RouteCipher

1.0

Doxygen 1.9.1

1

1.1

Иерархия классов.

2

2.1

Классы с их кратким описанием.

3

3.1

Полный список документированных файлов.

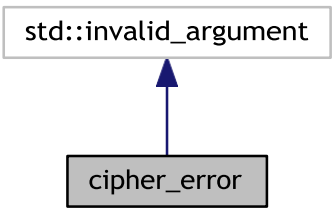
4

4.1 cipher_error

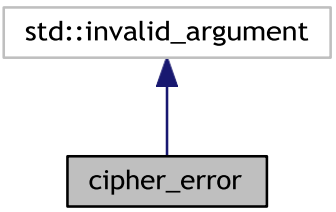
Класс исключения для ошибок шифрования

#include <module.h>

Граф наследования:cipher_error:



Граф связей класса cipher_error:



- cipher_error (const std::string &what_arg)
- cipher_error (const char *what_arg)
  (C-)

4.1.1

Класс исключения для ошибок шифрования

Наследуется от std::invalid_argument

4.1.2 ( )

4.1.2.1 cipher_error() [1/2]

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

Конструктор исключения с передачей строки

Аргументы

what_arg	- сообщение об ошибке

4.1.2.2 cipher_error() [2/2]

cipher_error::cipher_error (
const char * what_arg) [inline], [explicit]

Конструктор исключения с передачей строки (C-стиль)

Аргументы

what_arg	- сообщение об ошибке

Объявления и описания членов класса находятся в файле:

- module.h

4.2 RouteCipher

Класс для шифрования методом маршрутной перестановки

#include <module.h>

- RouteCipher ()=delete
- RouteCipher (const std::wstring &key)
- void setKey (const std::wstring &key)
- std::wstring getKey () const
- std::wstring encrypt (const std::wstring &text)
- std::wstring decrypt (const std::wstring &text)

- std::wstring getValidKey (const std::wstring &s)
- std::wstring getValidOpenText (const std::wstring &s)
- std::wstring getValidCipherText (const std::wstring &s)

- int columns

( )

4.2.1

Класс для шифрования методом маршрутной перестановки

Использует таблицу с заданным числом столбцов. Запись: слева направо, сверху вниз. Считывание: сверху вниз, справа налево.

4.2.2 ( )

4.2.2.1 RouteCipher()

```
RouteCipher::RouteCipher (
            const std::wstring & key )
```

Конструктор с установкой ключа

Конструктор RouteCipher.

Аргументы

key	- ключ шифрования (количество столбцов)

Исключения

cipher_error	если ключ невалиден

4.2.3

4.2.3.1 decrypt()

```
std::wstring RouteCipher::decrypt (
            const std::wstring & text )
```

Дешифрование текста

Дешифрование текста методом маршрутной перестановки

Аргументы

text	- шифртекст

Возвращает

Расшифрованная строка

Исключения

cipher_error	если текст пуст или содержит не заглавные буквы

4.2.3.2 encrypt()

```
std::wstring RouteCipher::encrypt (
            const std::wstring & text )
```

Шифрование текста

Шифрование текста методом маршрутной перестановки

Аргументы

text	- открытый текст

Возвращает

Зашифрованная строка

Исключения

cipher_error	если текст пуст после очистки

4.2.3.3 getKey()

std::wstring RouteCipher::getKey ( ) const

Получение текущего ключа

Возвращает

Строковое представление ключа

4.2.3.4 getValidCipherText()

std::wstring RouteCipher::getValidCipherText (
const std::wstring & s) [private]

Проверка шифртекста

Аргументы

s	- шифртекст

Возвращает

Текст без изменений

Исключения

cipher_error	если текст пуст или содержит не заглавные буквы

4.2.3.5 getValidKey()

std::wstring RouteCipher::getValidKey (
const std::wstring & s) [private]

Проверка и нормализация ключа

Проверка ключа на валидность

Аргументы

s	- исходный ключ (строка с числом)

Возвращает

Валидная строка ключа

Исключения

cipher_error	если ключ пустой, содержит не цифры, или число <= 0

Аргументы

s	- исходный ключ

Возвращает

Нормализованная строка ключа

Исключения

cipher_error	если ключ невалиден

4.2.3.6 getValidOpenText()

std::wstring RouteCipher::getValidOpenText (
const std::wstring & s) [private]

Проверка и нормализация открытого текста

Проверка и очистка открытого текста

Аргументы

s	- исходный текст

Возвращает

Текст в верхнем регистре без не-букв

Исключения

cipher_error	если после очистки текст пуст

Аргументы

s	- исходный текст

Возвращает

Текст в верхнем регистре без не-букв

Исключения

cipher_error	если текст пуст после очистки

4.2.3.7 setKey()

```
void RouteCipher::setKey (
            const std::wstring & key )
```

