My Project

Создано системой Doxygen 1.9.8

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Connection
: Класс для управления сетевым соединением с сервером . . . . . ??

Params
: Структура для хранения параметров командной строки . . . . . ??

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

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

connection.cpp
: Реализация класса Connection для сетевого взаимодействия . . . . . ??

connection.h
: Заголовочный файл класса Connection для сетевого взаимодействия . . . . . ??

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

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

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

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

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

Глава 3

Классы

3.1 Класс Connection

Класс для управления сетевым соединением с сервером

#include <connection.h>

Открытые статические члены

- static int conn (const Params *p)
  Установка соединения и обмен данными с сервером

3.1.1 Подробное описание

Класс для управления сетевым соединением с сервером

Обеспечивает установку соединения, аутентификацию и передачу данных

3.1.2 Методы

3.1.2.1 conn()

int Connection::conn (
const Params * p) [static]

Установка соединения и обмен данными с сервером

Аргументы

	in	p	Указатель на параметры соединения

Возвращает

0 при успешном выполнении

Исключения

	system_error	при ошибках сетевого взаимодействия

Аргументы

	in	p	Указатель на параметры соединения

Возвращает

0 при успешном выполнении

Исключения

	system_error	при ошибках сетевого взаимодействия

Метод выполняет:

- Создание сокета и установку соединения
- Аутентификацию с использованием соли и хеширования пароля
- Чтение векторов из файла и передачу на сервер
- Получение результатов от сервера и запись в файл

< Семейство адресов IPv4

< Автоматический выбор порта

< Автоматический выбор адреса

< Семейство адресов IPv4

< Порт сервера

< IP-адрес сервера

< Чтение логина из файла

< Чтение пароля из файла

< Количество векторов для обработки

< Размер текущего вектора

< Вектор числовых значений

< Число в сетевом порядке байт

< Результат обработки от сервера

Объявления и описания членов классов находятся в файлах:

- connection.h
- connection.cpp

3.2 Структура Params

Структура для хранения параметров командной строки

#include <interface.h>

Открытые атрибуты

- string inFileName
  Имя входного файла с данными
- string inFileResult
  Имя файла для результатов
- string inFileData
  Имя файла с данными аутентификации
- int Port
  Порт сервера
- string Address
  Адрес сервера

3.2.1 Подробное описание

Структура для хранения параметров командной строки

Объявления и описания членов структуры находятся в файле:

- interface.h

3.3 Класс UserInterface

Класс для обработки параметров командной строки

#include <interface.h>

Открытые члены

- UserInterface ()
  Конструктор класса UserInterface.
- bool Parser (int argc, const char **argv)
  Парсинг аргументов командной строки
- string getDescription ()
  Получение описания параметров
- Params getParams ()
  Получение параметров

3.3.1 Подробное описание

Класс для обработки параметров командной строки

Использует boost::program_options для парсинга аргументов

3.3.2 Конструктор(ы)

3.3.2.1 UserInterface()

```
UserInterface::UserInterface ( )
```

Конструктор класса UserInterface.

Инициализирует опции командной строки < Опция помощи

< Входной файл

< Файл результатов

< Файл с данными аутентификации

< Порт сервера

< Адрес сервера

3.3.3 Методы

3.3.3.1 getDescription()

```
std::string UserInterface::getDescription ( )
```

Получение описания параметров

Возвращает

Строка с описанием поддерживаемых опций

3.3.3.2 getParams()

```
Params UserInterface::getParams ( )  [inline]
```

Получение параметров

Возвращает

Структура Params с распарсенными значениями

3.3.3.3 Parser()

```
bool UserInterface::Parser (
            int argc,
            const char ** argv )
```

Парсинг аргументов командной строки

Аргументы

	in	argc	Количество аргументов
	in	argv	Массив аргументов

Возвращает

true если парсинг успешен, false если требуется показать справку

Объявления и описания членов классов находятся в файлах:

