

Kursovay

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 Client	7
4.1.1 Подробное описание	8
4.1.2 Методы	8
4.1.2.1 Server()	8
4.2 Класс client_error	9
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	10
4.2.2.1 client_error() [1/2]	10
4.2.2.2 client_error() [2/2]	10
5 Файлы	11
5.1 Файл Client.cpp	11
5.1.1 Подробное описание	11
5.1.2 Функции	12
5.1.2.1 file_exists()	12
5.2 Файл Client.h	12
5.2.1 Подробное описание	13
5.3 Файл interface.cpp	14
5.3.1 Подробное описание	15
5.3.2 Функции	15
5.3.2.1 main()	15
5.4 Файл md5.cpp	15
5.4.1 Подробное описание	16
5.4.2 Функции	17
5.4.2.1 MD5_hash()	17
5.5 Файл md5.h	17
5.5.1 Подробное описание	18
5.5.2 Функции	19
5.5.2.1 MD5_hash()	19
Предметный указатель	21



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Client . . . . .	7
invalid_argument	
client_error . . . . .	9



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">Client</a>	Класс клиента с переменными . . . . .	<a href="#">7</a>
<a href="#">client_error</a>	Класс обработки ошибок . . . . .	<a href="#">9</a>





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">Client.cpp</a>	
Модуль клиента . . . . .	11
<a href="#">Client.h</a>	
Заголовочный файл для модуля <a href="#">Client</a> . . . . .	12
<a href="#">interface.cpp</a>	
Интерфейс программы . . . . .	14
<a href="#">md5.cpp</a>	
Модуль хэширования . . . . .	15
<a href="#">md5.h</a>	
Заголовочный файл для модуля MD5 . . . . .	17
<a href="#">termcolor.h</a> . . . . .	??



## Глава 4

# Классы

### 4.1 Класс Client

Класс клиента с переменными.

```
#include <Client.h>
```

Открытые члены

- int `Server` (string str1, string str2)  
Функция клиента для взаимодействия с сервером.

Открытые атрибуты

- int `port`  
Порт
- uint32\_t `summa`  
Сумма в кодировке uint32\_t.
- string `message`  
Сообщение
- string `vectora`  
Переменная для векторов
- string `result`  
Переменная для записи результата
- string `autification`  
Переменная для аутентификации
- string `username`  
Переменная для записи Логина
- string `password`  
Переменная для записи Пароля
- string `line`

### 4.1.1 Подробное описание

Класс клиента с переменными.

Этот класс предоставляет функции для взаимодействия с сервером и включает переменные для различных параметров.

### 4.1.2 Методы

#### 4.1.2.1 Server()

```
int Client::Server (
    string ip_address,
    string port )
```

Функция клиента для взаимодействия с сервером.

Функция клиента для соединения с сервером, аутентификации и обработки данных.

Аргументы

str1	Адрес сервера.
str2	Порт сервера.

Возвращает

int Возвращает 0 в случае успешного выполнения.

Аргументы

ip_address	IP-адрес сервера.
port	Порт сервера.

Возвращает

0 в случае успешного выполнения, 1 в случае ошибки.

Объявления и описания членов классов находятся в файлах:

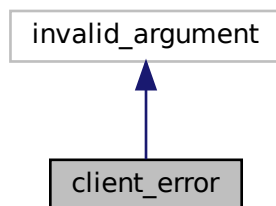
- [Client.h](#)
- [Client.cpp](#)

## 4.2 Класс client\_error

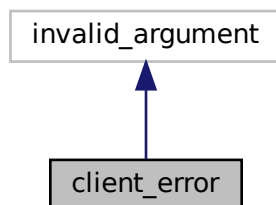
Класс обработки ошибок.

```
#include <Client.h>
```

Граф наследования:client\_error:



Граф связей класса client\_error:



Открытые члены

- `client_error` (`const string &what_arg`)  
Конструктор с аргументом типа `string`.
- `client_error` (`const char *what_arg`)  
Конструктор с аргументом типа `const char*`.

### 4.2.1 Подробное описание

Класс обработки ошибок.

Этот класс используется для обработки ошибок, возникающих в процессе работы клиента.

## 4.2.2 Конструктор(ы)

### 4.2.2.1 client\_error() [1/2]

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

Конструктор с аргументом типа string.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

### 4.2.2.2 client\_error() [2/2]

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

Конструктор с аргументом типа const char\*.

