

SERVER

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 Calculator	7
4.1.1 Подробное описание	7
4.1.2 Методы	7
4.1.2.1 math()	7
4.2 Класс ClientBase	8
4.2.1 Подробное описание	8
4.2.2 Методы	8
4.2.2.1 find()	8
4.2.2.2 read()	9
4.3 Класс Communicator	9
4.3.1 Подробное описание	10
4.3.2 Методы	10
4.3.2.1 authorized()	10
4.3.2.2 client_addr()	11
4.3.2.3 self_addr()	11
4.4 Класс LogWriter	11
4.4.1 Подробное описание	12
4.4.2 Методы	12
4.4.2.1 log()	12
4.5 Класс ServerError	13
4.6 Класс User_Interface	13
4.6.1 Подробное описание	14
4.6.2 Конструктор(ы)	14
4.6.2.1 User_Interface()	14
4.6.3 Методы	15
4.6.3.1 operator()()	15
4.6.4 Данные класса	15
4.6.4.1 descr	15
5 Файлы	17
5.1 Файл Calculator.h	17
5.1.1 Подробное описание	18
5.2 Файл client_base.h	18
5.2.1 Подробное описание	19
5.3 Файл Communicator.h	19

5.3.1 Подробное описание . . . . .	20
5.4 Файл mdfile.h . . . . .	20
5.4.1 Подробное описание . . . . .	21
5.5 Файл User_Interface.h . . . . .	22
5.5.1 Подробное описание . . . . .	22
Предметный указатель . . . . .	23

# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

Calculator . . . . .	7
ClientBase . . . . .	8
Communicator . . . . .	9
LogWriter . . . . .	11
std::runtime_error	
ServerError . . . . .	13
User_Interface . . . . .	13



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">Calculator</a>	
Класс <a href="#">Calculator</a> . . . . .	7
<a href="#">ClientBase</a>	
Класс <a href="#">ClientBase</a> . . . . .	8
<a href="#">Communicator</a>	
Класс <a href="#">Communicator</a> . . . . .	9
<a href="#">LogWriter</a>	
Класс <a href="#">LogWriter</a> . . . . .	11
<a href="#">ServerError</a> . . . . .	13
<a href="#">User_Interface</a>	
Класс <a href="#">User_Interface</a> . . . . .	13





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">Calculator.h</a>	
Заголовочный файл для модуля <a href="#">Calculator</a>	17
<a href="#">client_base.h</a>	
Заголовочный файл для модуля <a href="#">client_base</a>	18
<a href="#">Communicator.h</a>	
Заголовочный файл для модуля <a href="#">Communicator</a>	19
<a href="#">mdfile.h</a>	
Заголовочный файл для модуля <a href="#">mdfile</a>	20
<a href="#">User_Interface.h</a>	
Заголовочный файл для модуля <a href="#">User_Interface</a>	22



## Глава 4

# Классы

### 4.1 Класс Calculator

Класс [Calculator](#).

```
#include <Calculator.h>
```

Открытые статические члены

- static void [math](#) (int work\_sock, std::string &err\_way)

Метод `math` для выполнения операции над данными и их отправки

#### 4.1.1 Подробное описание

Класс [Calculator](#).

Для выполнения операции над данными и их отправки предназначен метод `math`

Предупреждения

Реализация только для типа данных `int64_t`

#### 4.1.2 Методы

##### 4.1.2.1 `math()`

```
void Calculator::math (
    int work_sock,
    std::string & err_way ) [static]
```

Метод `math` для выполнения операции над данными и их отправки

Аргументы

in	work_sock	сокет
in	err_way	файл записи логов

Возвращает

Метод ничего не возвращает

Объявления и описания членов классов находятся в файлах:

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

## 4.2 Класс ClientBase

Класс [ClientBase](#).

```
#include <client_base.h>
```

Открытые статические члены

- static std::map< std::string, std::string > [read](#) (std::string way, std::string err\_way)  
Чтение базы данных
- static std::string [find](#) (std::string &name, std::map< std::string, std::string > &all)  
Поиск клиента в базе данных

### 4.2.1 Подробное описание

Класс [ClientBase](#).

