Программная реализация сетевого сервера. Сервер, вычисляющий среднее арифметическое вектора типа double.

1.0

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

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс Average

Класс для вычисления среднего арифметического вектора

#include <calculator.h>

Открытые члены

- Average ()
  Конструтор класса без параметров
- double average (std::vector< double > &arr)
  Функция вычисления среднего арифметического вектора

4.1.1 Подробное описание

Класс для вычисления среднего арифметического вектора

Метод для вычисления

4.1.2 Методы

4.1.2.1 average()

```
double Average::average (
            std::vector< double > & arr )
```

Функция вычисления среднего арифметического вектора

Аргументы

in	vector<double>&	arr вектор со значениями типа double

Возвращает

result

Объявления и описания членов классов находятся в файлах:

- calculator.h
- calculator.cpp

4.2 Класс communicator

Класс для связи с клиентом

#include <communicator.h>

Открытые члены

- communicator ()
  Конструтор класса без параметров
- void conversation (unsigned int port, std::map< std::string, std::string > DataBaseP)
  Функция "разговора" с клиентом.
- std::string GenSALT ()
  Функция генерации соли
- std::string GenHash (const std::string &password)
  Генерация хэша из пароля и соли.
- bool CompareHashes (std::string ClientHash)
  Сравнение хэшей клиента и сервера.
- void getpass (std::string pass)
- void setSALT (const std::string &salt)

4.2.1 Подробное описание

Класс для связи с клиентом

Методы: установка бинда, установка сервера в режим ожидания соединения принятие соединения получение данных от клиента отправка данных клиенту "разговор" с клиентом

4.2.2 Методы

4.2.2.1 CompareHashes()

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

Сравнение хэшей клиента и сервера.

Данный метод сравнивает хэш, полученный от клиента, с хэшем, сгенерированным на сервере. Если хэши не совпадают, то генерируется исключение server_error с сообщением "Invalid Hash". Затем выводится на экран хэш клиента и сервера.*

Аргументы

in	ClientHash	Хэш, полученный от клиента.

Возвращает

true, если хэши совпадают, иначе false.

Исключения

server_error,если	хэши не совпадают.

4.2.2.2 conversation()

```
void communicator::conversation (
                unsigned int port,
                std::map< std::string, std::string > DataBaseP )
```

Функция "разговора" с клиентом.

Аргументы

in	unsigned	int port,
in	map	<string,string> DataBaseP

В этом методе происходит все взаимодействие с клиентом. Ничего не возвращает.

4.2.2.3 GenHash()

```
std::string communicator::GenHash (
                const std::string & password )
```

Генерация хэша из пароля и соли.

Данный метод использует библиотеку Crypto++ для генерации хэша MD5 из пароля и соли. Затем хэш преобразуется в шестнадцатеричную строку.

Аргументы

in	password	Пароль для хэширования.

Возвращает

Сгенерированный хэш в шестнадцатеричной строке.

4.2.2.4 GenSALT()

std::string communicator::GenSALT ( )

Функция генерации соли

В этом методе происходит генерация соли.

Возвращает

SALT

Объявления и описания членов классов находятся в файлах:

- communicator.h
- communicator.cpp

4.3 Класс DB

Класс для получения данных из файла базы клиентов

#include <userbase.h>

Открытые члены

- DB (std::string DBName)
  Конструтор класса
- std::map< std::string, std::string > get_data ()
  Метод для получения данных из словаря

4.3.1 Подробное описание

Класс для получения данных из файла базы клиентов

Методы: проверка ID

4.3.2 Конструктор(ы)

4.3.2.1 DB()

DB::DB (
std::string DBName)

Конструтор класса

Аргументы

in	DBName	- путь до файла с данными клиентов

В этом конструкторе происходит открытие файла, получение данных, заполнение словаря, закрытие словаря

4.3.3 Методы

4.3.3.1 get_data()

std::map<std::string,std::string> DB::get_data ( ) [inline]

Метод для получения данных из словаря

Возвращает

DataBaseP Словарь с парами идентификатор:пароль

Объявления и описания членов классов находятся в файлах:

- userbase.h
- userbase.cpp

4.4 Класс interface

Класс для разбора командной строки и включения других модулей

#include <interface.h>

Открытые члены

- int Opts (int argc, char **argv)
  Функция разбора ПКС и включение модулей
- std::string getLogFileName ()
  Функция получения пути до журнала работы

4.4.1 Подробное описание

Класс для разбора командной строки и включения других модулей

Методы: разбор ПКС проверка файлов получение пути до файла с логами

4.4.2 Методы

4.4.2.1 getLogFileName()

std::string interface::getLogFileName ( ) [inline]

Функция получения пути до журнала работы

В этом методе возвращается путь до файла с журналом работы

Возвращает

LogFileName Возвращает путь к файлу для записи логов

4.4.2.2 Opts()

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

Функция разбора ПКС и включение модулей

Аргументы

in	int	argc,
in	char	**argv

В этом методе происходит разбор ПКС и включение модулей

Возвращает

1 or 0

Объявления и описания членов классов находятся в файлах:

- interface.h
- interface.cpp

4.5 Класс Logger

Класс для формирования и записи логов

#include <logger.h>

Открытые члены

- int writelog (std::string s)
  Функция записи лога
- void set_path (std::string path_file)
  Функция получение пути до файла журнала работы
- Logger ()
  Конструтор класса без параметров
- Logger (std::string s)
  Конструтор класса

Открытые статические члены

- static std::string getDateTime ()
  Функция получения времени

4.5.1 Подробное описание

Класс для формирования и записи логов

Методы: запись лога, получение пути до файла журнала работы получение даты и времени

4.5.2 Конструктор(ы)

4.5.2.1 Logger()

```
Logger::Logger (
            std::string s )   [inline]
```

Конструтор класса

Аргументы

in	s	- путь до файла с логами

4.5.3 Методы

4.5.3.1 getDateTime()

```
std::string Logger::getDateTime ( )   [static]
```

Функция получения времени

В этом методе происходит получение времени и вывод его в нужном формате

Возвращает

string(buf)

4.5.3.2 set_path()

void Logger::set_path (
std::string path_file)

Функция получение пути до файла журнала работы

В этом методе происходит получение пути до файла журнала работы. Ничего не возвращает.

4.5.3.3 writelog()

int Logger::writelog (
std::string s)

Функция записи лога

В этом методе происходит открытие файла, запись, закрытие файла

Возвращает

1 or 0

Объявления и описания членов классов находятся в файлах:

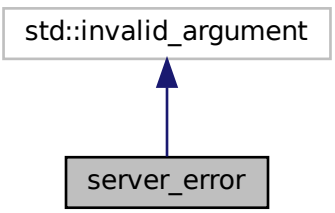
- logger.h
- logger.cpp

4.6 Класс server_error

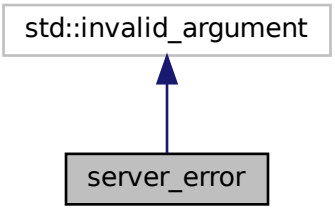
Класс вызова исключений

#include <programmerror.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
- std::string getState () const
  Возвращает статус критичности ошибки

4.6.1 Подробное описание

Класс вызова исключений

Класс server_error

4.6.2 Конструктор(ы)

4.6.2.1 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,
in	const	std::string, critical, критическая ошибка - true, штатная - false, bool

4.6.2.2 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,
in	const	char*, critical, критическая ошибка - true, штатная - false, bool

Объявления и описания членов класса находятся в файле:

- programmerror.h

Глава 5

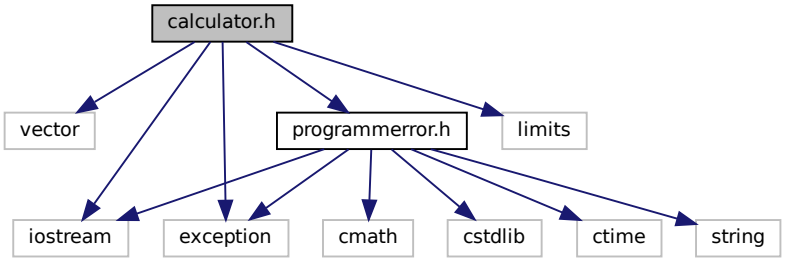
Файлы

5.1 Файл calculator.h

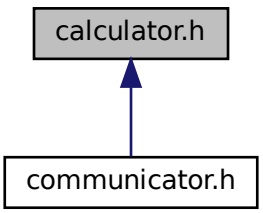
Заголовочный файл для модуля calculator, отвечающий за вычисления

```
#include <vector>
#include <iostream>
#include <exception>
#include <limits>
#include "programmerror.h"
```

Граф включаемых заголовочных файлов для calculator.h:



Граф файлов, в которые включается этот файл:



Классы

- class Average

  Класс для вычисления среднего арифметического вектора

5.1.1 Подробное описание

Заголовочный файл для модуля calculator, отвечающий за вычисления

Автор

Крестина С.Д.

Версия

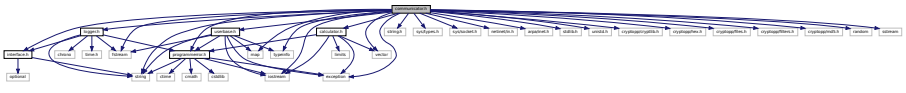
1.0

5.2 Файл communicator.h

Заголовочный файл для модуля communacator, отвечающий за связь с клиентом

```
#include <iostream>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>
#include <vector>
#include <string>
#include <map>
#include <cryptopp/cryptlib.h>
#include <cryptopp/hex.h>
#include <cryptopp/files.h>
#include <cryptopp/filters.h>
#include <cryptopp/md5.h>
#include <fstream>
#include <random>
#include <sstream>
#include <exception>
#include <typeinfo>
#include "programmerror.h"
#include "logger.h"
#include "calculator.h"
#include "userbase.h"
#include "interface.h"
```

Граф включаемых заголовочных файлов для communicator.h:



Классы

- class communicator

  Класс для связи с клиентом

Макросы

- #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1

5.2.1 Подробное описание

Заголовочный файл для модуля communacator, отвечающий за связь с клиентом

Автор

Крестина С.Д.

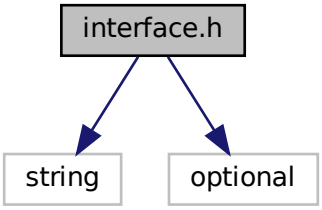
Версия

1.0

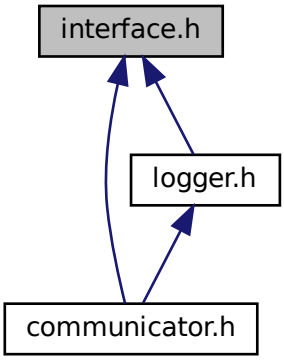
5.3 Файл interface.h

Заголовочный файл для модуля interface, отвечающий за разбор ПКС, и включение других модулей

#include <string>
#include <optional>
Граф включаемых заголовочных файлов для interface.h:



Граф файлов, в которые включается этот файл:



Классы

- class interface

  Класс для разбора командной строки и включения других модулей

5.3.1 Подробное описание

Заголовочный файл для модуля interface, отвечающий за разбор ПКС, и включение других модулей

Автор

Крестина С.Д.

Версия

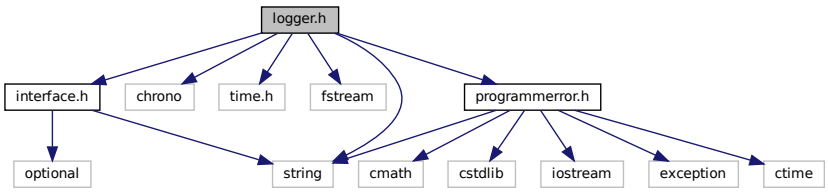
1.0

5.4 Файл logger.h

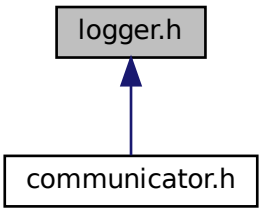
Заголовочный файл для модуля logger, отвечающий за запись логов

```
#include <string>
#include <chrono>
#include <time.h>
#include <fstream>
#include "interface.h"
```

#include "programmerror.h"
Граф включаемых заголовочных файлов для logger.h:



Граф файлов, в которые включается этот файл:



Классы

- class Logger

  Класс для формирования и записи логов

5.4.1 Подробное описание

Заголовочный файл для модуля logger, отвечающий за запись логов

Автор

Крестина С.Д.