Аргументы

what_arg	Сообщение об ошибке.
----------	----------------------

Объявления и описания членов класса находятся в файле:

- [Client.h](#)

## Глава 5

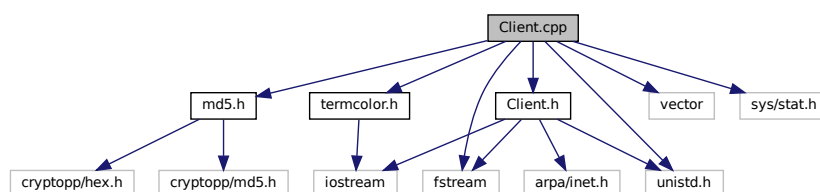
# Файлы

### 5.1 Файл Client.cpp

Модуль клиента

```
#include "md5.h"  
#include "Client.h"  
#include "termcolor.h"  
#include <fstream>  
#include <vector>  
#include <unistd.h>  
#include <sys/stat.h>
```

Граф включаемых заголовочных файлов для Client.cpp:



### Функции

- bool `file_exists` (const string &name)  
Функция для проверки существования файла.

#### 5.1.1 Подробное описание

Модуль клиента

Модуль для подключения клиента к серверу, аутентификации и отправки данных.

Версия

1.0

Дата

05.11.2024

Автор

Мочалин А.А.

Авторство

ИБСТ ПГУ

## 5.1.2 Функции

### 5.1.2.1 file\_exists()

```
bool file_exists (
    const string & name )
```

Функция для проверки существования файла.

Аргументы

name	Имя файла.
------	------------

Возвращает

true, если файл существует, false в противном случае.

## 5.2 Файл Client.h

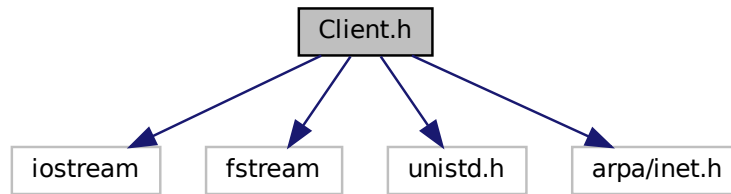
Заголовочный файл для модуля [Client](#).

```
#include <iostream>
#include <fstream>
#include <unistd.h>
```

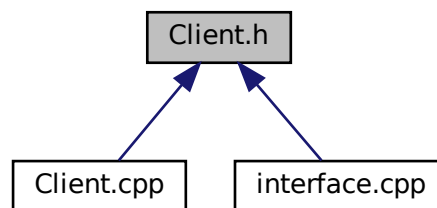


```
#include <arpa/inet.h>
```

Граф включаемых заголовочных файлов для Client.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Client](#)  
Класс клиента с переменными.
- class [client\\_error](#)  
Класс обработки ошибок.

### 5.2.1 Подробное описание

Заголовочный файл для модуля [Client](#).

Этот файл содержит объявления классов и функций для клиента.

Версия

1.0

Дата

05.11.2023

Автор

Мочалин А.А.

Авторство

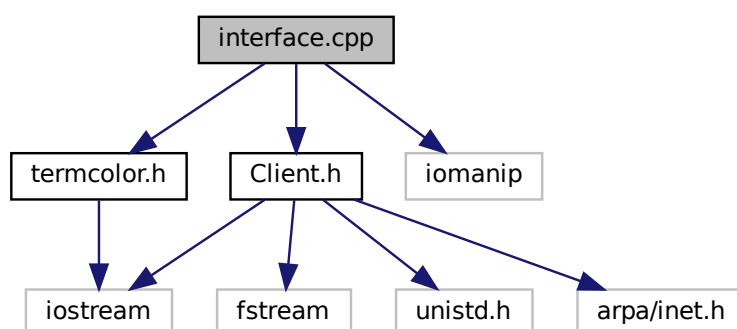
ИБСТ ПГУ

### 5.3 Файл interface.cpp

Интерфейс программы.

```
#include "Client.h"  
#include "termcolor.h"  
#include <iomanip>
```

Граф включаемых заголовочных файлов для interface.cpp:



#### Функции

- void `print_border()`  
Печатает красивую рамку.
- void `print_header()`  
Печатает заголовок программы.
- void `print_help()`  
Печатает справку по использованию программы.
- int `main(int argc, char *argv[])`  
Основная функция интерфейса

### 5.3.1 Подробное описание

Интерфейс программы.

Этот файл содержит код интерфейса для взаимодействия с пользователем.

Версия

2.0

Дата

05.11.2024

Автор

Мочалин А.А.

