

Курсовая работа Метальников М.А. гр. 22ПИ2

1

Создано системой Doxygen 1.9.1



1	Алфавитный указатель классов	1
1.1	Классы	1
2	Список файлов	3
2.1	Файлы	3
3	Классы	5
3.1	Класс <code>ErrorLog</code>	5
3.1.1	Подробное описание	5
3.1.2	Методы	5
3.1.2.1	<code>logError()</code>	5
3.2	Класс <code>Server</code>	6
3.2.1	Подробное описание	7
3.2.2	Конструктор(ы)	7
3.2.2.1	<code>Server()</code>	7
3.2.2.2	<code>~Server()</code>	7
3.2.3	Методы	7
3.2.3.1	<code>authentication()</code>	7
3.2.3.2	<code>get_base()</code>	8
3.2.3.3	<code>operator()()</code>	8
3.2.3.4	<code>quad()</code>	8
3.2.3.5	<code>startListening()</code>	9
3.2.3.6	<code>str_read()</code>	9
3.2.4	Данные класса	10
3.2.4.1	<code>db_file</code>	10
3.2.4.2	<code>foreign_addr</code>	10
3.2.4.3	<code>login</code>	10
3.2.4.4	<code>password</code>	10
3.2.4.5	<code>queueLen</code>	10
3.2.4.6	<code>self_addr</code>	10
3.2.4.7	<code>sock</code>	10
3.2.4.8	<code>userCredentials</code>	10
3.2.4.9	<code>work_sock</code>	10
4	Файлы	11
4.1	Файл <code>ErrorLog.cpp</code>	11
4.1.1	Подробное описание	11
4.2	Файл <code>ErrorLog.h</code>	12
4.2.1	Подробное описание	12
4.3	Файл <code>main.cpp</code>	13
4.3.1	Подробное описание	13
4.3.2	Функции	14
4.3.2.1	<code>main()</code>	14
4.4	Файл <code>Server.cpp</code>	14

4.4.1 Подробное описание . . . . .	15
4.5 Файл Server.h . . . . .	15
4.5.1 Подробное описание . . . . .	16
Предметный указатель	17

# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">ErrorLog</a>	Модуль для работы с логированием ошибок . . . . .	<a href="#">5</a>
<a href="#">Server</a>	Модуль для прослушки сокета, чтения БД, аутентификации и вычислений . . . .	<a href="#">6</a>



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">ErrorLog.cpp</a>	Реализация модуля обработки ошибок для сервера . . . . .	11
<a href="#">ErrorLog.h</a>	Модуль обработки ошибок для сервера . . . . .	12
<a href="#">main.cpp</a>	Главный модуль сервера . . . . .	13
<a href="#">Server.cpp</a>	Реализация модуля для прослушки сокета, чтения БД, аутентификации и вычислений . . . . .	14
<a href="#">Server.h</a>	Модуль для прослушки сокета, чтения БД, аутентификации и вычислений . . . .	15





# Глава 3

## Классы

### 3.1 Класс ErrorLog

Модуль для работы с логированием ошибок.

```
#include <ErrorLog.h>
```

Открытые статические члены

- static void [logError](#) (bool critical, const std::string &errorMessage)  
Записывает ошибку в лог-файл.

#### 3.1.1 Подробное описание

Модуль для работы с логированием ошибок.

#### 3.1.2 Методы

##### 3.1.2.1 logError()

```
void ErrorLog::logError (  
    bool critical,  
    const std::string & errorMessage ) [static]
```

Записывает ошибку в лог-файл.

Модуль записи ошибки в лог файл.

Аргументы

critical	Флаг, указывающий на критичность ошибки.
errorMessage	Сообщение об ошибке.