Для чтения базы данных предназначен метод read Для поиска клиента в базе данных предназначен метод find

### 4.2.2 Методы

#### 4.2.2.1 find()

```
std::string ClientBase::find (
    std::string & name,
    std::map< std::string, std::string > & all ) [static]
```

Поиск клиента в базе данных

## Аргументы

in	name	login клиента
in	all	словарь с базой данных

## Возвращает

строку с паролем

## Исключения

<a href="#">ServerError</a> ,если	в базе данных отсутствует пользователь
-----------------------------------	--

## 4.2.2.2 read()

```
std::map< std::string, std::string > ClientBase::read (
    std::string way,
    std::string err_way ) [static]
```

## Чтение базы данных

## Аргументы

in	way	путь к базе данных
in	err_way	файл записи логов

## Возвращает

Словарь с парой login,password

## Исключения

<a href="#">ServerError</a> ,если	база данных пустая
-----------------------------------	--------------------

Объявления и описания членов классов находятся в файлах:

- [client\\_base.h](#)
- [client\\_base.cpp](#)

## 4.3 Класс Communicator

Класс [Communicator](#).

```
#include <Communicator.h>
```

## Открытые члены

- `int self_addr (int port, std::string &err_way)`  
Создание сокета сервера
- `int client_addr (int s, std::string &err_way)`  
Создание сокета клиента
- `int authorized (int work_sock, std::string file_name, std::map< std::string, std::string > &all_names, std::string &err_way)`  
Авторизация

### 4.3.1 Подробное описание

Класс `Communicator`.

Для создания сокета сервера и слушания запросов предназначен метод `self_addr`

Для создания сокета клиента и подключения его к серверу метод `client_addr`

Для авторизации клиента предназначен метод `client_addr`

### 4.3.2 Методы

#### 4.3.2.1 `authorized()`

```
int Communicator::authorized (
    int work_sock,
    std::string file_name,
    std::map< std::string, std::string > & all_names,
    std::string & err_way )
```

Авторизация

Аргументы

in	work_sock	сокет клиента
in	file_name	файл записи логов
in	all_names	словарь с базой данных
in	err_way	файл записи логов

Возвращает

Сокет клиента

## 4.3.2.2 client\_addr()

```
int Communicator::client_addr (
    int s,
    std::string & err_way )
```

Создание сокета клиента

Аргументы

in	s	сокет сервера
in	err_way	файл записи логов

Возвращает

Сокет клиента

## 4.3.2.3 self\_addr()

```
int Communicator::self_addr (
    int port,
    std::string & err_way )
```

Создание сокета сервера

Аргументы

in	port	порт
in	err_way	файл записи логов

Возвращает

Сокет

Объявления и описания членов классов находятся в файлах:

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

## 4.4 Класс LogWriter

Класс [LogWriter](#).

```
#include <mdfile.h>
```

## Открытые статические члены

- static void [log](#) (std::string &way, std::string msg)  
Запись ошибок

### 4.4.1 Подробное описание

Класс [LogWriter](#).

Для записи ошибок предназначен метод `log`

### 4.4.2 Методы

#### 4.4.2.1 `log()`

```
void LogWriter::log (  
    std::string & way,  
    std::string msg ) [static]
```

Запись ошибок

Аргументы

in	way	путь к файлу для записи
in	msg	текст ошибки

Возвращает

Метод ничего не возвращает

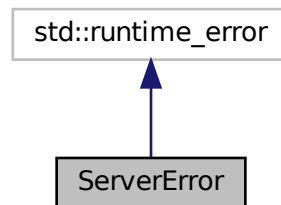
Объявления и описания членов классов находятся в файлах:

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

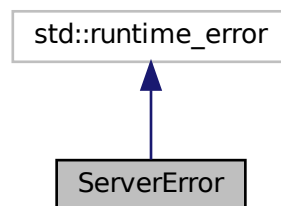


## 4.5 Класс `ServerError`

Граф наследования: `ServerError`:



Граф связей класса `ServerError`:



### Открытые члены

- [ServerError](#) (`const char *msg`)  
Класс [ServerError](#).
- `ServerError` (`const std::string &msg`)

Объявления и описания членов класса находятся в файле:

- [mdfile.h](#)

## 4.6 Класс `User_Interface`

Класс [User\\_Interface](#).

```
#include <User_Interface.h>
```

## Открытые члены

- void Input ()
- [User\\_Interface](#) (int argc, char \*\*argv)  
Метод [User\\_Interface](#) обработки параметров командной строки
- void [operator](#)() ()  
Метод [operator](#) для запуска сервера

## Открытые атрибуты

- std::string file
- std::string LogWrite
- int port
- po::options\_description [descr](#)  
Метод [UI\\_err](#) для обработки ошибок
- po::variables\_map vm
- bool flag = true

### 4.6.1 Подробное описание

Класс [User\\_Interface](#).