Установка нового ключа

Аргументы

key	- новый ключ

Исключения

cipher_error	если ключ невалиден

Объявления и описания членов классов находятся в файлах:

- module.h
- module.cpp

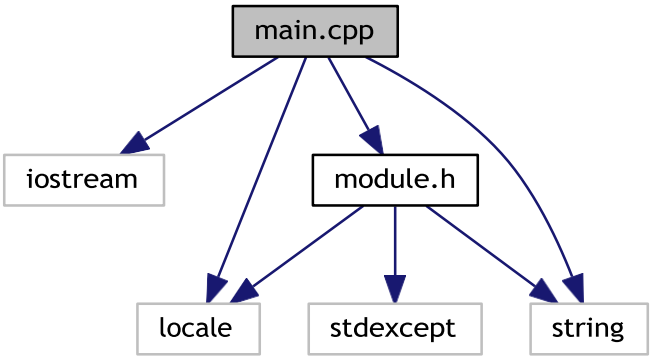
5

5.1 main.cpp

Главный модуль программы для шифрования методом маршрутной перестановки

```
#include <iostream>
#include <locale>
#include <string>
#include "module.h"
```

Граф включаемых заголовочных файлов для main.cpp:



- bool isValidKey (const std::wstring &s)
- int main ()

5.1.1

Главный модуль программы для шифрования методом маршрутной перестановки

(RouteCipher)

Автор

Пресняков Александр

Версия

1.0

Дата

2025

Программа предоставляет интерактивный интерфейс для шифрования и дешифрования текста методом табличной маршрутной перестановки.

5.1.2

5.1.2.1 isValidKey()

```
bool isValidKey (
            const std::wstring & s )
```

Проверка корректности ключа

Аргументы

s	- строка с ключом

Возвращает

true если ключ валиден, false в противном случае

Ключ должен быть положительным целым числом, состоящим только из цифр

5.1.2.2 main()

```
int main ( )
```

Главная функция программы

Возвращает

0 при успешном завершении, 1 при ошибке

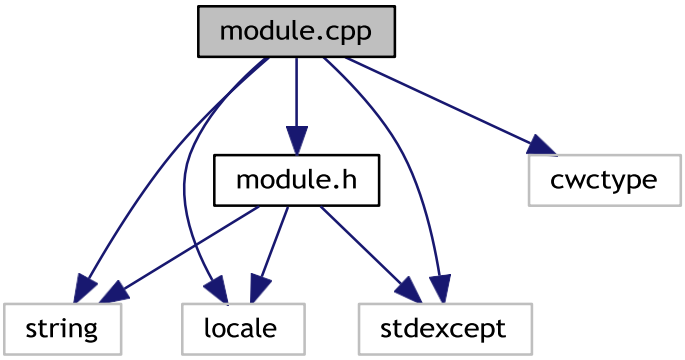
Устанавливает локаль для работы с русским языком, принимает ключ от пользователя, создаёт экземпляр шифра RouteCipher и предоставляет интерактивное меню для операций шифрования/дешифрования

5.2 module.cpp

Реализация класса RouteCipher.

```
#include "module.h"
#include <stdexcept>
#include <locale>
#include <string>
#include <cwctype>
```

Граф включаемых заголовочных файлов для module.cpp:



5.2.1

Реализация класса RouteCipher.

