

# Server TCP

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 Calculator	5
3.1.1 Подробное описание	5
3.1.2 Методы	5
3.1.2.1 Calculate()	5
3.2 Класс Comunicator	6
3.2.1 Подробное описание	6
3.2.2 Методы	6
3.2.2.1 CreateSocket()	6
3.2.2.2 isAuthorized()	7
3.2.2.3 WaitClient()	7
3.3 Класс Logger	8
3.3.1 Подробное описание	8
3.3.2 Методы	8
3.3.2.1 Error()	8
3.3.2.2 Info()	9
3.3.2.3 Initialize()	9
3.3.2.4 Warning()	9
3.4 Класс Server	10
3.4.1 Подробное описание	11
3.4.2 Методы	11
3.4.2.1 Initialize()	11
3.4.2.2 ParseCmdLine()	11
3.4.2.3 Run()	12
3.5 Класс UsersBase	12
3.5.1 Подробное описание	12
3.5.2 Методы	13
3.5.2.1 AddUser()	13
3.5.2.2 getPassword()	14
3.5.2.3 isHasUser()	14
3.5.2.4 ParseFile()	15
4 Файлы	17
4.1 Файл Calculator.h	17
4.1.1 Подробное описание	17
4.2 Файл Comunicator.h	18
4.2.1 Подробное описание	18
4.3 Файл Logger.h	19

4.3.1 Подробное описание . . . . .	20
4.4 Файл Server.h . . . . .	20
4.4.1 Подробное описание . . . . .	21
4.5 Файл UsersBase.h . . . . .	22
4.5.1 Подробное описание . . . . .	22
Предметный указатель . . . . .	25

# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">Calculator</a>		
Класс <a href="#">Calculator</a>	. . . . .	5
<a href="#">Communicator</a>		
Класс <a href="#">Communicator</a>	. . . . .	6
<a href="#">Logger</a>		
Класс <a href="#">Logger</a>	. . . . .	8
<a href="#">Server</a>		
Класс <a href="#">Server</a>	. . . . .	10
<a href="#">UsersBase</a>		
Класс <a href="#">UsersBase</a>	. . . . .	12



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">Calculator.h</a>	
Заголовочный файл для модуля <a href="#">Calculator</a>	17
<a href="#">Communicator.h</a>	
Заголовочный файл для модуля <a href="#">Communicator</a>	18
<a href="#">Logger.h</a>	
Заголовочный файл для модуля <a href="#">Logger</a>	19
<a href="#">Server.h</a>	
Заголовочный файл для модуля <a href="#">Server</a>	20
<a href="#">UsersBase.h</a>	
Заголовочный файл для модуля <a href="#">UsersBase</a>	22





## Глава 3

# Классы

### 3.1 Класс Calculator

Класс [Calculator](#).

```
#include <Calculator.h>
```

Открытые статические члены

- static bool [Calculate](#) (const int \_clientSocket)  
Метод Calculate для выполнения операции над данными и их отправки

#### 3.1.1 Подробное описание

Класс [Calculator](#).

Для выполнения операции над данными и их отправки предназначен метод Calculate

Предупреждения

Реализация только для типа данных uint32\_t

#### 3.1.2 Методы

##### 3.1.2.1 Calculate()

```
static bool Calculator::Calculate (  
    const int _clientSocket ) [static]
```

Метод Calculate для выполнения операции над данными и их отправки

Аргументы

in	_clientSocket	сокет
----	---------------	-------

Возвращает

Метод ничего не возвращает

Объявления и описания членов класса находятся в файле:

- [Calculator.h](#)

## 3.2 Класс Communicator

Класс [Communicator](#).

```
#include <Communicator.h>
```

Открытые члены

- int [CreateSocket](#) (const int \_port)  
Создание сокета сервера
- int [WaitClient](#) (const int \_socket)  
Создание сокета клиента
- bool [isAuthorized](#) (int work\_sock, [UsersBase](#) \_usersBase)  
Авторизация

### 3.2.1 Подробное описание

Класс [Communicator](#).

