CLIENT:INBIN:OUTBIN:FLOAT

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 управления подключением и взаимодействием с сервером

#include <client.h>

Открытые члены

- Client (const string &address, uint16_t port)
  Конструктор класса Client.
- void conn ()
  Установить соединение с сервером
- void auth (const string &username, const string &password)
  Аутентификация пользователя на сервере
- vector< float > calc (const vector< vector< float > > &data)
  Выполнение расчетов на сервере
- void close ()
  Закрытие соединения с сервером
- string getAddress () const
  Получить адрес сервера
- uint16_t getPort () const
  Получить порт сервера

4.1.1 Подробное описание

Класс для управления подключением и взаимодействием с сервером

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 auth()

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

Аутентификация пользователя на сервере

Аргументы

username	Имя пользователя
password	Пароль пользователя

Исключения

ClientError	Если аутентификация не удалась

4.1.3.2 calc()

```
vector< float > Client::calc (
                const vector< vector< float > > & data )
```

Выполнение расчетов на сервере

Аргументы

data	Входные данные для расчетов

Возвращает

Результаты расчетов

Исключения

ClientError	Если выполнение расчетов не удалось

4.1.3.3 conn()

void Client::conn ( )

Установить соединение с сервером

Исключения

ClientError	Если не удается установить соединение

4.1.3.4 getAddress()

string Client::getAddress ( ) const

Получить адрес сервера

Возвращает

Адрес сервера

4.1.3.5 getPort()

uint16_t Client::getPort ( ) const

Получить порт сервера

Возвращает

Порт сервера

Объявления и описания членов классов находятся в файлах:

- client.h
- client.cpp

4.2 Класс ClientError

Класс исключений для работы с клиентом

#include <error.h>

Граф наследования:ClientError:

4.3 Класс CMDParser

Класс для разбора аргументов командной строки

#include <cli.h>

Открытые члены

- CMDParser ()
  Конструктор класса CMDParser.
- string getAddress ()
  Получить адрес сервера
- int getPort ()
  Получить порт сервера
- string getInputPath ()
  Получить путь к входному файлу данных
- string getOutputPath ()
  Получить путь к выходному файлу данных
- string getConfigPath ()
  Получить путь к файлу конфигурации
- void parseArgs (int argc, char *argv[])
  Разбор аргументов командной строки
- void showHelp ()
  Показ справки по использованию программы

4.3.1 Подробное описание

Класс для разбора аргументов командной строки

4.3.2 Методы

4.3.2.1 getAddress()

string CMDParser::getAddress ( )

Получить адрес сервера

Возвращает

Адрес сервера

4.3.2.2 getConfigPath()

string CMDParser::getConfigPath ( )

Получить путь к файлу конфигурации

Возвращает

Путь к файлу конфигурации

4.3.2.3 getInputPath()

string CMDParser::getInputPath ( )

Получить путь к входному файлу данных

Возвращает

Путь к входному файлу данных

4.3.2.4 getOutputPath()

string CMDParser::getOutputPath ( )

Получить путь к выходному файлу данных

Возвращает

Путь к выходному файлу данных

4.3.2.5 getPort()

int CMDParser::getPort ( )

Получить порт сервера

Возвращает

Порт сервера

4.3.2.6 parseArgs()

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

Разбор аргументов командной строки

Аргументы

argc	Количество аргументов
argv	Массив аргументов

Исключения

ClientError	Если аргументы некорректные или отсутствуют

Объявления и описания членов классов находятся в файлах:

- cli.h
- cli.cpp

4.4 Класс Data

Класс для управления конфигурационными данными и данными для расчетов

#include <data.h>

Открытые члены

- Data (const string &path_to_conf, const string &path_to_in, const string &path_to_out)
  Конструктор класса Data.
- array< string, 2 > conf ()
  Чтение конфигурационных данных
- vector< vector< float > > read ()
  Чтение входных данных
- void write (const vector< float > &data)
  Запись выходных данных
- string getConfigPath () const
  Получить путь к файлу конфигурации
- string getInputPath () const
  Получить путь к входному файлу данных
- string getOutputPath () const
  Получить путь к выходному файлу данных

4.4.1 Подробное описание

Класс для управления конфигурационными данными и данными для расчетов

4.4.2 Конструктор(ы)

4.4.2.1 Data()

```
Data::Data (
            const string & path_to_conf,
            const string & path_to_in,
            const string & path_to_out )
```

Конструктор класса Data.

