

Course paper

1.0

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 Authorized	5
3.1.1 Подробное описание	6
3.1.2 Конструктор(ы)	6
3.1.2.1 Authorized()	6
3.1.3 Методы	6
3.1.3.1 authorized()	6
3.2 Класс Calculator	7
3.2.1 Подробное описание	8
3.2.2 Конструктор(ы)	8
3.2.2.1 Calculator()	8
3.2.3 Методы	8
3.2.3.1 calc()	8
3.3 Класс Error	9
3.3.1 Подробное описание	9
3.4 Класс Server	9
3.4.1 Подробное описание	10
3.4.2 Конструктор(ы)	10
3.4.2.1 Server()	10
3.4.3 Методы	10
3.4.3.1 client_addr()	10
3.4.3.2 self_addr()	11
4 Файлы	13
4.1 Файл calc.h	13
4.1.1 Подробное описание	14
Предметный указатель	15



# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">Authorized</a>	Класс авторизации . . . . .	5
<a href="#">Calculator</a>	Класс калькулятор . . . . .	7
<a href="#">Error</a>	Класс ошибок . . . . .	9
<a href="#">Server</a>	Класс Сервер . . . . .	9



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">calc.h</a>	
Заголовочный файл программы сервера . . . . .	13





## Глава 3

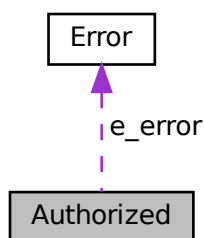
# Классы

### 3.1 Класс Authorized

Класс авторизации

```
#include <calc.h>
```

Граф связей класса Authorized:



Открытые члены

- [Authorized](#) ([Error](#) err)
- [int authorized](#) (int work\_sock, string file\_name, string file\_error)  
Функция авторизации

Закрытые члены

- [std::string salt\\_generator](#) (const [std::size\\_t](#) length)  
Метод для генерации соли
- [void msgsend](#) (int work\_sock, const [std::string](#) &mess)  
Метод для отправки сообщения
- [std::string MD](#) (const [std::string](#) &sah)  
Метод для вычисления хэша

## Закрытые данные

- [Error e\\_error](#)

Приватный член класса [Error](#).

### 3.1.1 Подробное описание

#### Класс авторизации

Данный класс создан для авторизации клиента на сервере и проверки логина и пароля в базе данных при подключении пользователя

#### Предупреждения

Пользователь должен быть в базе данных

### 3.1.2 Конструктор(ы)

#### 3.1.2.1 Authorized()

```
Authorized::Authorized (
    Error err ) [inline]
```

< Конструктор класса [Authorized](#)

< Присваивание объекта err приватному члену e\_error

### 3.1.3 Методы

#### 3.1.3.1 authorized()

```
int Authorized::authorized (
    int work_sock,
    string file_name,
    string file_error )
```

#### Функция авторизации

Функция выполняет авторизации клиента на сервере

#### Аргументы

in	work_sock	Передаваемый сокетом для работы с сетью
in	file_name	Имя файла базы данных
in	file_error	Имя файла для записи ошибок

Возвращает

Информацию о статусе авторизации пользователя

Исключения

error,если	введены некорректные данные логина или пароля Метод для авторизации пользователя
------------	--

Объявления и описания членов класса находятся в файле:

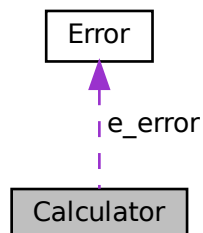
- [calc.h](#)

## 3.2 Класс Calculator

Класс калькулятор

```
#include <calc.h>
```

Граф связей класса Calculator:



Открытые члены

- [Calculator](#) ([Error](#) err)
- int [calc](#) (int work\_sock)  
Функция вычисления

Закрытые данные

- [Error](#) e\_error  
Приватный член класса [Error](#).

