

My Project

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 Error	5
3.1.1 Подробное описание	5
3.1.2 Конструктор(ы)	5
3.1.2.1 Error() [1/2]	5
3.1.2.2 Error() [2/2]	6
3.2 Класс Interface	6
3.2.1 Подробное описание	6
3.2.2 Методы	6
3.2.2.1 get_file_error()	7
3.2.2.2 get_file_name()	7
3.2.2.3 get_port()	7
3.3 Структура InterfaceTest	7
3.4 Класс Server	8
3.4.1 Подробное описание	8
3.4.2 Конструктор(ы)	8
3.4.2.1 Server()	8
4 Файлы	9
4.1 Файл error.h	9
4.1.1 Подробное описание	10
4.2 Файл interface.cpp	10
4.2.1 Подробное описание	10
4.3 Файл interface.h	11
4.3.1 Подробное описание	11
4.4 Файл main.cpp	12
4.4.1 Подробное описание	12
4.5 Файл mdfile.cpp	13
4.5.1 Подробное описание	13
4.6 Файл mdfile.h	14
4.6.1 Подробное описание	15
4.7 Файл Server.h	15
4.7.1 Подробное описание	16
4.8 Файл TEST.cpp	16
4.8.1 Подробное описание	17
Предметный указатель	19



# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">Error</a>	Класс описания ошибок . . . . .	5
<a href="#">Interface</a>	Класс для описывания интерфейса для взаимодействия с командной строкой . .	6
<a href="#">InterfaceTest</a>	. . . . .	7
<a href="#">Server</a>	Класс с методами для сервера . . . . .	8



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">error.h</a>	Файл с классом <a href="#">Error</a> . . . . .	9
<a href="#">interface.cpp</a>	File <a href="#">interface.cpp</a> . . . . .	10
<a href="#">interface.h</a>	Главный файл сервера . . . . .	11
<a href="#">main.cpp</a>	File <a href="#">main.cpp</a> . . . . .	12
<a href="#">mdfile.cpp</a>	File <a href="#">mdfile.cpp</a> . . . . .	13
<a href="#">mdfile.h</a>	File <a href="#">mdfile.h</a> . . . . .	14
<a href="#">Server.h</a>	Файл с классом <a href="#">Server</a> . . . . .	15
<a href="#">TEST.cpp</a>	File <a href="#">TEST.cpp</a> . . . . .	16





## Глава 3

# Классы

### 3.1 Класс Error

Класс описания ошибок

```
#include <error.h>
```

Открытые члены

- [Error](#) (const string &error)  
Конструктор класса [Error](#), который инициализирует поля `err_file` и `errMessage`.
- [Error](#) (const [Error](#) &name)  
Конструктор копирования класса [Error](#).
- void [errors](#) (string &error)  
объявление функции `errors` в классе [Error](#). Функция ожидает один параметр - ссылку на объект типа `string` с именем `error` и не возвращает никакого значения.

#### 3.1.1 Подробное описание

Класс описания ошибок

#### 3.1.2 Конструктор(ы)

##### 3.1.2.1 Error() [1/2]

```
Error::Error (  
    const string & error )    [inline]
```

Конструктор класса [Error](#), который инициализирует поля `err_file` и `errMessage`.

Конструктор принимает константную ссылку на объект типа `string` и использует её для инициализации поля `err_file`. Поле `errMessage` инициализируется пустой строкой.

### 3.1.2.2 Error() [2/2]

```
Error::Error (
    const Error & name ) [inline]
```

Конструктор копирования класса [Error](#).

Конструктор копирования принимает ссылку на объект типа [Error](#) и использует её для инициализации поля `err_file`. Таким образом, он создает новый объект [Error](#), который является копией существующего объекта [Error](#), используя значение поля `err_file` из переданного объекта.

Объявления и описания членов классов находятся в файлах:

- [error.h](#)
- [mdfile.cpp](#)

## 3.2 Класс Interface

Класс для описывания интерфейса для взаимодействия с командной строкой

```
#include <interface.h>
```

Открытые члены

- `bool get\_options\_from\_comline (int argc, char *argv[])`  
Определяется сигнатура (прототип) метода `get_options_from_comline` класса [Interface](#).
- `int get\_port ()`  
Эта строчка определяет метод `get_port`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает целочисленное значение (`int`).
- `string get\_file\_name ()`  
Эта строчка определяет метод `get_file_name`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает значение `string`.
- `string get\_file\_error ()`  
Эта строчка определяет метод `get_file_error`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает значение `string`.

### 3.2.1 Подробное описание

Класс для описывания интерфейса для взаимодействия с командной строкой

### 3.2.2 Методы

#### 3.2.2.1 get\_file\_error()

```
string Interface::get_file_error ( ) [inline]
```

Эта строчка определяет метод `get_file_error`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает значение `string`.