Для взаимодействия сервера с командной строкой предназначены методы: [operator](#) для запуска сервера и [User\\_Interface](#) для обработки параметров командной строки и [UI\\_err](#) для обработки ошибок парсера

#### Предупреждения

Все параметры для ввода являются обязательными

### 4.6.2 Конструктор(ы)

#### 4.6.2.1 User\_Interface()

```
User_Interface::User_Interface (
    int argc,
    char ** argv )
```

Метод [User\\_Interface](#) обработки параметров командной строки

#### Аргументы

Параметры	командной строки
-----------	------------------

Возвращает

Метод ничего не возвращает

### 4.6.3 Методы

#### 4.6.3.1 operator()

```
void User_Interface::operator() ( )
```

Метод operator для запуска сервера

Аргументы

Без	параметров
-----	------------

Возвращает

Метод ничего не возвращает

### 4.6.4 Данные класса

#### 4.6.4.1 descr

```
po::options_description User_Interface::descr
```

Метод UI\_err для обработки ошибок

Аргументы

Без	параметров
-----	------------

Возвращает

Метод ничего не возвращает

Исключения

ServerError,если	отсутствует обязательный параметр
------------------	-----------------------------------

Объявления и описания членов классов находятся в файлах:

- [User\\_Interface.h](#)
- User\_Interface.cpp

## Глава 5

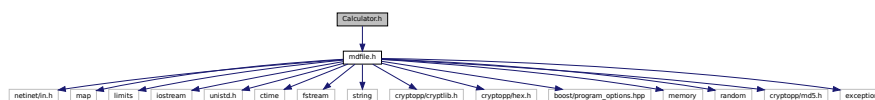
# Файлы

### 5.1 Файл Calculator.h

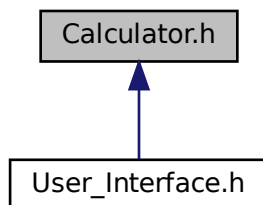
Заголовочный файл для модуля [Calculator](#).

```
#include "mdfile.h"
```

Граф включаемых заголовочных файлов для Calculator.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Calculator](#)  
Класс [Calculator](#).

### 5.1.1 Подробное описание

Заголовочный файл для модуля [Calculator](#).

Автор

Подмарькова В.О.

Версия

1.0

Дата

20.02.2024

Авторство

ИБСТ ПГУ

Предупреждения

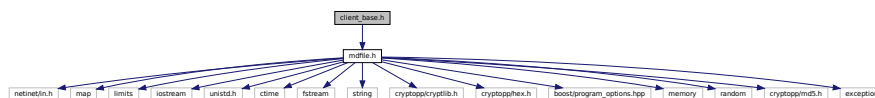
Курсовая работа студента

## 5.2 Файл client\_base.h

Заголовочный файл для модуля client\_base.

```
#include "mdfile.h"
```

Граф включаемых заголовочных файлов для client\_base.h:



Классы

- class [ClientBase](#)  
Класс [ClientBase](#).

### 5.2.1 Подробное описание

Заголовочный файл для модуля `client_base`.

Автор

Подмарькова В.О.

Версия

1.0

Дата

20.02.2024

Авторство

ИБСТ ПГУ

Предупреждения

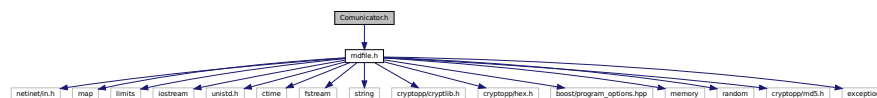
Курсовая работа студента

## 5.3 Файл Communicator.h

Заголовочный файл для модуля `Communicator`.

```
#include "mdfile.h"
```

Граф включаемых заголовочных файлов для `Communicator.h`:



Классы

- class `Communicator`

Класс `Communicator`.