Объявления и описания членов классов находятся в файлах:

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

## 3.2 Класс Server

Модуль для прослушки сокета, чтения БД, аутентификации и вычислений.

```
#include <Server.h>
```

### Открытые члены

- [Server](#) (unsigned short port, int qlen, const std::string &db)  
Конструктор сервера.
- void [connecting](#) ()  
Метод для установления соединения.
- void [get\\_base](#) (const std::string &db\_file)  
Метод для получения данных из базы.
- bool [authentication](#) (int [work\\_sock](#), const std::unordered\_map< std::string, std::string > &credentials)  
Метод аутентификации пользователя.
- float [quad](#) (int [work\\_sock](#))  
Метод для выполнения операции квадрата числа.
- std::string [str\\_read](#) (int [work\\_sock](#))  
Метод для чтения строки из сокета.
- [~Server](#) ()  
Деструктор сервера.
- void [startListening](#) ([Server](#) &server)  
Метод для запуска прослушивания подключений.

### Открытые атрибуты

- std::unordered\_map< std::string, std::string > [userCredentials](#)

### Закрытые члены

- void [operator\(\)](#) (int [sock](#))  
Оператор () для обработки подключений.

### Закрытые данные

- std::string [db\\_file](#)
- int [work\\_sock](#)
- int [sock](#)
- std::string [login](#)
- std::string [password](#)
- std::unique\_ptr< sockaddr\_in > [self\\_addr](#)
- std::unique\_ptr< sockaddr\_in > [foreign\\_addr](#)
- int [queueLen](#)

### 3.2.1 Подробное описание

Модуль для прослушки сокета, чтения БД, аутентификации и вычислений.

### 3.2.2 Конструктор(ы)

#### 3.2.2.1 Server()

```
Server::Server (
    unsigned short port,
    int qlen,
    const std::string & db )
```

Конструктор сервера.

Аргументы

port	Порт, на котором будет работать сервер.
qlen	Длина очереди подключений.
db	Путь к файлу базы данных.
port	Порт, на котором будет работать сервер.
qlen	Длина очереди.
db	Путь к файлу базы данных.

#### 3.2.2.2 ~Server()

```
Server::~Server ( )
```

Деструктор сервера.

Метод для закрытия сокета и освобождения ресурсов.

### 3.2.3 Методы

#### 3.2.3.1 authentication()

```
bool Server::authentication (
    int work_sock,
    const std::unordered_map< std::string, std::string > & credentials )
```

Метод аутентификации пользователя.

Аргументы

work_sock	Сокет для работы с пользователем.
credentials	Список учетных данных пользователя.

Возвращает

Возвращает true, если аутентификация прошла успешно.

### 3.2.3.2 get\_base()

```
void Server::get_base (
    const std::string & db_file )
```

Метод для получения данных из базы.

Аргументы

db_file	Путь к файлу базы данных.
---------	---------------------------

### 3.2.3.3 operator()()

```
void Server::operator() (
    int work_sock ) [private]
```

Оператор () для обработки подключений.

Аргументы

sock	Сокет для работы с пользователем.
work_sock	Сокет для работы с пользователем.

### 3.2.3.4 quad()

```
float Server::quad (
    int work_sock )
```

Метод для выполнения операции квадрата числа.

Метод для выполнения операции сумма квадратов значений вектора.

Аргументы

work_sock	Сокет для работы с пользователем.
-----------	-----------------------------------

Возвращает

Возвращает результат операции квадрата числа.

Аргументы

work_sock	Сокет для работы с пользователем.
-----------	-----------------------------------

Возвращает

Возвращает результат операции суммы квадратов значений вектора.

### 3.2.3.5 startListening()

```
void Server::startListening (  
    Server & server )
```

Метод для запуска прослушивания подключений.

Аргументы

server	Сервер, который будет прослушивать подключения.
server	ссылка на объект класса, который будет прослушивать подключения.

### 3.2.3.6 str\_read()

```
std::string Server::str_read (  
    int work_sock )
```

Метод для чтения строки из сокета.

Аргументы

work_sock	Сокет для работы с пользователем.
-----------	-----------------------------------

Возвращает

Возвращает строку, прочитанную из сокета.

### 3.2.4 Данные класса

#### 3.2.4.1 db\_file

`std::string Server::db_file [private]`

Путь к файлу базы данных.

#### 3.2.4.2 foreign\_addr

`std::unique_ptr<sockaddr_in> Server::foreign_addr [private]`

Указатель на структуру с адресом клиента.

#### 3.2.4.3 login

`std::string Server::login [private]`

Логин пользователя.

