ClientDoubleMD5

Создано системой Doxygen 1.9.4

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 Client

Класс для работы с сервером.

#include <client.h>

Открытые члены

- Client (const string &address, uint16_t port)
  Конструктор класса Client.
- void connectToServer ()
  Устанавливает соединение с сервером.
- void authenticate (const string &username, const string &password)
  Аутентифицирует пользователя на сервере.
- vector< double > calculate (const vector< vector< double > > &data)
  Выполняет вычисления на сервере.
- void closeConnection ()
  Закрывает соединение с сервером.
- const string & getAddress () const
  Возвращает адрес сервера.
- uint16_t getPort () const
  Возвращает порт сервера.

4.1.1 Подробное описание

Класс для работы с сервером.

Этот класс предоставляет методы для установки соединения с сервером, аутентификации пользователя, выполнения вычислений и закрытия соединения.

4.1.2 Конструктор(ы)

4.1.2.1 Client()

```
Client::Client (
                const string & address,
                uint16_t port )
```

Конструктор класса Client.

Аргументы

address	Адрес сервера.
port	Порт сервера.

4.1.3 Методы

4.1.3.1 authenticate()

```
void Client::authenticate (
                const string & username,
                const string & password )
```

Аутентифицирует пользователя на сервере.

Аргументы

username	Имя пользователя.
password	Пароль пользователя.

Исключения

RuntimeError	Если аутентификация не удалась.

4.1.3.2 calculate()

```
vector< double > Client::calculate (
                const vector< vector< double > > & data )
```

Выполняет вычисления на сервере.

Аргументы

data	Данные для вычислений в виде вектора векторов.

Возвращает

Результаты вычислений в виде вектора.

Исключения

RuntimeError	Если не удалось передать данные или получить результат.

4.1.3.3 connectToServer()

void Client::connectToServer ( )

Устанавливает соединение с сервером.

Исключения

RuntimeError	Если не удалось создать сокет или подключиться к серверу.

4.1.3.4 getAddress()

const string & Client::getAddress ( ) const

Возвращает адрес сервера.

Возвращает

Адрес сервера.

4.1.3.5 getPort()

uint16_t Client::getPort ( ) const

Возвращает порт сервера.

Возвращает

Порт сервера.

Объявления и описания членов классов находятся в файлах:

- client.h
- client.cpp

4.2 Класс DataHandler

Класс для работы с данными.

#include <data.h>

Открытые члены

- DataHandler (const string &config_path, const string &input_path, const string &output_path)

  Конструктор класса DataHandler.
- array< string, 2 > loadConfig () const

  Загружает конфигурационные данные из файла.
- vector< vector< double > > readData () const

  Читает данные из входного файла.
- void writeData (const vector< double > &data) const

  Записывает данные в выходной файл.
- const string & getConfigPath () const

  Возвращает путь к файлу конфигурации.
- const string & getInputPath () const

  Возвращает путь к входному файлу.
- const string & getOutputPath () const

  Возвращает путь к выходному файлу.

4.2.1 Подробное описание

Класс для работы с данными.

Этот класс предоставляет методы для загрузки конфигурации, чтения и записи данных.

4.2.2 Конструктор(ы)

4.2.2.1 DataHandler()

```
DataHandler::DataHandler (
            const string & config_path,
            const string & input_path,
            const string & output_path )
```

Конструктор класса DataHandler.

Аргументы

config_path	Путь к файлу конфигурации.
input_path	Путь к входному файлу.
output_path	Путь к выходному файлу.

4.2.3 Методы

4.2.3.1 getConfigPath()

const string & DataHandler::getConfigPath ( ) const

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.2.3.2 getInputPath()

const string & DataHandler::getInputPath ( ) const

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу.

4.2.3.3 getOutputPath()

const string & DataHandler::getOutputPath ( ) const

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу.

4.2.3.4 loadConfig()

array< string, 2 > DataHandler::loadConfig ( ) const

Загружает конфигурационные данные из файла.

Возвращает

Массив строк, содержащий логин и пароль.

Исключения

RuntimeError	Если не удалось открыть файл конфигурации или отсутствует логин или пароль.

4.2.3.5 readData()

vector< vector< double > > DataHandler::readData ( ) const

Читает данные из входного файла.

Возвращает

Вектор векторов данных.

Исключения

RuntimeError	Если не удалось открыть входной файл или произошла ошибка чтения данных.

4.2.3.6 writeData()

void DataHandler::writeData (
const vector< double > & data) const

Записывает данные в выходной файл.

Аргументы

data	Вектор данных для записи.

Исключения

RuntimeError	Если не удалось открыть выходной файл или произошла ошибка записи данных.

Объявления и описания членов классов находятся в файлах:

- data.h
- data.cpp

4.3 Класс RuntimeError

Класс для обработки ошибок времени выполнения.

#include <error.h>

Граф наследования:RuntimeError:

4.4 Класс Terminal

Класс для работы с параметрами командной строки и конфигурацией.

