Курсовая работа на тему Программная реализация сетевого сервера

Создано системой Doxygen 1.9.4

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс CmdLine

Открытые члены

- void addParam (std::string name, std::string shortName, std::string description=std::string(), param_type type=param_type::string)

  Добавляет параметр к объекту CmdLine.
- void parse (int argc, char **argv)

  Анализирует аргументы командной строки.
- void printHelp ()

  Выводит справочное сообщение.
- int getInt (std::string name)

  Возвращает целочисленное значение указанного параметра.
- std::string getString (std::string name)

  Возвращает строковое значение указанного параметра.
- bool isHasHelp ()

  Проверяет, присутствует ли опция «–help» или «-h».

4.1.1 Методы

4.1.1.1 addParam()

```
void CmdLine::addParam (
                std::string name,
                std::string shortName,
                std::string description = std::string(),
                param_type type = param_type::string )
```

Добавляет параметр к объекту CmdLine.

Аргументы

name	Имя параметра.
shortName	Краткое имя (один символ) параметра.
описание	Описание параметра.
type	Тип параметра.

4.1.1.2 getInt()

```
int CmdLine::getInt (
            std::string name )
```

Возвращает целочисленное значение указанного параметра.

Аргументы

name	Имя параметра.

Возвращает

Целочисленное значение параметра.

4.1.1.3 getString()

```
std::string CmdLine::getString (
            std::string name )
```

Возвращает строковое значение указанного параметра.

Аргументы

name	Имя параметра.

Возвращает

Строковое значение параметра.

4.1.1.4 isHasHelp()

```
bool CmdLine::isHasHelp ( )  [inline]
```

Проверяет, присутствует ли опция «–help» или «-h».

Возвращает

True, если присутствует опция справки, в противном случае — false.

4.1.1.5 parse()

```
void CmdLine::parse (
                int argc,
                char ** argv )
```

Анализирует аргументы командной строки.

Аргументы

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

Объявления и описания членов класса находятся в файле:

- CmdLine.h

4.2 Класс Communication

Класс Communication.

#include <Communication.h>

Открытые члены

- void createSocket (int port)
  Создает сокет на указанном порту.
- int waitClient ()
  Ожидает подключения клиента и принимает его
- void autorize (int clientSocket, UsersParser users)
  Авторизует клиента, сравнивая его учетные данные с информацией пользователя.

4.2.1 Подробное описание

Класс Communication.

Класс Communication отвечает за связь между клиентами и сервером.

Предупреждения

Реализация только для типа данных int16_t

4.2.2 Методы

4.2.2.1 autorize()

```
void Communication::autorize (
            int clientSocket,
            UsersParser users )
```

Авторизует клиента, сравнивая его учетные данные с информацией пользователя.

Аргументы

clientSocket	Дескриптор сокета клиента. @paramusers Экземпляр класса UsersParser, хранящий информацию о пользователях.

4.2.2.2 createSocket()

```
void Communication::createSocket (
            int port )
```

Создает сокет на указанном порту.

Аргументы

port	Номер порта, на котором будет создан сокет.

4.2.2.3 waitClient()

```
int Communication::waitClient ( )
```

Ожидает подключения клиента и принимает его

Возвращает

Дескриптор клиентского сокета.

Объявления и описания членов класса находятся в файле:

- Communication.h

4.3 Класс Logger

Класс Logger.

#include <Logger.h>

Открытые члены

- bool Initialize (std::string logPath)

  Инициализирует регистратор с указанным путем к журналу.
- void info (std::string str)

  Регистрирует информационное сообщение.
- void serverError (ServerError serverError)

  Регистрирует ошибку сервера.

Открытые статические члены

- static Logger & get ()

  Возвращает ссылку на одноэлементный экземпляр средства ведения журнала.
- static Logger * getPtr ()

  Возвращает указатель на одноэлементный экземпляр средства ведения журнала.

4.3.1 Подробное описание

Класс Logger.

Класс Logger отвечает за регистрацию информации и ошибок сервера.

Предупреждения

Реализация только для типа данных int16_t

4.3.2 Методы

4.3.2.1 get()

static Logger & Logger::get ( ) [static]

Возвращает ссылку на одноэлементный экземпляр средства ведения журнала.

