

modAlphaCipher

Создано системой Doxygen 1.9.1

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 cipher_error	7
4.1.1 Подробное описание	8
4.2 Класс modAlphaCipher	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 modAlphaCipher()	9
4.2.3 Методы	10
4.2.3.1 convert() [1/2]	10
4.2.3.2 convert() [2/2]	10
4.2.3.3 decrypt()	10
4.2.3.4 encrypt()	11
4.2.3.5 getValidCipherText()	11
4.2.3.6 getValidKey()	12
4.2.3.7 getValidOpenText()	13
4.2.4 Данные класса	13
4.2.4.1 numAlpha	13
5 Файлы	15
5.1 Файл main.cpp	15
5.1.1 Подробное описание	15
5.2 Файл modAlphaCipher.cpp	16
5.2.1 Подробное описание	16
5.3 Файл modAlphaCipher.h	17
5.3.1 Подробное описание	17
Предметный указатель	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
cipher_error	7
modAlphaCipher	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	
Созданное исключение	7
modAlphaCipher	
Шифрование методом Гронсфельда	8

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	
Главный модуль проекта	15
modAlphaCipher.cpp	
Модуль реализации методов класса modAlphaCipher	16
modAlphaCipher.h	
Заголовочный файл проекта	17

Глава 4

Классы

4.1 Класс `cipher_error`

Созданное исключение

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

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- [cipher_error](#) (const char *what_arg)
ошибка, возвращающая символ
- [cipher_error](#) (const std::string &what_arg)
ошибка, возвращающая строку

4.1.1 Подробное описание

Созданное исключение

Создано для отличия программных исключений от исключений, возбужденных модулем.

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Шифрование методом Гронсфелда

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

Открытые члены

- [modAlphaCipher](#) ()=delete
запрет конструктора без параметров
- [modAlphaCipher](#) (const std::string &skey)
Конструктор
- std::string [encrypt](#) (const std::string &open_text)
Зашифровывание
- std::string [decrypt](#) (const std::string &cipher_text)
Расшифровывание

Закрытые члены

- std::vector< int > [convert](#) (const std::string &s)
Преобразование строка-вектор
- std::string [convert](#) (const std::vector< int > &v)
Преобразование вектор-строка
- std::string [getValidKey](#) (const std::string &s)
Проверка и преобразование ключа
- std::string [getValidOpenText](#) (const std::string &s)
Проверка и преобразование нормального текста
- std::string [getValidCipherText](#) (const std::string &s)
Проверка зашифрованного текста

Закрытые данные

- `std::wstring_convert< std::codecvt_utf8< wchar_t >, wchar_t > codec`
переменная для преобразования широких строк в обычные;.
- `std::wstring numAlpha`
алфавит по порядку
- `std::map< wchar_t, int > alphaNum`
ассоциативный массив "номер по символу".
- `std::vector< int > key`
ключ

4.2.1 Подробное описание

Шифрование методом Гронсфельда

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

Предупреждения

Реализация только для русского языка

4.2.2 Конструктор(ы)

4.2.2.1 modAlphaCipher()

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

Конструктор

Аргументы

in	skey	Ключ. Не должен быть пустой строкой и содержать недопустимые символы. Строчные символы автоматически преобразуются к прописным.
----	------	---

Предупреждения

Запрещёнными символами считаются все символы кроме букв русского языка

Исключения

<code>cipher_error</code> , если	ключ пустой или содержит запрещённые символы конструктор для установки ключа
----------------------------------	--

4.2.3 Методы

4.2.3.1 convert() [1/2]

```
std::vector< int > modAlphaCipher::convert (
    const std::string & s )    [inline], [private]
```

Преобразование строка-вектор

Аргументы

in	s	Строка из символов
----	---	--------------------

Возвращает

Вектор числовых значений, соответствующих символам преобразование строка-вектор

4.2.3.2 convert() [2/2]

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

Преобразование вектор-строка

Аргументы

in	v	Вектор числовых значений
----	---	--------------------------

Возвращает

Строка из символов, соответствующих числовым значениям преобразование вектор-строка

4.2.3.3 decrypt()

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

Расшифровывание

Аргументы

in	cipher_text	Зашифрованный текст. Не должен быть пустой строкой и содержать недопустимые символы.
----	-------------	--

Предупреждения

Запрещёнными символами считаются все символы кроме букв русского языка

Возвращает

Расшифрованная строка

Исключения

cipher_error ,если	текст пустой или содержит запрещённые символы функция расшифрования
------------------------------------	---

4.2.3.4 encrypt()

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

Зашифровывание

Аргументы

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

Возвращает

Зашифрованная строка

Исключения

cipher_error ,если	текст пустой функция зашифрования
------------------------------------	-----------------------------------

4.2.3.5 getValidCipherText()

```
std::string modAlphaCipher::getValidCipherText (
    const std::string & s ) [inline], [private]
```

Проверка зашифрованного текста

Текст проверяется на пустоту и наличие запрещённых символов.