Когда этот метод вызывается для объекта класса [Interface](#), он возвращает значение переменной `file_error`, которое является приватным членом класса. В данном случае, метод `get_file_error` предоставляет возможность получения значения порта, хранящегося в объекте [Interface](#).

#### 3.2.2.2 get\_file\_name()

```
string Interface::get_file_name ( ) [inline]
```

Эта строчка определяет метод `get_file_name`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает значение `string`.

Когда этот метод вызывается для объекта класса [Interface](#), он возвращает значение переменной `file_name`, которое является приватным членом класса. В данном случае, метод `get_file_name` предоставляет возможность получения значения порта, хранящегося в объекте [Interface](#).

#### 3.2.2.3 get\_port()

```
int Interface::get_port ( ) [inline]
```

Эта строчка определяет метод `get_port`, который принадлежит классу [Interface](#). Метод не принимает аргументов и возвращает целочисленное значение (`int`).

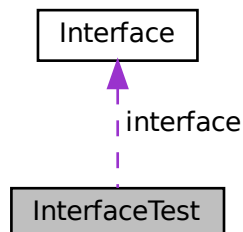
Когда этот метод вызывается для объекта класса [Interface](#), он возвращает значение переменной `port`, которое является приватным членом класса. В данном случае, метод `get_port` предоставляет возможность получения значения порта, хранящегося в объекте [Interface](#).

Объявления и описания членов классов находятся в файлах:

- [interface.h](#)
- [interface.cpp](#)

### 3.3 Структура InterfaceTest

Граф связей класса `InterfaceTest`:



## Открытые атрибуты

- [Interface](#) interface

Объявления и описания членов структуры находятся в файле:

- [TEST.cpp](#)

## 3.4 Класс Server

Класс с методами для сервера

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

### Открытые члены

- `int self_addr (string error, int port)`  
Объявляется функция `self_addr` в классе [Server](#). Функция ожидает два параметра: строку `error` и целочисленный `port`, и возвращает целочисленное значение.
- `int client_addr (int s, string error)`  
Объявляется функция `client_addr` в классе [Server](#). Функция ожидает два параметра: строку `error` и целочисленный `s`, и возвращает целочисленное значение.
- `int authorized (int work_sock, string file_name)`  
Объявляется функция `authorized` в классе [Server](#). Функция ожидает два параметра: строку `file_name` и целочисленный `work_sock`, и возвращает целочисленное значение.
- `int math (int work_sock)`  
Объявляется функция `authorized` в классе `math`. Функция ожидает целочисленный параметр `work_sock`, и возвращает целочисленное значение.
- [Server](#) ([Error](#) &wer)  
Конструктор класса [Server](#), который инициализирует поле `eg` объектом типа [Error](#) по ссылке.

### 3.4.1 Подробное описание

Класс с методами для сервера

### 3.4.2 Конструктор(ы)

#### 3.4.2.1 Server()

```
Server::Server (  
    Error & wer ) [inline]
```

Конструктор класса [Server](#), который инициализирует поле `eg` объектом типа [Error](#) по ссылке.

Конструктор принимает ссылку на объект типа [Error](#) и использует её для инициализации поля `eg`.

Объявления и описания членов классов находятся в файлах:

- [Server.h](#)
- [mdfile.cpp](#)

## Глава 4

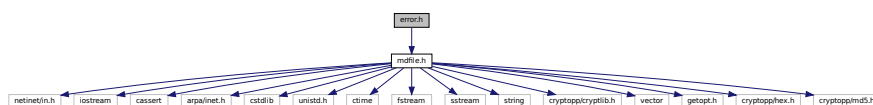
# Файлы

### 4.1 Файл error.h

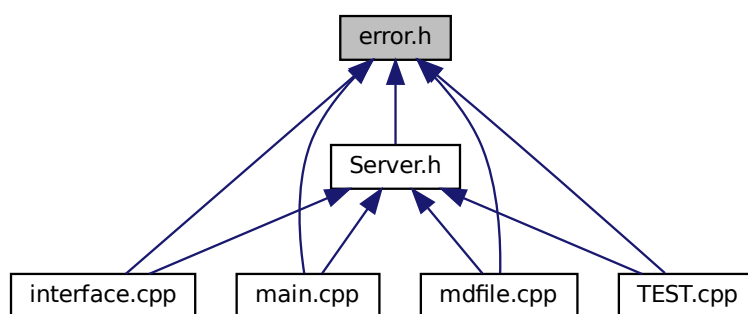
Файл с классом [Error](#).

```
#include "mdfile.h"
```

Граф включаемых заголовочных файлов для error.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Error](#)

Класс описания ошибок

### 4.1.1 Подробное описание

Файл с классом [Error](#).

