My Project

Создано системой Doxygen 1.9.8

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Configuration
: < Псевдоним для пространства имен program_options . . . . . ??

ConnectionManager
: Менеджер сетевых соединений . . . . . ??

UserInterface
: Класс для обработки аргументов командной строки . . . . . ??

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

connection.cpp
: Реализация менеджера сетевых соединений . . . . . ??

connection.h
: Заголовочный файл для управления сетевыми соединениями . . . . . ??

crypto.cpp
: Реализация криптографических функций . . . . . ??

crypto.h
: Заголовочный файл для криптографических функций . . . . . ??

interface.cpp
: Реализация пользовательского интерфейса . . . . . ??

interface.h
: Заголовочный файл для пользовательского интерфейса . . . . . ??

main.cpp
: Главный файл приложения . . . . . ??

Глава 3

Классы

3.1 Структура Configuration

< Псевдоним для пространства имен program_options

#include <interface.h>

Открытые атрибуты

- string input_filename
  Имя входного файла с данными
- string output_filename
  Имя выходного файла с результатами
- string credentials_filename
  Имя файла с учетными данными
- int port_number
  Номер порта сервера
- string server_address
  Адрес сервера

3.1.1 Подробное описание

< Псевдоним для пространства имен program_options

Структура для хранения параметров конфигурации

Объявления и описания членов структуры находятся в файле:

- interface.h

3.2 Класс ConnectionManager

Менеджер сетевых соединений

#include <connection.h>

Открытые статические члены

- static int establish_connection (const Configuration *params)
  Установка соединения с сервером

3.2.1 Подробное описание

Менеджер сетевых соединений

Осуществляет установку соединения с сервером, аутентификацию и передачу данных

3.2.2 Методы

3.2.2.1 establish_connection()

```
int ConnectionManager::establish_connection (
                const Configuration * params )   [static]
```

Установка соединения с сервером

Аргументы

	in	params	Параметры конфигурации соединения

Возвращает

0 в случае успеха

Исключения

	system_error	при ошибках сети или файловой системы

Объявления и описания членов классов находятся в файлах:

- connection.h
- connection.cpp

3.3 Класс UserInterface

Класс для обработки аргументов командной строки

#include <interface.h>

Открытые члены

- UserInterface ()

  Конструктор класса UserInterface.
- bool parse_arguments (int arg_count, const char **arg_values)

  Парсинг аргументов командной строки
- string get_help_text ()

  Получение текста справки
- Configuration get_configuration ()

  Получение конфигурации

3.3.1 Подробное описание

Класс для обработки аргументов командной строки

3.3.2 Методы

3.3.2.1 get_configuration()

```
Configuration UserInterface::get_configuration ( )  [inline]
```

Получение конфигурации

Возвращает

Структура Configuration с настройками

3.3.2.2 get_help_text()

```
string UserInterface::get_help_text ( )
```

Получение текста справки

Возвращает

Строка с текстом справки

3.3.2.3 parse_arguments()

```
bool UserInterface::parse_arguments (
            int arg_count,
            const char ** arg_values )
```

Парсинг аргументов командной строки

Аргументы

	in	arg_count	Количество аргументов
	in	arg_values	Массив аргументов

Возвращает

true если парсинг успешен, false в противном случае

Объявления и описания членов классов находятся в файлах:

- interface.h
- interface.cpp

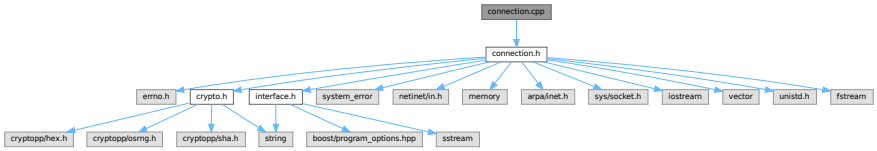
Глава 4

Файлы

4.1 Файл connection.cpp

Реализация менеджера сетевых соединений

#include "connection.h"
Граф включаемых заголовочных файлов для connection.cpp:



Функции

- void readCredentialsFromFile (const std::string &credentials_filename, std::string &username, std::string &password)

  Чтение учетных данных из файла (интегрированная версия)

4.1.1 Подробное описание

Реализация менеджера сетевых соединений

Автор

Сочилов Н.М.