Автор

Пресняков Александр

Версия

1.0

Дата

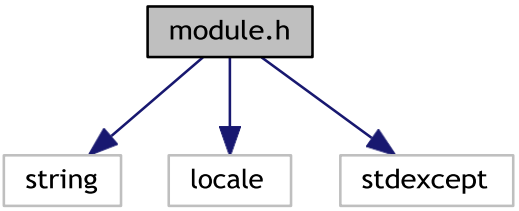
2025

5.3 module.h

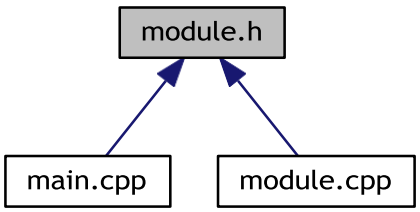
Заголовочный файл класса RouteCipher для шифрования маршрутной перестановкой

```
#include <string>
#include <locale>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для module.h:



Граф файлов, в которые включается этот файл:



- class cipher_error
- class RouteCipher

5.3.1

Заголовочный файл класса RouteCipher для шифрования маршрутной перестановкой

Автор

Пресняков Александр

Версия

1.0

Дата

2025

Предупреждения

Реализация для работы с текстами в кодировке UTF-8 (wstring)

CryptoLab -
1.0

Doxygen 1.9.1

1.1

Иерархия классов.

1

2.1

Классы с их кратким описанием.

2

3.1

Полный список документированных файлов.

3

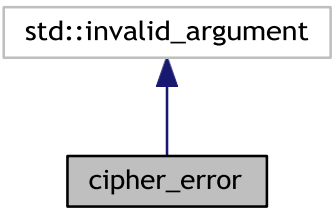
4

4.1 cipher_error

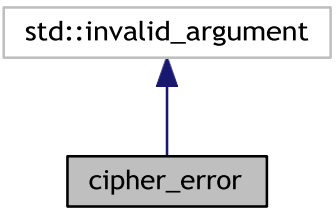
Класс исключения для ошибок шифрования

#include <modAlphaCipher.h>

Граф наследования:cipher_error:



Граф связей класса cipher_error:



- cipher_error (const std::string &what_arg)
- cipher_error (const char *what_arg)
  (C-)

4.1.1

Класс исключения для ошибок шифрования

Наследуется от std::invalid_argument

4.1.2 ( )

4.1.2.1 cipher_error() [1/2]

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

Конструктор исключения с передачей строки

Аргументы

what_arg	- сообщение об ошибке

4.1.2.2 cipher_error() [2/2]

cipher_error::cipher_error (
const char * what_arg) [inline], [explicit]

Конструктор исключения с передачей строки (C-стиль)

Аргументы

what_arg	- сообщение об ошибке

Объявления и описания членов класса находятся в файле:

- modAlphaCipher.h

4.2 modAlphaCipher

Класс для шифрования методом Гронсфельда

#include <modAlphaCipher.h>

- modAlphaCipher ()=delete
- modAlphaCipher (const std::wstring &skey)
- std::wstring encrypt (const std::wstring &open_text)
- std::wstring decrypt (const std::wstring &cipher_text)

- std::vector< int > convert (const std::wstring &s)
- std::wstring convert (const std::vector< int > &v)
- std::wstring getValidKey (const std::wstring &s)
- std::wstring getValidOpenText (const std::wstring &s)
- std::wstring getValidCipherText (const std::wstring &s)

- std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
- std::map< wchar_t, int > alphaNum
  '' ''.
- std::vector< int > key

4.2.1

Класс для шифрования методом Гронсфельда

Использует сложение символов сообщения с символами ключа по модулю 33 Алфавит: АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ (33 символа)

4.2.2 ( )

4.2.2.1 modAlphaCipher()

modAlphaCipher::modAlphaCipher (
const std::wstring & skey)

Конструктор с установкой ключа

Конструктор класса

Аргументы

skey	- ключ шифрования

Исключения

cipher_error	если ключ невалиден

Аргументы

skey	- ключ шифрования

Исключения

cipher_error	если ключ невалиден

Инициализирует алфавит и преобразует ключ в числовой вид

4.2.3

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 чисел

Возвращает

Строка, соответствующая вектору

4.2.3.2 convert() [2/2]

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

Преобразование строки в числовой вектор

Аргументы

s	- строка для преобразования

Возвращает

Вектор чисел (номеров символов в алфавите)

Аргументы

s	- строка для преобразования

Возвращает

Вектор номеров символов

4.2.3.3 decrypt()

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

Дешифрование текста

Дешифрование текста методом Гронсфельда

Аргументы

cipher_text	- шифртекст

Возвращает

Расшифрованная строка

Исключения

cipher_error	если текст невалиден

Аргументы

cipher_text	- шифртекст

Возвращает

Расшифрованная строка

Исключения

cipher_error	если текст пуст или содержит не заглавные буквы

Выполняется обратное сложение с ключом по модулю 33

4.2.3.4 encrypt()

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

Шифрование текста

Шифрование текста методом Гронсфельда

Аргументы

open_text	- открытый текст

Возвращает

Зашифрованная строка

Исключения

cipher_error	если текст невалиден

Аргументы

open_text	- открытый текст

Возвращает

Зашифрованная строка

Исключения

cipher_error	если текст пуст после очистки

Текст очищается, преобразуется в числа, складывается с ключом по модулю 33

4.2.3.5 getValidCipherText()

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

Проверка шифртекста

Аргументы

s	- шифртекст

Возвращает

Текст без изменений

Исключения

cipher_error	если текст пуст или содержит не заглавные буквы

4.2.3.6 getValidKey()

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

Проверка и нормализация ключа

Аргументы

s	- исходный ключ

Возвращает

Валидный ключ в верхнем регистре

Исключения

cipher_error	если ключ пустой или содержит не-буквы

Аргументы

s	- исходный ключ

Возвращает

Ключ в верхнем регистре

Исключения

cipher_error	если ключ пустой или содержит не-буквы

4.2.3.7 getValidOpenText()

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

Проверка и нормализация открытого текста

Проверка и очистка открытого текста

Аргументы

s	- исходный текст

Возвращает

Текст в верхнем регистре без не-букв

Исключения

cipher_error	если после очистки текст пуст

Аргументы

s	- исходный текст

Возвращает

Текст в верхнем регистре без не-букв

Исключения

cipher_error	если текст пуст после очистки

Объявления и описания членов классов находятся в файлах:

- modAlphaCipher.h
- modAlphaCipher.cpp

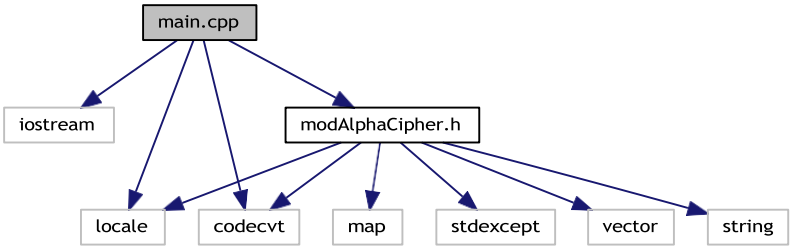
5

main.cpp

Главный модуль программы для тестирования шифра Гронсфельда

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

Граф включаемых заголовочных файлов для main.cpp:



- void check (const std::wstring &Text, const std::wstring &key, bool destructCipherText=false)
- void interactiveMode ()
- int main ()

Главный модуль программы для тестирования шифра Гронсфельда

Автор

Пресняков Александр

Версия

1.0

Дата

2025

check()

