

Cursovaya

Создано системой Doxygen 1.9.4

1	Алфавитный указатель классов	1
1.1	Классы	1
2	Список файлов	3
2.1	Файлы	3
3	Классы	5
3.1	Класс Client	5
3.1.1	Подробное описание	5
3.1.2	Конструктор(ы)	5
3.1.2.1	Client()	5
3.1.3	Методы	6
3.1.3.1	Authenticate()	6
3.1.3.2	Connect()	6
3.1.3.3	SendVectors()	6
3.1.3.4	SHA224_hash()	8
3.2	Класс Interface	8
3.2.1	Подробное описание	9
3.2.2	Конструктор(ы)	9
3.2.2.1	Interface()	9
3.2.3	Методы	9
3.2.3.1	GetConfigFile()	9
3.2.3.2	GetInputFile()	10
3.2.3.3	GetOutputFile()	10
3.2.3.4	GetServerAddress()	10
3.2.3.5	GetServerPort()	10
3.2.3.6	ParseCommandLine()	10
3.2.3.7	ReadConfigFile()	11
3.2.3.8	ReadVectorsFromFile()	11
4	Файлы	13
4.1	Файл /home/stud/Desktop/клиент/client_new/interface.cpp	13
4.1.1	Подробное описание	13
4.2	Файл /home/stud/Desktop/клиент/client_new/main.cpp	14
4.2.1	Подробное описание	14
4.2.2	Функции	14
4.2.2.1	main()	14
4.3	Файл /home/stud/Desktop/клиент/client_new/client.h	15
4.3.1	Подробное описание	16
4.4	client.h	16
4.5	Файл /home/stud/Desktop/клиент/client_new/interface.h	17
4.5.1	Подробное описание	18
4.6	interface.h	18
4.7	Файл /home/stud/Desktop/клиент/client_new/client.cpp	19

4.7.1 Подробное описание	19
Предметный указатель	21

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Client	Класс для взаимодействия с сервером	5
Interface	Класс для обработки командной строки и чтения файлов	8

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

/home/stud/Desktop/клиент/client_new/client.cpp	
Реализация класса Client	19
/home/stud/Desktop/клиент/client_new/client.h	
Заголовочный файл для класса Client , который отвечает за взаимодействие с сервером	15
/home/stud/Desktop/клиент/client_new/interface.cpp	
Реализация класса Interface	13
/home/stud/Desktop/клиент/client_new/interface.h	
Заголовочный файл для класса Interface , который отвечает за обработку командной строки и чтение файлов	17
/home/stud/Desktop/клиент/client_new/main.cpp	
Основной файл программы	14

Глава 3

Классы

3.1 Класс Client

Класс для взаимодействия с сервером.

```
#include <client.h>
```

Открытые члены

- [Client](#) (const std::string &server_address, int server_port)
Конструктор класса [Client](#).
- [~Client](#) ()
Деструктор класса [Client](#).
- void [Connect](#) ()
Подключение к серверу.
- void [Authenticate](#) (const std::string &login, const std::string &password)
Аутентификация на сервере.
- void [SendVectors](#) (const std::vector< std::vector< float > > &vectors, const std::string &output←_file)
Отправка векторов на сервер.
- std::string [SHA224_hash](#) (const std::string &input)
Вычисление SHA-224 хеша.

3.1.1 Подробное описание

Класс для взаимодействия с сервером.

Этот класс предоставляет методы для подключения к серверу, аутентификации и отправки данных.

3.1.2 Конструктор(ы)

3.1.2.1 Client()

```
Client::Client (  
    const std::string & server_address,  
    int server_port )
```

Конструктор класса [Client](#).

Аргументы

server_address	Адрес сервера.
server_port	Порт сервера.

3.1.3 Методы

3.1.3.1 Authenticate()

```
void Client::Authenticate (
    const std::string & login,
    const std::string & password )
```

Аутентификация на сервере.

