

Server

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 AuthError	7
4.2 Класс BindingError	8
4.2.1 Подробное описание	9
4.3 Класс Server	9
4.3.1 Методы	9
4.3.1.1 client_addr()	9
4.3.1.2 comstr()	9
4.3.1.3 self_addr()	10
5 Файлы	11
5.1 Файл head.h	11
5.1.1 Подробное описание	12
5.1.2 Функции	13
5.1.2.1 authorized()	13
5.1.2.2 math()	13
5.1.2.3 MD()	13
5.1.2.4 msgsend()	14
5.2 Файл main.cpp	14
5.2.1 Подробное описание	14
5.2.2 Функции	15
5.2.2.1 main()	15
5.3 Файл work.cpp	15
5.3.1 Подробное описание	16
5.3.2 Функции	16
5.3.2.1 authorized()	16
5.3.2.2 errors()	16
5.3.2.3 math()	17
5.3.2.4 MD()	17
5.3.2.5 msgsend()	17
Предметный указатель	19

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

std::invalid_argument	
AuthError	7
BindingError	8
Server	9

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

AuthError	7
BindingError	
Классы ошибок	8
Server	9

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

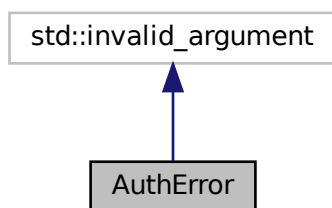
head.h	Заголовочный файл	11
main.cpp	Главный файл программы	14
work.cpp	Файл с функциями сервера	15

Глава 4

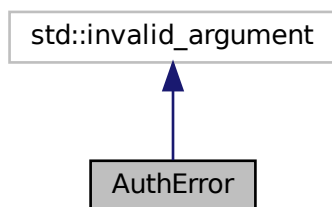
Классы

4.1 Класс AuthError

Граф наследования:AuthError:



Граф связей класса AuthError:



Открытые члены

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

Объявления и описания членов класса находятся в файле:

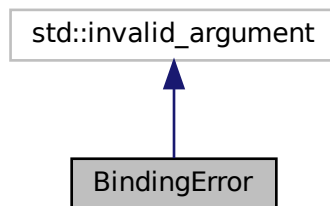
- [head.h](#)

4.2 Класс BindingError

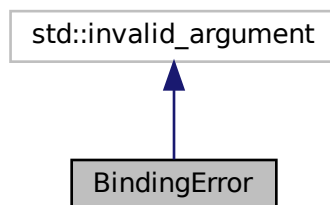
Классы ошибок

```
#include <head.h>
```

Граф наследования: BindingError:



Граф связей класса BindingError:



Открытые члены

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

4.2.1 Подробное описание

Классы ошибок

Данный блок нужен для выброса исключений и обработки ошибок.

Объявления и описания членов класса находятся в файле:

- [head.h](#)

4.3 Класс Server

Открытые члены

- `string * comstr (int argc, char *argv[])`
Обработка командной строки
- `int self_addr (string file_error, int port)`
Структура сервера
- `int client_addr (int s, string file_error)`
Структура клиента

4.3.1 Методы

4.3.1.1 client_addr()

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

Структура клиента

Подключается к серверу по уже созданному в структуре сервера сокету. В случае ошибки делает запись в лог.

4.3.1.2 comstr()

```
string * Server::comstr (  
    int argc,  
    char * argv[] )
```

Обработка командной строки

Обработка параметров командной строки и вывод справки в случае параметра -h

Предупреждения

Указание параметров -f(файл базы данных) и -e(файл с логами) обязательно!

4.3.1.3 self_addr()

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

Структура сервера

Для соединения с клиентом используется порт 127.0.0.1. Функция создает сокет и включает режим ожидания клиента

Объявления и описания членов классов находятся в файлах:

- [head.h](#)
- [work.cpp](#)

Глава 5

Файлы

5.1 Файл head.h

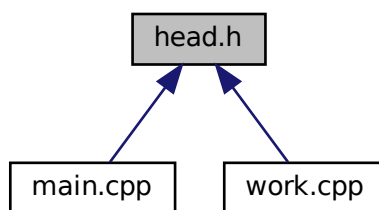
Заголовочный файл

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

Граф включаемых заголовочных файлов для head.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Server](#)
- class [BindingError](#)
 - Классы ошибок
- class [AuthError](#)

Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`
 - Используемые библиотеки

Функции

- `std::string MD (std::string sah)`
 - Структура сервера
- `void errors (std::string error, std::string name)`
- `void msgsend (int work_sock, string mess)`
 - Функция отправки сообщения клиенту
- `int authorized (int work_sock, string file_name, string file_error)`
 - Функция авторизации пользователя на сервере.
- `int math (int work_sock)`
 - Обработка принимаемых векторов

5.1.1 Подробное описание

Заголовочный файл

Автор

Фарафонов А.Е.

