

NoKurs

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 CmdLine	7
4.1.1 Методы	7
4.1.1.1 addParam()	7
4.1.1.2 getInt()	8
4.1.1.3 getString()	8
4.1.1.4 isHasHelp()	8
4.1.1.5 parse()	9
4.2 Класс Communication	9
4.2.1 Подробное описание	9
4.2.2 Методы	10
4.2.2.1 authorize()	10
4.2.2.2 createSocket()	10
4.2.2.3 waitClient()	10
4.3 Класс Logger	11
4.3.1 Подробное описание	11
4.3.2 Методы	11
4.3.2.1 get()	11
4.3.2.2 getPtr()	12
4.3.2.3 info()	12
4.3.2.4 Initialize()	12
4.3.2.5 serverError()	12
4.4 Класс Server	13
4.4.1 Подробное описание	13
4.4.2 Методы	13
4.4.2.1 Initialize()	13
4.5 Класс ServerError	14
4.5.1 Конструктор(ы)	15
4.5.1.1 ServerError() [1/2]	15
4.5.1.2 ServerError() [2/2]	15
4.5.2 Методы	15
4.5.2.1 getErrorLevel()	15
4.6 Класс UsersParser	16
4.6.1 Подробное описание	16
4.6.2 Методы	16

---

4.6.2.1 getPassword()	16
4.6.2.2 isHasUser()	17
4.6.2.3 parse()	17
5 Файлы	19
5.1 Файл CmdLine.h	19
5.1.1 Подробное описание	20
5.1.2 Перечисления	20
5.1.2.1 param_type	20
5.2 Файл Communication.h	21
5.2.1 Подробное описание	21
5.3 Файл Logger.h	22
5.3.1 Подробное описание	23
5.4 Файл Server.h	24
5.4.1 Подробное описание	24
5.5 Файл ServerError.h	25
5.5.1 Подробное описание	26
5.5.2 Перечисления	26
5.5.2.1 ErrorLevel	26
5.6 Файл UsersParser.h	27
5.6.1 Подробное описание	27
Предметный указатель	29

# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

CmdLine . . . . .	7
Communication . . . . .	9
Logger . . . . .	11
std::runtime_error	
ServerError . . . . .	14
Server . . . . .	13
UsersParser . . . . .	16



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">CmdLine</a>	7
<a href="#">Communication</a>	
Класс <a href="#">Communication</a>	9
<a href="#">Logger</a>	
Класс <a href="#">Logger</a>	11
<a href="#">Server</a>	
Класс <a href="#">Server</a>	13
<a href="#">ServerError</a>	14
<a href="#">UsersParser</a>	
Класс <a href="#">UsersParser</a>	16





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">CmdLine.h</a>	
Заголовочный файл для модуля <a href="#">CmdLine</a> . . . . .	19
<a href="#">Communication.h</a>	
Заголовочный файл для модуля <a href="#">Communication</a> . . . . .	21
<a href="#">Logger.h</a>	
Заголовочный файл для модуля <a href="#">Logger</a> . . . . .	22
<a href="#">Server.h</a>	
Заголовочный файл для модуля <a href="#">Server</a> . . . . .	24
<a href="#">ServerError.h</a>	
Заголовочный файл для модуля <a href="#">ServerError</a> . . . . .	25
<a href="#">UsersParser.h</a>	
Заголовочный файл для модуля <a href="#">UsersParser</a> . . . . .	27



## 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 [authorize](#) (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 authorize()

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

Авторизует клиента, сравнивая его учетные данные с информацией пользователя.

Аргументы

clientSocket	Дескриптор сокета клиента. @paramusers Экземпляр класса <a href="#">UsersParser</a> , 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

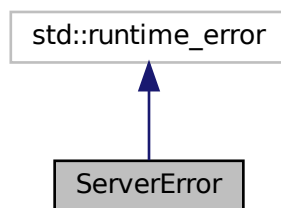
true, если инициализация прошла успешно, false в противном случае

Объявления и описания членов класса находятся в файле:

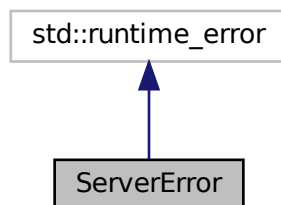
- [Server.h](#)