#### 3.2.4.4 password

`std::string Server::password [private]`

Пароль пользователя.

#### 3.2.4.5 queueLen

`int Server::queueLen [private]`

Длина очереди подключений.

#### 3.2.4.6 self\_addr

`std::unique_ptr<sockaddr_in> Server::self_addr [private]`

Указатель на структуру с адресом сервера.

#### 3.2.4.7 sock

`int Server::sock [private]`

Сокет сервера.

#### 3.2.4.8 userCredentials

`std::unordered_map<std::string, std::string> Server::userCredentials`

Список учетных данных пользователей.

#### 3.2.4.9 work\_sock

`int Server::work_sock [private]`

Сокет для работы с пользователем.

Объявления и описания членов классов находятся в файлах:

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

## Глава 4

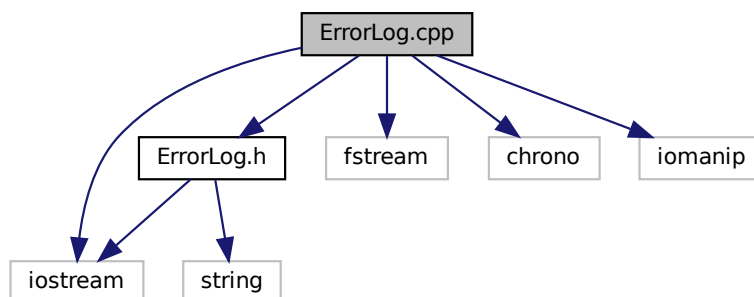
# Файлы

### 4.1 Файл ErrorLog.cpp

Реализация модуля обработки ошибок для сервера

```
#include "ErrorLog.h"  
#include <iostream>  
#include <fstream>  
#include <chrono>  
#include <iomanip>
```

Граф включаемых заголовочных файлов для ErrorLog.cpp:



#### 4.1.1 Подробное описание

Реализация модуля обработки ошибок для сервера

Автор

Метальников М.А.

Версия

1.0

Дата

22.12.2023

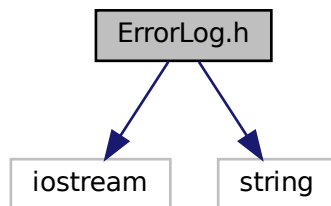
## 4.2 Файл ErrorLog.h

Модуль обработки ошибок для сервера

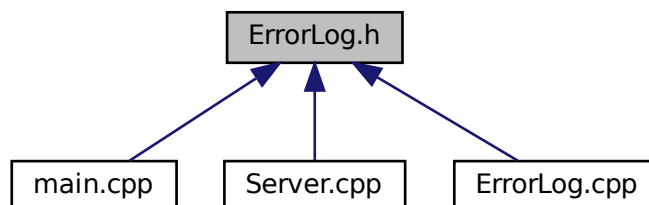
```
#include <iostream>
```

```
#include <string>
```

Граф включаемых заголовочных файлов для ErrorLog.h:



Граф файлов, в которые включается этот файл:



### Классы

- class [ErrorLog](#)

Модуль для работы с логированием ошибок.

#### 4.2.1 Подробное описание

Модуль обработки ошибок для сервера

Автор

Метальников М.А.



Версия

1.0

Дата

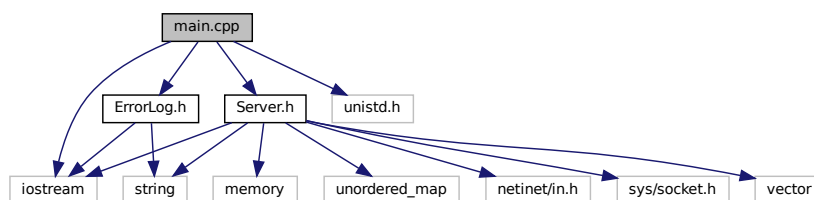
22.12.2023

## 4.3 Файл main.cpp

Главный модуль сервера

```
#include <iostream>
#include <unistd.h>
#include "Server.h"
#include "ErrorLog.h"
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

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

Главный модуль сервера.

### 4.3.1 Подробное описание

Главный модуль сервера

Автор

Метальников М.А.

Версия

1.0

Дата

22.12.2023

### 4.3.2 Функции

#### 4.3.2.1 main()

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

Главный модуль сервера.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив аргументов командной строки.

Возвращает

Возвращает 0 в случае успешного завершения.

< Порт по умолчанию

< Длина очереди по умолчанию

< Путь к файлу базы данных.

## 4.4 Файл Server.cpp

Раулизация модуля для прослушки сокета, чтения БД, аутентификации и вычислений