Версия

1.0

Дата

10.10.2024

Авторство

ИБСТ ПГУ

5.1.2 Функции

5.1.2.1 authorized()

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

Функция авторизации пользователя на сервере.

Поочередно получает логин и пароль клиента, замешивает в пароль соль для шифрования и сравнивает правильность полученных данных. В случае ошибок выбрасывается исключение и ведется запись в лог.

5.1.2.2 math()

```
int math (
    int work_sock )
```

Обработка принимаемых векторов

На вход принимается количество векторов kolvo, длина векторов numb и значения в каждом векторе.

Предупреждения

На вход принимаются значения типа uint16_t. При переполнении возвращается значение 65535.

5.1.2.3 MD()

```
std::string MD (
    std::string sah )
```

Структура сервера

Структура сервера с обозначением функций программы и требуемых параметров функций.

5.1.2.4 msgsend()

```
void msgsend (
    int work_sock,
    string mess )
```

Функция отправки сообщения клиенту

Создаёт буфер для сообщения и отправляет сообщение

5.2 Файл main.cpp

Главный файл программы

```
#include "head.h"
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

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

Функция запуска сервера

5.2.1 Подробное описание

Главный файл программы

Автор

Фарафонов А.Е.

Версия

1.0

Дата

10.10.2024

Авторство

ИБСТ ПГУ

5.2.2 Функции

5.2.2.1 main()

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

Функция запуска сервера

Функция отвечает за запуск сервера с параметрами командной строки. Для работы использует заголовочный файл со структурой сервера и файл с функциями для определения поведения сервера.

5.3 Файл work.cpp

Файл с функциями сервера

```
#include "head.h"
```

Граф включаемых заголовочных файлов для work.cpp:



Функции

- void `msgsend` (int work_sock, string mess)
Функция отправки сообщения клиенту
- string `MD` (string str)
Функция создания хэша
- void `errors` (string error, string name)
Обработка ошибок
- int `authorized` (int work_sock, string file_name, string file_error)
Функция авторизации пользователя на сервере.
- int `math` (int work_sock)
Обработка принимаемых векторов

5.3.1 Подробное описание

Файл с функциями сервера

Автор

Фарафонов А.Е.

Версия

1.0

Дата

10.10.2024

Авторство

ИБСТ ПГУ

5.3.2 Функции

5.3.2.1 authorized()

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

Функция авторизации пользователя на сервере.

Поочередно получает логин и пароль клиента, замешивает в пароль соль для шифрования и сравнивает правильность полученных данных. В случае ошибок выбрасывается исключение и ведется запись в лог.

5.3.2.2 errors()

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

Обработка ошибок

Открывает файл с логами(параметр -e) и записывает туда ошибку в формате: Ошибка:Время.

5.3.2.3 math()

```
int math (
    int work_sock )
```

Обработка принимаемых векторов

На вход принимается количество векторов kolvo, длина векторов numb и значения в каждом векторе.

Предупреждения

На вход принимаются значения типа uint16_t. При переполнении возвращается значение 65535.

5.3.2.4 MD()

```
string MD (
    string str )
```

Функция создания хэша

Принимает на вход текстовую строку и возвращает строку шестнадцатиричного хэша

5.3.2.5 msgsend()

```
void msgsend (
    int work_sock,
    string mess )
```

Функция отправки сообщения клиенту

Создаёт буфер для сообщения и отправляет сообщение

Предметный указатель

- AuthError, [7](#)
- authorized
 - head.h, [13](#)
 - work.cpp, [16](#)
- BindingError, [8](#)
- client_addr
 - Server, [9](#)
- comstr
 - Server, [9](#)
- errors
 - work.cpp, [16](#)
- head.h, [11](#)
 - authorized, [13](#)
 - math, [13](#)
 - MD, [13](#)
 - msgsend, [13](#)
- main
 - main.cpp, [15](#)
- main.cpp, [14](#)
 - main, [15](#)
- math
 - head.h, [13](#)
 - work.cpp, [16](#)
- MD
 - head.h, [13](#)
 - work.cpp, [17](#)
- msgsend
 - head.h, [13](#)
 - work.cpp, [17](#)
- self_addr
 - Server, [9](#)
- Server, [9](#)
 - client_addr, [9](#)
 - comstr, [9](#)
 - self_addr, [9](#)
- work.cpp, [15](#)
 - authorized, [16](#)
 - errors, [16](#)
 - math, [16](#)
 - MD, [17](#)
 - msgsend, [17](#)