

Шифратор Гронсфельда

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
4.1.2 Конструктор(ы)	8
4.1.2.1 cipher_error() [1/2]	8
4.1.2.2 cipher_error() [2/2]	8
4.2 Структура KeyBFixture	9
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 KeyBFixture()	9
4.2.2.2 ~KeyBFixture()	10
4.2.3 Данные класса	10
4.2.3.1 cipher	10
4.3 Класс modAlphaCipher	10
4.3.1 Подробное описание	10
4.3.2 Конструктор(ы)	10
4.3.2.1 modAlphaCipher() [1/2]	11
4.3.2.2 modAlphaCipher() [2/2]	11
4.3.3 Методы	12
4.3.3.1 decrypt()	12
4.3.3.2 encrypt()	13
5 Файлы	15
5.1 Файл Desktop/4лб/task1/main.cpp	15
5.1.1 Подробное описание	16
5.1.2 Функции	16
5.1.2.1 main()	16
5.1.2.2 str8_to_w()	16
5.1.2.3 w_to_str8()	17
5.2 Файл Desktop/4лб/task1/modAlphaCipher.cpp	17
5.2.1 Подробное описание	17
5.3 Файл Desktop/4лб/task1/modAlphaCipher.h	18
5.3.1 Подробное описание	19
5.4 Файл Desktop/4лб/task1/test.cpp	19
5.4.1 Подробное описание	20
5.4.2 Макросы	20

5.4.2.1 CHECK_WIDE_EQUAL	21
5.4.3 Функции	21
5.4.3.1 main()	21
5.4.3.2 SUITE() [1/3]	21
5.4.3.3 SUITE() [2/3]	21
5.4.3.4 SUITE() [3/3]	22
5.4.3.5 utf8ToWide()	22
5.4.3.6 wideToUtf8()	22
Предметный указатель	23

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

invalid_argument	
cipher_error	7
KeyBFixture	9
modAlphaCipher	10

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

cipher_error	Класс-исключение для ошибок шифрования	7
KeyBFixture	Фикстура для тестов с ключом "Б"	9
modAlphaCipher	Класс для шифрования методом Гронсфельда	10

Глава 3

Список файлов

3.1 Файлы

Полный список файлов.

Desktop/4лб/task1/ main.cpp	
Главный модуль программы для шифрования/расшифрования текста	15
Desktop/4лб/task1/ modAlphaCipher.cpp	
Реализация методов класса modAlphaCipher	17
Desktop/4лб/task1/ modAlphaCipher.h	
Заголовочный файл для модуля шифрования методом Гронсфельда	18
Desktop/4лб/task1/ test.cpp	
Модульные тесты для класса modAlphaCipher	19

Глава 4

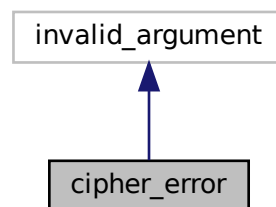
Классы

4.1 Класс `cipher_error`

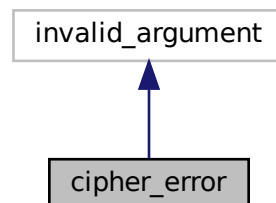
Класс-исключение для ошибок шифрования

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

Граф наследования: `cipher_error`:



Граф связей класса `cipher_error`:



Открытые члены

- [cipher_error](#) (const string &what_arg)
Конструктор с параметром string.
- [cipher_error](#) (const char *what_arg)
Конструктор с параметром const char*.

4.1.1 Подробное описание

Класс-исключение для ошибок шифрования

Наследуется от std::invalid_argument

4.1.2 Конструктор(ы)

4.1.2.1 cipher_error() [1/2]

```

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

```

Конструктор с параметром string.

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 с параметром const char*.

Аргументы

what_arg	сообщение об ошибке
----------	---------------------

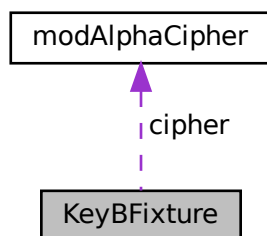
Объявления и описания членов класса находятся в файле:

- Desktop/4лб/task1/[modAlphaCipher.h](#)

4.2 Структура KeyBFixture

Фикстура для тестов с ключом "Б".

Граф связей класса KeyBFixture:



Открытые члены

- `KeyBFixture ()`
- `~KeyBFixture ()`

Открытые атрибуты

- `modAlphaCipher * cipher`

4.2.1 Подробное описание

Фикстура для тестов с ключом "Б".

