

Создано системой Doxygen 1.8.17

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	1
2.1 Классы	1
3 Список файлов	2
3.1 Файлы	2
4 Классы	2
4.1 Класс modAlphaCipher	2
4.1.1 Подробное описание	3
4.1.2 Конструктор(ы)	3
4.1.3 Методы	3
4.2 Kласс MyExceptions	6
4.2.1 Подробное описание	8
4.2.2 Конструктор(ы)	8
5 Φ айлы	8
5.1 Файл Exception.h	8
5.1.1 Подробное описание	9
5.2 Файл modAlphaCipher.h	10
5.2.1 Подробное описание	10
Предметный указатель	11
1 Иерархический список классов	
1.1 Иерархия классов	
Иерархия классов.	
$invalid_argument$	
$\operatorname{MyExceptions}$	6
$\bmod Alpha Cipher$	2
2 Алфавитный указатель классов	
2.1 Классы	
Классы с их кратким описанием.	
modAlphaCipher Класс, который реализует шифрование сообщений методом "Гронсвельда"	2

MyExceptions Класс для обработки ошибок, которые могут возникнуть при взаимодействии с программой	6
3 Список файлов	
3.1 Файлы	
Полный список документированных файлов.	
Exception.h Описание класса MyException	8
modAlphaCipher.h Описание класса modAlphaCipher	10
4 Классы	
4.1 Класс modAlphaCipher	
Класс, который реализует шифрование сообщений методом "Гронсвельда".	
# include < modAlphaCipher.h >	
Открытые члены	
 modAlphaCipher ()=delete Запрещающий конструктор без параметров modAlphaCipher (wstring skey) Конструктор для ключа wstring encrypt (const wstring &open_text) Метод, предназначенный для шифрования wstring decrypt (const wstring &cipher_text) Метод, предназначенный для расшифрования 	
Закрытые члены	
• vector< int > convert (const wstring &s)	
Праобразоранна строки в вактор(нисал)	

- Преобразование строки в вектор(чисел)
- wstring convert (const vector< int > &v)
 - Преобразование вектора(чисел) в строку
- - Валидация ключа
- wstring getValidAlphabetText (const wstring &s)

Валидация текста при шифровании или расшифровании

Закрытые данные

• wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"

Используемый алфавит для сообщений

• map< char, int > alphaNum

Ассоциативный массив "номер по символу".

• vector< int> key

Атрибут для ключа

4.1.1 Подробное описание

Класс, который реализует шифрование сообщений методом "Гронсвельда".

Предупреждения

Работает только с русскоязычными сообщениями

4.1.2 Конструктор(ы)

```
4.1.2.1 modAlphaCipher() modAlphaCipher::modAlphaCipher ( wstring skey )
```

Конструктор для ключа

Цикл for построен по строке-алфавиту и на каждом шаге добавляет в ассоциативный массив символ и его номер. Ассоциативный массив в данном случае удобен из-за синтаксиса доступа к элементам по индексу-символу

```
for (unsigned i=0; i<numAlpha.size(); i++) {
    alphaNum[numAlpha[i]]=i;
}
```

Аргументы

```
std::wstring | - ключ в виде строки
```

4.1.3 Методы

```
4.1.3.1 convert() [1/2] wstring modAlphaCipher::convert ( const vector< int > \& v) [inline], [private]
```

Преобразование вектора(чисел) в строку

В переменную типа "wstring" с именем "Result" формируется строка по индексам алфавита "num↔ Alpha". Индексы хранятся в векторе типа "int", который пришёл на вход.

```
wstring result;
for(auto i:v) {
    result.push_back(numAlpha[i]);
}
```

Возвращает

строка текста типа "wstring"

```
4.1.3.2 \quad convert() \ [2/2] \quad vector < int > modAlphaCipher::convert \ ( \\ const \ wstring \ \& \ s \ ) \quad [inline], \ [private]
```

Преобразование строки в вектор(чисел)

В вектор типа "int" с именем "Result" формируются числа, которые являются индексами алфавита "numAlpha" из строки, которая пришла на вход.

```
vector<int > result;
for(auto c:s) {
    result.push _back(alphaNum[c]);
}
```

Возвращает

std::vector <int>, в котором хранятся индексы букв сообщения из алфавита "numAlpha"

```
4.1.3.3 decrypt() wstring modAlphaCipher::decrypt(
const wstring & cipher text)
```

Метод, предназначенный для расшифрования

Здесь сначала формируется вектор work из строки щифратекста с помощью метода convert(). А также происходит проверка шифротекста на наличие ошибки при помощи метода getValidAlphabetText(). vector<int> work = convert(getValidAlphabetText(cipher_text));

Если при зашифровывании мы прибавляли значение ключа, то при расшифровывании значения ключа надо вычитать. А чтобы не получить отрицательных значений, выполняется еще прибавление значения модуля, так как такое прибавление не влияет на результат модулю.

```
for(unsigned i=0; i < work.size(); i++) {
      work[i] = (work[i] + alphaNum.size() - key[i % key.size()]) % alphaNum.size();
}</pre>
```

Аргументы

d::wstring cipher_text - сообщение, которое нужно расшифровать
--

Исключения

MyException,если	строка, которая пришла на вход оказывается пустой или в ней есть
	недопустимые символы

Возвращает

строка расшифрованного текста типа "wstring"

```
4.1.3.4 encrypt() wstring modAlphaCipher::encrypt(
const wstring & open text)
```

Метод, предназначенный для шифрования

Здесь сначала формируется вектор work из строки открытого текста с помощью метода convert(). А также происходит проверка текста на наличие ошибки при помощи метода getValidAlphabetText(). vector<int> work = convert(getValidAlphabetText(open text));

Далее, при возврате значения, вектор work опять преобразуется в строку.