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

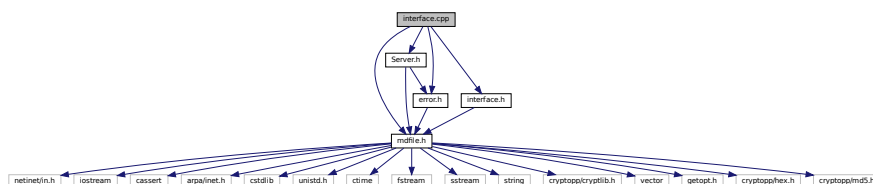
ИБСТ ПГУ

## 4.2 Файл interface.cpp

file [interface.cpp](#)

```
#include "mdfile.h"  
#include "Server.h"  
#include "error.h"  
#include "interface.h"
```

Граф включаемых заголовочных файлов для interface.cpp:



### 4.2.1 Подробное описание

file [interface.cpp](#)

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

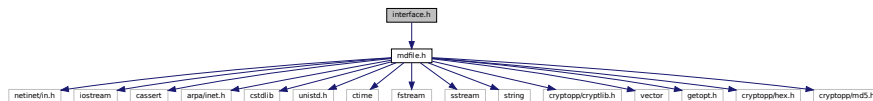
ИБСТ ПГУ

## 4.3 Файл interface.h

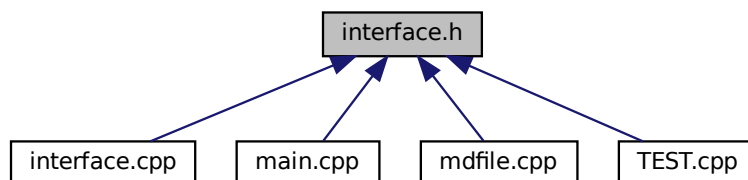
Главный файл сервера

```
#include "mdfile.h"
```

Граф включаемых заголовочных файлов для interface.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Interface](#)

Класс для описывания интерфейса для взаимодействия с командной строкой

### 4.3.1 Подробное описание

Главный файл сервера

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

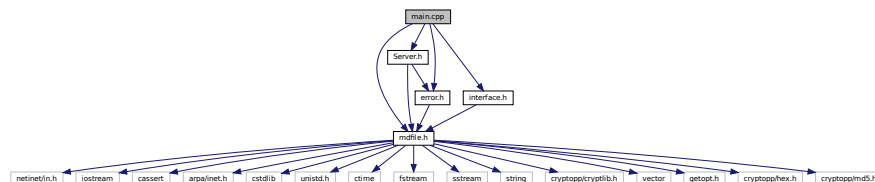
ИБСТ ПГУ

## 4.4 Файл main.cpp

file [main.cpp](#)

```
#include "mdfile.h"
#include "Server.h"
#include "error.h"
#include "interface.h"
```

Граф включаемых заголовочных файлов для main.cpp:



### Функции

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

### 4.4.1 Подробное описание

file [main.cpp](#)

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

ИБСТ ПГУ

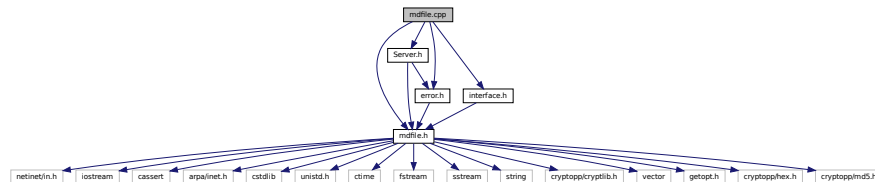


## 4.5 Файл mdfile.cpp

file [mdfile.cpp](#)

```
#include "mdfile.h"  
#include "Server.h"  
#include "error.h"  
#include "interface.h"
```

Граф включаемых заголовочных файлов для mdfile.cpp:



### Функции

- void msgsend (int work\_sock, string mess)
- string MD (string sah)

#### 4.5.1 Подробное описание

file [mdfile.cpp](#)