## 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

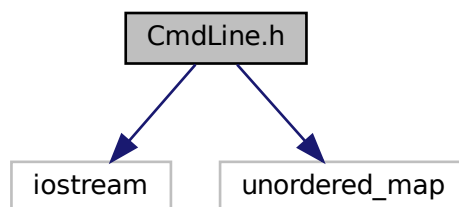
# Файлы

### 5.1 Файл CmdLine.h

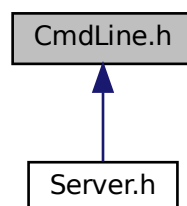
Заголовочный файл для модуля [CmdLine](#).

```
#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

Макеев В.Д.

#### Версия

1.0

#### Дата

20.12.2023

#### 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](#).

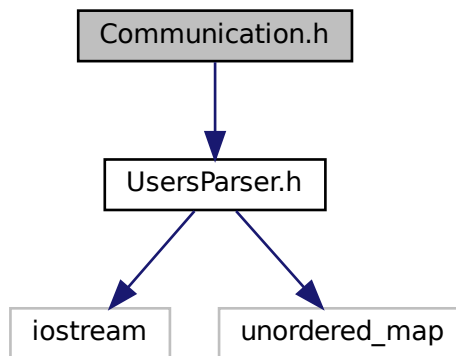


## 5.2 Файл Communication.h

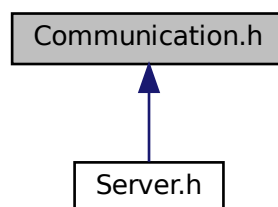
Заголовочный файл для модуля [Communication](#).

```
#include "UsersParser.h"
```

Граф включаемых заголовочных файлов для Communication.h:



Граф файлов, в которые включается этот файл:



### Классы

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

### 5.2.1 Подробное описание

Заголовочный файл для модуля [Communication](#).

Автор

Макеев В.Д.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

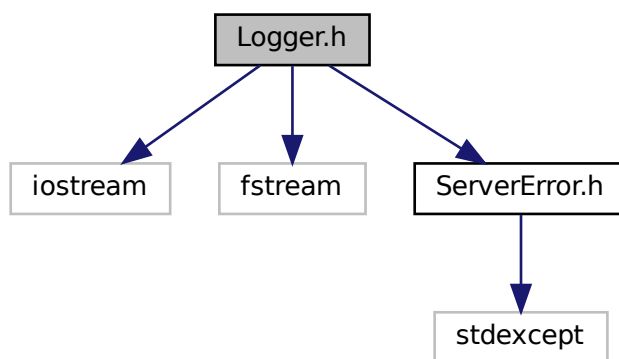
Курсовая работа студента

### 5.3 Файл `Logger.h`

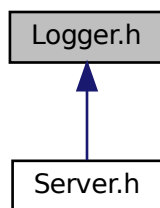
Заголовочный файл для модуля [Logger](#).

```
#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](#).

### 5.3.1 Подробное описание

Заголовочный файл для модуля [Logger](#).

Автор

Макеев В.Д.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

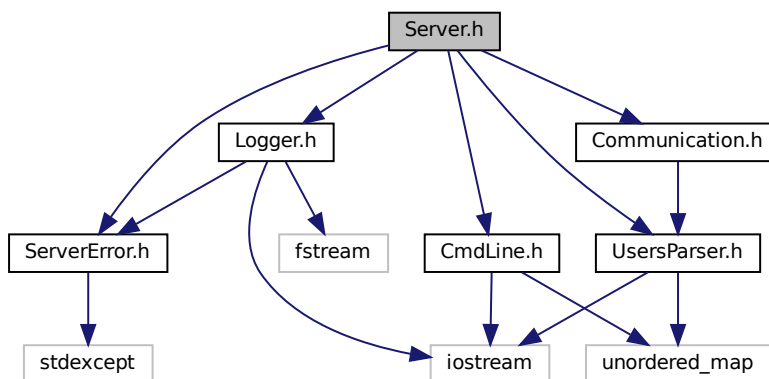
Курсовая работа студента

## 5.4 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](#).

### 5.4.1 Подробное описание

Заголовочный файл для модуля [Server](#).

