

Server TCP

1

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	12
4.5.1 Подробное описание	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
4.6.3.2 UI_err()	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	12
User_Interface	13

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Calculator		
Класс Calculator	7
ClientBase		
Класс ClientBase	8
Communicator		
Класс Communicator	9
LogWriter		
Класс LogWriter	11
ServerError		
Класс ServerError	12
User_Interface		
Класс User_Interface	13

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Calculator.h	
Заголовочный файл для модуля Calculator	17
client_base.h	
Заголовочный файл для модуля client_base	18
Communicator.h	
Заголовочный файл для модуля Communicator	19
mdfile.h	
Заголовочный файл для модуля mdfile	20
User_Interface.h	
Заголовочный файл для модуля User_Interface	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 `uint32_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

ServerError ,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

ServerError ,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	<code>work_sock</code>	сокет клиента
in	<code>file_name</code>	файл записи логов
in	<code>all_names</code>	словарь с базой данных
in	<code>err_way</code>	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:

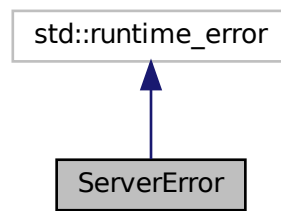
- [mdfile.h](#)
- [mdfile.cpp](#)

4.5 Класс `ServerError`

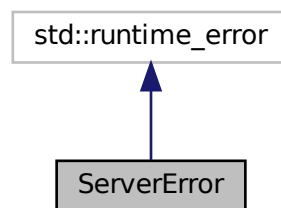
Класс [ServerError](#).

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


Граф наследования:ServerError:



Граф связей класса ServerError:



Открытые члены

- ServerError (const char *msg)
- ServerError (const std::string &msg)

4.5.1 Подробное описание

Класс [ServerError](#).

Объявления и описания членов класса находятся в файле:

- [mdfile.h](#)

4.6 Класс User_Interface

Класс [User_Interface](#).

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

Открытые члены

- `User_Interface` (int argc, char **argv)
Метод `User_Interface` обработки параметров командной строки
- void `operator()` ()
Метод `operator` для запуска сервера

Закрытые члены

- void `UI_err` ()
Метод `UI_err` для обработки ошибок

Закрытые данные

- `po::options_description` `descr`
описание параметров командной строки
- `po::variables_map` `vm`
значения параметров командной строки
- `std::string` `file`
путь к файлу с базой данных
- `std::string` `LogWrite`
путь к журналом работы
- int `port`
номер порта

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

4.6.3.2 UI_err()

```
void User_Interface::UI_err ( ) [private]
```

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

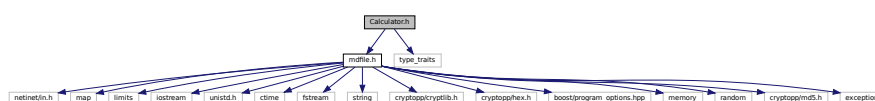
Файлы

5.1 Файл Calculator.h

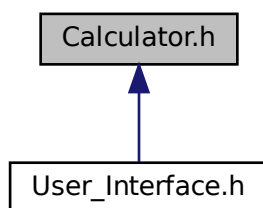
Заголовочный файл для модуля [Calculator](#).

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

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

Елинова А.С

Версия

1.0

Дата

27.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

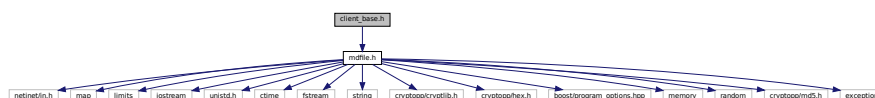
Курсовая работа студента

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

Елинова А.С

Версия

1.0

Дата

27.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

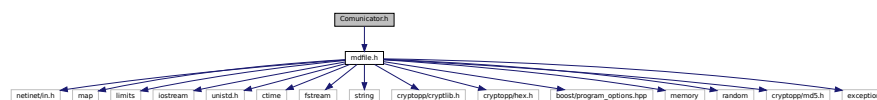
Курсовая работа студента

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

Елинова А.С

Версия

1.0

Дата

27.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

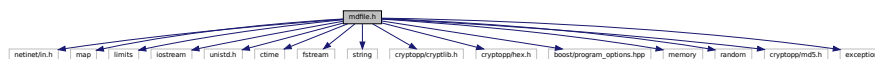
Курсовая работа студента

5.4 Файл mdfile.h

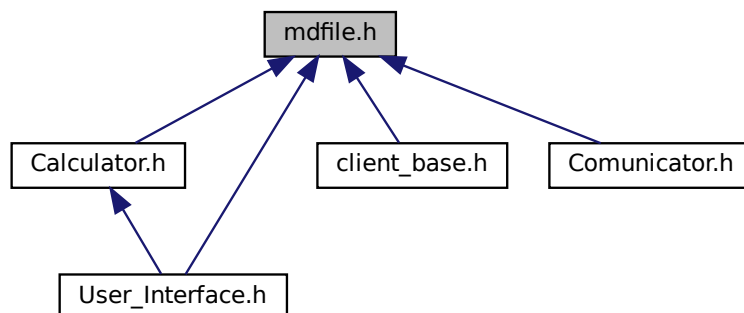
Заголовочный файл для модуля mdfile.

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

Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

5.4.1 Подробное описание

Заголовочный файл для модуля mdfile.

Автор

Елинова А.С

Версия

1.0

Дата

27.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

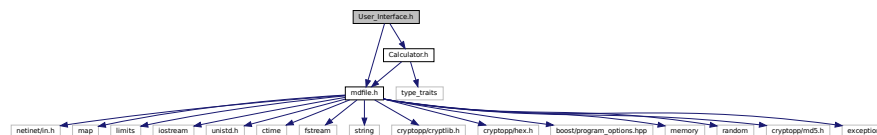
Курсовая работа студента

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

Елинова А.С

Версия

1.0

Дата

27.12.2023

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](#)
- 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, [12](#)
- UI_err
 - User_Interface, [15](#)
- User_Interface, [13](#)
 - operator(), [15](#)
 - UI_err, [15](#)
 - User_Interface, [14](#)
- User_Interface.h, [22](#)