Возвращает

Ссылка на экземпляр регистратора

4.3.2.2 getPtr()

static Logger * Logger::getPtr ( ) [static]

Возвращает указатель на одноэлементный экземпляр средства ведения журнала.

Возвращает

Указатель на экземпляр регистратора

4.3.2.3 info()

void Logger::info (
std::string str)

Регистрирует информационное сообщение.

Аргументы

str	Информационное сообщение, которое необходимо записать в журнал.

4.3.2.4 Initialize()

bool Logger::Initialize (
std::string logPath)

Инициализирует регистратор с указанным путем к журналу.

Аргументы

logPath	Путь к файлу журнала.

Возвращает

True, если регистратор успешно инициализирован, в противном случае — false.

4.3.2.5 serverError()

void Logger::serverError (
ServerError serverError)

Регистрирует ошибку сервера.

Аргументы

serverError	Ошибка сервера, которая должна быть зарегистрирована.

Объявления и описания членов класса находятся в файле:

- Logger.h

4.4 Класс Server

Класс Server.

#include <Server.h>

Открытые члены

- bool Initialize (int argc, char **argv)
  Инициализация сервера
- void start ()
  Запуск сервера

4.4.1 Подробное описание

Класс Server.

Класс Server отвечает за инициализацию и запуск сервера.

Предупреждения

Реализация только для типа данных int16_t

4.4.2 Методы

4.4.2.1 Initialize()

```
bool Server::Initialize (
                int argc,
                char ** argv )
```

Инициализация сервера

Аргументы

argc	Количество аргументов командной строки
argv	Массив с аргументами командной строки

Возвращает

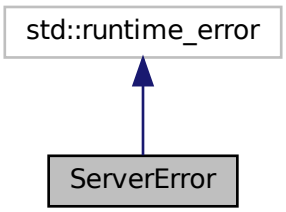
true, если инициализация прошла успешно, false в противном случае

Объявления и описания членов класса находятся в файле:

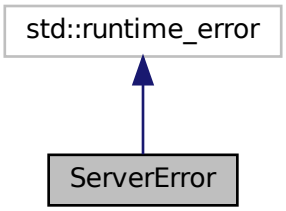
- Server.h

4.5 Класс ServerError

Граф наследования:ServerError:



Граф связей класса ServerError:



Открытые члены

- ServerError (const char *msg)

  Конструктор с сообщением об ошибке.
- ServerError (const std::string &msg, ErrorLevel level=ErrorLevel::Warning)

  Конструктор с сообщением об ошибке и уровнем ошибки.
- ErrorLevel getErrorLevel ()

  Получаем уровень ошибки.

4.5.1 Конструктор(ы)

4.5.1.1 ServerError() [1/2]

ServerError::ServerError (
const char * msg) [inline]

Конструктор с сообщением об ошибке.

Аргументы

msg	Сообщение об ошибке.

4.5.1.2 ServerError() [2/2]

ServerError::ServerError (
const std::string & msg,
ErrorLevel level = ErrorLevel::Warning) [inline]

Конструктор с сообщением об ошибке и уровнем ошибки.

Аргументы

msg	Сообщение об ошибке.
level	Уровень ошибки.

4.5.2 Методы

4.5.2.1 getErrorLevel()

ErrorLevel ServerError::getErrorLevel ( ) [inline]

Получаем уровень ошибки.

Возвращает

Уровень ошибки.

Объявления и описания членов класса находятся в файле:

- ServerError.h

4.6 Класс UsersParser

Класс UsersParser.

#include <UsersParser.h>

Открытые члены

- void parse (std::string pathToFile)
  Разберите файл и сохраните информацию о пользователе в хеш-карте.
- bool isHasUser (std::string username)
  Проверить, существует ли пользователь с указанным именем пользователя
- std::string getPassword (std::string username)
  Получить пароль пользователя с указанным именем пользователя

4.6.1 Подробное описание

Класс UsersParser.

Класс UsersParser анализирует файл, содержащий информацию о пользователе.