Для создания сокета сервера и слушания запросов предназначен метод [CreateSocket](#)

Для создания сокета клиента и подключения его к серверу метод [WaitClient](#)

Для авторизации клиента предназначен метод [isAuthorized](#)

### 3.2.2 Методы

#### 3.2.2.1 CreateSocket()

```
int Communicator::CreateSocket (  
    const int _port )
```

Создание сокета сервера

Аргументы

in	_port	порт
----	-------	------

Возвращает

Сокет

### 3.2.2.2 isAuthorized()

```
bool Communicator::isAuthorized (
    int work_sock,
    UsersBase _usersBase )
```

Авторизация

Аргументы

in	work_sock	сокет клиента
in	<a href="#">UsersBase</a>	база данных пользователя

Возвращает

Сокет клиента

### 3.2.2.3 WaitClient()

```
int Communicator::WaitClient (
    const int _socket )
```

Создание сокета клиента

Аргументы

in	_socket	сокет сервера
----	---------	---------------

Возвращает

Сокет клиента

Объявления и описания членов класса находятся в файле:

- [Communicator.h](#)

### 3.3 Класс Logger

Класс `Logger`.

```
#include <Logger.h>
```

#### Открытые члены

- `Logger()`=default  
Конструктор по умолчанию
- `bool Initialize (const std::string _pathToLogFile)`  
Проверка доступности файла
- `void Info (std::string _log)`  
Выводы в логи информации
- `void Error (std::string _log)`  
Выводы в логи ошибки
- `void Warning (std::string _log)`  
Выводы в логи предупреждения

#### Закрытые данные

- `std::string m_pathToFile`

#### 3.3.1 Подробное описание

Класс `Logger`.

Для записи ошибок предназначен методы `Info`, `Error`, `Warning`

#### 3.3.2 Методы

##### 3.3.2.1 `Error()`

```
void Logger::Error (
    std::string _log )
```

Выводы в логи ошибки

Аргументы

in	_log	запись в файл
----	------	---------------

Возвращает

Метод ничего не возвращает

#### 3.3.2.2 Info()

```
void Logger::Info (
    std::string _log )
```

Выводы в логи информации

Аргументы

in	_log	запись в файл
----	------	---------------

Возвращает

Метод ничего не возвращает

#### 3.3.2.3 Initialize()

```
bool Logger::Initialize (
    const std::string _pathToLogFile )
```

Проверка доступности файла

Аргументы

in	_pathToLogFile	путь до файла
----	----------------	---------------

Возвращает

Метод возвращает false, если файл недоступен для записи

#### 3.3.2.4 Warning()

```
void Logger::Warning (
    std::string _log )
```

Выводы в логи предупреждения

Аргументы

in	_log	запись в файл
----	------	---------------

Возвращает

Метод ничего не возвращает

Объявления и описания членов класса находятся в файле:

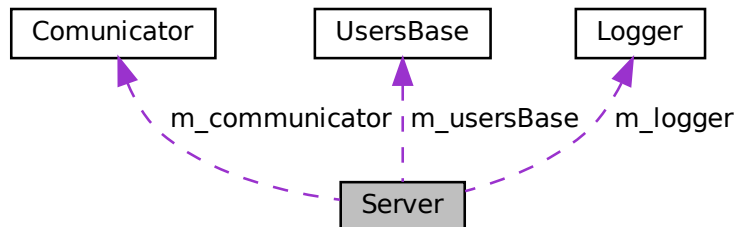
- [Logger.h](#)

### 3.4 Класс Server

Класс [Server](#).

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

Граф связей класса Server:



Открытые члены

- bool [ParseCmdLine](#) (int argc, char \*\*argv)  
Метод ParseCmdLine для парсинга
- bool [Initialize](#) ()  
Метод Initialize для инициализации
- int [Run](#) ()  
Метод Run для запуска и прослушки сообщений

Закрытые члены

- std::string getNextArg (int argc, char \*\*argv)
- bool isNumber (std::string \_str)

## Закрытые данные

- `std::string m_usersBasePath`
- `int m_targetPort`
- `std::string m_logFilePath`
- `UsersBase m_usersBase`
- `Communicator m_communicator`
- `Logger m_logger`

### 3.4.1 Подробное описание

Класс `Server`.

