

Документация кода программы курсовой работы
1.08

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 datareader	7
4.1.1 Подробное описание	7
4.1.2 Конструктор(ы)	7
4.1.2.1 datareader()	7
4.1.3 Методы	8
4.1.3.1 get()	8
4.2 Класс ErrorAuth	8
4.2.1 Подробное описание	9
4.3 Класс ErrorLogin	9
4.3.1 Подробное описание	10
4.4 Класс ErrorRead	10
4.4.1 Подробное описание	11
4.5 Класс md5h	11
4.5.1 Подробное описание	12
4.5.2 Конструктор(ы)	12
4.5.2.1 md5h()	12
4.5.3 Методы	12
4.5.3.1 generation_SALT()	12
4.5.3.2 hash()	12
4.6 Класс server	13
4.6.1 Подробное описание	14
4.6.2 Конструктор(ы)	14
4.6.2.1 server()	14
4.6.3 Методы	14
4.6.3.1 authentication()	15
4.6.3.2 connect()	15
4.6.3.3 operations()	15
4.6.3.4 start()	15
5 Файлы	17
5.1 Файл datareader.hpp	17
5.1.1 Подробное описание	18
5.2 Файл main.cpp	18
5.2.1 Подробное описание	19
5.2.2 Функции	19

5.2.2.1 main()	19
5.3 Файл md5h.cpp	20
5.3.1 Подробное описание	20
5.4 Файл md5h.hpp	21
5.4.1 Подробное описание	22
5.5 Файл ProgramError.hpp	22
5.5.1 Подробное описание	23
5.5.2 Функции	24
5.5.2.1 log_entry()	24
5.6 Файл server.hpp	24
5.6.1 Подробное описание	25
Предметный указатель	27

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

datareader	7
std::invalid_argument	
ErrorAuth	8
ErrorRead	10
md5h	11
std::runtime_error	
ErrorLogin	9
server	13

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

datareader	Класс для чтения данных из файла и формирования хранилища пользователей .	7
ErrorAuth	Класс обработки ошибок аутентификации	8
ErrorLogin	Класс обработки ошибок записи в журнал	9
ErrorRead	Класс обработки ошибок чтения файла данных пользователей	10
md5h	Класс обработки MD5 Hash	11
server	Класс управления работой сервера	13

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

datareader.hpp	
Считывание данных о клиентах	17
main.cpp	
Справка по использованию	18
md5h.cpp	
Класс обработки MD5 Hash	20
md5h.hpp	
Класс обработки MD5 Hash	21
ProgramError.hpp	
Обработка ошибок, с последующей записью в журнал	22
server.hpp	
Операции выполняемые сервером	24

Глава 4

Классы

4.1 Класс datareader

Класс для чтения данных из файла и формирования хранилища пользователей

```
#include <datareader.hpp>
```

Открытые члены

- `datareader` (const string &local_path)
Конструктор класса DataReader.
- `map< string, string > get` (const string &file_path)
Получает данные из указанного файла и формирует хранилище пользователей

4.1.1 Подробное описание

Класс для чтения данных из файла и формирования хранилища пользователей

4.1.2 Конструктор(ы)

4.1.2.1 datareader()

```
datareader::datareader (  
    const string & local_path )
```

Конструктор класса DataReader.

Аргументы

local_path	Путь к локальному файлу, который будет использоваться для чтения данных
------------	---

4.1.3 Методы

4.1.3.1 get()

```
map< string, string > datareader::get (
    const string & file_path )
```

Получает данные из указанного файла и формирует хранилище пользователей

Аргументы

file_path	Путь к файлу с данными пользователей
-----------	--------------------------------------

Возвращает

Хранилище данных пользователей в виде `std::map`, где ключ - идентификатор, значение - пароль

Исключения

ErrorRead	Если возникают проблемы при открытии или чтении файла
---------------------------	---

Объявления и описания членов классов находятся в файлах:

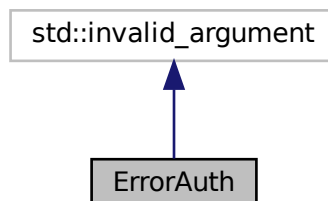
- [datareader.hpp](#)
- [datareader.cpp](#)