```
#include "ErrorLog.h"
#include "Server.h"
#include <iostream>
#include <unistd.h>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <iomanip>
#include <cstring>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <limits>
#include <random>
#include <cryptopp/md5.h>
#include <cryptopp/hex.h>
#include <sys/ioctl.h>
#include <cryptopp/filters.h>
```

```
#include <unordered_map>
```

Граф включаемых заголовочных файлов для Server.cpp:



## Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

#### 4.4.1 Подробное описание

## Реализация модуля для прослушки сокета, чтения БД, аутентификации и вычислений

Автор

Метальников М.А.

Версия

1.0

Data

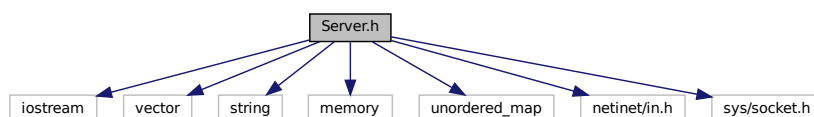
22.12.2023

## 4.5 Файл Server.h

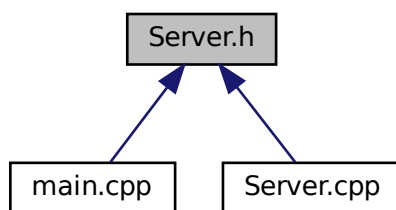
# Модуль для прослушки сокета, чтения БД, аутентификации и вычислений

```
#include <iostream>
#include <vector>
#include <string>
#include <memory>
#include <unordered_map>
#include <netinet/in.h>
#include <sys/socket.h>
```

Граф включаемых заголовочных файлов для Server.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Server](#)

Модуль для прослушки сокета, чтения БД, аутентификации и вычислений.

## Макросы

- `#define BUFLLEN 1024`

### 4.5.1 Подробное описание

Модуль для прослушки сокета, чтения БД, аутентификации и вычислений

Автор

Метальников М.А.

Версия

1.0

Дата

22.12.2023

# Предметный указатель

- ~Server
  - Server, [7](#)
- authentication
  - Server, [7](#)
- db\_file
  - Server, [10](#)
- ErrorLog, [5](#)
  - logError, [5](#)
- ErrorLog.cpp, [11](#)
- ErrorLog.h, [12](#)
- foreign\_addr
  - Server, [10](#)
- get\_base
  - Server, [8](#)
- logError
  - ErrorLog, [5](#)
- login
  - Server, [10](#)
- main
  - main.cpp, [14](#)
- main.cpp, [13](#)
  - main, [14](#)
- operator()
  - Server, [8](#)
- password
  - Server, [10](#)
- quad
  - Server, [8](#)
- queueLen
  - Server, [10](#)
- self\_addr
  - Server, [10](#)
- Server, [6](#)
  - ~Server, [7](#)
  - authentication, [7](#)
  - db\_file, [10](#)
  - foreign\_addr, [10](#)
  - get\_base, [8](#)
  - login, [10](#)
  - operator(), [8](#)
  - password, [10](#)
  - quad, [8](#)
  - queueLen, [10](#)
  - self\_addr, [10](#)
  - Server, [7](#)
  - sock, [10](#)
  - startListening, [9](#)
  - str\_read, [9](#)
  - userCredentials, [10](#)
  - work\_sock, [10](#)
- Server.cpp, [14](#)
- Server.h, [15](#)
- sock
  - Server, [10](#)
- startListening
  - Server, [9](#)
- str\_read
  - Server, [9](#)
- userCredentials
  - Server, [10](#)
- work\_sock
  - Server, [10](#)