Предупреждения

Реализация только для типа данных int16_t

4.6.2 Методы

4.6.2.1 getPassword()

```
std::string UsersParser::getPassword (
            std::string username )
```

Получить пароль пользователя с указанным именем пользователя

Аргументы

username	Имя пользователя.

Возвращает

Пароль пользователя

4.6.2.2 isHasUser()

```
bool UsersParser::isHasUser (
            std::string username )
```

Проверить, существует ли пользователь с указанным именем пользователя

Аргументы

username	Имя пользователя, которого нужно проверить.

Возвращает

true, если пользователь существует, в противном случае — false

4.6.2.3 parse()

```
void UsersParser::parse (
            std::string pathToFile )
```

Разберите файл и сохраните информацию о пользователе в хеш-карте.

Аргументы

pathToFile	Путь к файлу, который нужно проанализировать.

Объявления и описания членов класса находятся в файле:

- UsersParser.h

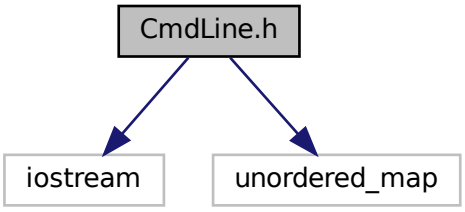
Глава 5

Файлы

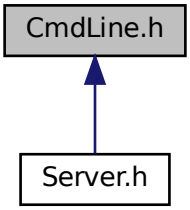
5.1 Файл CmdLine.h

Заголовочный файл для модуля CmdLine.

#include <iostream>
#include <unordered_map>
Граф включаемых заголовочных файлов для CmdLine.h:



Граф файлов, в которые включается этот файл:



Классы

- class CmdLine

Перечисления

- enum param_type { string , integer }
  Класс CmdLine.

5.1.1 Подробное описание

Заголовочный файл для модуля CmdLine.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.1.2 Перечисления

5.1.2.1 param_type

enum param_type

Класс CmdLine.

Класс CmdLine позволяет добавлять параметры, анализировать аргументы командной строки и получать значения параметров

Предупреждения

Реализация только для типа данных int16_t

Перечисление типов параметров

Это перечисление определяет типы параметров, которые может обрабатывать класс CmdLine.

5.2 CmdLine.h

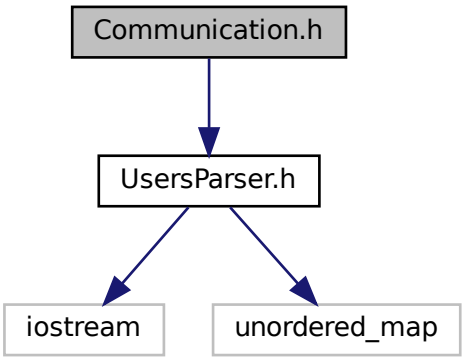
См. документацию.

```
1 #ifndef __CMDLINE_H__
2 #define __CMDLINE_H__
3
4 #include <iostream>
5 #include <unordered_map>
6
28 enum param_type{
29     string,
30     integer,
31 };
32
33 class CmdLine{
39     struct param{
40         std::string name;
41         std::string shortName;
42         std::string valStr;
43         std::string description;
44
45         param_type type;
46         int valInt;
47     };
48
49 public:
57     void addParam(std::string name, std::string shortName,
58                 std::string description = std::string(),param_type type = param_type::string);
64     void parse(int argc, char** argv);
68     void printHelp();
74     int getInt(std::string name);
80     std::string getString(std::string name);
85     bool isHasHelp(){return m_isHasHelp;}
86 private:
93     std::string getNextArg(int argc,char** argv);
99     bool isHasParam(std::string name);
105     bool isHasParamShort(std::string name);
111     bool isnumber(std::string str);
117     void modifyParam(param& parameter, std::string curArg);
118
119     std::unordered_map<std::string,param> m_params;
120     std::unordered_map<std::string,param> m_paramsShort;
121
122
123     int m_curArgIndx = 1;
124     bool m_isHasHelp = false;
125 };
126
127 #endif // __CMDLINE_H__
```

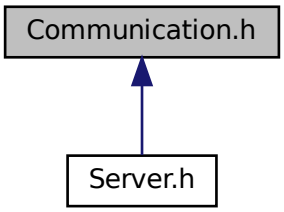
5.3 Файл Communication.h

Заголовочный файл для модуля Communication.

