

My Project  
333333

Створено системою Doxygen 1.8.13



# Зміст

1	Алфавітний покажчик класів	1
1.1	Класи	1
2	Показчик файлів	3
2.1	Файли	3
3	Класи	5
3.1	Клас <code>modAlphaCipher</code>	5
3.1.1	Детальний опис	6
3.1.2	Конструктор(и)	6
3.1.2.1	<code>modAlphaCipher()</code> [1/2]	6
3.1.2.2	<code>modAlphaCipher()</code> [2/2]	6
3.1.3	Опис методів компонент	7
3.1.3.1	<code>convert()</code> [1/2]	7
3.1.3.2	<code>convert()</code> [2/2]	7
3.1.3.3	<code>decrypt()</code>	7
3.1.3.4	<code>encrypt()</code>	8
3.1.3.5	<code>fromwstr()</code>	9
3.1.3.6	<code>towstr()</code>	9
3.1.4	Компонентні дані	10
3.1.4.1	<code>alphaNum</code>	10
3.1.4.2	<code>key</code>	10
3.1.4.3	<code>numAlpha</code>	10

---

4	Файли	11
4.1	Файл <code>main.cpp</code>	11
4.1.1	Детальний опис	12
4.1.2	Опис функцій	12
4.1.2.1	<code>isValid()</code>	12
4.1.2.2	<code>main()</code>	12
4.2	Файл <code>modAlphaCipher.cpp</code>	13
4.2.1	Детальний опис	13
4.3	Файл <code>modAlphaCipher.h</code>	13
4.3.1	Детальний опис	14
	Предметний покажчик	15

## Розділ 1

# Алфавітний покажчик класів

### 1.1 Класи

Класи, структури, об'єднання та інтерфейси з коротким описом.

`modAlphaCipher`

Шифрование методом Гронсфельда . . . . . 5



## Розділ 2

# Показчик файлів

### 2.1 Файли

Повний список файлів.

<a href="#">main.cpp</a>	
Заголовочный файл для модуля <a href="#">main.cpp</a> . . . . .	11
<a href="#">modAlphaCipher.cpp</a>	
Заголовочный файл для модуля <a href="#">modAlphaCipher.cpp</a> . . . . .	13
<a href="#">modAlphaCipher.h</a>	
Заголовочный файл для модуля <a href="#">modAlphaCipher.h</a> . . . . .	13





## Розділ 3

# Класи

### 3.1 Клас modAlphaCipher

Шифрование методом Гронсфельда

```
#include <modAlphaCipher.h>
```

Загальнодоступні елементи

- `modAlphaCipher ()=delete`  
Валидация зашифрованного текста
- `modAlphaCipher (const string &skkey)`  
Конструктор для установки ключа
- `string encrypt (const string &open_text)`  
Зашифровывание
- `string decrypt (const string &cipher_text)`  
Расшифровывание

Приватні елементи

- `vector< int > convert (const wstring &ws)`  
Преобразование строка-вектор
- `string convert (const vector< int > &v)`  
Преобразование вектор-строка
- `wstring towstr (const string &s)`  
Валидация ключа
- `string fromwstr (const wstring &ws)`  
Валидация открытого текста 4.

Приватні дані

- `wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`  
Русский алфавит по порядку
- `map< wchar_t, int > alphaNum`  
Ассоциативный массив "номер по символу".
- `vector< int > key`  
Ключ

### 3.1.1 Детальный опис

#### Шифрование методом Гронсфельда

Ключ устанавливается в конструкторе. Для зашифровывания и расшифровывания предназначены методы `encrypt` и `decrypt`.

#### Застережения

Реализация только для русского языка

### 3.1.2 Конструктор(и)

#### 3.1.2.1 `modAlphaCipher()` [1/2]

`modAlphaCipher::modAlphaCipher ( )` [delete]

#### Валидация зашифрованного текста

##### Аргументы

in	ws	Зашифрованный текст в кодировке UTF-32. Не должен быть пустой строкой. Не должен содержать строчные символы и небуквы.
----	----	--

#### Повертає

Валидный зашифрованный текст в кодировке UTF-32

#### Обробка виняткових ситуацій

<code>cipher_error</code>	текст пустой или содержит недопустимые символы Конструктор без параметров
---------------------------	---

#### Запрещен

#### 3.1.2.2 `modAlphaCipher()` [2/2]

`modAlphaCipher::modAlphaCipher (`  
     `const string & skey )`

#### Конструктор для установки ключа

Ключ проверяется на валидность. Переводится в кодировку UTF-32. Формируется вектор-ключ.