Создает экземпляр шифратора с ключом "Б" для повторного использования

4.2.2 Конструктор(ы)

4.2.2.1 KeyBFixture()

`KeyBFixture::KeyBFixture ()` [inline]

4.2.2.2 ~KeyBFixture()

KeyBFixture::~KeyBFixture () [inline]

4.2.3 Данные класса

4.2.3.1 cipher

`modAlphaCipher*` KeyBFixture::cipher

Объявления и описания членов структуры находятся в файле:

- Desktop/4лб/task1/[test.cpp](#)

4.3 Класс modAlphaCipher

Класс для шифрования методом Гронсфельда

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

Открытые члены

- `modAlphaCipher ()=delete`
Удаленный конструктор по умолчанию
- `modAlphaCipher (const wstring &keyStr)`
Конструктор с установкой ключа
- `wstring encrypt (const wstring &plain)`
Шифрование открытого текста
- `wstring decrypt (const wstring &cipher)`
Расшифрование зашифрованного текста

4.3.1 Подробное описание

Класс для шифрования методом Гронсфельда

Реализует шифрование и расшифрование текста на русском языке

4.3.2 Конструктор(ы)

4.3.2.1 modAlphaCipher() [1/2]

```
modAlphaCipher::modAlphaCipher ( ) [delete]
```

Удаленный конструктор по умолчанию

4.3.2.2 modAlphaCipher() [2/2]

```
modAlphaCipher::modAlphaCipher (
    const wstring & keyStr )
```

Конструктор с установкой ключа

Конструктор класса [modAlphaCipher](#).

Аргументы

keyStr	строковый ключ
--------	----------------

Исключения

cipher_error	если ключ невалиден
------------------------------	---------------------

Аргументы

keyStr	строковый ключ для шифрования
--------	-------------------------------

Исключения

cipher_error	если ключ невалиден
------------------------------	---------------------

4.3.3 Методы

4.3.3.1 decrypt()

```
wstring modAlphaCipher::decrypt (  
    const wstring & cipher )
```

Расшифрование зашифрованного текста

Аргументы

cipher	зашифрованный текст
--------	---------------------

Возвращает

расшифрованный текст

Исключения

cipher_error	если текст невалиден
------------------------------	----------------------

Аргументы

cipher	зашифрованный текст для расшифрования
--------	---------------------------------------

Возвращает

расшифрованный текст

Исключения

cipher_error	если зашифрованный текст невалиден
------------------------------	------------------------------------

4.3.3.2 encrypt()

```
wstring modAlphaCipher::encrypt (
    const wstring & plain )
```

Шифрование открытого текста

Аргументы

<code>plain</code>	открытый текст
--------------------	----------------

Возвращает

зашифрованный текст

Исключения

cipher_error	если текст невалиден
------------------------------	----------------------

Аргументы

<code>plain</code>	открытый текст для шифрования
--------------------	-------------------------------

Возвращает

зашифрованный текст

Исключения

cipher_error	если открытый текст невалиден
------------------------------	-------------------------------

Объявления и описания членов классов находятся в файлах:

- Desktop/4лб/task1/[modAlphaCipher.h](#)
- Desktop/4лб/task1/[modAlphaCipher.cpp](#)

Глава 5

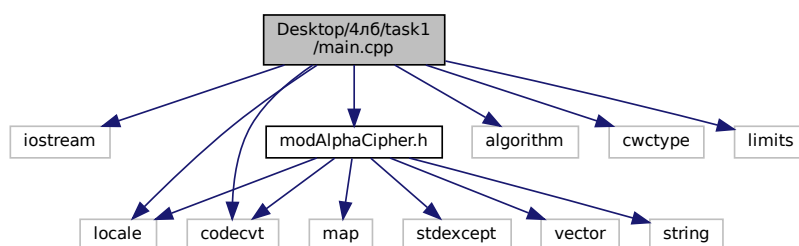
Файлы

5.1 Файл Desktop/4лб/task1/main.cpp

Главный модуль программы для шифрования/расшифрования текста

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

Граф включаемых заголовочных файлов для main.cpp:



Функции

- wstring `str8_to_w` (const string &s)
Преобразование строки UTF-8 в широкую строку
- string `w_to_str8` (const wstring &ws)
Преобразование широкой строки в строку UTF-8.
- int `main` ()
Главная функция программы

