

Шифрование методом Гронсфельда

1.0

Создано системой Doxygen 1.9.8

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
4.1.2 Конструктор(ы)	8
4.1.2.1 cipher_error()	8
4.1.3 Методы	8
4.1.3.1 what()	8
4.2 Класс modAlphaCipher	9
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()	11
4.2.3.4 encrypt()	11
4.2.3.5 toUpper()	12
5 Файлы	13
5.1 Файл main.cpp	13
5.1.1 Подробное описание	14
5.1.2 Функции	14
5.1.2.1 exception_handling()	14
5.1.2.2 main()	14
5.2 Файл modAlphaCipher.cpp	15
5.2.1 Подробное описание	15
5.3 Файл modAlphaCipher.h	15
5.3.1 Подробное описание	16
5.4 modAlphaCipher.h	17
Предметный указатель	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::exception	
cipher_error	7
modAlphaCipher	9

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	
Класс для исключений шифрования	7
modAlphaCipher	
Шифрование методом Гронсфельда для русского языка	9

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

main.cpp	Главный модуль программы шифрования методом Гронсфельда	13
modAlphaCipher.cpp	Реализация класса шифрования методом Гронсфельда	15
modAlphaCipher.h	Заголовочный файл для модуля шифрования методом Гронсфельда	15

Глава 4

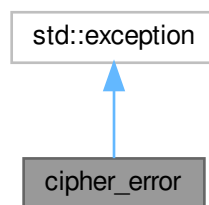
Классы

4.1 Класс cipher_error

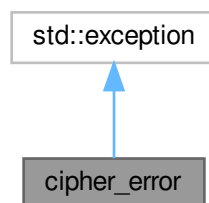
Класс для исключений шифрования

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

Граф наследования: cipher_error:



Граф связей класса cipher_error:



Открытые члены

- [cipher_error](#) (const std::string &msg)
Конструктор исключения
- const char * [what](#) () const noexcept override
Получить сообщение об ошибке

Закрытые данные

- std::string message

4.1.1 Подробное описание

Класс для исключений шифрования

Наследуется от std::exception для интеграции со стандартной системой исключений

4.1.2 Конструктор(ы)

4.1.2.1 cipher_error()

```

cipher_error::cipher_error (
    const std::string & msg )    [inline], [explicit]

```

Конструктор исключения

Аргументы

msg	Сообщение об ошибке
-----	---------------------

4.1.3 Методы

4.1.3.1 what()

```

const char * cipher_error::what ( ) const    [inline], [override], [noexcept]

```

Получить сообщение об ошибке

Возвращает

Указатель на строку с сообщением об ошибке

Объявления и описания членов класса находятся в файле:

- [modAlphaCipher.h](#)

4.2 Класс modAlphaCipher

Шифрование методом Гронсфельда для русского языка

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

Открытые члены

- `modAlphaCipher ()=delete`
Запрещенный конструктор без параметров
- `modAlphaCipher (const std::wstring &skey)`
Конструктор для установки ключа
- `std::wstring encrypt (const std::wstring &open_text)`
Зашифровывание текста
- `std::wstring decrypt (const std::wstring &cipher_text)`
Расшифровывание текста

Закрытые члены

- `std::vector< int > convert (const std::wstring &s)`
Преобразование строки в числовой вектор
- `std::wstring convert (const std::vector< int > &v)`
Преобразование числового вектора в строку
- `std::wstring toUpper (const std::wstring &s)`
Приведение строки к верхнему регистру

Закрытые данные

- `std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"`
Алфавит русского языка
- `std::map< wchar_t, int > alphaNum`
Ассоциативный массив "символ-номер".
- `std::vector< int > key`
Ключ шифрования в числовом виде

