Kursovaya_Ponomarev

Создано системой Doxygen 1.9.1

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Классы

3.1 Класс Auth

Открытые члены

- Auth (std::string ID, std::string pass)

  Конструктор для установки идентификатора и пароля клиента
- void GenSALT ()

  Генерация случайной соли для вычисления хэша
- bool CompareHashes (std::string ClientHash)

  Сравнение хэша, присылаемого клиентом и хэша, вычисляемого внутри метода
- std::string getSALT ()
- std::string getId ()
- std::string getpass ()
- std::string getstrHash ()

Открытые атрибуты

- char ERRmsg [3] = {'E', 'R','R'}

  Сообщение, отсылаемое клиенту при ошибке его обрабоки
- char OKmsg [2] = {'O', 'K'}

  Сообщение, отсылаемое клиенту при успешной авторизации

3.1.1 Конструктор(ы)

3.1.1.1 Auth()

```
Auth::Auth (
            std::string ID,
            std::string pass )
```

Конструктор для установки идентификатора и пароля клиента

Аргументы

in	ID,идентификатор	клиента, std::string.
in	pass,пароль	клиента, std::string.

3.1.2 Методы

3.1.2.1 CompareHashes()

```
bool Auth::CompareHashes (
            std::string ClientHash )
```

Сравнение хэша, присылаемого клиентом и хэша, вычисляемого внутри метода

Вычисляет MD5 хэш от строки SALT+password и сравнивает его с хэшем, который присылает клиент

Аргументы

in	ClientHash,хэш	клиента, std::string

Возвращает

bool, если хэши совпадают - true, иначе false

Исключения

std::server_error	в случае несовпадения хэшей, штатная type = invalid_argument, what ="Invalid hash"

3.1.2.2 GenSALT()

```
void Auth::GenSALT ( )
```

Генерация случайной соли для вычисления хэша

Соль - 64-х разрядное число, представленное в виде строки из 16-ти шестнадцатиричных цифр

Объявления и описания членов классов находятся в файлах:

- Auth.h
- Auth.cpp

3.2 Класс Calc

Класс для вычислений по вектору

#include <Calc.h>

Открытые члены

- int16_t * mean (std::vector< int16_t > arr)
  Конструктор без параметров

3.2.1 Подробное описание

Класс для вычислений по вектору

3.2.2 Методы

3.2.2.1 mean()

int16_t* Calc::mean (
std::vector< int16_t > arr) [inline]

Конструктор без параметров

Вычисляет суммы по вектору

Аргументы

in	arr,вектор,std::vector<int16_t>	

Возвращает

указатель на массив с результатом, int16_t *

Исключения

std::server_error	в случае ошибки, критическая @type = invalid_argument, what ="Error: Count " @details Если идет переполнение вектораю, выводит максимальное возможное если больше, и минимальное возможное если меньше

Объявления и описания членов класса находятся в файле:

- Calc.h

3.3 Класс Connect

Класс, обеспечивающий работу с сокетами и сетовое взаимодействие

#include <Connect.h>

Открытые члены

- Connect (unsigned int port)
  Конструктор
- void start_listening ()
  Установка сокета в режим ожидания
- int new_bind ()
  Привязка сокета к адресу
- int accepting ()
  Приём соединения
- int receiving (int sock, void *buf, int size)
  Приём данных
- void sending (int sock, void *buf, int sizeb)
  Отправка данных

3.3.1 Подробное описание

Класс, обеспечивающий работу с сокетами и сетовое взаимодействие

3.3.2 Конструктор(ы)

3.3.2.1 Connect()

```
Connect::Connect (
            unsigned int port )
```

Конструктор

Устанавливает порт, инициализирует основной сокет и структуру sockaddr_in

Аргументы

in	port,порт,на	котором работает сервер, int.

Исключения

std::server_error	в случае ошибки, критическая type = invalid_argument, what ="Error: Socket creation "

3.3.3 Методы

3.3.3.1 accepting()

int Connect::accepting ( )

Приём соединения

Возвращает

код сокета, int

Исключения

std::server_error	в случае ошибки, штатная type = invalid_argument, what ="Error: Accepting "

3.3.3.2 new_bind()

int Connect::new_bind ( )

Привязка сокета к адресу

Возвращает

код сокета, int

Исключения

std::server_error	в случае ошибки, критическая type = invalid_argument, what ="Error: Socket bind "

3.3.3.3 receiving()

int Connect::receiving (
int sock,
void * buf,
int size)

Приём данных

Аргументы

in	sock,сокет,int	
in	buf,буфер	для данных, void*
in	size,размер	буфера, int

Возвращает

количество полученных байт, int

Исключения

std::server_error	в случае ошибки, штатная type = invalid_argument, what ="Error: Receiving "

3.3.3.4 sending()

```
void Connect::sending (
                int sock,
                void * buf,
                int sizeb )
```

Отправка данных

Аргументы

in	sock,сокет,int	
in	buf,буфер	с данными, void*
in	sizeb,количество	отправляемых байт, int

Исключения

std::server_error	в случае ошибки, штатная type = invalid_argument, what ="Error: Sending "

3.3.3.5 start_listening()

```
void Connect::start_listening ( )
```

Установка сокета в режим ожидания

Исключения

std::server_error	в случае ошибки, критическая type = invalid_argument, what ="Error: Listening "

Объявления и описания членов классов находятся в файлах:

- Connect.h
- Connect.cpp

3.4 Класс Db

Класс для работы с базой данных пользователей

#include <DataBase.h>

Открытые члены

- Db (std::string Dbname)

  Конструктор, в котором считывается база данных и сохраняется в словарь
- bool IDcheck (std::string login)

  Проверка наличия идентификатора клиента в базе данных

Открытые атрибуты

- std::map< std::string, std::string > DatabaseP

  Словарь с парами идентификатор:пароль

3.4.1 Подробное описание

Класс для работы с базой данных пользователей

3.4.2 Конструктор(ы)

3.4.2.1 Db()

```
Db::Db (
            std::string Dbname )
```

Конструктор, в котором считывается база данных и сохраняется в словарь

Аргументы

in	Dbname,путь	к файлу с базой данных, std::string.