Автор

Макеев В.Д.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

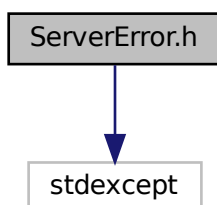
Курсовая работа студента

## 5.5 Файл ServerError.h

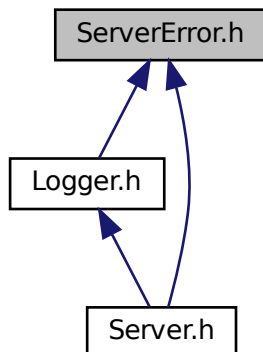
Заголовочный файл для модуля [ServerError](#).

```
#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](#).

### 5.5.1 Подробное описание

Заголовочный файл для модуля [ServerError](#).

Автор

Макеев В.Д.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

### 5.5.2 Перечисления

#### 5.5.2.1 ErrorLevel

enum [ErrorLevel](#)

Класс [ServerError](#).

Класс [ServerError](#) отвечает за обработку ошибок сервера

Предупреждения

Реализация только для типа данных `int16_t`

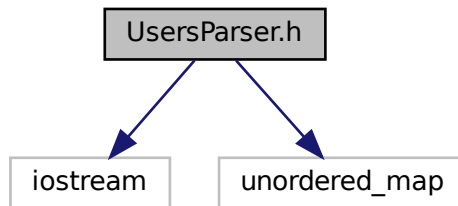
## 5.6 Файл UsersParser.h

Заголовочный файл для модуля `UsersParser`.

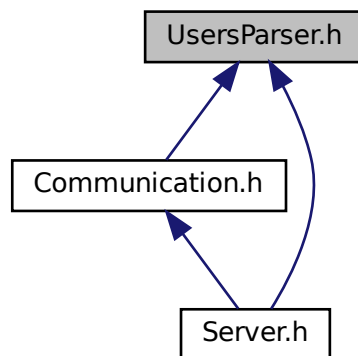
```
#include <iostream>
```

```
#include <unordered_map>
```

Граф включаемых заголовочных файлов для UsersParser.h:



Граф файлов, в которые включается этот файл:



### Классы

- class `UsersParser`  
Класс `UsersParser`.

#### 5.6.1 Подробное описание

Заголовочный файл для модуля `UsersParser`.

Автор

Макеев В.Д.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента



# Предметный указатель

- addParam
  - CmdLine, [7](#)
- authorize
  - Communication, [10](#)
- CmdLine, [7](#)
  - addParam, [7](#)
  - getInt, [8](#)
  - getString, [8](#)
  - isHasHelp, [8](#)
  - parse, [9](#)
- CmdLine.h, [19](#)
  - param\_type, [20](#)
- Communication, [9](#)
  - authorize, [10](#)
  - createSocket, [10](#)
  - waitClient, [10](#)
- Communication.h, [21](#)
- createSocket
  - Communication, [10](#)
- ErrorLevel
  - ServerError.h, [26](#)
- get
  - Logger, [11](#)
- getErrorLevel
  - ServerError, [15](#)
- getInt
  - CmdLine, [8](#)
- getPassword
  - UsersParser, [16](#)
- getPtr
  - Logger, [11](#)
- getString
  - CmdLine, [8](#)
- info
  - Logger, [12](#)
- Initialize
  - Logger, [12](#)
  - Server, [13](#)
- isHasHelp
  - CmdLine, [8](#)
- isHasUser
  - UsersParser, [17](#)
- Logger, [11](#)
  - get, [11](#)
  - getPtr, [11](#)
- info, [12](#)
- Initialize, [12](#)
- serverError, [12](#)
- Logger.h, [22](#)
- param\_type
  - CmdLine.h, [20](#)
- parse
  - CmdLine, [9](#)
  - UsersParser, [17](#)
- Server, [13](#)
  - Initialize, [13](#)
- Server.h, [24](#)
- ServerError, [14](#)
  - getErrorLevel, [15](#)
  - ServerError, [15](#)
- serverError
  - Logger, [12](#)
- ServerError.h, [25](#)
  - ErrorLevel, [26](#)
- UsersParser, [16](#)
  - getPassword, [16](#)
  - isHasUser, [17](#)
  - parse, [17](#)
- UsersParser.h, [27](#)
- waitClient
  - Communication, [10](#)