

Курсовая работа

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 Auth	7
4.1.1 Подробное описание	7
4.1.2 Методы	7
4.1.2.1 <code>authorized()</code>	7
4.1.2.2 <code>GetLogin()</code>	8
4.1.2.3 <code>GetPass()</code>	8
4.2 Класс <code>calculation</code>	9
4.2.1 Подробное описание	9
4.2.2 Методы	9
4.2.2.1 <code>math()</code>	9
4.2.2.2 <code>overflow()</code>	10
4.2.2.3 <code>summa()</code>	10
4.3 Класс <code>error_server</code>	11
4.3.1 Подробное описание	11
4.4 Структура <code>for_auth</code>	12
4.5 Структура <code>for_calc</code>	12
4.6 Класс <code>Server</code>	13
4.6.1 Подробное описание	13
4.6.2 Методы	13
4.6.2.1 <code>client_addr()</code>	13
4.6.2.2 <code>self_addr()</code>	14
4.7 Структура <code>things_fixture</code>	14
5 Файлы	17
5.1 Файл <code>auth.cpp</code>	17
5.2 Файл <code>comstr.cpp</code>	17
5.3 Файл <code>create_servers.cpp</code>	17
5.4 Файл <code>er.cpp</code>	18
5.4.1 Функции	18
5.4.1.1 <code>errors()</code>	18
5.5 Файл <code>main.cpp</code>	18
5.6 Файл <code>math.cpp</code>	19
5.7 Файл <code>mdfile.cpp</code>	19
5.7.1 Функции	19
5.7.1.1 <code>MD()</code>	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Auth	7
calculation	9
for_auth	12
for_calc	12
invalid_argument	
error_server	11
Server	13
things_fixture	14

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Auth	Класс для аутентификации пользователя	7
calculation	Класс для выполнения вычисления суммы вектора	9
error_server	Класс для обработки ошибок возникающих при работе сервера	11
for_auth	12
for_calc	12
Server	Класс в котором находятся переменные и функция для подключения сервера к клиенту	13
things_fixture	14

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

auth.cpp	17
auth.h	??
comstr.cpp	17
create_servers.cpp	17
er.cpp	18
error.h	??
main.cpp	18
math.cpp	19
math.h	??
md_gen.h	??
mdfile.cpp	19
msg_send.h	??
server.h	??

Глава 4

Классы

4.1 Класс Auth

Класс для аутентификации пользователя

```
#include <auth.h>
```

Открытые члены

- string [GetLogin](#) (string file_name, string file_error, string message)
Функция для получения и сравнения логина
- string [GetPass](#) (string file_name, string file_error, string salt, string msg)
Функция для получения и сравнения хэша
- int [authorized](#) (int work_sock, string file_name, string file_error)
Функция реализации авторизации пользователя

Открытые атрибуты

- string file_name
- string file_error
- string message
- string salt

4.1.1 Подробное описание

Класс для аутентификации пользователя

4.1.2 Методы

4.1.2.1 [authorized\(\)](#)

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

Функция реализации авторизации пользователя

Аргументы

work_sock	Рабочий сокет
file_name	Путь к файлу с базой данных пользователя
file_error	Путь к лог файлу для записи ошибки

Возвращает

true

4.1.2.2 GetLogin()

```
std::string Auth::GetLogin (
    string file_name,
    string file_error,
    string message )
```

Функция для получения и сравнения логина

Аргументы

file_name	путь к файлу с БД клиентов
file_error	описание ошибки
message	строка с логином и паролем

Возвращает

login логин пользователя
 "doesn't match" если логины не совпадают

4.1.2.3 GetPass()

```
std::string Auth::GetPass (
    string file_name,
    string file_error,
    string salt,
    string msg )
```

Функция для получения и сравнения хэша

Аргументы

file_name	путь к файлу с БД клиентов
file_error	описание ошибки
salt	соль
msg	полученный хэш от пользователя

Возвращает

```
digest MD хэш  
"doesn't match" если пароли не совпадают
```

Объявления и описания членов классов находятся в файлах:

- `auth.h`
- `auth.cpp`

4.2 Класс calculation

Класс для выполнения вычисления суммы вектора

```
#include <math.h>
```

Открытые члены

- `float summa (std::vector< float >vector)`
Функция для сложения значений
- `float overflow (float sum)`
Функция для проверки на переполнение

Открытые статические члены

- `static float math (int work_sock)`
Реализация функции математических вычислений