Авторство

ИБСТ ПГУ

### 5.3.2 Функции

#### 5.3.2.1 main()

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

Основная функция интерфейса

Аргументы

argc	Количество аргументов командной строки
argv	Массив аргументов командной строки

Возвращает

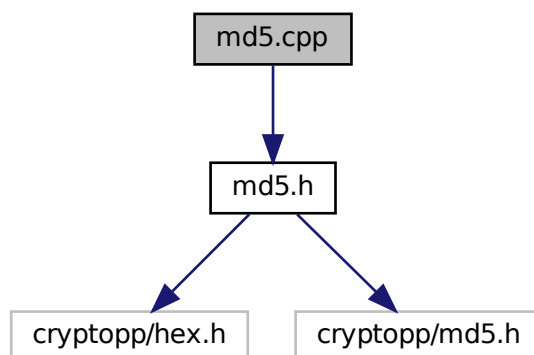
int Возвращает 0 в случае успешного выполнения

## 5.4 Файл md5.cpp

Модуль хэширования.

```
#include "md5.h"
```

Граф включаемых заголовочных файлов для md5.cpp:



## Функции

- string [MD5\\_hash](#) (string hash)  
Вычисление хеша MD5.

### 5.4.1 Подробное описание

Модуль хеширования.

Этот файл содержит определения функций для вычисления хешей MD5.

Версия

1.0

Дата

05.11.2024

Автор

Мочалин А.А.

Авторство

ИБСТ ПГУ

## 5.4.2 Функции

### 5.4.2.1 MD5\_hash()

```
string MD5_hash (  
    string hash )
```

Вычисление хеша MD5.

Эта функция принимает строку и возвращает её MD5-хеш.

Аргументы

hash	Сообщение, получаемое для хэширования.
------	----------------------------------------

Возвращает

string Хешированное сообщение.

Переменная для хранения хешированного сообщения.

Класс для чтения данных из источника и записи их в приемник.

Класс для преобразования двоичных данных в шестнадцатеричную строку.

Класс для записи данных в приемник.

Возвращает

Возвращает хешированное сообщение.

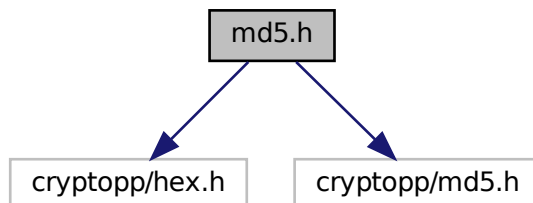
## 5.5 Файл md5.h

Заголовочный файл для модуля MD5.

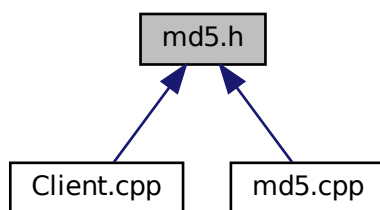
```
#include <cryptopp/hex.h>
```

```
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для md5.h:



Граф файлов, в которые включается этот файл:



## Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`  
Библиотеки Crypto++.

## Функции

- `string MD5_hash (string hash)`  
Вычисление хеша MD5.

### 5.5.1 Подробное описание

Заголовочный файл для модуля MD5.

Этот файл содержит объявления функций для вычисления хешей MD5.

#### Версия

1.0

#### Дата

05.11.2024

#### Автор

Мочалин А.А.

#### Авторство

ИБСТ ПГУ

## 5.5.2 Функции

### 5.5.2.1 MD5\_hash()

```
string MD5_hash (  
    string hash )
```

Вычисление хеша MD5.

Эта функция принимает строку и возвращает её MD5-хеш.

## Аргументы

hash	Входная строка для вычисления хеша.
------	-------------------------------------

## Возвращает

string MD5-хеш входной строки.

Эта функция принимает строку и возвращает её MD5-хеш.

## Аргументы

hash	Сообщение, получаемое для хэширования.
------	----------------------------------------

## Возвращает

string Хешированное сообщение.

Переменная для хранения хешированного сообщения.

Класс для чтения данных из источника и записи их в приемник.

Класс для преобразования двоичных данных в шестнадцатеричную строку.

Класс для записи данных в приемник.

## Возвращает

Возвращает хешированное сообщение.



# Предметный указатель

- Client, [7](#)
  - Server, [8](#)
- Client.cpp, [11](#)
  - file\_exists, [12](#)
- Client.h, [12](#)
- client\_error, [9](#)
  - client\_error, [10](#)
- file\_exists
  - Client.cpp, [12](#)
- interface.cpp, [14](#)
  - main, [15](#)
- main
  - interface.cpp, [15](#)
- md5.cpp, [15](#)
  - MD5\_hash, [17](#)
- md5.h, [17](#)
  - MD5\_hash, [19](#)
- MD5\_hash
  - md5.cpp, [17](#)
  - md5.h, [19](#)
- Server
  - Client, [8](#)