

Сетевой сервер

Создано системой Doxygen 1.9.1

1 Алфавитный указатель классов	1
1.1 Классы	1
2 Список файлов	2
2.1 Файлы	2
3 Классы	2
3.1 Класс Calculator	2
3.1.1 Подробное описание	3
3.1.2 Методы	3
3.2 Класс ErrorHandler	4
3.2.1 Подробное описание	4
3.3 Класс LogWriter	4
3.3.1 Подробное описание	4
3.3.2 Конструктор(ы)	4
3.3.3 Методы	5
3.4 Класс WaitingMode	5
3.4.1 Подробное описание	6
3.4.2 Конструктор(ы)	6
4 Файлы	7
4.1 Файл Calculator.h	7
4.1.1 Подробное описание	8
4.2 Файл Communicator.h	8
4.2.1 Подробное описание	9
4.3 Файл ErrorHandler.h	9
4.3.1 Подробное описание	10
4.4 Файл LogWriter.h	10
4.4.1 Подробное описание	11
4.5 Файл mdfile.h	11
4.5.1 Подробное описание	12
4.6 Файл WaitingMode.h	13
4.6.1 Подробное описание	13
Предметный указатель	15

## 1 Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

#### Calculator

Класс для вычисления произведений

2

<a href="#">ErrorHandling</a>	4
Класс для обработки ошибок	
<a href="#">LogWriter</a>	4
Класс для записи логов	
<a href="#">WaitingMode</a>	5
Класс для работы сервера в режиме ожидания	

## 2 Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">Calculator.h</a>	7
Заголовочный файл для модуля <a href="#">Calculator</a>	
<a href="#">Communicator.h</a>	8
Заголовочный файл для модуля <a href="#">WaitingMode</a>	
<a href="#">ErrorHandling.h</a>	9
Заголовочный файл для модуля <a href="#">ErrorHandling</a>	
<a href="#">LogWriter.h</a>	10
Заголовочный файл для модуля <a href="#">LogWriter</a>	
<a href="#">mdfile.h</a>	11
Заголовочный файл для модуля <a href="#">mdfile.h</a>	
<a href="#">WaitingMode.h</a>	13
Заголовочный файл для модуля <a href="#">WaitingMode</a>	

## 3 Классы

### 3.1 Класс Calculator

Класс для вычисления произведений

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

Открытые члены

- `std::vector< int32_t > calculateProducts (const std::vector< std::vector< int32_t >> &vectors)`  
Вычисление произведений для каждого вектора
- `int32_t calculateProduct (const std::vector< int32_t > &numbers)`  
Вычисление произведения чисел в векторе

### 3.1.1 Подробное описание

Класс для вычисления произведений

Класс содержит два метода для вычисления произведений чисел.

Предупреждения

Реализация только для типа данных `int32_t`

### 3.1.2 Методы

3.1.2.1 `calculateProduct()` `int32_t Calculator::calculateProduct (`  
`const std::vector< int32_t > & numbers )`

Вычисление произведения чисел в векторе

Аргументы

in	numbers	Вектор чисел, для которых нужно вычислить произведение
----	---------	--

Возвращает

Произведение чисел

3.1.2.2 `calculateProducts()` `std::vector<int32_t> Calculator::calculateProducts (`  
`const std::vector< std::vector< int32_t >> & vectors )`

Вычисление произведений для каждого вектора

Аргументы

in	vectors	Вектор векторов, для которых нужно вычислить произведения
----	---------	---

Возвращает

Вектор произведений

Объявления и описания членов классов находятся в файлах:

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

## 3.2 Класс `ErrorHandling`

Класс для обработки ошибок

```
#include <ErrorHandling.h>
```

Открытые члены

- [ErrorHandling](#) ()  
Конструктор по умолчанию
- [ErrorHandling](#) (const [ErrorHandling](#) &)  
Конструктор копирования
- void [watchCritical](#) ()  
Отслеживание критических ошибок
- void [watchMinor](#) ()  
Отслеживание незначительных ошибок
- void [watchInfo](#) ()  
Отслеживание информационных сообщений

### 3.2.1 Подробное описание

Класс для обработки ошибок

Класс содержит методы для отслеживания критических, незначительных ошибок и информационных сообщений.

Объявления и описания членов класса находятся в файле:

- [ErrorHandling.h](#)

## 3.3 Класс `LogWriter`

Класс для записи логов

```
#include <LogWriter.h>
```

Открытые члены

- [LogWriter](#) (const std::string &errorLogFileName)  
Конструктор класса
- [~LogWriter](#) ()  
Деструктор класса
- void [writeLog](#) (const std::string &logMessage)  
Запись лога

### 3.3.1 Подробное описание

Класс для записи логов

Класс содержит методы для записи логов в файл.