Аргументы

login	Логин пользователя.
password	Пароль пользователя.

Исключения

std::runtime_error	В случае ошибки аутентификации.
--------------------	---------------------------------

3.1.3.2 Connect()

```
void Client::Connect ( )
```

Подключение к серверу.

Исключения

std::runtime_error	В случае ошибки подключения.
--------------------	------------------------------

3.1.3.3 SendVectors()

```
void Client::SendVectors (
    const std::vector< std::vector< float > > & vectors,
    const std::string & output_file )
```

Отправка векторов на сервер.

Аргументы

vectors	Векторы для отправки.
output_file	Файл для записи результатов.

Исключения

std::runtime_error	В случае ошибки отправки данных.
--------------------	----------------------------------

3.1.3.4 SHA224_hash()

```
std::string Client::SHA224_hash (
    const std::string & input )
```

Вычисление SHA-224 хеша.

Аргументы

input	Входная строка для хеширования.
-------	---------------------------------

Возвращает

Хеш в виде строки.

Объявления и описания членов классов находятся в файлах:

- /home/stud/Desktop/клиент/client_new/[client.h](#)
- /home/stud/Desktop/клиент/client_new/[client.cpp](#)

3.2 Класс Interface

Класс для обработки командной строки и чтения файлов.

```
#include <interface.h>
```

Открытые члены

- [Interface](#) (int argc, char *argv[])
Конструктор класса [Interface](#).
- void [ParseCommandLine](#) ()
Парсинг командной строки.
- std::vector< std::vector< float > > [ReadVectorsFromFile](#) (const std::string &filename)
Чтение векторов из файла.
- void [ReadConfigFile](#) (const std::string &config_file, std::string &login, std::string &password)

- Чтение конфигурационного файла.
- `std::string GetInputFile () const`
Получение имени входного файла.
- `std::string GetOutputFile () const`
Получение имени выходного файла.
- `std::string GetConfigFile () const`
Получение имени конфигурационного файла.
- `std::string GetServerAddress () const`
Получение адреса сервера.
- `int GetServerPort () const`
Получение порта сервера.

3.2.1 Подробное описание

Класс для обработки командной строки и чтения файлов.

Этот класс предоставляет методы для парсинга командной строки, чтения векторов из файла и чтения конфигурационного файла.

3.2.2 Конструктор(ы)

3.2.2.1 Interface()

```
Interface::Interface (  
    int argc,  
    char * argv[] )
```

Конструктор класса [Interface](#).

Аргументы

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

3.2.3 Методы

3.2.3.1 GetConfigFile()

```
std::string Interface::GetConfigFile ( ) const    [inline]
```

Получение имени конфигурационного файла.

Возвращает

Имя конфигурационного файла.

3.2.3.2 GetInputFile()

```
std::string Interface::GetInputFile ( ) const [inline]
```

Получение имени входного файла.

Возвращает

Имя входного файла.

3.2.3.3 GetOutputFile()

```
std::string Interface::GetOutputFile ( ) const [inline]
```

Получение имени выходного файла.

Возвращает

Имя выходного файла.

3.2.3.4 GetServerAddress()

```
std::string Interface::GetServerAddress ( ) const [inline]
```

Получение адреса сервера.

Возвращает

Адрес сервера.

3.2.3.5 GetServerPort()

```
int Interface::GetServerPort ( ) const [inline]
```

Получение порта сервера.

Возвращает

Порт сервера.

3.2.3.6 ParseCommandLine()

```
void Interface::ParseCommandLine ( )
```

Парсинг командной строки.

Исключения

<code>std::runtime_error</code>	В случае ошибки парсинга.
---------------------------------	---------------------------

3.2.3.7 ReadConfigFile()

```
void Interface::ReadConfigFile (
    const std::string & config_file,
    std::string & login,
    std::string & password )
```

Чтение конфигурационного файла.