Для управления сервером предназначены методы: `ParseCmdLine` для парсинга командной строки и `Initialize` для инициализации и `Run` для запуска сервера

### 3.4.2 Методы

#### 3.4.2.1 `Initialize()`

```
bool Server::Initialize ( )
```

Метод `Initialize` для инициализации

Возвращает

Метод возвращает `false`, если указан неверный параметр

#### 3.4.2.2 `ParseCmdLine()`

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

Метод `ParseCmdLine` для парсинга

Аргументы

Параметры	командной строки
-----------	------------------

Возвращает

Метод возвращает `false`, если нарушена правильность написания аргументов

## Предупреждения

Вызов до инициализации

### 3.4.2.3 Run()

```
int Server::Run ( )
```

Метод Run для запуска и прослушки сообщений

Возвращает

Метод ничего не возвращает

Объявления и описания членов класса находятся в файле:

- [Server.h](#)

## 3.5 Класс UsersBase

Класс [UsersBase](#).

```
#include <UsersBase.h>
```

### Открытые члены

- bool [ParseFile](#) (const std::string &\_pathToFile)  
Метод ParseFile для обработки базы данных пользователей
- bool [AddUser](#) (const std::string &\_login, const std::string &\_password)  
Метод AddUser для добавления пользователя в базу данных
- std::string [getPassword](#) (const std::string &\_login)  
Метод getPassword возвращает пароль для логина
- const bool [isHasUser](#) (const std::string &\_login)  
Метод isHasUser для проверки наличия пользователя

### Закрытые данные

- std::unordered\_map< std::string, std::string > m\_users

### 3.5.1 Подробное описание

Класс [UsersBase](#).

Для чтения базы данных предназначен метод ParseFile Для проверки наличия клиента в базе данных предназначен метод isHasUser



### 3.5.2 Методы

#### 3.5.2.1 AddUser()

```
bool UsersBase::AddUser (
    const std::string & _login,
    const std::string & _password )
```

Метод AddUser для добавления пользователя в базу данных

Аргументы

in	login	- логин
in	password	- пароль

Возвращает

Метод возвращает false, если пользователь уже существует

### 3.5.2.2 getPassword()

```
std::string UsersBase::getPassword (
    const std::string & _login )
```

Метод getPassword возвращает пароль для логина

Аргументы

in	login	- логин
----	-------	---------

Возвращает

Метод возвращает пустую строку, если пользователь отсутствует

### 3.5.2.3 isHasUser()

```
const bool UsersBase::isHasUser (
    const std::string & _login )
```

Метод isHasUser для проверки наличия пользователя

Аргументы

in	login	- логин
----	-------	---------

Возвращает

Метод возвращает false, если пользователь отсутствует

#### 3.5.2.4 ParseFile()

```
bool UsersBase::ParseFile (
    const std::string & _pathToFile )
```

Метод ParseFile для обработки базы данных пользователей

Аргументы

in	_pathToUser	base - путь до базы данных пользователей
----	-------------	--

Возвращает

Метод возвращает false, если файл недоступен для чтения и если содержится ошибка в базе данных

Объявления и описания членов класса находятся в файле:

- [UsersBase.h](#)



# Глава 4

## Файлы

### 4.1 Файл Calculator.h

Заголовочный файл для модуля [Calculator](#).

#### Классы

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

#### 4.1.1 Подробное описание

Заголовочный файл для модуля [Calculator](#).

#### Автор

Спирин И.А.

#### Версия

1.0

#### Дата

18.12.2023

#### Авторство

ИБСТ ПГУ

#### Предупреждения

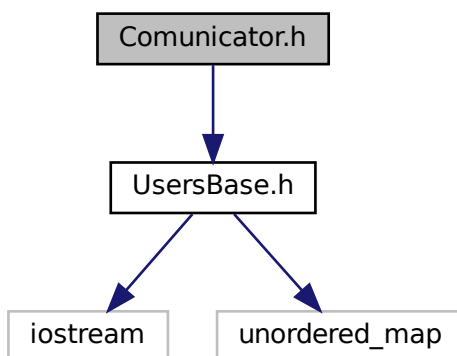
Курсовая работа студента

## 4.2 Файл Communicator.h

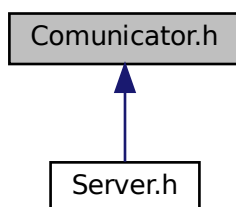
Заголовочный файл для модуля [Communicator](#).

```
#include "UsersBase.h"
```

Граф включаемых заголовочных файлов для Communicator.h:



Граф файлов, в которые включается этот файл:



### Классы

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

#### 4.2.1 Подробное описание

Заголовочный файл для модуля [Communicator](#).

Автор

Спирин И.А.