4.2 Класс ErrorAuth

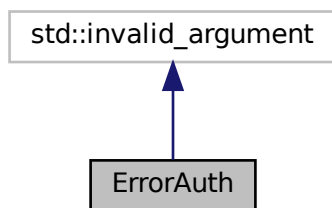
Класс обработки ошибок аутентификации

```
#include <ProgramError.hpp>
```

Граф наследования:ErrorAuth:



Граф связей класса ErrorAuth:



Открытые члены

- ErrorAuth (const std::string &s)
- ErrorAuth (const char *s)

4.2.1 Подробное описание

Класс обработки ошибок аутентификации

Объявления и описания членов класса находятся в файле:

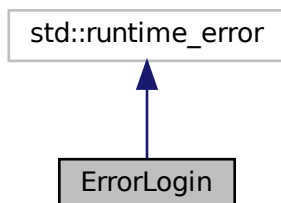
- [ProgramError.hpp](#)

4.3 Класс ErrorLogin

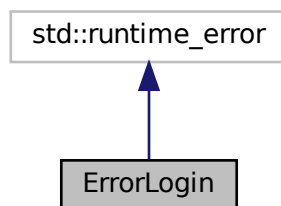
Класс обработки ошибок записи в журнал

```
#include <ProgramError.hpp>
```

Граф наследования:ErrorLogin:



Граф связей класса ErrorLogin:



Открытые члены

- `ErrorLogin (const std::string &s)`
- `ErrorLogin (const char *s)`

4.3.1 Подробное описание

Класс обработки ошибок записи в журнал

Объявления и описания членов класса находятся в файле:

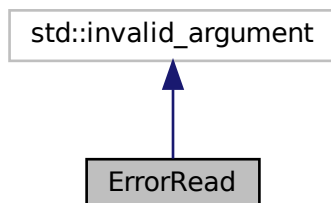
- [ProgramError.hpp](#)

4.4 Класс ErrorRead

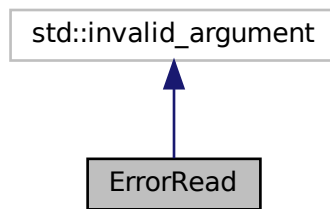
Класс обработки ошибок чтения файла данных пользователей

```
#include <ProgramError.hpp>
```

Граф наследования:ErrorRead:



Граф связей класса ErrorRead:



Открытые члены

- `ErrorRead (const std::string &s)`
- `ErrorRead (const char *s)`

4.4.1 Подробное описание

Класс обработки ошибок чтения файла данных пользователей

Объявления и описания членов класса находятся в файле:

- [ProgramError.hpp](#)

4.5 Класс md5h

Класс обработки MD5 Hash.

```
#include <md5h.hpp>
```

Открытые члены

- [md5h \(\)](#)
Конструктор класса [md5h](#).
- `string hash (const string &PASS)`
Получение хэша пароля

Открытые атрибуты

- `string salt`
SALT для хеширования

Закрытые члены

- `string generation_SALT ()`
Генератор SALT.

4.5.1 Подробное описание

Класс обработки MD5 Hash.

4.5.2 Конструктор(ы)

4.5.2.1 `md5h()`

`md5h::md5h ()`

Конструктор класса `md5h`.

Инициализирует соль для использования при генерации хеша.

4.5.3 Методы

4.5.3.1 `generation_SALT()`

`std::string md5h::generation_SALT () [private]`

Генератор SALT.

Генерация случайной соли

Возвращает

`tmp_s SALT`

Сгенерированная строка с солью

4.5.3.2 `hash()`

`std::string md5h::hash (`
`const string & PASS)`

Получение хэша пароля

Генерация хеша MD5.