5.1.1 Подробное описание

Главный модуль программы для шифрования/расшифрования текста

Автор

Соловьев Арсений Евгеньевич

Версия

1.0

Дата

2025

Реализует пользовательский интерфейс для работы с шифром Гронсфельда

5.1.2 Функции

5.1.2.1 main()

```
int main ( )
```

Главная функция программы

Возвращает

код завершения программы

Реализует диалоговый интерфейс для шифрования/расшифрования

5.1.2.2 str8_to_w()

```
wstring str8_to_w (
    const string & s )
```

Преобразование строки UTF-8 в широкую строку

Аргументы

s	строка в UTF-8
---	----------------

Возвращает

широкая строка

5.1.2.3 w_to_str8()

```
string w_to_str8 (
    const wstring & ws )
```

Преобразование широкой строки в строку UTF-8.

Аргументы

ws	широкая строка
----	----------------

Возвращает

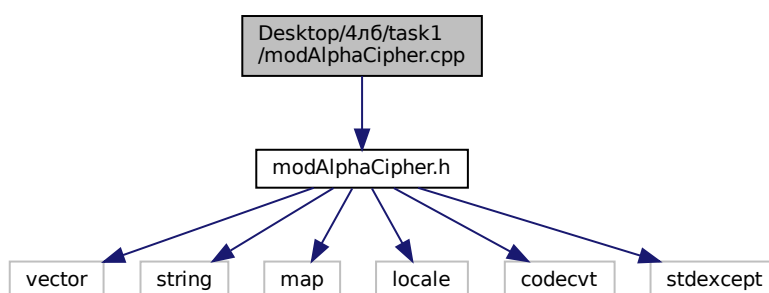
строка в UTF-8

5.2 Файл Desktop/4лб/task1/modAlphaCipher.cpp

Реализация методов класса [modAlphaCipher](#).

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

Граф включаемых заголовочных файлов для modAlphaCipher.cpp:



5.2.1 Подробное описание

Реализация методов класса [modAlphaCipher](#).

Автор

Соловьев Арсений Евгеньевич

Версия

1.0

Дата

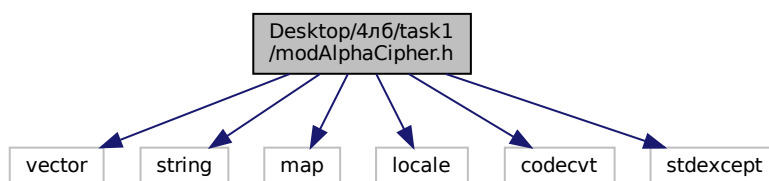
2025

5.3 Файл Desktop/4лб/task1/modAlphaCipher.h

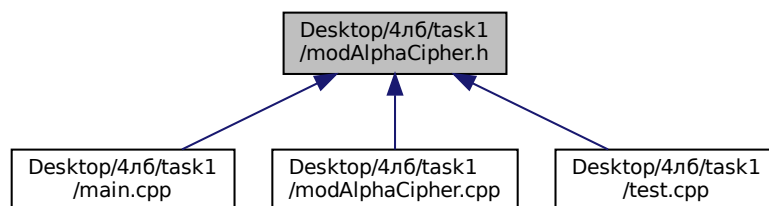
Заголовочный файл для модуля шифрования методом Гронсфелда

```
#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-исключение для ошибок шифрования
- class `modAlphaCipher`
Класс для шифрования методом Гронсфельда

5.3.1 Подробное описание

Заголовочный файл для модуля шифрования методом Гронсфельда

Автор

Соловьев Арсений Евгеньевич

Версия

1.0

Дата

2025

Авторство

ПГУ

Предупреждения

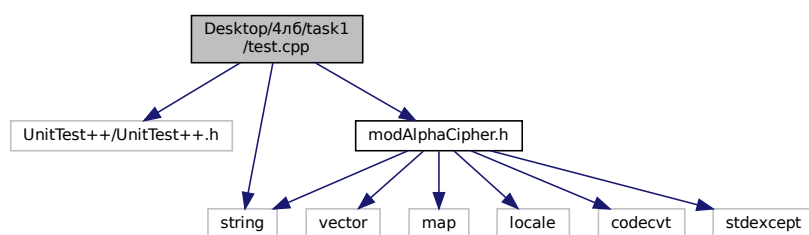
Реализация для русского языка

5.4 Файл Desktop/4лб/task1/test.cpp

Модульные тесты для класса `modAlphaCipher`.