4.2.1 Подробное описание

Шифрование методом Гронсфельда для русского языка

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::wstring & skey )
```

Конструктор для установки ключа

Аргументы

skey	Ключ шифрования
------	-----------------

Исключения

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

4.2.3 Методы

4.2.3.1 convert() [1/2]

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

Преобразование числового вектора в строку

Аргументы

v	Вектор числовых представлений
---	-------------------------------

Возвращает

Результирующая строка

Исключения

cipher_error	Если индекс выходит за границы алфавита
------------------------------	---

4.2.3.2 convert() [2/2]

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

Преобразование строки в числовой вектор

Аргументы

s	Исходная строка
---	-----------------

Возвращает

Вектор числовых представлений символов

4.2.3.3 decrypt()

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

Расшифровывание текста

Аргументы

<code>cipher_text</code>	Зашифрованный текст. Не должен быть пустой строкой. Должен содержать только буквы русского алфавита
--------------------------	---

Возвращает

Расшифрованная строка

Исключения

<code>cipher_error</code>	Если текст пустой или содержит недопустимые символы
---------------------------	---

Аргументы

<code>cipher_text</code>	Зашифрованный текст
--------------------------	---------------------

Возвращает

Расшифрованная строка

Исключения

<code>cipher_error</code>	Если текст пустой или содержит недопустимые символы
---------------------------	---

4.2.3.4 encrypt()

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

Зашифровывание текста

Аргументы

<code>open_text</code>	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 или содержит недопустимые символы
------------------------------	---

Аргументы

open_text	Открытый текст
-----------	----------------

Возвращает

Зашифрованная строка

Исключения

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

4.2.3.5 toUpper()

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

Приведение строки к верхнему регистру

Аргументы

s	Исходная строка
---	-----------------

Возвращает

Строка в верхнем регистре

Объявления и описания членов классов находятся в файлах:

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

Глава 5

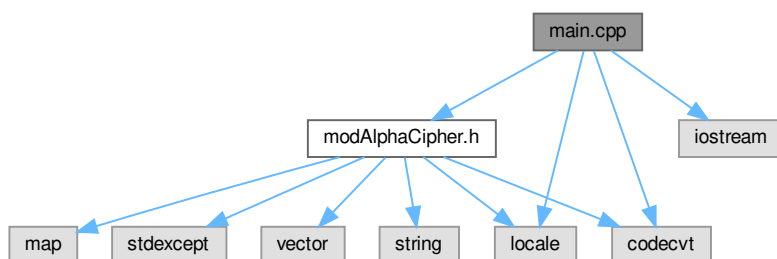
Файлы

5.1 Файл main.cpp

Главный модуль программы шифрования методом Гронсфельда

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- void `exception_handling` (const std::wstring &Text, const std::wstring &key, const std::string &testName)
Обработка исключений при тестировании шифрования
- void `testCorrectCases` ()
Тестирование корректных случаев шифрования
- void `testErrorCases` ()
Тестирование ошибочных случаев шифрования
- void `demonstrateErrorTypes` ()
Демонстрация возможных типов ошибок
- int `main` (int argc, char **argv)
Главная функция программы

5.1.1 Подробное описание

Главный модуль программы шифрования методом Гронсфельда

Автор

Ганьшин В.А.

Версия

1.0

Дата

17.12.2025

Содержит функции для тестирования модуля шифрования

5.1.2 Функции

5.1.2.1 exception_handling()