Аргументы

PASS	- пароль
------	----------

Возвращает

digest - хэш от пароля

Аргументы

PASS	Пароль для хеширования
------	------------------------

Возвращает

Строка с хешем

Объявления и описания членов классов находятся в файлах:

- [md5h.hpp](#)
- [md5h.cpp](#)

4.6 Класс server

Класс управления работой сервера

```
#include <server.hpp>
```

Открытые члены

- [server](#) (std::string file_path, int port, std::string [local_path](#))
Конструктор класса
- int [start](#) (int port, const char *address)
Запуск сервера и привязка порта
- int [connect](#) ()
Функция ожидания и принятия соединения с клиентом
- bool [authentication](#) ()
Функция аутентификации пользователя
- bool [operations](#) ()
Функция обработки векторов

Закрытые данные

- `std::string local_path`
Путь к файлу с логом
- `int sckt`
Сокет
- `int wrkr`
Действующее соединение с клиентом
- `int rc`
Ответ от клиента
- `char * buffer = new char[4096]`
Буфер
- `std::map< std::string, std::string > vault`
Данные клиента ID:PASS.

4.6.1 Подробное описание

Класс управления работой сервера

4.6.2 Конструктор(ы)

4.6.2.1 server()

```
server::server (
    std::string file_path,
    int port,
    std::string local_path )
```

Конструктор класса

Конструктор класса server.

Аргументы

file_path	Путь к файлу с ID:PASS клиентов
port	Порт для приема соединения
local_path	Путь к файлу журнала
file_path	- путь к файлу с данными
port	- порт сервера
local_path	- путь к локальному файлу

4.6.3 Методы

4.6.3.1 authentication()

```
bool server::authentication ( )
```

Функция аутентификации пользователя

Метод аутентификации клиента

Возвращает

True, если аутентификация удалась, False - если нет

bool - результат аутентификации (true - успешно, false - ошибка)

4.6.3.2 connect()

```
int server::connect ( )
```

Функция ожидания и принятия соединения с клиентом

Метод для соединения с клиентом

Возвращает

wrkr Действующее соединение с клиентом

wrkr - рабочий сокет

4.6.3.3 operations()

```
bool server::operations ( )
```

Функция обработки векторов

Метод для выполнения операций

Возвращает

True, если обработка удалась, False - если нет

bool - результат выполнения операций (true - успешно, false - ошибка)

4.6.3.4 start()

```
int server::start (
    int port,
    const char * address )
```

Запуск сервера и привязка порта

Метод для запуска сервера

Аргументы

port	Порт для приема соединения
address	Сетевой адрес для приема соединения

Возвращает

sckt Сокет

Аргументы

port	- порт сервера
address	- IP-адрес сервера

Возвращает

sckt - сокет сервера

Объявления и описания членов классов находятся в файлах:

- [server.hpp](#)
- [server.cpp](#)

Глава 5

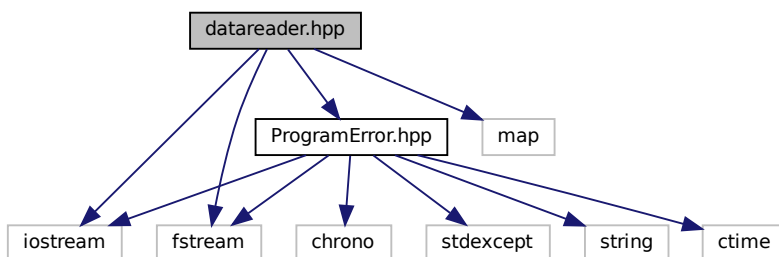
Файлы

5.1 Файл datareader.hpp

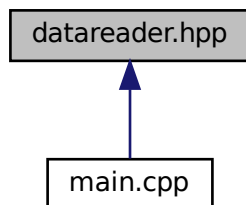
Считывание данных о клиентах

```
#include <iostream>
#include <fstream>
#include <map>
#include "ProgramError.hpp"
```

Граф включаемых заголовочных файлов для datareader.hpp:



Граф файлов, в которые включается этот файл:



Классы

- class [datareader](#)

Класс для чтения данных из файла и формирования хранилища пользователей

5.1.1 Подробное описание

Считывание данных о клиентах

Автор

Шатов В.С.