```
void check (
        const std::wstring & Text,
        const std::wstring & key,
        bool destructCipherText = false )
```

Функция для проверки работы шифра

Аргументы

Text	- исходный текст
key	- ключ шифрования
destructCipherText	- флаг для порчи шифртекста (тестирование обработки ошибок)

Шифрует текст, опционально портит его, затем расшифровывает и сравнивает

interactiveMode()

```
void interactiveMode ( )
```

Интерактивный режим работы программы

Позволяет пользователю вводить ключ, шифровать/дешифровать тексты и запускать автоматическое тестирование

main()

int main ( )

Главная функция программы

Возвращает

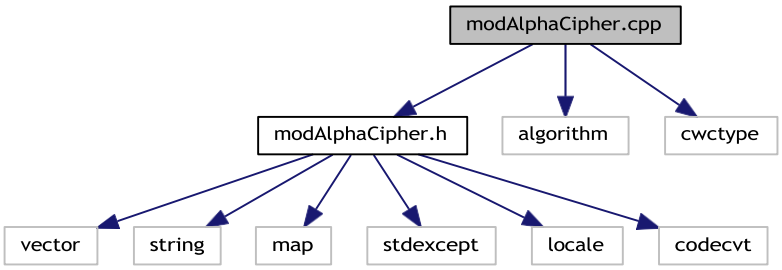
0 при успешном завершении

Устанавливает локаль и запускает интерактивный режим

modAlphaCipher.cpp

Реализация класса modAlphaCipher.

#include "modAlphaCipher.h"
#include <algorithm>
#include <cwctype>
Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



Реализация класса modAlphaCipher.

Автор

Пресняков Александр

Версия

1.0

Дата

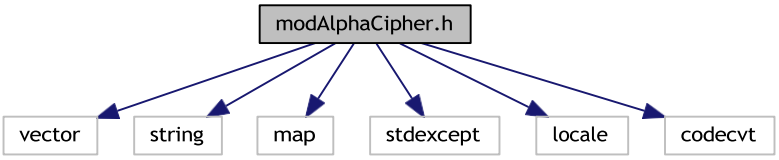
2025

modAlphaCipher.h

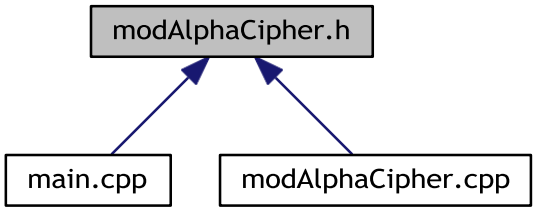
Заголовочный файл класса modAlphaCipher для шифрования методом Гронсфельда

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

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



- class cipher_error
- class modAlphaCipher

Заголовочный файл класса modAlphaCipher для шифрования методом Гронсфельда

Автор

Пресняков Александр

Версия

1.0

Дата

2025

Предупреждения

Реализация для русского алфавита (33 буквы)

Ссылка на репозиторий : https://github.com/24pt208/lab_4