```
void exception_handling (
    const std::wstring & Text,
    const std::wstring & key,
    const std::string & testName )
```

Обработка исключений при тестировании шифрования

Аргументы

Text	Текст для обработки
key	Ключ шифрования
testName	Название теста

5.1.2.2 main()

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

Главная функция программы

Аргументы

argc	Количество аргументов командной строки
argv	Массив аргументов командной строки

Возвращает

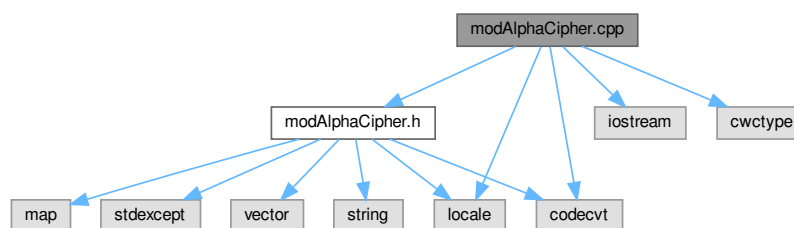
Код завершения программы

5.2 Файл modAlphaCipher.cpp

Реализация класса шифрования методом Гронсфеляда

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

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



5.2.1 Подробное описание

Реализация класса шифрования методом Гронсфеляда

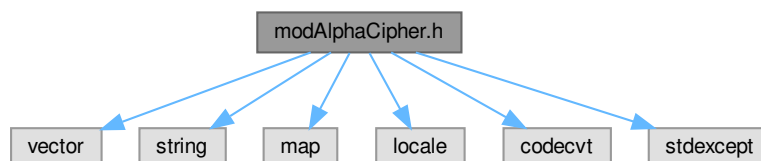
Содержит реализацию всех методов класса [modAlphaCipher](#)

5.3 Файл modAlphaCipher.h

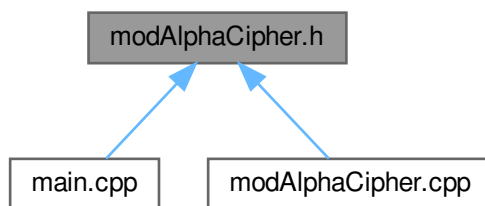
Заголовочный файл для модуля шифрования методом Гронсфеляда

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



Классы

- class `cipher_error`
Класс для исключений шифрования
- class `modAlphaCipher`
Шифрование методом Гронсфельда для русского языка

5.3.1 Подробное описание

Заголовочный файл для модуля шифрования методом Гронсфельда

Автор

Ганьшин В.А.

Версия

1.0

Дата

17.12.2025

Предупреждения

Это документирование программы лабораторной работы

5.4 modAlphaCipher.h

[См. документацию.](#)

```
00001 #pragma once
00002 #include <vector>
00003 #include <string>
00004 #include <map>
00005 #include <locale>
00006 #include <codecvt>
00007 #include <stdexcept>
00008
00022 class cipher_error : public std::exception {
00023 private:
00024     std::string message;
00025 public:
00030     explicit cipher_error(const std::string& msg) : message(msg) {}
00031
00036     const char* what() const noexcept override {
00037         return message.c_str();
00038     }
00039 };
00040
00047 class modAlphaCipher
00048 {
00049 private:
00050     std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
00051     std::map <wchar_t,int> alphaNum;
00052     std::vector <int> key;
00053
00059     std::vector<int> convert(const std::wstring& s);
00060
00067     std::wstring convert(const std::vector<int>& v);
00068
00074     std::wstring toUpper(const std::wstring& s);
00075
00076 public:
00080     modAlphaCipher()=delete;
00081
00087     modAlphaCipher(const std::wstring& skey);
00088
00097     std::wstring encrypt(const std::wstring& open_text);
00098
00106     std::wstring decrypt(const std::wstring& cipher_text);
00107 };
```


Предметный указатель

- cipher_error, [7](#)
 - cipher_error, [8](#)
 - what, [8](#)
- convert
 - modAlphaCipher, [10](#)
- decrypt
 - modAlphaCipher, [10](#)
- encrypt
 - modAlphaCipher, [11](#)
- exception_handling
 - main.cpp, [14](#)
- main
 - main.cpp, [14](#)
- main.cpp, [13](#)
 - exception_handling, [14](#)
 - main, [14](#)
- modAlphaCipher, [9](#)
 - convert, [10](#)
 - decrypt, [10](#)
 - encrypt, [11](#)
 - modAlphaCipher, [9](#)
 - toUpper, [12](#)
- modAlphaCipher.cpp, [15](#)
- modAlphaCipher.h, [15](#)
- toUpper
 - modAlphaCipher, [12](#)
- what
 - cipher_error, [8](#)