#include "UsersParser.h"
Граф включаемых заголовочных файлов для Communication.h:



Граф файлов, в которые включается этот файл:



Классы

- class Communication

  Класс Communication.

5.3.1 Подробное описание

Заголовочный файл для модуля Communication.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.4 Communication.h

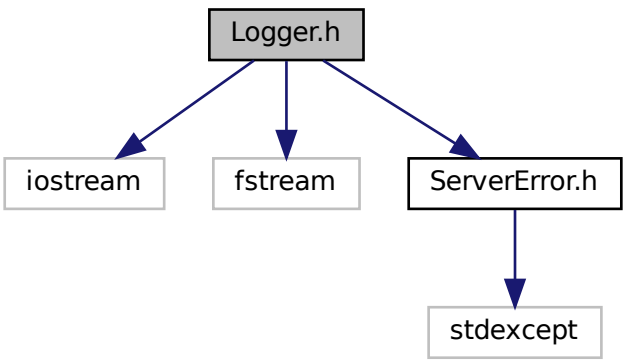
См. документацию.

```
1 #ifndef __COMMUNICATION_H__
2 #define __COMMUNICATION_H__
3
4 #include "UsersParser.h"
20 class Communication{
21 public:
26     void createSocket(int port);
31     int waitClient();
37     void autorize(int clientSocket,UsersParser users);
38
39 private:
40     int m_socket = 0;
41 };
42
43 #endif // __COMMUNICATION_H__
```

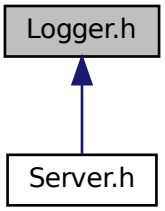
5.5 Файл Logger.h

Заголовочный файл для модуля Logger.

#include <iostream>
#include <fstream>
#include "ServerError.h"
Граф включаемых заголовочных файлов для Logger.h:



Граф файлов, в которые включается этот файл:



Классы

- class Logger
  Класс Logger.

5.5.1 Подробное описание

Заголовочный файл для модуля Logger.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.6 Logger.h

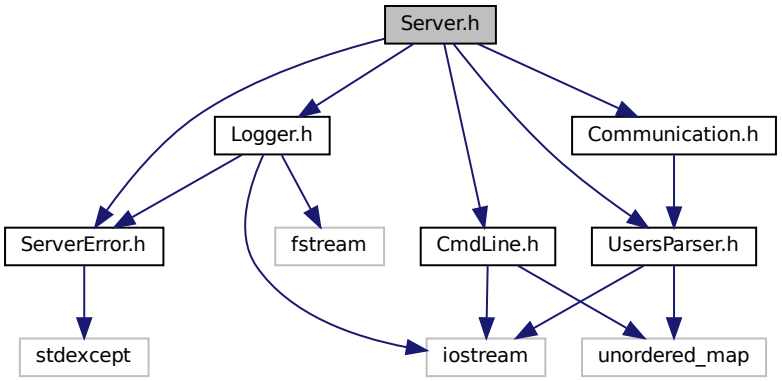
См. документацию.

```
1 #ifndef __LOGGER_H__
2 #define __LOGGER_H__
3
4 #include <iostream>
5 #include <fstream>
6 #include "ServerError.h"
22 class Logger{
23     public:
29         bool Initialize(std::string logPath);
34         void info(std::string str);
39         void serverError(ServerError serverError);
44         static Logger &get();
49         static Logger *getPtr();
50
51     private:
52         static Logger* m_state;
53
54         std::string m_pathToFile;
55         std::ofstream m_logFile;
56 };
57 #endif // __LOGGER_H__
```

5.7 Файл Server.h

Заголовочный файл для модуля Server.

#include "ServerError.h"
#include "CmdLine.h"
#include "Logger.h"
#include "Communication.h"
#include "UsersParser.h"
Граф включаемых заголовочных файлов для Server.h:



Классы

- class Server

  Класс Server.