#include <terminal.h>

Открытые члены

- Terminal ()
  Конструктор класса Terminal.
- string getAddress () const
  Возвращает адрес сервера.
- int getPort () const
  Возвращает порт сервера.
- string getInputPath () const
  Возвращает путь к входному файлу.
- string getOutputPath () const
  Возвращает путь к выходному файлу.
- string getConfigPath () const
  Возвращает путь к файлу конфигурации.
- void parseArgs (int argc, char *argv[])
  Разбирает аргументы командной строки и устанавливает соответствующие параметры.
- void showHelp () const
  Показывает справочную информацию.

4.4.1 Подробное описание

Класс для работы с параметрами командной строки и конфигурацией.

Этот класс предоставляет методы для разбора аргументов командной строки, получения параметров конфигурации и отображения справки.

4.4.2 Конструктор(ы)

4.4.2.1 Terminal()

Terminal::Terminal ( )

Конструктор класса Terminal.

Устанавливает значения по умолчанию для адреса сервера, порта и пути к файлу конфигурации.

4.4.3 Методы

4.4.3.1 getAddress()

string Terminal::getAddress ( ) const

Возвращает адрес сервера.

Возвращает

Адрес сервера.

4.4.3.2 getConfigPath()

string Terminal::getConfigPath ( ) const

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.4.3.3 getInputPath()

string Terminal::getInputPath ( ) const

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу.

4.4.3.4 getOutputPath()

string Terminal::getOutputPath ( ) const

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу.

4.4.3.5 getPort()

int Terminal::getPort ( ) const

Возвращает порт сервера.

Возвращает

Порт сервера.

4.4.3.6 parseArgs()

void Terminal::parseArgs (
int argc,
char * argv[])

Разбирает аргументы командной строки и устанавливает соответствующие параметры.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив аргументов командной строки.

Исключения

RuntimeError	Если отсутствует значение для параметра или обнаружен неизвестный параметр.

4.4.3.7 showHelp()

void Terminal::showHelp ( ) const

Показывает справочную информацию.

Выводит справочное сообщение о доступных параметрах командной строки.

Объявления и описания членов классов находятся в файлах:

- terminal.h
- terminal.cpp

Глава 5

Файлы

5.1 client.h

```
1 #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
2 #pragma once
3
4 #include "error.h"
5 #include <string>
6 #include <vector>
7 #include <cstdint>
8 #include <cstring>
9 #include <unistd.h>
10 #include <iostream>
11
12 #include <sys/types.h>
13 #include <sys/socket.h>
14 #include <arpa/inet.h>
15
16 #include <cryptopp/hex.h>
17 #include <cryptopp/md5.h>
18 #include <cryptopp/osrng.h>
19
20 using namespace std;
21 using namespace CryptoPP;
22 using namespace CryptoPP::Weak1;
23
31 class Client
32 {
33 public:
40     Client(const string &address, uint16_t port);
41
47     void connectToServer();
48
56     void authenticate(const string &username, const string &password);
57
65     vector<double> calculate(const vector<vector<double» &data);
66
70     void closeConnection();
71
77     const string &getAddress() const;
78
84     uint16_t getPort() const;
85
86 private:
87     string address_;
88     uint16_t port_;
89     int socket_;
90 };
```

5.2 data.h

```
1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <array>
6 #include <fstream>
```

```
7 #include <iostream>
8 #include <iomanip>
9 #include <sstream>
10 #include "error.h"
11
12 using namespace std;
13
20 class DataHandler
21 {
22 public:
30     DataHandler(const string &config_path, const string &input_path, const string &output_path);
31
38     array<string, 2> loadConfig() const;
39
46     vector<vector<double» readData() const;
47
54     void writeData(const vector<double> &data) const;
55
61     const string &getConfigPath() const;
62
68     const string &getInputPath() const;
69
75     const string &getOutputPath() const;
76
77 private:
78     string config_path;
79     string input_path;
80     string output_path;
81 };
82
88 void PrintVector(const vector<double> &data);
89
95 void PrintVectors(const vector<vector<double» &data);
```

5.3 error.h

```
1 #pragma once
2
3 #include <stdexcept>
4 #include <string>
5
6 using namespace std;
7
14 class RuntimeError : public runtime_error
15 {
16 public:
24     RuntimeError(const string &message, const string &func);
25 };
```

5.4 terminal.h

```
1 #pragma once
2
3 #include "error.h"
4 #include <string>
5 #include <vector>
6
7 using namespace std;
8
16 class Terminal
17 {
18 public:
24     Terminal();
25
31     string getAddress() const;
32
38     int getPort() const;
39
45     string getInputPath() const;
46
52     string getOutputPath() const;
53
59     string getConfigPath() const;
60
68     void parseArgs(int argc, char *argv[]);
69
75     void showHelp() const;
76
77 private:
78     string address_;
```

```
79     uint16_t port_;
80     string input_path_;
81     string output_path_;
82     string config_path_;
83 };
```

Предметный указатель