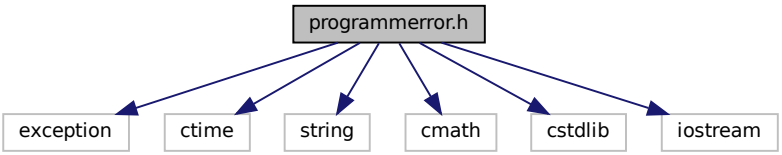
Версия

1.0

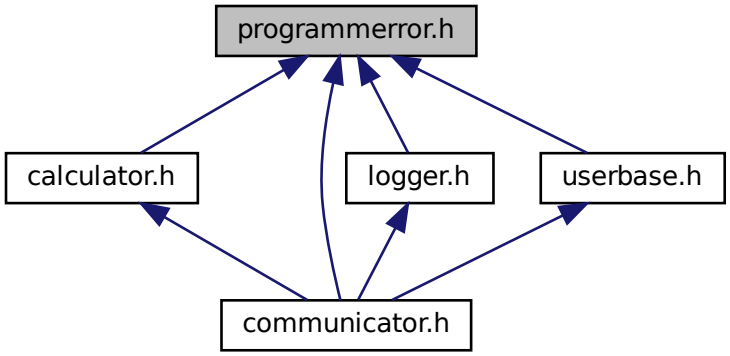
5.5 Файл programmerror.h

Заголовочный файл для модуля server_error, отвечающий за ошибки

#include <exception>
#include <ctime>
#include <string>
#include <cmath>
#include <cstdlib>
#include <iostream>
Граф включаемых заголовочных файлов для programmerror.h:



Граф файлов, в которые включается этот файл:



Классы

- class server_error

  Класс вызова исключений

5.5.1 Подробное описание

Заголовочный файл для модуля server_error, отвечающий за ошибки

Автор

Крестина С.Д.

Версия

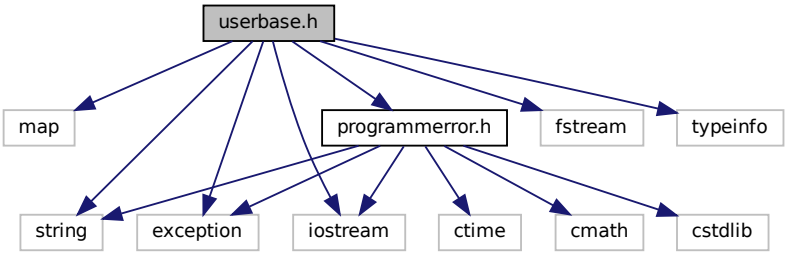
1.0

5.6 Файл userbase.h

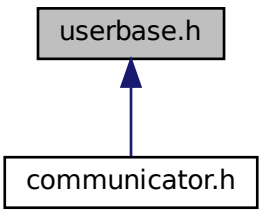
Заголовочный файл для модуля userbase, отвечающий за получение данных из базы клиентов

```
#include <map>
#include <string>
#include <fstream>
#include <exception>
#include <typeinfo>
#include <iostream>
#include "programmerror.h"
```

Граф включаемых заголовочных файлов для userbase.h:



Граф файлов, в которые включается этот файл:



Классы

- class DB

  Класс для получения данных из файла базы клиентов

5.6.1 Подробное описание

Заголовочный файл для модуля userbase, отвечающий за получение данных из базы клиентов

Автор

Крестина С.Д.

Версия

1.0

Предметный указатель