Версия

1.08

Дата

26.12.2023

Авторство

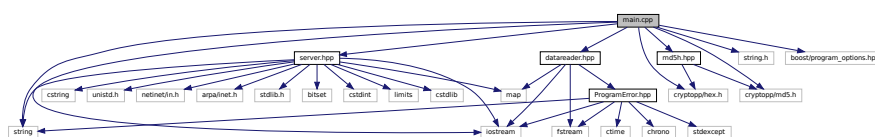
ИБСТ ПГУ

5.2 Файл main.cpp

Справка по использованию

```
#include <iostream>
#include <string>
#include "server.hpp"
#include "datareader.hpp"
#include <cryptopp/hex.h>
#include "md5h.hpp"
#include <string.h>
#include <cryptopp/md5.h>
#include <boost/program_options.hpp>
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- void `help` ()
Выводит справку по использованию аргументов командной строки
- int `main` (int argc, const char *argv[])
Главная функция приложения сервера

Переменные

- ```
struct {
 uint32_t p = 33333
 Порт по умолчанию
 std::string d = "data.txt"
 Путь к файлу данных по умолчанию
 std::string l = "log.txt"
 Путь к файлу журнала по умолчанию
} parameters
```

Структура для хранения параметров сервера по умолчанию

### 5.2.1 Подробное описание

#### Справка по использованию

##### Автор

Шатов В.С.

##### Версия

1.08

##### Дата

26.12.2023

##### Авторство

ИБСТ ПГУ

### 5.2.2 Функции

#### 5.2.2.1 main()

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

Главная функция приложения сервера

## Аргументы

|      |                                        |
|------|----------------------------------------|
| argc | Количество аргументов командной строки |
| argv | Массив аргументов командной строки     |

## Возвращает

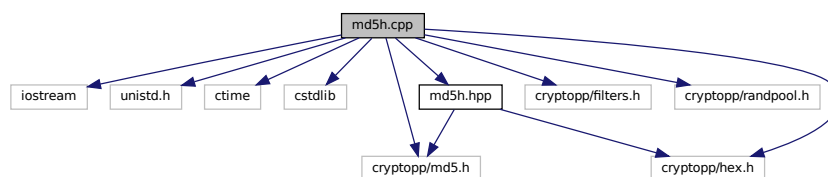
Код завершения

## 5.3 Файл md5h.cpp

Класс обработки MD5 Hash.

```
#include <iostream>
#include <unistd.h>
#include <ctime>
#include <cstdlib>
#include <cryptopp/md5.h>
#include <cryptopp/hex.h>
#include <cryptopp/filters.h>
#include <cryptopp/randpool.h>
#include "md5h.hpp"
```

Граф включаемых заголовочных файлов для md5h.cpp:



## Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

## 5.3.1 Подробное описание

Класс обработки MD5 Hash.

## Автор

Шатов В.С.

## Версия

1.08

## Дата

26.12.2023

## Авторство

ИБСТ ПГУ Содержит главную функцию для запуска сервера, включая разбор аргументов командной строки и обработку ошибок.

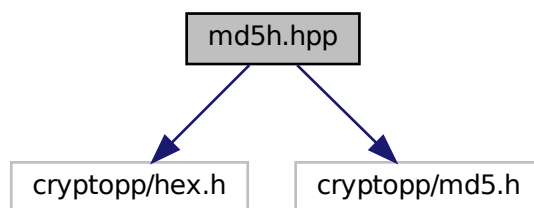


## 5.4 Файл md5h.hpp

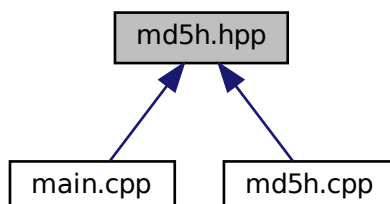
Класс обработки MD5 Hash.

```
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для md5h.hpp:



Граф файлов, в которые включается этот файл:



### Классы

- class `md5h`  
Класс обработки MD5 Hash.

### Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

### 5.4.1 Подробное описание

Класс обработки MD5 Hash.

Автор

Шатов В.С.