### 3.2.1 Подробное описание

Класс калькулятор

Данный класс создан для вычисления среднего арифметического элементов вектора

Предупреждения

Поддерживаемый тип данных float

### 3.2.2 Конструктор(ы)

#### 3.2.2.1 Calculator()

```
Calculator::Calculator (
    Error err ) [inline]
```

< Конструктор класса [Calculator](#)

< Присваивание объекта err приватному члену e\_error

### 3.2.3 Методы

#### 3.2.3.1 calc()

```
int Calculator::calc (
    int work_sock )
```

Функция вычисления

Функция вычисляет значение среднего арифметического элементов вектора

Аргументы

in	work_sock	Передаваемый сокет через который осуществляется обмен данными между клиентом и сервером
----	-----------	---

Возвращает

Результат математических вычислений Метод для выполнения вычислений

Объявления и описания членов класса находятся в файле:

- [calc.h](#)

## 3.3 Класс Error

Класс ошибок

```
#include <calc.h>
```

Открытые члены

- `Error ()`  
Конструктор по умолчанию

Открытые статические члены

- `static void errors (std::string error, std::string name)`  
Статический метод для вывода ошибки
- `static int er (std::string file_name, std::string file_error)`  
Статический метод для обработки ошибок

### 3.3.1 Подробное описание

Класс ошибок

Данный класс создан для отслеживания исключений в программе и записи в файл с ошибками

Объявления и описания членов класса находятся в файле:

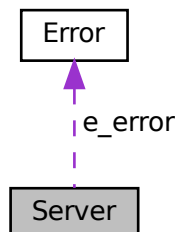
- `calc.h`

## 3.4 Класс Server

Класс Сервер

```
#include <calc.h>
```

Граф связей класса Server:



## Открытые члены

- [Server](#) ([Error](#) err)
- int [self\\_addr](#) (std::string &error, std::string &file\_error, int port)  
Функция установки сетевого соединения
- int [client\\_addr](#) (int s, std::string &error, std::string &file\_error)  
Функция установки сетевого соединения

## Закрытые данные

- [Error e\\_error](#)  
Приватный член класса [Error](#).

### 3.4.1 Подробное описание

#### Класс Сервер

Данный класс создан для для установки адреса сервера и клиента

### 3.4.2 Конструктор(ы)

#### 3.4.2.1 Server()

```
Server::Server (  
    Error err ) [inline]
```

< Конструктор класса [Server](#)

< Присваивание объекта err приватному члену e\_error

### 3.4.3 Методы

#### 3.4.3.1 client\_addr()

```
int Server::client_addr (  
    int s,  
    std::string & error,  
    std::string & file_error )
```

Функция установки сетевого соединения

Функция используется для получения адреса клиента, с которым установлено сетевое соединение по заданному сокету

## Аргументы

in	s	Передаваемый сокетом для работы с сетью
in	error	Имя файла базы данных
in	file_error	Имя файла для записи ошибок

## Возвращает

Информацию о выполнении операции

## Исключения

error,если	имеются ошибки при подключении Метод для установки адреса клиента
------------	---

## 3.4.3.2 self\_addr()

```
int Server::self_addr (
    std::string & error,
    std::string & file_error,
    int port )
```

## Функция установки сетевого соединения

Функция используется для определения собственного сетевого адреса и порта

## Аргументы

in	error	Строка с сообщением об ошибки
in	file_error	Имя файла для записи ошибок
in	port	Порт

## Возвращает

Информацию о выполнении операции

## Исключения

error,если	имеются ошибки при подключении Метод для установки адреса сервера
------------	---

Объявления и описания членов класса находятся в файле:

- [calc.h](#)





## Глава 4

# Файлы

### 4.1 Файл calc.h

Заголовочный файл программы сервера

```
#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 <random>
#include <cryptopp/cryptlib.h>
#include <vector>
#include <getopt.h>
#include <csignal>
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для calc.h:



## Классы

- class [Error](#)  
Класс ошибок
- class [Server](#)  
Класс Сервер
- class [Authorized](#)  
Класс авторизации
- class [Calculator](#)  
Класс калькулятор

## Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

### 4.1.1 Подробное описание

Заголовочный файл программы сервера

Автор

Гурин Д.А.

Версия

1.0

Дата

17.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа

# Предметный указатель

Authorized, [5](#)  
    Authorized, [6](#)  
    authorized, [6](#)  
authorized  
    Authorized, [6](#)

calc  
    Calculator, [8](#)  
calc.h, [13](#)  
Calculator, [7](#)  
    calc, [8](#)  
    Calculator, [8](#)  
client\_addr  
    Server, [10](#)

Error, [9](#)

self\_addr  
    Server, [11](#)  
Server, [9](#)  
    client\_addr, [10](#)  
    self\_addr, [11](#)  
    Server, [10](#)