валидация ключа

Проверяет, является ли ключ допустимым для шифрования текста.

Аргументы

key	Ключ для проверки.
text	Открытый текст, для которого проверяется ключ.

Возвращает

Допустимый ключ.

Исключения

CryptException	Если ключ недействителен.
--------------------------------	---------------------------

4.1.3.5 getValidOpenText()

```
std::wstring Crypt::getValidOpenText (
    const std::wstring & s ) [inline]
```

Валидация открытого текста

Проверяет, что открытый текст содержит только допустимые символы.

Аргументы

s	Открытый текст для проверки.
---	------------------------------

Возвращает

Проверенный открытый текст.

Исключения

CryptException	Если открытый текст содержит недопустимые символы.
--------------------------------	--

Объявления и описания членов классов находятся в файлах:

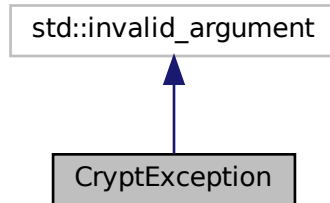
- [crypt.h](#)
- [crypt.cpp](#)

4.2 Класс CryptException

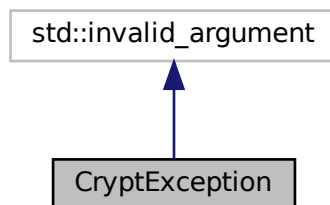
Исключение для ошибок в классе [Crypt](#).

```
#include <crypt.h>
```

Граф наследования: CryptException:



Граф связей класса CryptException:



Открытые члены

- [CryptException](#) (const std::string &what_arg)
Конструктор с аргументом типа std::string.
- [CryptException](#) (const char *what_arg)
Конструктор с аргументом типа const char*.

4.2.1 Подробное описание

Исключение для ошибок в классе [Crypt](#).

Класс исключений, используемый для обработки ошибок, связанных с шифрованием и расшифрованием.

4.2.2 Конструктор(ы)

4.2.2.1 CryptException() [1/2]

```
CryptException::CryptException (
    const std::string & what_arg )    [inline], [explicit]
```

Конструктор с аргументом типа `std::string`.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

4.2.2.2 CryptException() [2/2]

```
CryptException::CryptException (
    const char * what_arg )    [inline], [explicit]
```

Конструктор с аргументом типа `const char*`.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

Объявления и описания членов класса находятся в файле:

- [crypt.h](#)

Глава 5

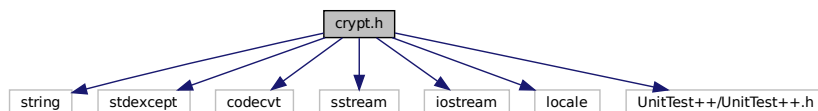
Файлы

5.1 Файл crypt.h

Описание класса [Crypt](#).

```
#include <string>
#include <stdexcept>
#include <codecvt>
#include <sstream>
#include <iostream>
#include <locale>
#include <UnitTest++/UnitTest++.h>
```

Граф включаемых заголовочных файлов для crypt.h:



Классы

- class [Crypt](#)
Шифрование методом табличной маршрутной перестановки
- class [CryptException](#)
Исключение для ошибок в классе [Crypt](#).

5.1.1 Подробное описание

Описание класса [Crypt](#).

Автор

Федякина Т.Н.

Версия

1.0

Дата

21.11.2024

5.2 crypt.h

[См. документацию.](#)

```
1
9 #pragma once
10 #include <string>
11 #include <stdexcept>
12 #include <codecvt>
13 #include <sstream>
14 #include <iostream>
15 #include <locale>
16 #include <UnitTest++/UnitTest++.h>
17
28 class Crypt {
29 private:
30     int key;
31
41     bool isInteger(const std::string& str);
42
43 public:
44     Crypt() = delete;
45
52     Crypt(const std::string& keyStr);
53
62     std::wstring encrypt(const std::wstring& text);
63
72     std::wstring decrypt(const std::wstring& encrypted_text);
73
84     int getValidKey(const int& key, const std::wstring& text);
85
95     std::wstring getValidOpenText(const std::wstring& s);
96
106     std::wstring getValidCipherText(const std::wstring& s);
107 };
108
115 class CryptException : public std::invalid_argument {
116 public:
122     explicit CryptException(const std::string& what_arg)
123         : std::invalid_argument(what_arg) {}
124
130     explicit CryptException(const char* what_arg)
131         : std::invalid_argument(what_arg) {}
132 };
```

Предметный указатель

Crypt, [7](#)

 Crypt, [8](#)

 decrypt, [8](#)

 encrypt, [8](#)

 getValidCipherText, [9](#)

 getValidKey, [9](#)

 getValidOpenText, [10](#)

crypt.h, [13](#)

CryptException, [10](#)

 CryptException, [11](#), [12](#)

decrypt

 Crypt, [8](#)

encrypt

 Crypt, [8](#)

getValidCipherText

 Crypt, [9](#)

getValidKey

 Crypt, [9](#)

getValidOpenText

 Crypt, [10](#)