### 5.3.1 Подробное описание

Заголовочный файл для модуля [Communicator](#).

Автор

Подмарькова В.О.

Версия

1.0

Дата

20.02.2024

Авторство

ИБСТ ПГУ

Предупреждения

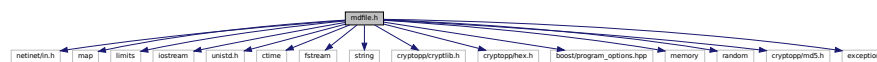
Курсовая работа студента

## 5.4 Файл mdfile.h

Заголовочный файл для модуля mdfile.

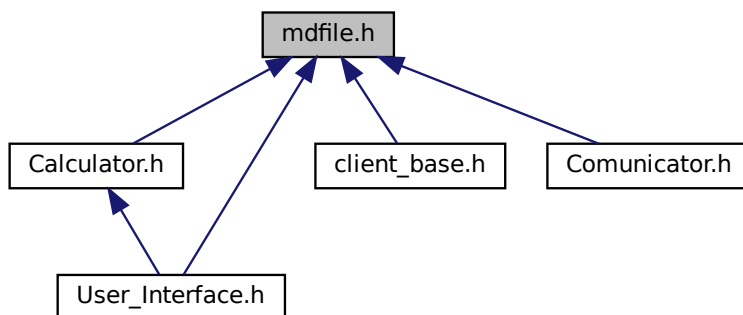
```
#include <netinet/in.h>
#include <map>
#include <limits>
#include <iostream>
#include <unistd.h>
#include <ctime>
#include <fstream>
#include <string>
#include <cryptopp/cryptlib.h>
#include <cryptopp/hex.h>
#include <boost/program_options.hpp>
#include <memory>
#include <random>
#include <cryptopp/md5.h>
#include <exception>
```

Граф включаемых заголовочных файлов для mdfile.h:





Граф файлов, в которые включается этот файл:



## Классы

- class [ServerError](#)
  - class [LogWriter](#)
- Класс [LogWriter](#).

## Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

### 5.4.1 Подробное описание

Заголовочный файл для модуля mdfile.

Автор

Подмарькова В.О.

Версия

1.0

Дата

20.02.2024

Авторство

ИБСТ ПГУ

Предупреждения

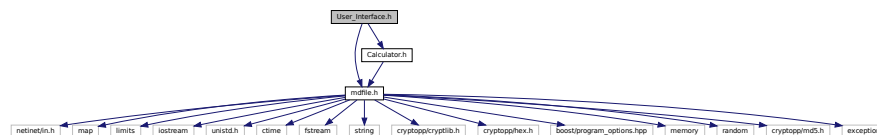
Курсовая работа студента

## 5.5 Файл User\_Interface.h

Заголовочный файл для модуля [User\\_Interface](#).

```
#include "mdfile.h"
#include "Calculator.h"
```

Граф включаемых заголовочных файлов для User\_Interface.h:



### Классы

- class [User\\_Interface](#)  
Класс [User\\_Interface](#).

#### 5.5.1 Подробное описание

Заголовочный файл для модуля [User\\_Interface](#).

Автор

Подмарькова В.О.

Версия

1.0

Дата

20.02.2024

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

# Предметный указатель

- authorized
  - Communicator, [10](#)
- Calculator, [7](#)
  - math, [7](#)
- Calculator.h, [17](#)
- client\_addr
  - Communicator, [10](#)
- client\_base.h, [18](#)
- ClientBase, [8](#)
  - find, [8](#)
  - read, [9](#)
- Communicator, [9](#)
  - authorized, [10](#)
  - client\_addr, [10](#)
  - self\_addr, [11](#)
- Communicator.h, [19](#)
- descr
  - User\_Interface, [15](#)
- find
  - ClientBase, [8](#)
- log
  - LogWriter, [12](#)
- LogWriter, [11](#)
  - log, [12](#)
- math
  - Calculator, [7](#)
- mdfile.h, [20](#)
- operator()
  - User\_Interface, [15](#)
- read
  - ClientBase, [9](#)
- self\_addr
  - Communicator, [11](#)
- ServerError, [13](#)
- User\_Interface, [13](#)
  - descr, [15](#)
  - operator(), [15](#)
  - User\_Interface, [14](#)
- User\_Interface.h, [22](#)