### 3.3.2 Конструктор(ы)

3.3.2.1 `LogWriter()` `LogWriter::LogWriter (`  
                  const std::string & errorLogFileName ) [inline]

Конструктор класса

## Аргументы

in	errorLogFileName	Имя файла для записи логов
----	------------------	----------------------------

## 3.3.3 Методы

3.3.3.1 writeLog() void LogWriter::writeLog (  
const std::string & logMessage )

## Запись лога

## Аргументы

in	logMessage	Сообщение для записи в лог
----	------------	----------------------------

Объявления и описания членов классов находятся в файлах:

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

## 3.4 Класс WaitingMode

Класс для работы сервера в режиме ожидания

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

## Открытые члены

- [WaitingMode](#) ()=delete  
Запретим конструктор без параметров
- [WaitingMode](#) (int port, int queueLength, [ErrorHandling](#) &errorHandling, [LogWriter](#) &logWriter, std::string clients)  
Конструктор класса
- void [runServer](#) ()  
Запуск сервера
- [WaitingMode](#) ()=delete  
Запретим конструктор без параметров
- [WaitingMode](#) (int port, int queueLength, [ErrorHandling](#) &errorHandling, [LogWriter](#) &logWriter, std::string clients)  
Конструктор класса
- void [runServer](#) ()  
Запуск сервера

### 3.4.1 Подробное описание

Класс для работы сервера в режиме ожидания

Класс содержит методы для авторизации клиента, выполнения вычислений, отправки сообщений и запуска сервера.

### 3.4.2 Конструктор(ы)

3.4.2.1 `WaitingMode()` [1/2] `WaitingMode::WaitingMode (`  
    `int port,`  
    `int queueLength,`  
    [`ErrorHandling`](#) `& errorHandler,`  
    [`LogWriter`](#) `& logWriter,`  
    `std::string clients )`

Конструктор класса

Аргументы

in	port	Порт для подключения
in	queueLength	Длина очереди
in	errorHandling	Обработчик ошибок
in	logWriter	Записыватель логов
in	clients	Клиенты

3.4.2.2 `WaitingMode()` [2/2] `WaitingMode::WaitingMode (`  
    `int port,`  
    `int queueLength,`  
    [`ErrorHandling`](#) `& errorHandler,`  
    [`LogWriter`](#) `& logWriter,`  
    `std::string clients )`

Конструктор класса

Аргументы

in	port	Порт для подключения
in	queueLength	Длина очереди
in	errorHandling	Обработчик ошибок
in	logWriter	Записыватель логов
in	clients	Клиенты

Объявления и описания членов классов находятся в файлах:

- [Communicator.h](#)
- [WaitingMode.h](#)
- `worker.cpp`

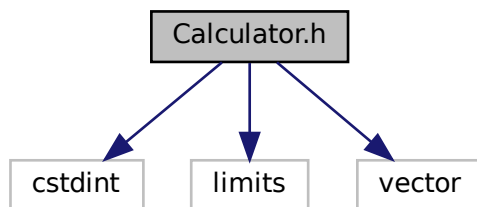
## 4 Файлы

### 4.1 Файл Calculator.h

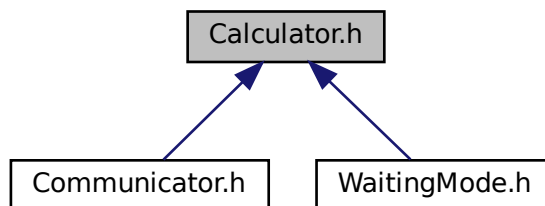
Заголовочный файл для модуля [Calculator](#).

```
#include <cstdint>
#include <limits>
#include <vector>
```

Граф включаемых заголовочных файлов для Calculator.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Calculator](#)  
Класс для вычисления произведений



#### 4.1.1 Подробное описание

Заголовочный файл для модуля [Calculator](#).

Автор

Казанкин Максим

Версия

1.0

Дата

22.12.23

Авторство

ИБСТ ПГУ

Предупреждения

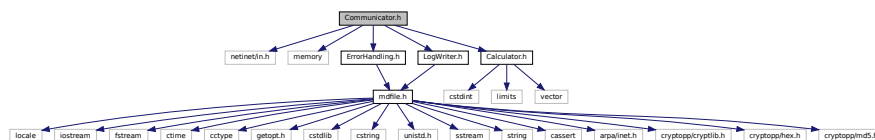
Курсовая работа

## 4.2 Файл Communicator.h

Заголовочный файл для модуля [WaitingMode](#).

```
#include <netinet/in.h>
#include <memory>
#include "ErrorHandling.h"
#include "Calculator.h"
#include "LogWriter.h"
```

Граф включаемых заголовочных файлов для Communicator.h:



Классы

- class [WaitingMode](#)  
Класс для работы сервера в режиме ожидания

## 4.2.1 Подробное описание

Заголовочный файл для модуля [WaitingMode](#).

