

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 ErrorLog	5
3.1.1 Подробное описание	5
3.1.2 Методы	5
3.1.2.1 logError()	5
3.1.3 Данные класса	6
3.1.3.1 LogFile	6
3.2 Класс Server	6
3.2.1 Подробное описание	7
3.2.2 Конструктор(ы)	7
3.2.2.1 Server()	7
3.2.2.2 ~Server()	8
3.2.3 Методы	8
3.2.3.1 authentication()	8
3.2.3.2 get_base()	8
3.2.3.3 operator()()	8
3.2.3.4 quad()	9
3.2.3.5 startListening()	9
3.2.3.6 str_read()	10
3.2.4 Данные класса	10
3.2.4.1 db_file	10
3.2.4.2 foreign_addr	10
3.2.4.3 login	10
3.2.4.4 password	10
3.2.4.5 queueLen	11
3.2.4.6 self_addr	11
3.2.4.7 sock	11
3.2.4.8 userCredentials	11
3.2.4.9 work_sock	11
4 Файлы	13
4.1 Файл ErrorLog.cpp	13
4.1.1 Подробное описание	13
4.2 Файл ErrorLog.h	14
4.2.1 Подробное описание	14
4.3 Файл main.cpp	15
4.3.1 Подробное описание	15
4.3.2 Функции	16

4.3.2.1 <code>main()</code>	16
4.4 Файл <code>Server.cpp</code>	16
4.4.1 Подробное описание	17
4.5 Файл <code>Server.h</code>	17
4.5.1 Подробное описание	18
Предметный указатель	19

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

ErrorLog	Модуль для работы с логированием ошибок	5
Server	Модуль для прослушки сокета, чтения БД, аутентификации и вычислений	6

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

ErrorLog.cpp	Реализация модуля обработки ошибок для сервера	13
ErrorLog.h	Модуль обработки ошибок для сервера	14
main.cpp	Главный модуль сервера	15
Server.cpp	Реализация модуля для прослушки сокета, чтения БД, аутентификации и вычислений	16
Server.h	Модуль для прослушки сокета, чтения БД, аутентификации и вычислений	17

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.

Статические открытые данные

- static std::string [LogFile](#) = "default_log_file.txt"
Модуль записи ошибки 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.

Аргументы

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

3.1.3 Данные класса

3.1.3.1 LogFile

```
std::string ErrorLog::LogFile = "default_log_file.txt" [static]
```

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

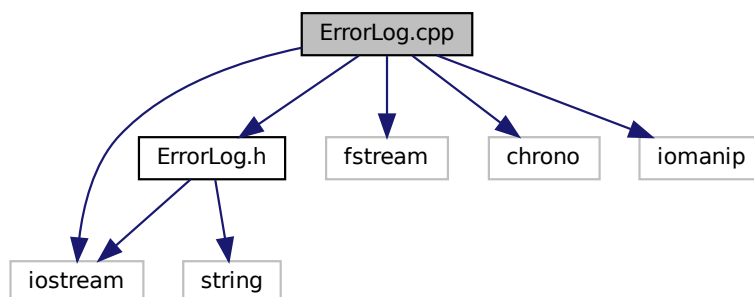
Файлы

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

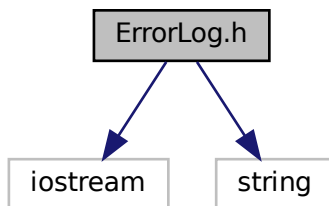
4.2 Файл ErrorLog.h

Модуль обработки ошибок для сервера

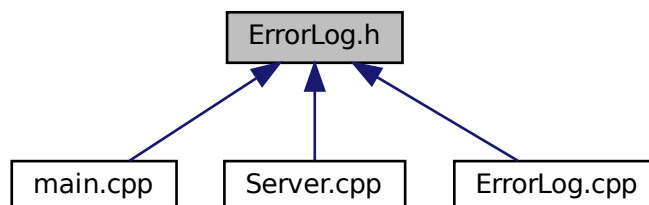
```
#include <iostream>
```

```
#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

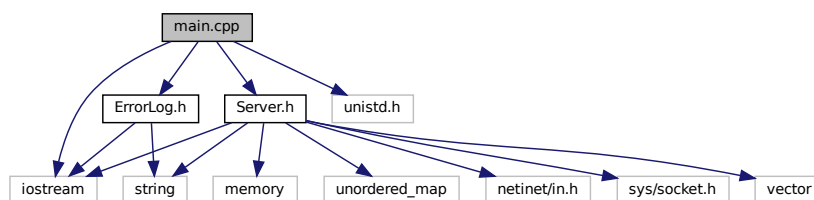
22.12.2023

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

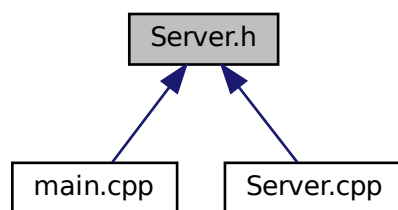
< Путь к файлу базы данных.

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>
```


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, [8](#)
- db_file
 - Server, [10](#)
- ErrorLog, [5](#)
 - logError, [5](#)
 - LogFile, [6](#)
- ErrorLog.cpp, [13](#)
- ErrorLog.h, [14](#)
- foreign_addr
 - Server, [10](#)
- get_base
 - Server, [8](#)
- logError
 - ErrorLog, [5](#)
- LogFile
 - ErrorLog, [6](#)
- login
 - Server, [10](#)
- main
 - main.cpp, [16](#)
 - main.cpp, [15](#)
 - main, [16](#)
- operator()
 - Server, [8](#)
- password
 - Server, [10](#)
- quad
 - Server, [9](#)
- queueLen
 - Server, [10](#)
- self_addr
 - Server, [11](#)
- Server, [6](#)
 - ~Server, [7](#)
 - authentication, [8](#)
 - db_file, [10](#)
 - foreign_addr, [10](#)
 - get_base, [8](#)
 - login, [10](#)
 - operator(), [8](#)
 - password, [10](#)
 - quad, [9](#)
 - queueLen, [10](#)
 - self_addr, [11](#)
 - Server, [7](#)
 - sock, [11](#)
 - startListening, [9](#)
 - str_read, [10](#)
 - userCredentials, [11](#)
 - work_sock, [11](#)
- Server.cpp, [16](#)
- Server.h, [17](#)
- sock
 - Server, [11](#)
- startListening
 - Server, [9](#)
- str_read
 - Server, [10](#)
- userCredentials
 - Server, [11](#)
- work_sock
 - Server, [11](#)