Предупреждения

Запрещёнными символами считаются все символы кроме букв русского языка

Аргументы

in	s	Строка с введённым текстом
----	---	----------------------------

Возвращает

Строка с проверенным текстом

Исключения

cipher_error , если	текст пустой или содержит запрещённые символы проверка зашифрованного текста
-------------------------------------	--

4.2.3.6 getValidKey()

```
std::string modAlphaCipher::getValidKey (
    const std::string & s ) [inline], [private]
```

Проверка и преобразование ключа

Ключ проверяется на наличие запрещённых символов и пустоту и преобразуется. Строчные буквы преобразуются в заглавные

Предупреждения

Запрещёнными символами считаются все символы кроме букв русского языка

Аргументы

in	s	Строка с введённым ключом
----	---	---------------------------

Возвращает

Строка с преобразованным ключом

Исключения

cipher_error , если	ключ пустой, слабый или имеет недопустимые символы проверка и преобразование ключа
-------------------------------------	--

4.2.3.7 getValidOpenText()

```
std::string modAlphaCipher::getValidOpenText (
    const std::string & s ) [inline], [private]
```

Проверка и преобразование нормального текста

Текст проверяется на пустоту и преобразуется. Строчные буквы преобразуются в заглавные, запрещённые символы удаляются из текста

Предупреждения

Запрещёнными символами считаются все символы кроме букв русского языка

Аргументы

in	s	Строка с введённым текстом
----	---	----------------------------

Возвращает

Строка с преобразованным текстом

Исключения

cipher_error ,если	текст пустой проверка и преобразование нормального текста
------------------------------------	---

4.2.4 Данные класса

4.2.4.1 numAlpha

```
std::wstring modAlphaCipher::numAlpha [private]
```

Инициализатор

```
= L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
```

алфавит по порядку

Объявления и описания членов классов находятся в файлах:

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

Глава 5

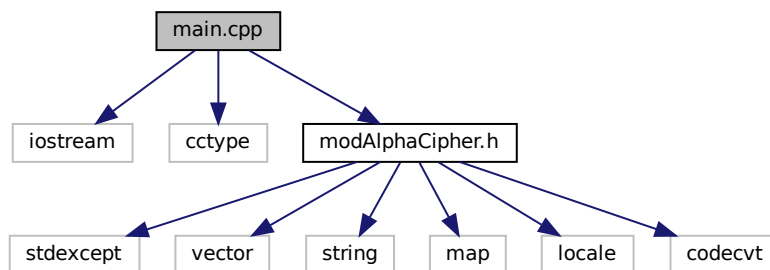
Файлы

5.1 Файл main.cpp

Главный модуль проекта

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- int main ()

5.1.1 Подробное описание

Главный модуль проекта

Автор

Воронин Н.А.

Дата

11.12.2022

Авторство

ИБСТ ПГУ

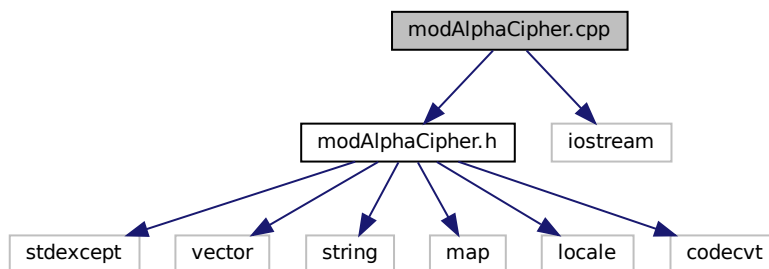
5.2 Файл modAlphaCipher.cpp

Модуль реализации методов класса [modAlphaCipher](#).

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

```
#include <iostream>
```

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



5.2.1 Подробное описание

Модуль реализации методов класса [modAlphaCipher](#).

Автор

Воронин Н.А.

Дата

11.12.2022

Авторство

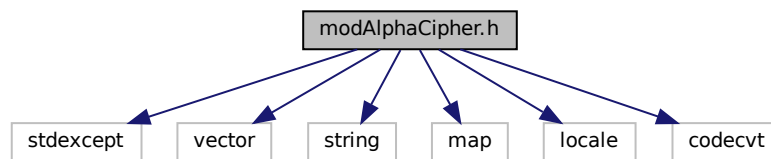
ИБСТ ПГУ

5.3 Файл modAlphaCipher.h

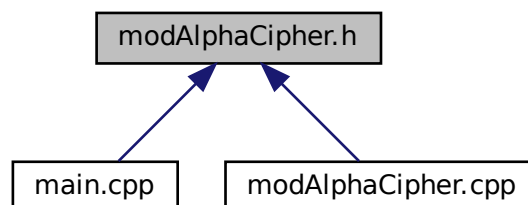
Заголовочный файл проекта

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



Классы

- class `modAlphaCipher`
Шифрование методом Гронсфельда
- class `cipher_error`
Созданное исключение

5.3.1 Подробное описание

Заголовочный файл проекта

Автор

Воронин Н.А.

Дата

11.12.2022

Авторство

ИБСТ ПГУ

Предметный указатель

- cipher_error, [7](#)
- convert
 - modAlphaCipher, [10](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [11](#)
- getValidCipherText
 - modAlphaCipher, [11](#)
- getValidKey
 - modAlphaCipher, [12](#)
- getValidOpenText
 - modAlphaCipher, [12](#)
- main.cpp, [15](#)
- modAlphaCipher, [8](#)
 - convert, [10](#)
 - decrypt, [10](#)
 - encrypt, [11](#)
 - getValidCipherText, [11](#)
 - getValidKey, [12](#)
 - getValidOpenText, [12](#)
 - modAlphaCipher, [9](#)
 - numAlpha, [13](#)
- modAlphaCipher.cpp, [16](#)
- modAlphaCipher.h, [17](#)
- numAlpha
 - modAlphaCipher, [13](#)