```
#include <UnitTest++/UnitTest++.h>
#include <string>
#include "modAlphaCipher.h"
```

Граф включаемых заголовочных файлов для test.cpp:



Классы

- struct [KeyBFixture](#)
Фикстура для тестов с ключом "Б".

Макросы

- `#define CHECK_WIDE_EQUAL(expected, actual) CHECK_EQUAL(wideToUtf8(expected), wideToUtf8(actual))`
Макрос для сравнения широких строк в тестах

Функции

- string [wideToUtf8](#) (const wstring &ws)
Преобразование широкой строки в UTF-8.
- wstring [utf8ToWide](#) (const string &s)
Преобразование строки UTF-8 в широкую строку
- [SUITE](#) (ConstructorTest)
Тестовый набор для конструктора
- [SUITE](#) (EncryptTest)
Тестовый набор для метода encrypt.
- [SUITE](#) (DecryptTest)
Тестовый набор для метода decrypt.
- int [main](#) (int argc, char **argv)
Главная функция тестов

5.4.1 Подробное описание

Модульные тесты для класса [modAlphaCipher](#).

Автор

Соловьев Арсений Евгеньевич

Версия

1.0

Дата

2025

Тестирование функциональности шифрования и обработки ошибок

5.4.2 Макросы

5.4.2.1 CHECK_WIDE_EQUAL

```
#define CHECK_WIDE_EQUAL(  
    expected,  
    actual ) CHECK_EQUAL(wideToUtf8(expected), wideToUtf8(actual))
```

Макрос для сравнения широких строк в тестах

5.4.3 Функции

5.4.3.1 main()

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

Главная функция тестов

Аргументы

argc	количество аргументов
argv	массив аргументов

Возвращает

результат выполнения тестов

5.4.3.2 SUITE() [1/3]

```
SUITE (  
    ConstructorTest )
```

Тестовый набор для конструктора

Проверяет различные сценарии инициализации ключа

5.4.3.3 SUITE() [2/3]

```
SUITE (  
    DecryptTest )
```

Тестовый набор для метода decrypt.

Проверяет различные сценарии расшифрования

5.4.3.4 SUITE() [3/3]

```
SUITE (
    EncryptTest )
```

Тестовый набор для метода encrypt.

Проверяет различные сценарии шифрования

5.4.3.5 utf8ToWide()

```
wstring utf8ToWide (
    const string & s )
```

Преобразование строки UTF-8 в широкую строку

Аргументы

s	строка в UTF-8
---	----------------

Возвращает

широкая строка

5.4.3.6 wideToUtf8()

```
string wideToUtf8 (
    const wstring & ws )
```

Преобразование широкой строки в UTF-8.

Аргументы

ws	широкая строка
----	----------------

Возвращает

строка в UTF-8

Предметный указатель

- `~KeyBFixture`
 - `KeyBFixture`, [9](#)
- `CHECK_WIDE_EQUAL`
 - `test.cpp`, [20](#)
- `cipher`
 - `KeyBFixture`, [10](#)
- `cipher_error`, [7](#)
 - `cipher_error`, [8](#)
- `decrypt`
 - `modAlphaCipher`, [12](#)
- `Desktop/4л6/task1/main.cpp`, [15](#)
- `Desktop/4л6/task1/modAlphaCipher.cpp`, [17](#)
- `Desktop/4л6/task1/modAlphaCipher.h`, [18](#)
- `Desktop/4л6/task1/test.cpp`, [19](#)
- `encrypt`
 - `modAlphaCipher`, [13](#)
- `KeyBFixture`, [9](#)
 - `~KeyBFixture`, [9](#)
 - `cipher`, [10](#)
 - `KeyBFixture`, [9](#)
- `main`
 - `main.cpp`, [16](#)
 - `test.cpp`, [21](#)
- `main.cpp`
 - `main`, [16](#)
 - `str8_to_w`, [16](#)
 - `w_to_str8`, [17](#)
- `modAlphaCipher`, [10](#)
 - `decrypt`, [12](#)
 - `encrypt`, [13](#)
 - `modAlphaCipher`, [10](#), [11](#)
- `str8_to_w`
 - `main.cpp`, [16](#)
- `SUITE`
 - `test.cpp`, [21](#)
- `test.cpp`
 - `CHECK_WIDE_EQUAL`, [20](#)
 - `main`, [21](#)
 - `SUITE`, [21](#)
 - `utf8ToWide`, [22](#)
 - `wideToUtf8`, [22](#)
- `utf8ToWide`
 - `test.cpp`, [22](#)
- `w_to_str8`
 - `main.cpp`, [17](#)
- `wideToUtf8`
 - `test.cpp`, [22](#)