Аргументы

<code>config_file</code>	Имя конфигурационного файла.
<code>login</code>	Ссылка на строку для сохранения логина.
<code>password</code>	Ссылка на строку для сохранения пароля.

Исключения

<code>std::runtime_error</code>	В случае ошибки чтения файла.
---------------------------------	-------------------------------

3.2.3.8 ReadVectorsFromFile()

```
std::vector< std::vector< float > > Interface::ReadVectorsFromFile (
    const std::string & filename )
```

Чтение векторов из файла.

Аргументы

<code>filename</code>	Имя файла.
-----------------------	------------

Возвращает

Вектор векторов типа `float`.

Исключения

<code>std::runtime_error</code>	В случае ошибки чтения файла.
---------------------------------	-------------------------------

Объявления и описания членов классов находятся в файлах:

- `/home/stud/Desktop/клиент/client_new/interface.h`
- `/home/stud/Desktop/клиент/client_new/interface.cpp`

Глава 4

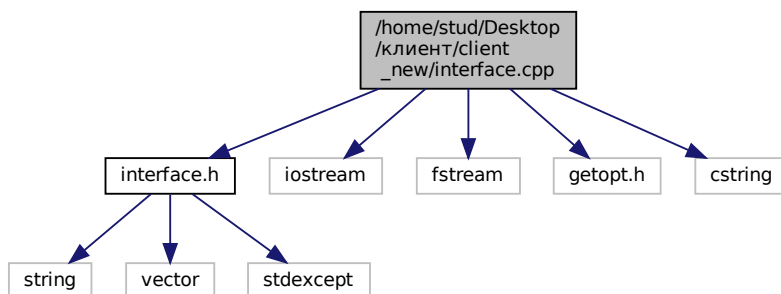
Файлы

4.1 Файл /home/stud/Desktop/клиент/client_new/interface.cpp

Реализация класса [Interface](#).

```
#include "interface.h"  
#include <iostream>  
#include <fstream>  
#include <getopt.h>  
#include <cstring>
```

Граф включаемых заголовочных файлов для interface.cpp:



4.1.1 Подробное описание

Реализация класса [Interface](#).

Автор

Донскова А.Д.