Версия

1.0

Дата

18.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

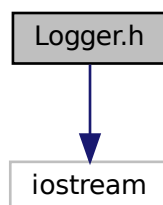
Курсовая работа студента

## 4.3 Файл Logger.h

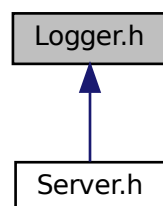
Заголовочный файл для модуля [Logger](#).

```
#include <iostream>
```

Граф включаемых заголовочных файлов для Logger.h:



Граф файлов, в которые включается этот файл:



## Классы

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

### 4.3.1 Подробное описание

Заголовочный файл для модуля [Logger](#).

Автор

Спирин И.А.

Версия

1.0

Дата

18.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

## 4.4 Файл Server.h

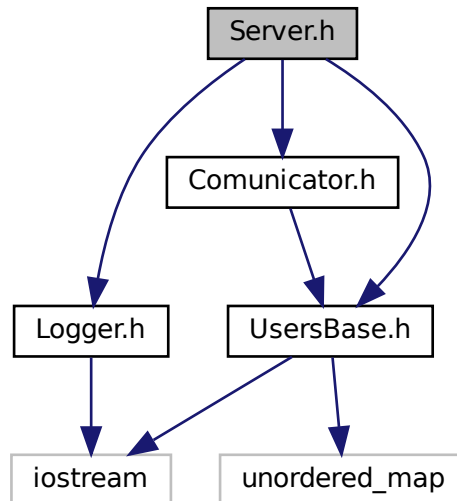
Заголовочный файл для модуля [Server](#).

```
#include "Logger.h"  
#include "Communicator.h"
```



```
#include "UsersBase.h"
```

Граф включаемых заголовочных файлов для Server.h:



## Классы

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

### 4.4.1 Подробное описание

Заголовочный файл для модуля [Server](#).

Автор

Спирин И.А.

Версия

1.0

Дата

18.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

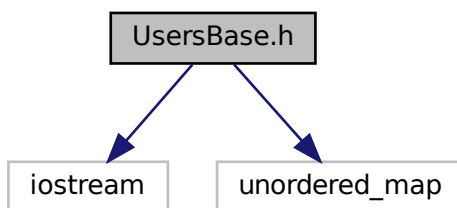
## 4.5 Файл UsersBase.h

Заголовочный файл для модуля [UsersBase](#).

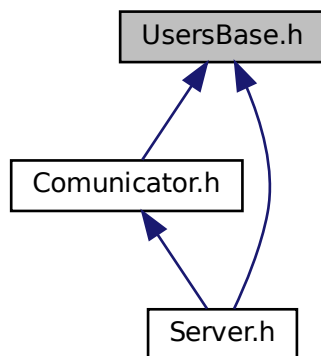
```
#include <iostream>
```

```
#include <unordered_map>
```

Граф включаемых заголовочных файлов для UsersBase.h:



Граф файлов, в которые включается этот файл:



### Классы

- class [UsersBase](#)

Класс [UsersBase](#).

#### 4.5.1 Подробное описание

Заголовочный файл для модуля [UsersBase](#).

Автор

Спирин И.А.

Версия

1.0

Дата

18.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента



# Предметный указатель

- AddUser
  - UsersBase, [13](#)
- Calculate
  - Calculator, [5](#)
- Calculator, [5](#)
  - Calculate, [5](#)
- Calculator.h, [17](#)
- Communicator, [6](#)
  - CreateSocket, [6](#)
  - isAuthorized, [7](#)
  - WaitClient, [7](#)
- Communicator.h, [18](#)
- CreateSocket
  - Communicator, [6](#)
- Error
  - Logger, [8](#)
- getPassword
  - UsersBase, [14](#)
- Info
  - Logger, [9](#)
- Initialize
  - Logger, [9](#)
  - Server, [11](#)
- isAuthorized
  - Communicator, [7](#)
- isHasUser
  - UsersBase, [14](#)
- Logger, [8](#)
  - Error, [8](#)
  - Info, [9](#)
  - Initialize, [9](#)
  - Warning, [9](#)
- Logger.h, [19](#)
- ParseCmdLine
  - Server, [11](#)
- ParseFile
  - UsersBase, [14](#)
- Run
  - Server, [12](#)
- Server, [10](#)
  - Initialize, [11](#)
  - ParseCmdLine, [11](#)
  - Run, [12](#)
- Server.h, [20](#)
- UsersBase, [12](#)
  - AddUser, [13](#)
  - getPassword, [14](#)
  - isHasUser, [14](#)
  - ParseFile, [14](#)
- UsersBase.h, [22](#)
- WaitClient
  - Communicator, [7](#)
- Warning
  - Logger, [9](#)