modAlphaCipher

1.0

Создано системой Doxygen 1.9.1

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс modAlphaCipher

Класс, реализует шифрование 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 для шифрования
- wstring decrypt (const wstring &cipher_text)
  Метод для расшифрования

Закрытые члены

- vector< int > convert (const wstring &s)
  Преобразование строки в вектор(чисел)
- wstring convert (const vector< int > &v)
  Преобразование вектора(чисел) в строку
- wstring getValidKey (const wstring &s)
  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, реализует шифрование методом "Гронсвельда".

Автор

Kirill Koltunov 20PT1

Дата

06/01/21

4.1.2 Конструктор(ы)

4.1.2.1 modAlphaCipher()

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

Конструктор для ключа

Цикл for по строке-алфавиту на каждом шаге добавляет в массив символ и его номер. массив в данном случае удобен из-за 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 convert() [2/2]

```
vector< int > modAlphaCipher::convert (
                const wstring & s )   [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 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 прибавляли значение ключа, то при расшифровывании значения ключа надо вычитать. 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();
}
```

Аргументы

std::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 для шифрования

Здесь формируется вектор work из строки открытого текста с помощью метода convert().
```
vector<int> work = convert(getValidAlphabetText(open_text));
```

в цикле к каждому элементу вектора прибавляется элемент ключа по модулю размера алфавита. Так как ключ может быть короче текста, то при индексации ключа выполняется операция по модулю размера ключа.
```
for(unsigned i=0; i < work.size(); i++) {
  work[i] = (work[i] + key[i % key.size()]) % alphaNum.size();
}
```

вектор work опять преобразуется в строку.

Аргументы

std::wstring	open_text - сообщение, которое нужно зашифровать

Исключения

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

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:

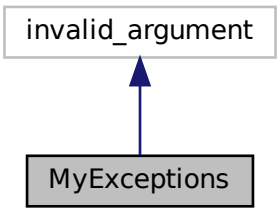
- modAlphaCipher.h
- modAlphaCipher.cpp

4.2 Класс MyExceptions

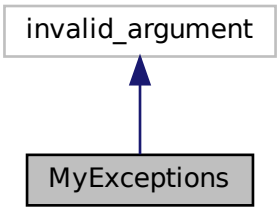
Класс для обработки ошибок

#include <Exceptions.h>

Граф наследования:MyExceptions:



Граф связей класса MyExceptions:



Открытые члены

- MyExceptions ()=delete
  Запрещающий конструктор
- 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

Автор

Kirill Koltunov 20PT1

Дата

06/01/21

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.
correction	- строка, хранящая информацию об исправлении ошибки

Объявления и описания членов классов находятся в файлах:

- Exceptions.h
- Exceptions.cpp

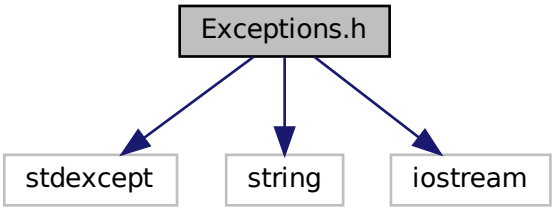
Глава 5

Файлы

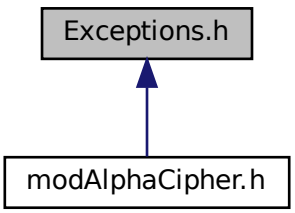
5.1 Файл Exceptions.h

Header file for MyExpections.

#include <stdexcept>
#include <string>
#include <iostream>
Граф включаемых заголовочных файлов для Exceptions.h:



Граф файлов, в которые включается этот файл:



Классы

- class MyExceptions

  Класс для обработки ошибок

5.1.1 Подробное описание

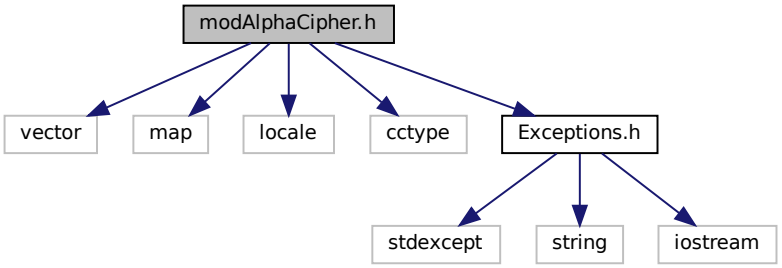
Header file for MyExpections.

5.2 Файл modAlphaCipher.h

Header file for modAlphaCipher.

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Классы

- class modAlphaCipher

  Класс, реализует шифрование методом "Гронсвельда".

5.2.1 Подробное описание

Header file for modAlphaCipher.

Предметный указатель