Аргументы

st	d::wstring	$open_t$	ext - сообщение,	которое	нужно	зашифровать	
----	------------	----------	------------------	---------	-------	-------------	--

Исключения

MyException,если	строка, которая пришла на вход оказывается пустой или в ней есть	1
	недопустимые символы	

Возвращает

строка шифротекста типа "wstring"

```
4.1.3.5 getValidAlphabetText() wstring modAlphaCipher::getValidAlphabetText( const wstring & s ) [inline], [private]
```

Валидация текста при шифровании или расшифровании

Сначала текст проверяется на пустоту при помощи обычного условия. Если проверка закончилась успешно, то текст проверяется на наличие недопустимых символов.

Предупреждения

Письменные буквы алфавита переводятся в прописные. То есть регистр букв русского алфавита не влияет на появление исключений.

Аргументы

std::wstring	s - строка текста для шифрования или расшифрования, которую нужно
	проверить на наличие ошибок

Исключения

MyExceptions, если текст является пустым или в нём присутствуют недопустимые символы

Возвращает

Текст в виде строки типа "wstring", который успешно прошёл валидацию

4.1.3.6 getValidKey() wstring modAlphaCipher::getValidKey(const wstring & s) [inline], [private]

Валидация ключа

Сначала ключ проверяется на пустоту при помощи обычного условия. Если проверка закончилась успешно, то ключ проверяется на наличие недопустимых символов.

Предупреждения

Письменные буквы алфавита переводятся в прописные. То есть регистр букв русского алфавита не влияет на появление исключений.

Аргументы

std::wstring s - ключ в виде строки, который нужно проверить на наличие ошибок

Исключения

MyExceptions, если ключ является пустым или в нём присутствуют недопустимые символы

Возвращает

Ключ в виде строки типа "wstring", который успешно прошёл валидацию

Объявления и описания членов классов находятся в файлах:

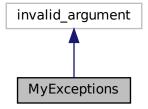
- · modAlphaCipher.h
- $\bullet \ \, mod Alpha Cipher.cpp$

4.2 Класс MyExceptions

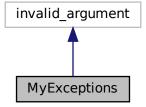
Класс для обработки ошибок, которые могут возникнуть при взаимодействии с программой

#include < Exception.h >

Граф наследования: MyExceptions:



Граф связей класса MyExceptions:



Открытые члены

• MyExceptions ()=delete

Запрещающий конструктор без параметров

• MyExceptions (const string &error, const int &num, const string &fix)

Конструктор с параметрами

• void fix ()

Предназначен для вывода информации об исправлении ошибки

• void code ()

Предназначен для вывода кода ошибки

Закрытые данные

• int num

атрибут, хранящий код ошибки

• string correction

атрибут, хранящий информацию об исправлении ошибки

4.2.1 Подробное описание

Класс для обработки ошибок, которые могут возникнуть при взаимодействии с программой

Класс наследует существующий класс обработки исключений с именем "invalid_argument" из библиотеки "stdexcept".

4.2.2 Конструктор(ы)

```
4.2.2.1 MyExceptions() MyExceptions::MyExceptions (
const string & error,
const int & num,
const string & fix )
```

Конструктор с параметрами

Аргументы

num	- целочисленное число, хранящее информацию о коде ошибки
error	- строка, хранящая описание ошибки. Данный параметр перегружается конструктором класса "invalid_argument"
correction	- строка, хранящая информацию об исправлении ошибки

Объявления и описания членов классов находятся в файлах:

- Exception.h
- \bullet Exception.cpp

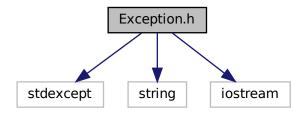
5 Файлы

5.1 Файл Exception.h

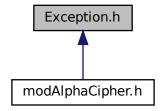
Описание класса MyException.

```
#include <stdexcept>
#include <string>
#include <iostream>
```

Граф включаемых заголовочных файлов для Exception.h:



Граф файлов, в которые включается этот файл:



Классы

• class MyExceptions

Класс для обработки ошибок, которые могут возникнуть при взаимодействии с программой

5.1.1 Подробное описание

Описание класса MyException.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

Авторство

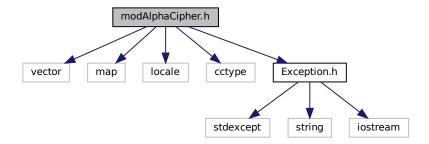
ивст пгу

5.2 Файл modAlphaCipher.h

Описание класса modAlphaCipher.

```
#include <vector>
#include <map>
#include <locale>
#include <cctype>
#include "Exception.h"
```

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

 $\bullet \ class \ modAlphaCipher \\$

Класс, который реализует шифрование сообщений методом "Гронсвельда".

5.2.1 Подробное описание

Описание класса modAlphaCipher.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

Авторство

ИБСТ ПГУ

Предметный указатель

```
convert
     \bmod Alpha Cipher,\, {\color{red} 3},\, {\color{red} 4}
decrypt
     modAlphaCipher, 4
encrypt
     modAlphaCipher, 4
Exception.h, 8
{\it getValidAlphabetText}
     modAlphaCipher, 5
\operatorname{getValidKey}
     modAlphaCipher, 6
\bmod Alpha Cipher,~{\color{red}2}
     convert, 3, 4
     decrypt, 4
     encrypt, \frac{4}{}
     {\tt getValidAlphabetText},\, {\tt 5}
     getValidKey, 6
     modAlphaCipher, 3
modAlphaCipher.h, 10
MyExceptions, 6
     MyExceptions, 8
```