Версия

1.0

Дата

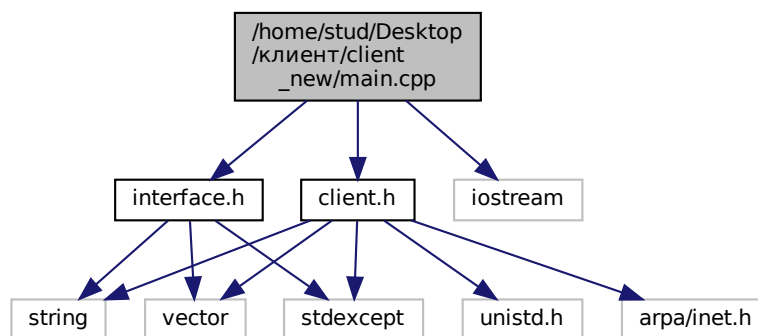
22.12.2024

4.2 Файл /home/stud/Desktop/клиент/client_new/main.cpp

Основной файл программы.

```
#include "interface.h"  
#include "client.h"  
#include <iostream>
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- `int main (int argc, char *argv[])`
Точка входа в программу.

4.2.1 Подробное описание

Основной файл программы.

Автор

Донскова А.Д.

Версия

1.0

Дата

22.12.2024

4.2.2 Функции

4.2.2.1 main()

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

Точка входа в программу.

Аргументы

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

Возвращает

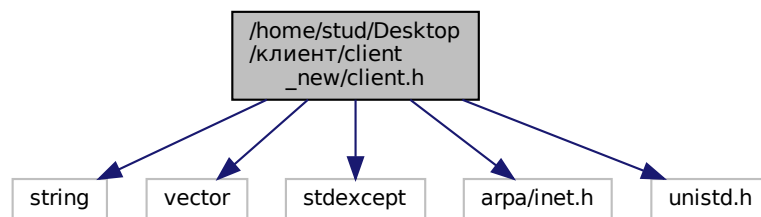
Код возврата.

4.3 Файл /home/stud/Desktop/клиент/client_new/client.h

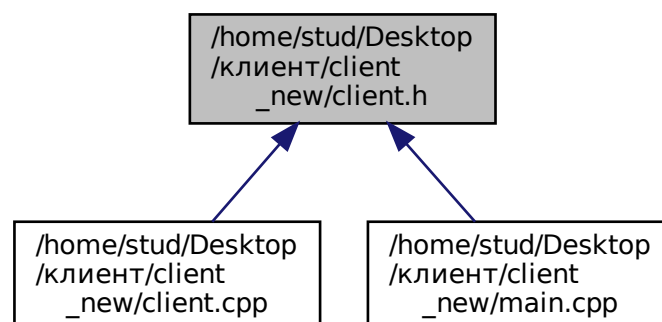
Заголовочный файл для класса [Client](#), который отвечает за взаимодействие с сервером.

```
#include <string>
#include <vector>
#include <stdexcept>
#include <arpa/inet.h>
#include <unistd.h>
```

Граф включаемых заголовочных файлов для client.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Client](#)

Класс для взаимодействия с сервером.

4.3.1 Подробное описание

Заголовочный файл для класса [Client](#), который отвечает за взаимодействие с сервером.

Автор

Донскова А.Д.

Версия

1.0

Дата

22.12.2024

4.4 client.h

[См. документацию.](#)

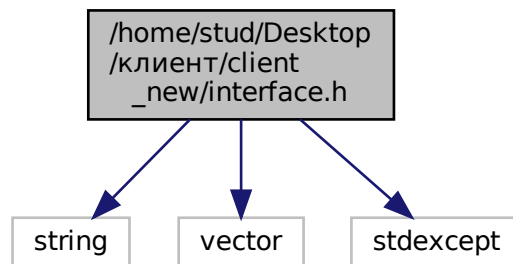
```
1
2
3
4
5
6
7
8 #ifndef CLIENT_H
9 #define CLIENT_H
10
11 #include <string>
12 #include <vector>
13 #include <stdexcept>
14 #include <arpa/inet.h>
15 #include <unistd.h>
16
17
18
19
20
21
22
23 class Client {
24 public:
25     Client(const std::string& server_address, int server_port);
26
27     ~Client();
28
29
30     void Connect();
31
32
33     void Authenticate(const std::string& login, const std::string& password);
34
35
36     void SendVectors(const std::vector<std::vector<float>>& vectors, const std::string& output_file);
37
38
39     std::string SHA224_hash(const std::string& input);
40
41 private:
42     std::string server_address_;
43     int server_port_;
44     int sock_;
45 };
46
47 #endif // CLIENT_H
```

4.5 Файл /home/stud/Desktop/клиент/client_new/interface.h

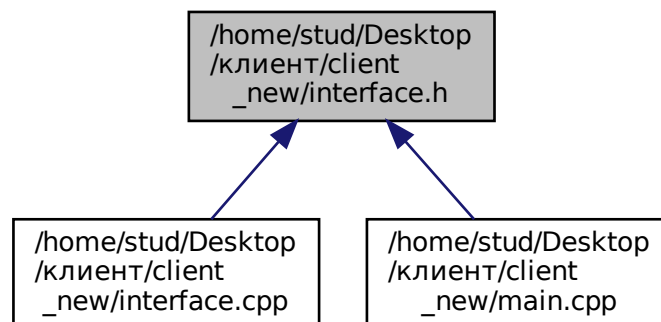
Заголовочный файл для класса [Interface](#), который отвечает за обработку командной строки и чтение файлов.

```
#include <string>
#include <vector>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для interface.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Interface](#)

Класс для обработки командной строки и чтения файлов.

4.5.1 Подробное описание

Заголовочный файл для класса [Interface](#), который отвечает за обработку командной строки и чтение файлов.