Автор

Казанкин Максим

Версия

1.0

Дата

22.12.23

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа

## 4.3 Файл ErrorHandler.h

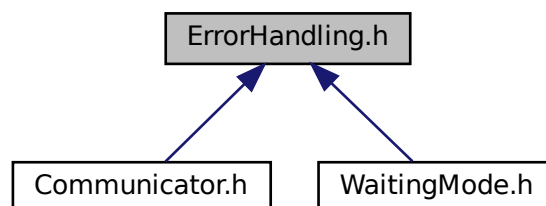
Заголовочный файл для модуля [ErrorHandling](#).

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

Граф включаемых заголовочных файлов для ErrorHandler.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [ErrorHandling](#)  
Класс для обработки ошибок

### 4.3.1 Подробное описание

Заголовочный файл для модуля [ErrorHandling](#).

Автор

Казанкин Максим

Версия

1.0

Дата

22.12.23

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа

## 4.4 Файл LogWriter.h

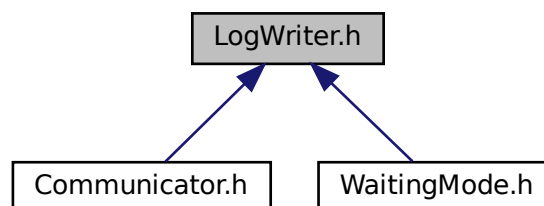
Заголовочный файл для модуля [LogWriter](#).

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

Граф включаемых заголовочных файлов для LogWriter.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [LogWriter](#)  
Класс для записи логов

### 4.4.1 Подробное описание

Заголовочный файл для модуля [LogWriter](#).

## Автор

Казанкин Максим

## Версия

1.0

## Дата

22.12.23

## Авторство

ИБСТ ПГУ

## Предупреждения

Курсовая работа

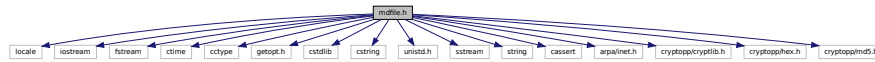
## 4.5 Файл mdfile.h

Заголовочный файл для модуля [mdfile.h](#).

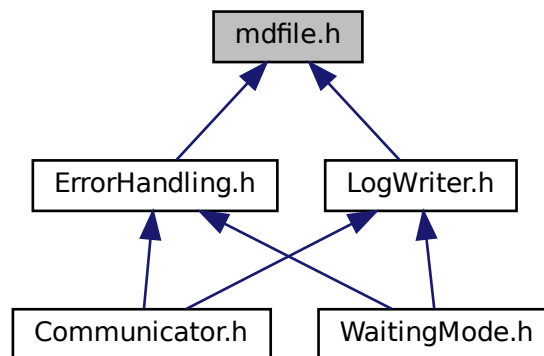
```
#include <locale>
#include <iostream>
#include <fstream>
#include <ctime>
#include <cctype>
#include <getopt.h>
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <sstream>
#include <string>
#include <cassert>
#include <arpa/inet.h>
#include <cryptopp/cryptlib.h>
#include <cryptopp/hex.h>
```

```
#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`

### 4.5.1 Подробное описание

Заголовочный файл для модуля [mdfile.h](#).

Автор

Казанкин Максим

Версия

1.0

Дата

22.12.23

Авторство

ИБСТ ПГУ

Предупреждения

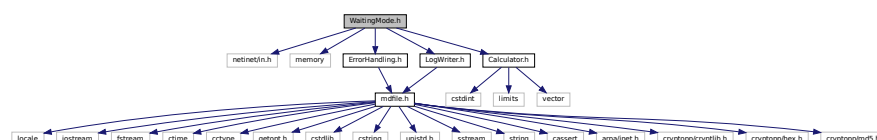
Курсовая работа

## 4.6 Файл WaitingMode.h

Заголовочный файл для модуля [WaitingMode](#).

```
#include <netinet/in.h>
#include <memory>
#include "ErrorHandling.h"
#include "Calculator.h"
#include "LogWriter.h"
```

Граф включаемых заголовочных файлов для WaitingMode.h:



### Классы

- class [WaitingMode](#)  
Класс для работы сервера в режиме ожидания

#### 4.6.1 Подробное описание

Заголовочный файл для модуля [WaitingMode](#).

### Автор

Казанкин Максим

### Версия

1.0

### Дата

22.12.23

### Авторство

ИБСТ ПГУ

### Предупреждения

Курсовая работа



## Предметный указатель

- calculateProduct
  - Calculator, [3](#)
- calculateProducts
  - Calculator, [3](#)
- Calculator, [2](#)
  - calculateProduct, [3](#)
  - calculateProducts, [3](#)
- Calculator.h, [7](#)
- Communicator.h, [8](#)
- ErrorHandling, [4](#)
- ErrorHandling.h, [9](#)
- LogWriter, [4](#)
  - LogWriter, [4](#)
  - writeLog, [5](#)
- LogWriter.h, [10](#)
- mdfile.h, [11](#)
- WaitingMode, [5](#)
  - WaitingMode, [6](#)
- WaitingMode.h, [13](#)
- writeLog
  - LogWriter, [5](#)