##### Аргументы

in	skey	Ключ в кодировке UTF-8
----	------	------------------------

Обробка виняткових ситуацій

cipher_error	ключ вырожденный
--------------	------------------

### 3.1.3 Опис методів компонент

#### 3.1.3.1 convert() [1/2]

```
vector<int> modAlphaCipher::convert (  
    const wstring & ws ) [private]
```

Преобразование строка-вектор

Аргументы

in	ws	Строка в кодировке UTF-32
----	----	---------------------------

Повертає

Целочисленный вектор

#### 3.1.3.2 convert() [2/2]

```
string modAlphaCipher::convert (  
    const vector< int > & v ) [private]
```

Преобразование вектор-строка

Аргументы

in	v	Целочисленный вектор
----	---	----------------------

Повертає

Строка в кодировке UTF-8

#### 3.1.3.3 decrypt()

```
std::string modAlphaCipher::decrypt (  
    const string & cipher_text )
```

Расшифровывание

## Аргументи

in	cipher_text	Зашифрованный текст в кодировке UTF-8. Не должен быть пустой строкой. Не должен содержать строчные символы и небуквы.
----	-------------	---

## Повертає

Расифрованная строка в кодировке UTF-8

## Обробка виняткових ситуацій

cipher_error	текст пустой
--------------	--------------

/\*\*

## Аргументи

in	cipher_text	Зашифрованный текст в кодировке UTF-8. Не должен быть пустой строкой. Не должен содержать строчные символы и небуквы.
----	-------------	---

## Повертає

Расифрованная строка в кодировке UTF-8

## Обробка виняткових ситуацій

cipher_error	текст пустой
--------------	--------------

## 3.1.3.4 encrypt()

```
std::string modAlphaCipher::encrypt (
    const string & open_text )
```

## Зашифровывание

## Аргументи

in	open_text	Открытый текст в кодировке UTF-8. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются.
----	-----------	--

## Повертає

Зашифрованная строка в кодировке UTF-8

Обробка виняткових ситуацій

cipher_error	текст пустой
--------------	--------------

Аргументи

in	open_text	Открытый текст. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным. Все не-буквы удаляются
----	-----------	---

Повертає

Зашифрованная строка

Обробка виняткових ситуацій

cipher_error,если	текст пустой
-------------------	--------------

### 3.1.3.5 fromwstr()

```
std::string modAlphaCipher::fromwstr (
    const wstring & ws ) [private]
```

Валидация открытого текста 4.

Аргументи

in	ws	Открытый текст в кодировке UTF32. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным.
----	----	--

Повертає

Валидный открытый текст в кодировке UTF-32

Обробка виняткових ситуацій

cipher_error	текст пустой или содержит недопустимые символы
--------------	--

### 3.1.3.6 towstr()

```
std::wstring modAlphaCipher::towstr (
    const string & s ) [private]
```

Валидация ключа

## Аргументи

in	ws	Ключ в кодировке UTF-32. Не должен быть пустой строкой. Строчные символы автоматически преобразуются к прописным.
----	----	---

## Повертає

Валидный ключ в кодировке UTF-32

## Обробка виняткових ситуацій

cipher_error	ключ пустой или содержит недопустимые символы
--------------	---

## 3.1.4 Компонентні дані

## 3.1.4.1 alphaNum

```
map<wchar_t,int> modAlphaCipher::alphaNum [private]
```

Ассоциативный массив "номер по символу".

## 3.1.4.2 key

```
vector<int> modAlphaCipher::key [private]
```

Ключ

## 3.1.4.3 numAlpha

```
wstring modAlphaCipher::numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ" [private]
```

Русский алфавит по порядку

Документація цих класів була створена з файлів:

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

## Розділ 4

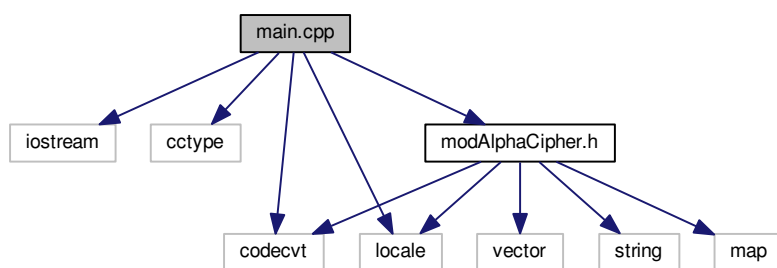
# Файли

### 4.1 Файл main.cpp

Заголовочный файл для модуля `main.cpp`.

```
#include <iostream>
#include <cctype>
#include <codecvt>
#include <locale>
#include "modAlphaCipher.h"
```

Діаграма включених заголовочних файлів для `main.cpp`:



### Функції

- bool `isValid` (const string &s)
- int `main` (int argc, char \*\*argv)

Интерфейс программы

#### 4.1.1 Детальний опис

Заголовочный файл для модуля `main.cpp`.