Автор

Донскова А.Д.

Версия

1.0

Дата

22.12.2024

4.6 interface.h

[См. документацию.](#)

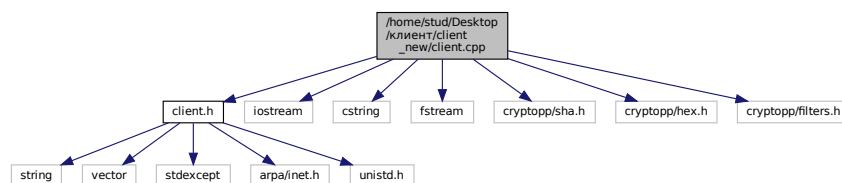
```
1
2
3 8 #ifndef INTERFACE_H
4 9 #define INTERFACE_H
5 10
6 11 #include <string>
7 12 #include <vector>
8 13 #include <stdexcept>
9 14
10 21 class Interface {
11 22 public:
12 23     Interface(int argc, char* argv[]);
13 24
14 25     void ParseCommandLine();
15 26
16 27     std::vector<std::vector<float>> ReadVectorsFromFile(const std::string& filename);
17 28
18 29     void ReadConfigFile(const std::string& config_file, std::string& login, std::string& password);
19 30
20 31     std::string GetInputFile()const { return input_file_; }
21 32
22 33     std::string GetOutputFile()const { return output_file_; }
23 34
24 35     std::string GetConfigFile()const { return config_file_; }
25 36
26 37     std::string GetServerAddress()const { return server_address_; }
27 38
28 39     int GetServerPort()const { return server_port_; }
29 40
30 41 private:
31 42     int argc_;
32 43     char** argv_;
33 44     std::string input_file_;
34 45     std::string output_file_;
35 46     std::string config_file_;
36 47     std::string server_address_;
37 48     int server_port_;
38 49 };
39 50
40 93 #endif // INTERFACE_H
```

4.7 Файл /home/stud/Desktop/клиент/client_new/client.cpp

Реализация класса [Client](#).

```
#include "client.h"
#include <iostream>
#include <cstring>
#include <fstream>
#include <cryptopp/sha.h>
#include <cryptopp/hex.h>
#include <cryptopp/filters.h>
```

Граф включаемых заголовочных файлов для client.cpp:



4.7.1 Подробное описание

Реализация класса [Client](#).

Автор

Донскова А.Д.

Версия

1.0

Дата

22.12.2024

Предметный указатель

/home/stud/Desktop/клиент/client_new/client.cpp,	Interface, 10
19	
/home/stud/Desktop/клиент/client_new/client.h,	ReadConfigFile
15, 16	Interface, 11
/home/stud/Desktop/клиент/client_new/interface.cpp,	ReadVectorsFromFile
13	Interface, 11
/home/stud/Desktop/клиент/client_new/interface.h,	SendVectors
17, 18	
/home/stud/Desktop/клиент/client_new/main.cpp,	Client, 6
14	SHA224_hash
	Client, 8

Authenticate
Client, 6

```
Client, 5
  Authenticate, 6
    Client, 5
    Connect, 6
    SendVectors, 6
    SHA224 hash, 8
```

Connect
Client, 6

GetConfigFile Interface, 9

GetInputFile
Interface, 10

GetOutputFile
Interface, [10](#)

GetServerAddress
Interface, [10](#)

GetServerPort
Interface, [10](#)

- Interface, [8](#)
- GetConfigFile, [9](#)
- GetInputFile, [10](#)
- GetOutputFile, [10](#)
- GetServerAddress, [10](#)
- GetServerPort, [10](#)
- Interface, [9](#)
- ParseCommandLine, [10](#)
- ReadConfigFile, [11](#)
- ReadVectorsFromFile, [11](#)

```
main
    main.cpp, 14
main.cpp
    main, 14
```

ParseCommandLine