Исключения

std::server_error	в случае проблем с файлом базы данных, критическая

3.4.3 Методы

3.4.3.1 IDcheck()

```
bool Db::IDcheck (
            std::string login )
```

Проверка наличия идентификатора клиента в базе данных

Аргументы

in	login,идентификатора	клиента, std::string

Возвращает

bool, если идентификатор есть в базе - true, иначе false

Исключения

std::server_error	в случае отсутствия идентификатора в базе, штатная type = invalid_argument, what ="Invalid ID"

Объявления и описания членов классов находятся в файлах:

- DataBase.h
- DataBase.cpp

3.5 Класс Error

Класс для обработки ошибок

#include <Error.h>

Открытые члены

- void setLogname (std::string Logname)
  Конструктор без параметров
- void write_log (std::string what, bool Critical)
  Запись ошибки в лог

3.5.1 Подробное описание

Класс для обработки ошибок

3.5.2 Методы

3.5.2.1 setLogname()

```
void Error::setLogname (
            std::string Logname )   [inline]
```

Конструктор без параметров

Функция, устанавливающая путь к файлу с логом ошибок

3.5.2.2 write_log()

```
void Error::write_log (
            std::string what,
            bool Critical )
```

Запись ошибки в лог

Записывает время, тип и критичность ошибки

Аргументы

in	what,тип	ошибки, std::string
in	Critical,критичность	ошибки (Критическая - true, Штатная - false), std::string

Объявления и описания членов классов находятся в файлах:

- Error.h
- Error.cpp

3.6 Класс Opts

Класс для получения параметров коммандной строки

#include <interface.h>

Открытые члены

- Opts (int argc, char **argv)
  Конструктор, внутри которого считываются параметры командной строки
- bool Checkfiles ()
  Проверка роботоспособности файлов базы данных и лога
- string getDatabase ()
- string getLogfile ()
- int getPort ()

3.6.1 Подробное описание

Класс для получения параметров коммандной строки

3.6.2 Конструктор(ы)

3.6.2.1 Opts()

```
Opts::Opts (
            int argc,
            char ** argv )
```

Конструктор, внутри которого считываются параметры командной строки

Параметры командной строки: 1)-b Путь к файлу с базой данных, необязательный 2)-l Путь к файлу для записи логов, необязательный 3)-p Порт, на котором работает сервер, необязательный 4)-h вызов подсказки При ошибках в параметрах вызывается справка и программа завершает работу

Аргументы

in	int	argc
in	char	**argv

3.6.3 Методы

3.6.3.1 Checkfiles()

```
bool Opts::Checkfiles ( )
```

Проверка роботоспособности файлов базы данных и лога

Возвращает

bool, если нет ошибок в фалах - true, иначе false

Исключения

std::invalid_argument	в случае проблем с файлами, критическая type = invalid_argument, what ="Wrong DB File Name" или what ="Wrong Log File Name"

Объявления и описания членов классов находятся в файлах:

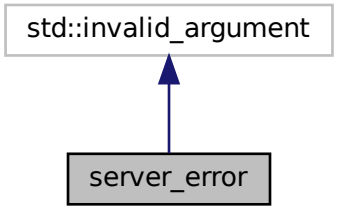
- interface.h
- interface.cpp

3.7 Класс server_error

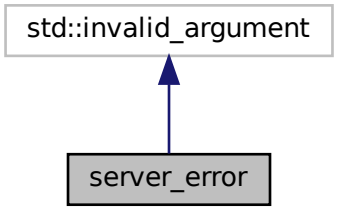
Класс ошибок

#include <Error.h>

Граф наследования:server_error:



Граф связей класса server_error:



Открытые члены

- server_error (const std::string &what_arg, bool critical=false)
  Конструктор ошибок с строкой в качестве параметра
- server_error (const char *what_arg, bool critical=false)
  Конструктор ошибок с си-строкой в качестве параметра
- bool getState () const
  Возвращает статус критичности ошибки

3.7.1 Подробное описание

Класс ошибок

Наследует от класса std::invalid_argument

3.7.2 Конструктор(ы)

3.7.2.1 server_error() [1/2]

```
server_error::server_error (
            const std::string & what_arg,
            bool critical = false )   [inline], [explicit]
```

Конструктор ошибок с строкой в качестве параметра

Аргументы

in	what_arg,тип	ошибки, const std::string.
in	critical,критическа	ошибка - true, штатная - false, bool

3.7.2.2 server_error() [2/2]

```
server_error::server_error (
            const char * what_arg,
            bool critical = false )   [inline], [explicit]
```

Конструктор ошибок с си-строкой в качестве параметра

Аргументы

in	what_arg,тип	ошибки, const char*.
in	critical,критическа	ошибка - true, штатная - false, bool

Объявления и описания членов класса находятся в файле:

- Error.h

Предметный указатель