5.7.1 Подробное описание

Заголовочный файл для модуля Server.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.8 Server.h

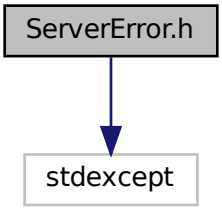
См. документацию.

```
1 #ifndef __SERVER_H__
2 #define __SERVER_H__
3
4 #include "ServerError.h"
5 #include "CmdLine.h"
6 #include "Logger.h"
7 #include "Communication.h"
8 #include "UsersParser.h"
24 class Server{
25
26 public:
33     bool Initialize(int argc,char** argv);
37     void start();
38
39 private:
44     void executeCalculations(int client);
45
46     CmdLine m_cmd;
47     UsersParser m_usersParser;
48     Communication m_com;
49
50     bool m_mustDie = false;
51 };
52 #endif // __SERVER_H__
```

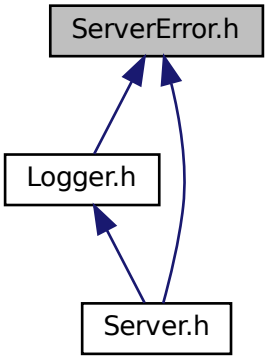
5.9 Файл ServerError.h

Заголовочный файл для модуля ServerError.

#include <stdexcept>
Граф включаемых заголовочных файлов для ServerError.h:



Граф файлов, в которые включается этот файл:



Классы

- class ServerError

Перечисления

- enum ErrorLevel { Warning , Critical }
  Класс ServerError.

5.9.1 Подробное описание

Заголовочный файл для модуля ServerError.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.9.2 Перечисления

5.9.2.1 ErrorLevel

enum ErrorLevel

Класс ServerError.

Класс ServerError отвечает за обработку ошибок сервера

Предупреждения

Реализация только для типа данных int16_t

5.10 ServerError.h

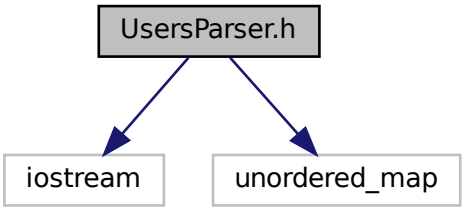
См. документацию.

```
1 #ifndef __SERVERERROR_H__
2 #define __SERVERERROR_H__
3
4 #include <stdexcept>
19 enum ErrorLevel
20 {
21     Warning,
22     Critical
23 };
24
25 class ServerError : public std::runtime_error
26 {
27
28 public:
32     ServerError(const char *msg) : std::runtime_error(msg){};
37     ServerError(const std::string &msg, ErrorLevel level = ErrorLevel::Warning) : std::runtime_error(msg),
38                                                 m_errorLevel(level){
39
40                                                 };
44     ErrorLevel getErrorLevel() {return m_errorLevel;}
45
46 private:
47     ErrorLevel m_errorLevel;
48 };
49
50 #endif // __SERVERERROR_H__
```

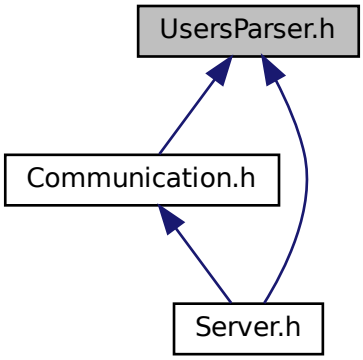
5.11 Файл UsersParser.h

Заголовочный файл для модуля UsersParser.

#include <iostream>
#include <unordered_map>
Граф включаемых заголовочных файлов для UsersParser.h:



Граф файлов, в которые включается этот файл:



Классы

- class UsersParser
  Класс UsersParser.

5.11.1 Подробное описание

Заголовочный файл для модуля UsersParser.

Автор

Павлова В.М.

Версия

1.0

Дата

20.12.2025

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

5.12 UsersParser.h

См. документацию.

```
1 #ifndef __USERSPARSER_H__
2 #define __USERSPARSER_H__
3
4 #include <iostream>
5 #include <unordered_map>
20 class UsersParser{
21
22 public:
28     void parse(std::string pathToFile);
35     bool isHasUser(std::string username);
42     std::string getPassword(std::string username);
43
44 private:
45
46     std::unordered_map<std::string,std::string> m_users;
47 };
48
49 #endif // __USERSPARSER_H__
```

Предметный указатель