- interface.h
- interface.cpp

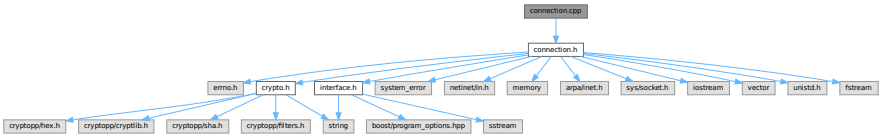
Глава 4

Файлы

4.1 Файл connection.cpp

Реализация класса Connection для сетевого взаимодействия

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



4.1.1 Подробное описание

Реализация класса Connection для сетевого взаимодействия

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

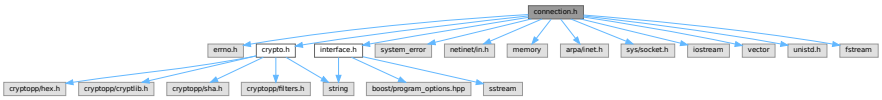
Ваша организация

4.2 Файл connection.h

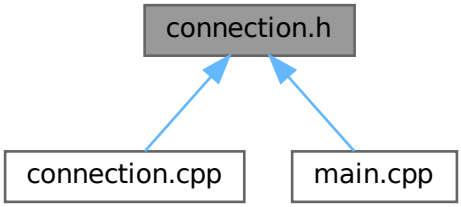
Заголовочный файл класса Connection для сетевого взаимодействия

```
#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 Connection

  Класс для управления сетевым соединением с сервером

Макросы

- #define BUFFER_SIZE 1024

  Размер буфера для сетевого обмена

4.2.1 Подробное описание

Заголовочный файл класса Connection для сетевого взаимодействия

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

Ваша организация

4.3 connection.h

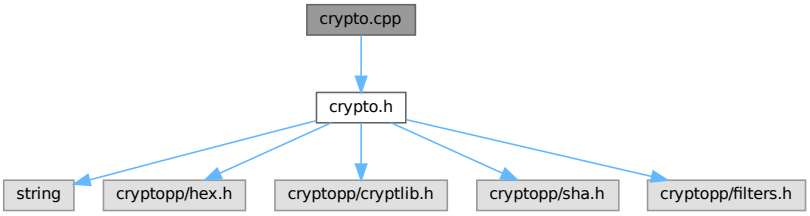
См. документацию.

```
00001
00010 #pragma once
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 using namespace std;
00024
00025 #define BUFFER_SIZE 1024
00026
00032 class Connection{
00033 private:
00034     static string salt;
00035
00036 public:
00043     static int conn(const Params* p);
00044 };
```

4.4 Файл crypto.cpp

Реализация криптографических функций

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



Функции

- string auth (string salt, string pass)

  Функция аутентификации с использованием хеширования

4.4.1 Подробное описание

Реализация криптографических функций

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

Ваша организация

4.4.2 Функции

4.4.2.1 auth()

```
string auth (
            string salt,
            string pass )
```

Функция аутентификации с использованием хеширования

Аргументы

	in	salt	Соль для хеширования
	in	pass	Пароль пользователя

Возвращает

Хеш-строка в hex-формате

Использует алгоритм SHA256 для создания хеша комбинации соли и пароля < Объект для вычисления SHA256

< Результирующий хеш

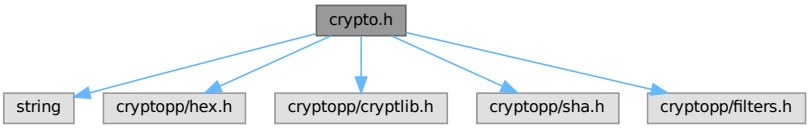
< Конкатенация соли и пароля

< Возврат хеша в hex-формате

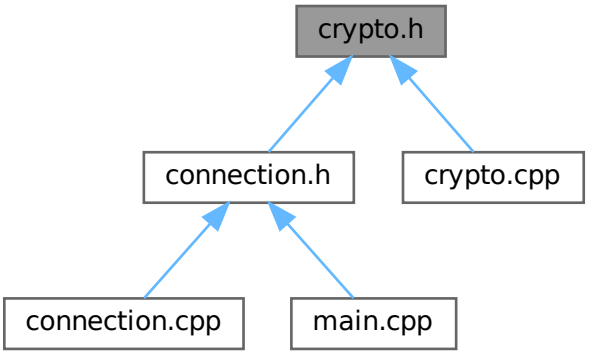
4.5 Файл crypto.h

Заголовочный файл для криптографических функций

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



Граф файлов, в которые включается этот файл:



Макросы

- #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1

Функции

- string auth (string salt, string pass)

  Функция аутентификации с использованием хеширования

4.5.1 Подробное описание

Заголовочный файл для криптографических функций

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

Ваша организация

4.5.2 Функции

4.5.2.1 auth()