Версия

1.0

Дата

2025

4.1.2 Функции

4.1.2.1 readCredentialsFromFile()

void readCredentialsFromFile (
const std::string & credentials_filename,
std::string & username,
std::string & password)

Чтение учетных данных из файла (интегрированная версия)

Аргументы

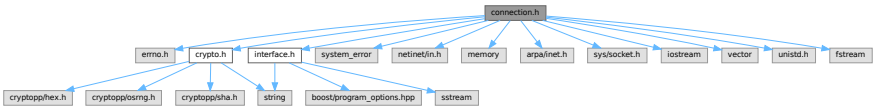
	in	credentials_filename	Имя файла с учетными данными
	out	username	Логин пользователя
	out	password	Пароль пользователя

4.2 Файл connection.h

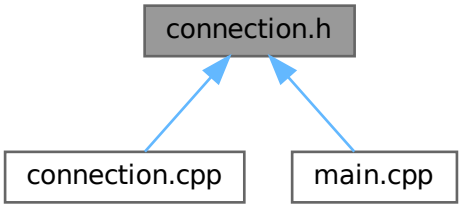
Заголовочный файл для управления сетевыми соединениями

```
#include "errno.h"
#include "crypto.h"
#include "interface.h"
#include <system_error>
#include <netinet/in.h>
#include <memory>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <iostream>
#include <vector>
#include <unistd.h>
#include <fstream>
```

Граф включаемых заголовочных файлов для connection.h:



Граф файлов, в которые включается этот файл:



Классы

- class ConnectionManager

  Менеджер сетевых соединений

Макросы

- #define BUFFER_SIZE 1024

  Размер буфера для сетевых операций

4.2.1 Подробное описание

Заголовочный файл для управления сетевыми соединениями

Версия

1.0

Дата

2025

Предупреждения

Реализация для Linux-систем

4.3 connection.h

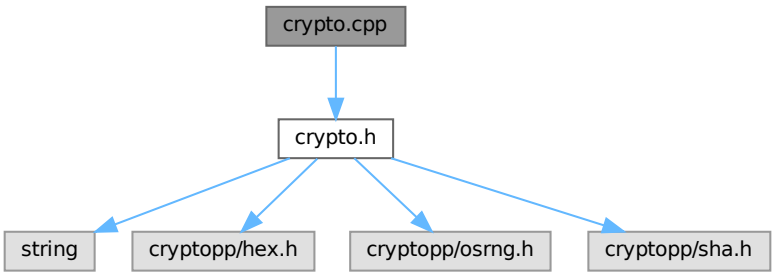
См. документацию.

```
00001
00009 #pragma once
00010
00011 #include "errno.h"
00012 #include "crypto.h"
00013 #include "interface.h"
00014 #include <system_error>
00015 #include <netinet/in.h>
00016 #include <memory>
00017 #include <arpa/inet.h>
00018 #include <sys/socket.h>
00019 #include <iostream>
00020 #include <vector>
00021 #include <unistd.h>
00022 #include <fstream>
00023 #include <vector>
00024
00025 using namespace std;
00026
00027 #define BUFFER_SIZE 1024
00028
00034 class ConnectionManager {
00035 private:
00036     static string salt;
00037
00038 public:
00045     static int establish_connection(const Configuration* params);
00046 };
```

4.4 Файл crypto.cpp

Реализация криптографических функций

#include "crypto.h"
Граф включаемых заголовочных файлов для crypto.cpp:



Функции

- string generate_hash (string salt_value, string password_value)
  < Псевдоним для пространства имен CryptoPP

4.4.1 Подробное описание

Реализация криптографических функций

Автор

Сочилов Н.М.