Аргументы

path_to_conf	Путь к файлу конфигурации
path_to_in	Путь к входному файлу данных
path_to_out	Путь к выходному файлу данных

4.4.3 Методы

4.4.3.1 conf()

array< string, 2 > Data::conf ( )

Чтение конфигурационных данных

Возвращает

Пара логин и пароль

Исключения

ClientError	Если не удается прочитать конфигурационный файл

4.4.3.2 getConfigPath()

string Data::getConfigPath ( ) const

Получить путь к файлу конфигурации

Возвращает

Путь к файлу конфигурации

4.4.3.3 getInputPath()

string Data::getInputPath ( ) const

Получить путь к входному файлу данных

Возвращает

Путь к входному файлу данных

4.4.3.4 getOutputPath()

string Data::getOutputPath ( ) const

Получить путь к выходному файлу данных

Возвращает

Путь к выходному файлу данных

4.4.3.5 read()

vector< vector< float > > Data::read ( )

Чтение входных данных

Возвращает

Входные данные для расчетов

Исключения

ClientError	Если не удается прочитать входной файл

4.4.3.6 write()

void Data::write (
const vector< float > & data)

Запись выходных данных

Аргументы

data	Данные для записи

Исключения

ClientError	Если не удается записать в выходной файл

Объявления и описания членов классов находятся в файлах:

- data.h
- data.cpp

Глава 5

Файлы

5.1 cli.h

```
1 #pragma once
2
3 #include "error.h"
4 #include <string>
5 #include <vector>
6 #include <iostream>
7 #include <cstring>
8
9 using namespace std;
10
14 class CMDParser
15 {
16 public:
20     CMDParser();
21
27     string getAddress();
28
34     int getPort();
35
41     string getInputPath();
42
48     string getOutputPath();
49
55     string getConfigPath();
56
64     void parseArgs(int argc, char *argv[]);
65
69     void showHelp();
70
71 private:
72     string address = "127.0.0.1";
73     uint16_t port = 33333;
74     string input_path = "input.bin";
75     string output_path = "output.bin";
76     string config_path = "config.txt";
77 };
```

5.2 client.h

```
1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <cstdint>
6
7 using namespace std;
8
12 class Client
13 {
14 public:
21     Client(const string &address, uint16_t port);
22
28     void conn();
29
```

```
37     void auth(const string &username, const string &password);
38
46     vector<float> calc(const vector<vector<float>> &data);
47
51     void close();
52
58     string getAddress() const;
59
65     uint16_t getPort() const;
66
67 private:
68     int socket;
69     string address;
70     uint16_t port;
71 };
```

5.3 data.h

```
1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <array>
6 #include "error.h"
7 #include <fstream>
8 #include <sstream>
9 #include <iostream>
10
11 using namespace std;
12
16 class Data
17 {
18 public:
26     Data(const string &path_to_conf, const string &path_to_in, const string &path_to_out);
27
34     array<string, 2> conf();
35
42     vector<vector<float>> read();
43
50     void write(const vector<float> &data);
51
57     string getConfigPath() const;
58
64     string getInputPath() const;
65
71     string getOutputPath() const;
72
73 private:
74     string path_to_conf;
75     string path_to_in;
76     string path_to_out;
77 };
78
84 void printVector(const vector<float>& vec);
85
91 void printVectors(const vector<vector<float>>& vectors);
```

5.4 error.h

```
1 #pragma once
2
3 #include <exception>
4 #include <string>
5
6 using namespace std;
7
11 class ClientError : public exception
12 {
13 public:
20     ClientError(const string &errorName, const string &func);
21
27     const char *what() const noexcept override;
28
34     string getName() const;
35
41     string getFunc() const;
42
43 protected:
44     string name;
45     string func;
```

```
46     mutable string message;
47 };
```

Предметный указатель