Закрытые данные

- `int work_sock`
- `float summ_of_vector`

4.2.1 Подробное описание

Класс для выполнения вычисления суммы вектора

4.2.2 Методы

4.2.2.1 math()

```
float calculation::math (  
    int work_sock ) [static]
```

Реализация функции математических вычислений

Аргументы

work_sock	Рабочий сокет
-----------	---------------

Возвращает

true

4.2.2.2 overflow()

```
float calculation::overflow (
    float sum )
```

Функция для проверки на переполнение

Аргументы

sum	сумма значений для проверки
-----	-----------------------------

Возвращает

sum сумма значений

4.2.2.3 summa()

```
float calculation::summa (
    std::vector< float > vector )
```

Функция для сложения значений

Аргументы

vector	вектор со значениями
--------	----------------------

Возвращает

sum сумма значений

Объявления и описания членов классов находятся в файлах:

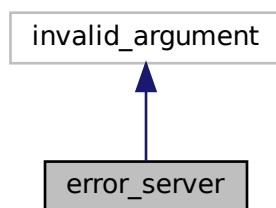
- math.h
- [math.cpp](#)

4.3 Класс error_server

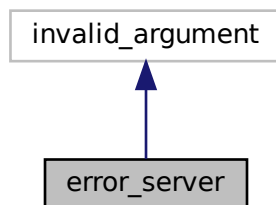
Класс для обработки ошибок возникающих при работе сервера

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

Граф наследования: error_server:



Граф связей класса error_server:



Открытые члены

- error_server (const string &what_arg)
- error_server (const char *what_arg)

4.3.1 Подробное описание

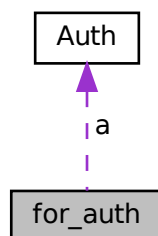
Класс для обработки ошибок возникающих при работе сервера

Объявления и описания членов класса находятся в файле:

- error.h

4.4 Структура for_auth

Граф связей класса for_auth:



Открытые атрибуты

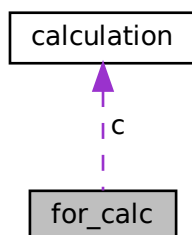
- `Auth * a`

Объявления и описания членов структуры находятся в файле:

- `UnitTest.cpp`

4.5 Структура for_calc

Граф связей класса for_calc:



Открытые атрибуты

- `calculation * c`

Объявления и описания членов структуры находятся в файле:

- `UnitTest.cpp`

4.6 Класс Server

Класс в котором находятся переменные и функция для подключения сервера к клиенту

```
#include <server.h>
```

Открытые члены

- string * [comstr](#) (int argc, char *argv[])
Реализация ком. строки
- string * adding_com (string file_name, string file_error, int port)
- int [self_addr](#) (string error, string file_error, int port)
Реализация метода self_addr.
- int [client_addr](#) (int s, string error, string file_error)
Реализация метода client_addr.

Открытые атрибуты

- int port
- string file_name
- string file_error
- string error
- string error_name
- string * com = new string[4]

4.6.1 Подробное описание

Класс в котором находятся переменные и функция для подключения сервера к клиенту

4.6.2 Методы

4.6.2.1 client_addr()

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

Реализация метода client_addr.

Аргументы

error	Описание ошибки
s	Сокет полученный из метода self_addr
file_error	Путь к лог файлу для записи ошибки

Возвращает

`work_sock` Возвращает рабочий сокет

4.6.2.2 `self_addr()`

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

Реализация метода `self_addr`.

Аргументы

<code>error</code>	Описание ошибки
<code>port</code>	Порт для соединения
<code>file_error</code>	Путь к лог файлу для записи ошибки

Возвращает

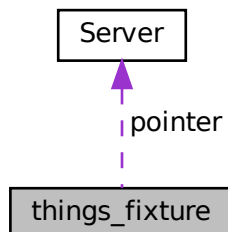
`s` Возвращает рабочий сокет

Объявления и описания членов классов находятся в файлах:

- `server.h`
- [comstr.cpp](#)
- [create_servers.cpp](#)

4.7 Структура `things_fixture`

Граф связей класса `things_fixture`:



Открытые атрибуты

- `Server * pointer`

Объявления и описания членов структуры находятся в файле:

- `UnitTest.cpp`

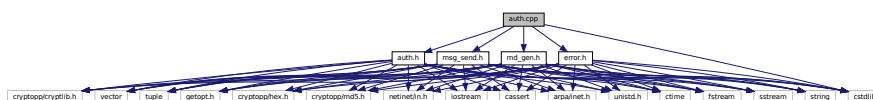
Глава 5

Файлы

5.1 Файл auth.cpp

```
#include "error.h"  
#include "auth.h"  
#include "msg_send.h"  
#include "md_gen.h"  
#include <cstdlib>
```