Версия

1.0

Дата

2025

4.4.2 Функции

4.4.2.1 generate_hash()

```
string generate_hash (
                string salt_value,
                string password_value )
```

< Псевдоним для пространства имен CryptoPP

Генерация хеша для аутентификации

Аргументы

	in	salt_value	Соль для хеширования
	in	password_value	Пароль пользователя

Возвращает

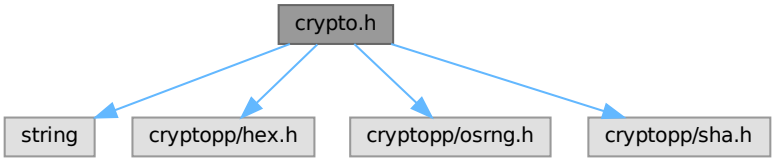
Хеш в шестнадцатеричном формате

4.5 Файл crypto.h

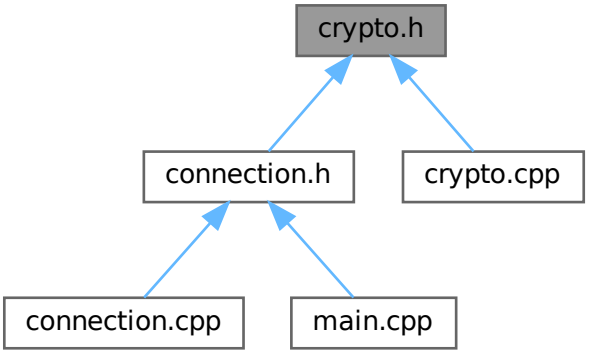
Заголовочный файл для криптографических функций

```
#include <string>
#include <cryptopp/hex.h>
#include <cryptopp/osrng.h>
```

#include <cryptopp/sha.h>
Граф включаемых заголовочных файлов для crypto.h:



Граф файлов, в которые включается этот файл:



Функции

- string generate_hash (string salt_value, string password_value)
  < Псевдоним для пространства имен CryptoPP

4.5.1 Подробное описание

Заголовочный файл для криптографических функций

Версия

1.0

Дата

2024

Заметки

Используется библиотека CryptoPP

4.5.2 Функции

4.5.2.1 generate_hash()

```
string generate_hash (
                string salt_value,
                string password_value )
```

< Псевдоним для пространства имен CryptoPP

Генерация хеша для аутентификации

Аргументы

	in	salt_value	Соль для хеширования
	in	password_value	Пароль пользователя

Возвращает

Хеш в шестнадцатеричном формате

4.6 crypto.h

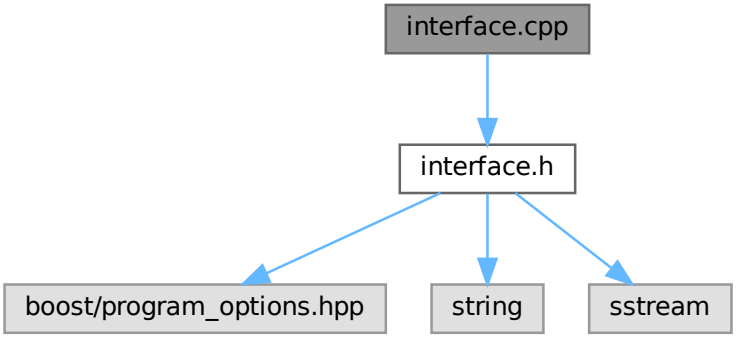
См. документацию.

```
00001
00009 #pragma once
00010
00011 #include <string>
00012 #include <cryptopp/hex.h>
00013 #include <cryptopp/osrng.h>
00014 #include <cryptopp/sha.h>
00015
00016 using namespace std;
00017
00018 namespace CPP = CryptoPP;
00019
00026 string generate_hash(string salt_value, string password_value);
```

4.7 Файл interface.cpp

Реализация пользовательского интерфейса

#include "interface.h"
Граф включаемых заголовочных файлов для interface.cpp:



4.7.1 Подробное описание

Реализация пользовательского интерфейса

Автор

Сочилов Н.М.