```
string auth (
            string salt,
            string pass )
```

Функция аутентификации с использованием хеширования

Аргументы

	in	salt	Соль для хеширования
	in	pass	Пароль пользователя

Возвращает

Хеш-строка в hex-формате

Использует алгоритм SHA256 для создания хеша соли и пароля

Аргументы

	in	salt	Соль для хеширования
	in	pass	Пароль пользователя

Возвращает

Хеш-строка в hex-формате

Использует алгоритм SHA256 для создания хеша комбинации соли и пароля < Объект для вычисления SHA256

< Результирующий хеш

< Конкатенация соли и пароля

< Возврат хеша в hex-формате

4.6 crypto.h

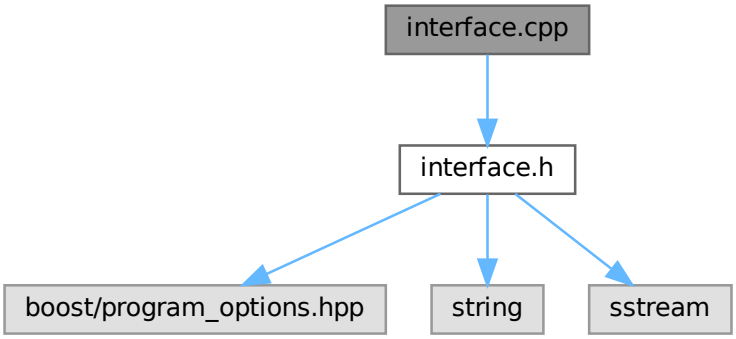
См. документацию.

```
00001
00010 #pragma once
00011 #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
00012 #include <string>
00013 #include <cryptopp/hex.h>
00014 #include <cryptopp/cryptlib.h>
00015 #include <cryptopp/sha.h>
00016 #include <cryptopp/filters.h>
00017 using namespace std;
00018 namespace CPP = CryptoPP;
00019
00027 string auth(string salt, string pass);
```

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

Кошкин Е.В.

Версия

1.0

Дата

2025

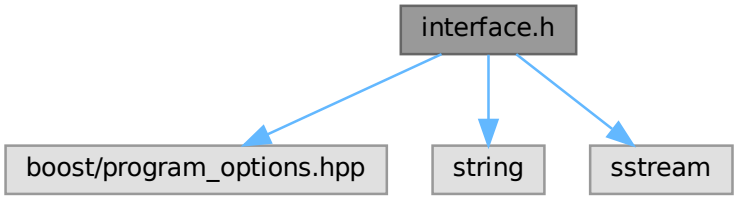
Авторство

Ваша организация

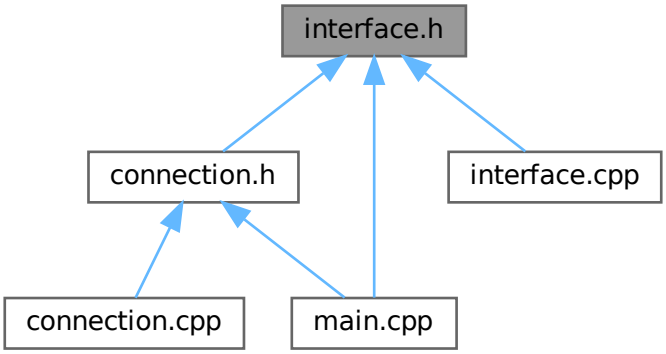
4.8 Файл interface.h

Заголовочный файл 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 Params

  Структура для хранения параметров командной строки
- class UserInterface

  Класс для обработки параметров командной строки

4.8.1 Подробное описание

Заголовочный файл пользовательского интерфейса

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

Ваша организация

4.9 interface.h

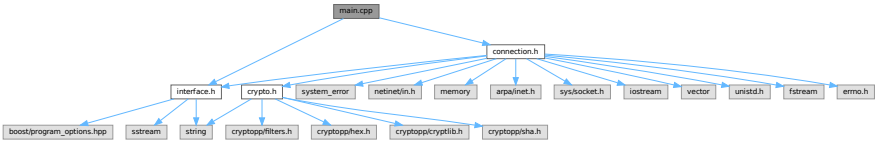
См. документацию.

```
00001
00010 #pragma once
00011 #include <boost/program_options.hpp>
00012 #include <string>
00013 #include <sstream>
00014 using namespace std;
00015 namespace po = boost::program_options;
00016
00021 struct Params {
00022     string inFileName;
00023     string inFileResult;
00024     string inFileData;
00025     int Port;
00026     string Address;
00027 };
00028
00034 class UserInterface {
00035 private:
00036     po::options_description desc;
00037     po::variables_map vm;
00038     Params params;
00039
00040 public:
00044     UserInterface();
00045
00052     bool Parser(int argc, const char** argv);
00053
00058     string getDescription();
00059
00064     Params getParams() {
00065         return params;
00066     };
00067 };
```

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 argc, const char **argv)

  Главная функция приложения

4.10.1 Подробное описание

Главный файл приложения

Автор

Кошкин Е.В.

Версия

1.0

Дата

2025

Авторство

Ваша организация

Точка входа в программу, обработка параметров и запуск соединения

4.10.2 Функции

4.10.2.1 main()

```
int main (
                int argc,
                const char ** argv )
```

Главная функция приложения

Аргументы

	in	argc	Количество аргументов командной строки
	in	argv	Массив аргументов командной строки

Возвращает

0 при успешном выполнении, 1 при ошибке

< Объект пользовательского интерфейса