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

ИБСТ ПГУ

## 4.6 Файл mdfile.h

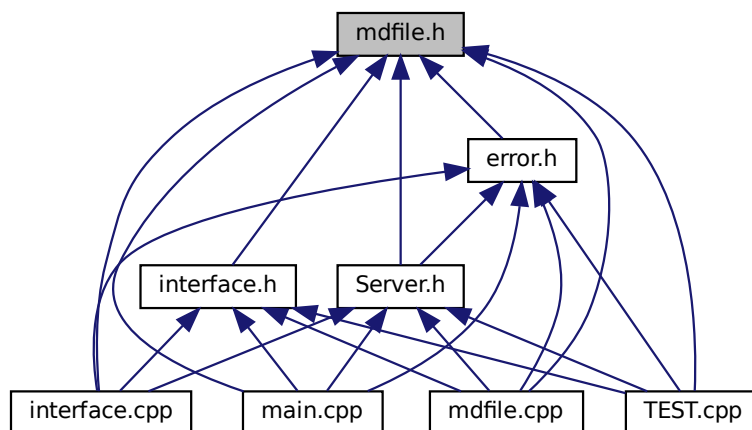
file [mdfile.h](#)

```
#include <netinet/in.h>
#include <iostream>
#include <cassert>
#include <arpa/inet.h>
#include <cstdlib>
#include <unistd.h>
#include <ctime>
#include <fstream>
#include <sstream>
#include <string>
#include <cryptopp/cryptlib.h>
#include <vector>
#include <getopt.h>
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для mdfile.h:



Граф файлов, в которые включается этот файл:



Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

## Функции

- string MD (string sah)
- int er (string file\_name, string file\_error)
- void msgsend (int work\_sock, string mess)
- int authorized (int work\_sock, string file\_name, string file\_error)
- int math (int work\_sock)

### 4.6.1 Подробное описание

file [mdfile.h](#)

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

ИБСТ ПГУ

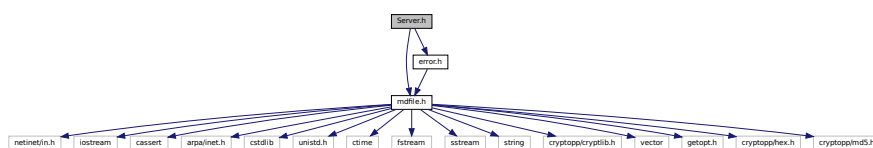
## 4.7 Файл Server.h

Файл с классом [Server](#).

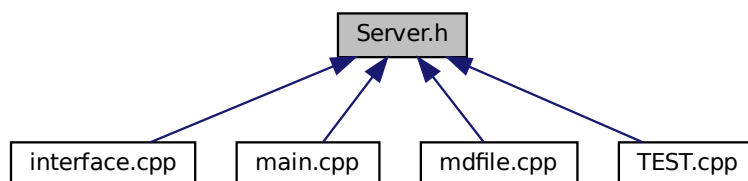
```
#include "mdfile.h"
```

```
#include "error.h"
```

Граф включаемых заголовочных файлов для Server.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Server](#)

Класс с методами для сервера

### 4.7.1 Подробное описание

Файл с классом [Server](#).

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

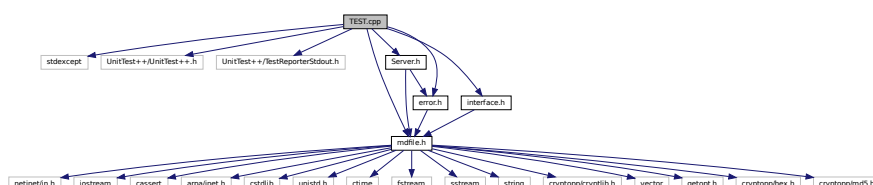
ИБСТ ПГУ

## 4.8 Файл TEST.cpp

file [TEST.cpp](#)

```
#include <stdexcept>
#include <UnitTest++/UnitTest++.h>
#include <UnitTest++/TestReporterStdout.h>
#include "mdfile.h"
#include "Server.h"
#include "error.h"
#include "interface.h"
```

Граф включаемых заголовочных файлов для TEST.cpp:



## Классы

- struct [InterfaceTest](#)

## Функции

- SUITE (ComLineTests)
- int main (int argc, char \*\*argv)

### 4.8.1 Подробное описание

file [TEST.cpp](#)

Автор

Коросткин А.П.

Версия

1.0

Дата

22.12.2023

Авторство

ИБСТ ПГУ

нужен для модульного тестирования



# Предметный указатель

- Error, [5](#)
  - Error, [5](#)
- error.h, [9](#)
- get\_file\_error
  - Interface, [6](#)
- get\_file\_name
  - Interface, [7](#)
- get\_port
  - Interface, [7](#)
- Interface, [6](#)
  - get\_file\_error, [6](#)
  - get\_file\_name, [7](#)
  - get\_port, [7](#)
- interface.cpp, [10](#)
- interface.h, [11](#)
- InterfaceTest, [7](#)
- main.cpp, [12](#)
- mdfile.cpp, [13](#)
- mdfile.h, [14](#)
- Server, [8](#)
  - Server, [8](#)
- Server.h, [15](#)
- TEST.cpp, [16](#)