Версия

1.0

Дата

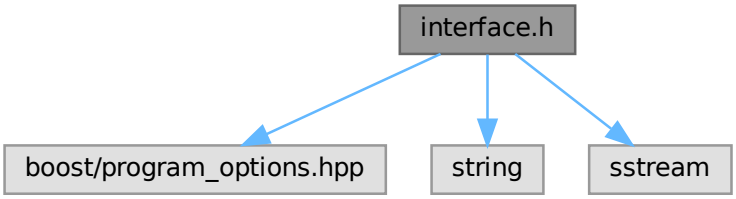
2025

4.8 Файл interface.h

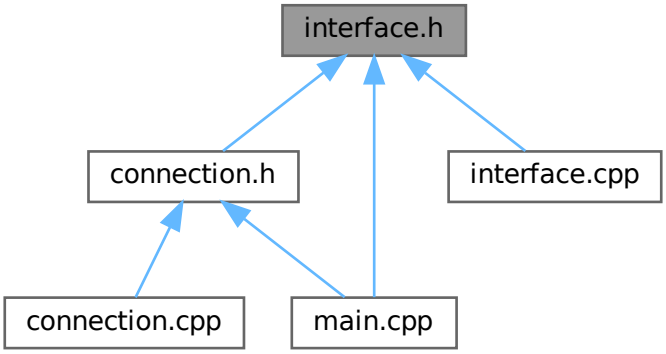
Заголовочный файл для пользовательского интерфейса

#include <boost/program_options.hpp>
#include <string>

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



Граф файлов, в которые включается этот файл:



Классы

- struct Configuration

  < Псевдоним для пространства имен program_options
- class UserInterface

  Класс для обработки аргументов командной строки

4.8.1 Подробное описание

Заголовочный файл для пользовательского интерфейса

Версия

1.0

Дата

2024

Заметки

Используется библиотека Boost.Program_options

4.9 interface.h

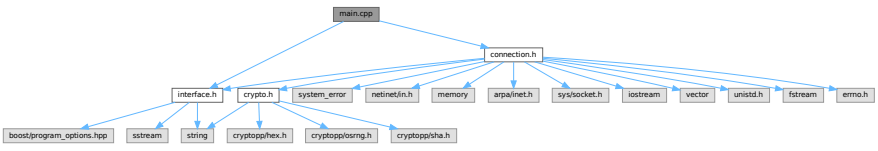
См. документацию.

```
00001
00009 #pragma once
00010
00011 #include <boost/program_options.hpp>
00012 #include <string>
00013 #include <sstream>
00014
00015 using namespace std;
00016
00017 namespace po = boost::program_options;
00018
00023 struct Configuration {
00024     string input_filename;
00025     string output_filename;
00026     string credentials_filename;
00027     int port_number;
00028     string server_address;
00029 };
00030
00035 class UserInterface {
00036 private:
00037     po::options_description options_description;
00038     po::variables_map variables_map;
00039     Configuration settings;
00040
00041 public:
00045     UserInterface();
00046
00053     bool parse_arguments(int arg_count, const char** arg_values);
00054
00059     string get_help_text();
00060
00065     Configuration get_configuration() {
00066         return settings;
00067     };
00068 };
```

4.10 Файл main.cpp

Главный файл приложения

#include "connection.h"
#include "interface.h"
Граф включаемых заголовочных файлов для main.cpp:



Функции

- int main (int arg_count, const char **arg_values)
  Главная функция приложения

4.10.1 Подробное описание

Главный файл приложения

Автор

Сочилов Н.М.

Версия

1.0

Дата

2025

Клиент для вычислений на сервере

4.10.2 Функции

4.10.2.1 main()

```
int main (
            int arg_count,
            const char ** arg_values )
```

Главная функция приложения

Аргументы

	in	arg_count	Количество аргументов командной строки
	in	arg_values	Массив аргументов командной строки

Возвращает

0 при успешном выполнении, 1 при ошибке