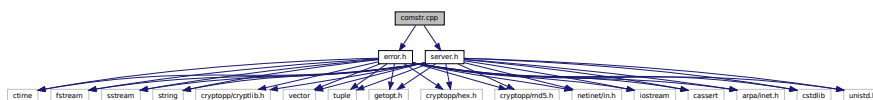
Граф включаемых заголовочных файлов для auth.cpp:



5.2 Файл comstr.cpp

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

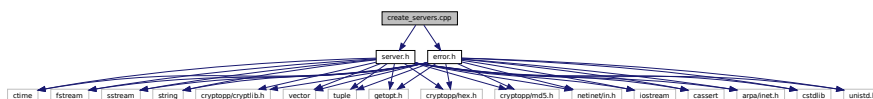
Граф включаемых заголовочных файлов для comstr.cpp:



5.3 Файл create_servers.cpp

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

Граф включаемых заголовочных файлов для create_servers.cpp:



5.4 Файл `er.cpp`

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

Граф включаемых заголовочных файлов для `er.cpp`:



Функции

- `void errors (string error, string file_error)`
Функция для обработки ошибок

5.4.1 Функции

5.4.1.1 `errors()`

```
void errors (
    string error,
    string file_error )
```

Функция для обработки ошибок

Аргументы

<code>error</code>	Описание ошибки
<code>file_error</code>	Путь к лог файлу для записи ошибки

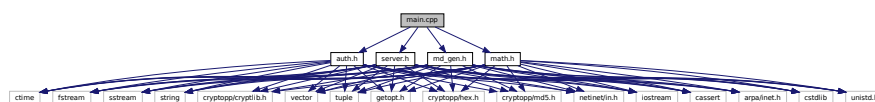
Возвращает

`true`

5.5 Файл `main.cpp`

```
#include "md_gen.h"
#include "math.h"
#include "auth.h"
#include "server.h"
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

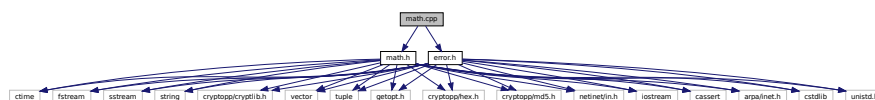
- `int main (int argc, char *argv[])`

5.6 Файл math.cpp

```
#include "math.h"
```

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

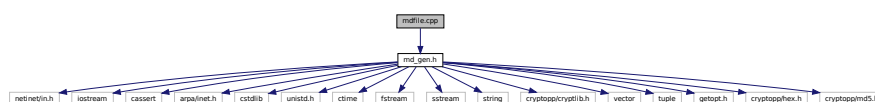
Граф включаемых заголовочных файлов для math.cpp:



5.7 Файл mdfile.cpp

```
#include "md_gen.h"
```

Граф включаемых заголовочных файлов для mdfile.cpp:



Функции

- `string MD (string sah)`
Функция вычисления хэша

5.7.1 Функции

5.7.1.1 MD()

```
string MD (
    string sah )
```

Функция вычисления хэша

Аргументы

sal	Переменная содержащая соль и хэш
-----	----------------------------------

Возвращает

digest MD5 хэш

Предметный указатель

- Auth, [7](#)
 - authorized, [7](#)
 - GetLogin, [8](#)
 - GetPass, [8](#)
- auth.cpp, [17](#)
- authorized
 - Auth, [7](#)
- calculation, [9](#)
 - math, [9](#)
 - overflow, [10](#)
 - summa, [10](#)
- client_addr
 - Server, [13](#)
- comstr.cpp, [17](#)
- create_servers.cpp, [17](#)
- er.cpp, [18](#)
 - errors, [18](#)
- error_server, [11](#)
- errors
 - er.cpp, [18](#)
- for_auth, [12](#)
- for_calc, [12](#)
- GetLogin
 - Auth, [8](#)
- GetPass
 - Auth, [8](#)
- main.cpp, [18](#)
- math
 - calculation, [9](#)
- math.cpp, [19](#)
- MD
 - mdfile.cpp, [19](#)
- mdfile.cpp, [19](#)
 - MD, [19](#)
- overflow
 - calculation, [10](#)
- self_addr
 - Server, [14](#)
- Server, [13](#)
 - client_addr, [13](#)
 - self_addr, [14](#)
- summa
 - calculation, [10](#)
- things_fixture, [14](#)