Автор

Козин А.В.

Версія

1.0.0

Дата

13.06.2019

#### 4.1.2 Опис функцій

##### 4.1.2.1 isValid()

```
bool isValid (
    const string & s )
```

##### 4.1.2.2 main()

```
int main (
    int argc,
    char ** argv )
```

#### Интерфейс программы

Осуществление выбора ключа и операции 0, 1 или 2. В зависимости от выбора выполняются следующие действия: выход, зашифровка, расшифровка. ввод ключа

ввод числа

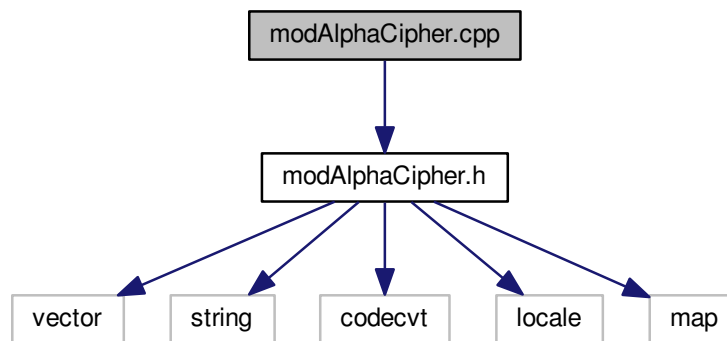


## 4.2 Файл modAlphaCipher.cpp

Заголовочный файл для модуля [modAlphaCipher.cpp](#).

```
#include "modAlphaCipher.h"
```

Диаграмма включений заголовочных файлов для modAlphaCipher.cpp:



### 4.2.1 Детальный опис

Заголовочный файл для модуля [modAlphaCipher.cpp](#).

Автор

Максимов О.В.

Версія

1.0.0

Дата

13.06.2019

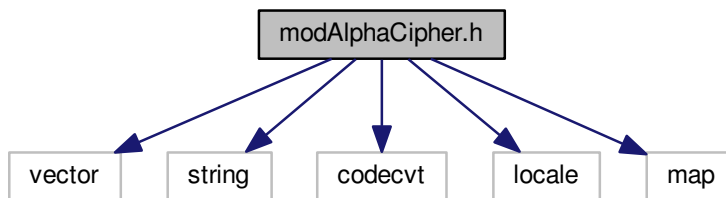
## 4.3 Файл modAlphaCipher.h

Заголовочный файл для модуля [modAlphaCipher.h](#).

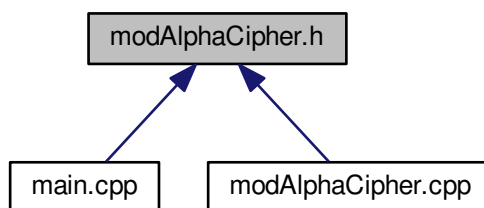
```
#include <vector>
#include <string>
#include <codecvt>
#include <locale>
```

```
#include <map>
```

Діаграма включених заголовочних файлів для `modAlphaCipher.h`:



Граф файлів, які включають цей файл:



## Класи

- class `modAlphaCipher`  
Шифрование методом Гронсфельда

### 4.3.1 Детальний опис

Заголовочный файл для модуля `modAlphaCipher.h`.

Автор

Максимов О.В.

Версія

1.0.0

Дата

13.06.2019

# Покажчик

- alphaNum
  - modAlphaCipher, [10](#)
- convert
  - modAlphaCipher, [7](#)
- decrypt
  - modAlphaCipher, [7](#)
- encrypt
  - modAlphaCipher, [8](#)
- fromwstr
  - modAlphaCipher, [9](#)
- isValid
  - main.cpp, [12](#)
- key
  - modAlphaCipher, [10](#)
- main
  - main.cpp, [12](#)
- main.cpp, [11](#)
  - isValid, [12](#)
  - main, [12](#)
- modAlphaCipher, [5](#)
  - alphaNum, [10](#)
  - convert, [7](#)
  - decrypt, [7](#)
  - encrypt, [8](#)
  - fromwstr, [9](#)
  - key, [10](#)
  - modAlphaCipher, [6](#)
  - numAlpha, [10](#)
  - towstr, [9](#)
- modAlphaCipher.cpp, [13](#)
- modAlphaCipher.h, [13](#)
- numAlpha
  - modAlphaCipher, [10](#)
- towstr
  - modAlphaCipher, [9](#)