Версия

1.08

Дата

26.12.2023

Авторство

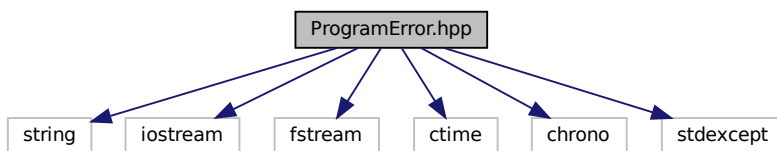
ИБСТ ПГУ

## 5.5 Файл ProgramError.hpp

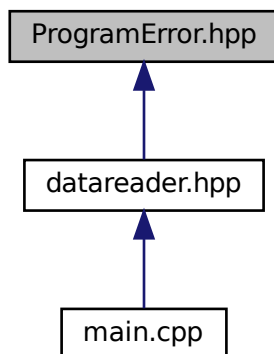
Обработка ошибок, с последующей записью в журнал

```
#include <string>
#include <iostream>
#include <fstream>
#include <ctime>
#include <chrono>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для ProgramError.hpp:



Граф файлов, в которые включается этот файл:



## Классы

- class [ErrorRead](#)  
Класс обработки ошибок чтения файла данных пользователей
- class [ErrorLogin](#)  
Класс обработки ошибок записи в журнал
- class [ErrorAuth](#)  
Класс обработки ошибок аутентификации

## Функции

- std::string [log\\_entry](#) (std::string local\_path, std::string what)  
Функция записи ошибок в журнал

### 5.5.1 Подробное описание

Обработка ошибок, с последующей записью в журнал

Автор

Шатов В.С.

Версия

1.08

Дата

26.12.2023

Авторство

ИБСТ ПГУ

## 5.5.2 Функции

### 5.5.2.1 log\_entry()

```
std::string log_entry (
 std::string local_path,
 std::string what) [inline]
```

Функция записи ошибок в журнал

Аргументы

|            |                        |
|------------|------------------------|
| local_path | - путь к файлу журнала |
| what       | - содержание ошибки    |

Возвращает

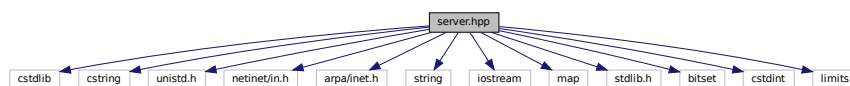
description\_string - строка с логом ошибки

## 5.6 Файл server.hpp

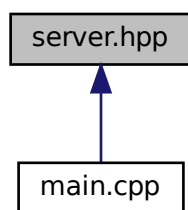
Операции выполняемые сервером

```
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string>
#include <iostream>
#include <map>
#include <stdlib.h>
#include <bitset>
#include <cstdint>
#include <limits>
```

Граф включаемых заголовочных файлов для server.hpp:



Граф файлов, в которые включается этот файл:



## Классы

- class `server`  
Класс управления работой сервера

### 5.6.1 Подробное описание

Операции выполняемые сервером

Автор

Шатов В.С.

Версия

1.08

Дата

26.12.2023

Авторство

ИБСТ ПГУ



# Предметный указатель

- authentication
  - server, [14](#)
- connect
  - server, [15](#)
- datareader, [7](#)
  - datareader, [7](#)
  - get, [8](#)
- datareader.hpp, [17](#)
- ErrorAuth, [8](#)
- ErrorLogin, [9](#)
- ErrorRead, [10](#)
- generation\_SALT
  - md5h, [12](#)
- get
  - datareader, [8](#)
- hash
  - md5h, [12](#)
- log\_entry
  - ProgramError.hpp, [24](#)
- main
  - main.cpp, [19](#)
- main.cpp, [18](#)
  - main, [19](#)
- md5h, [11](#)
  - generation\_SALT, [12](#)
  - hash, [12](#)
  - md5h, [12](#)
- md5h.cpp, [20](#)
- md5h.hpp, [21](#)
- operations
  - server, [15](#)
- ProgramError.hpp, [22](#)
  - log\_entry, [24](#)
- server, [13](#)
  - authentication, [14](#)
  - connect, [15](#)
  - operations, [15](#)
  - server, [14](#)
  - start, [15](#)
- server.hpp, [24](#)
- start
  - server, [15](#)