kursovaya_final

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

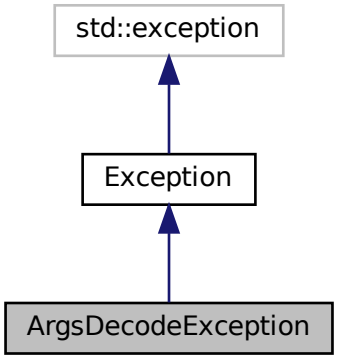
Классы

4.1 Класс ArgsDecodeException

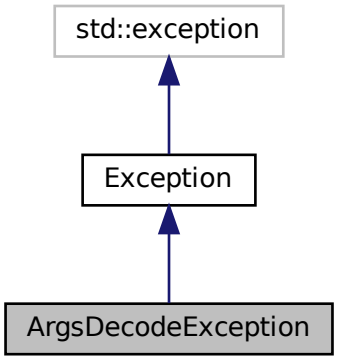
Класс для исключений при декодировании аргументов.

#include <exceptions.h>

Граф наследования:ArgsDecodeException:



Граф связей класса ArgsDecodeException:



Открытые члены

- ArgsDecodeException (const string &message, const string &func, const string &log_path, bool critical)

  Конструктор класса ArgsDecodeException.

Дополнительные унаследованные члены

4.1.1 Подробное описание

Класс для исключений при декодировании аргументов.

4.1.2 Конструктор(ы)

4.1.2.1 ArgsDecodeException()

```
ArgsDecodeException::ArgsDecodeException (
            const string & message,
            const string & func,
            const string & log_path,
            bool critical )
```

Конструктор класса ArgsDecodeException.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

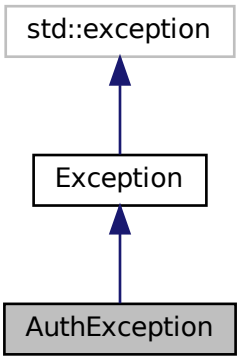
- exceptions.h
- exceptions.cpp

4.2 Класс AuthException

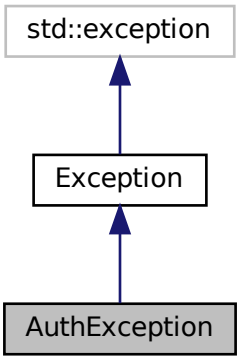
Класс для исключений аутентификации.

#include <exceptions.h>

Граф наследования:AuthException:



Граф связей класса AuthException:



Открытые члены

- AuthException (const string &message, const string &func, const string &log_path, bool critical)
  Конструктор класса AuthException.

Дополнительные унаследованные члены

4.2.1 Подробное описание

Класс для исключений аутентификации.

4.2.2 Конструктор(ы)

4.2.2.1 AuthException()

```
AuthException::AuthException (
                const string & message,
                const string & func,
                const string & log_path,
                bool critical )
```

Конструктор класса AuthException.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

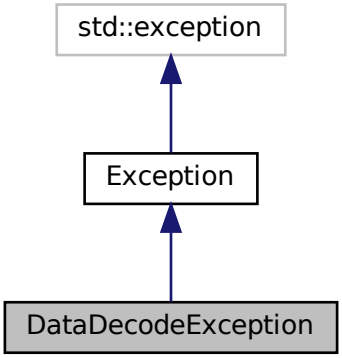
- exceptions.h
- exceptions.cpp

4.3 Класс DataDecodeException

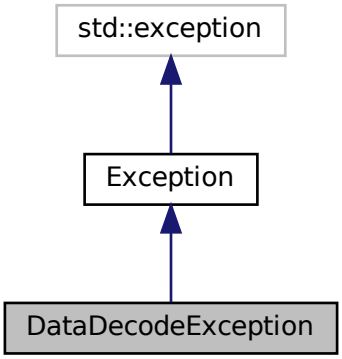
Класс для исключений при декодировании данных.

#include <exceptions.h>

Граф наследования:DataDecodeException:



Граф связей класса DataDecodeException:



Открытые члены

- DataDecodeException (const string &message, const string &func, const string &log_path, bool critical)

  Конструктор класса DataDecodeException.

Дополнительные унаследованные члены

4.3.1 Подробное описание

Класс для исключений при декодировании данных.

4.3.2 Конструктор(ы)

4.3.2.1 DataDecodeException()

```
DataDecodeException::DataDecodeException (
                const string & message,
                const string & func,
                const string & log_path,
                bool critical )
```

Конструктор класса DataDecodeException.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

- exceptions.h
- exceptions.cpp

4.4 Класс Decoder

Класс для управления интерфейсом программы.

#include <decoder.h>

Открытые члены

- Decoder ()
  Конструктор принимает параметры командной строки.
- void parse (int argc, char *argv[])
  Метод для парсинга аргументов командной строки.
- void showHelp () const
  Метод для получения справки.
- string & getDataBasePath ()
  Метод для получения пути к базе данных.
- string & getLogPath ()
  Метод для получения пути к файлу логов.
- string & getAddres ()
  Метод для получения адреса сервера.
- int & getPort ()
  Метод для получения порта сервера.

Закрытые данные

- string db_path

  Путь к файлу базы данных.
- string log_path

  Путь к файлу логов.
- int port

  Порт.
- string address

  Адрес.

4.4.1 Подробное описание

Класс для управления интерфейсом программы.

4.4.2 Конструктор(ы)

4.4.2.1 Decoder()

Decoder::Decoder ( )

Конструктор принимает параметры командной строки.

Исключения

ArgsDecodeException	Если аргументы командной строки некорректны.

4.4.3 Методы

4.4.3.1 getAddres()

string & Decoder::getAddres ( )

Метод для получения адреса сервера.

Возвращает

Ссылка на строку с адресом сервера.

4.4.3.2 getDataBasePath()

```
string & Decoder::getDataBasePath ( )
```

Метод для получения пути к базе данных.

Возвращает

Ссылка на строку с путем к базе данных.

4.4.3.3 getLogPath()

```
string & Decoder::getLogPath ( )
```

Метод для получения пути к файлу логов.

Возвращает

Ссылка на строку с путем к файлу логов.

4.4.3.4 getPort()

```
int & Decoder::getPort ( )
```

Метод для получения порта сервера.

Возвращает

Ссылка на целое число с номером порта сервера.

4.4.3.5 parse()

```
void Decoder::parse (
            int argc,
            char * argv[] )
```

Метод для парсинга аргументов командной строки.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Исключения

ArgsDecodeException	Если аргументы командной строки некорректны.

4.4.3.6 showHelp()

void Decoder::showHelp ( ) const

Метод для получения справки.

Возвращает

Справка по использованию.

Объявления и описания членов классов находятся в файлах:

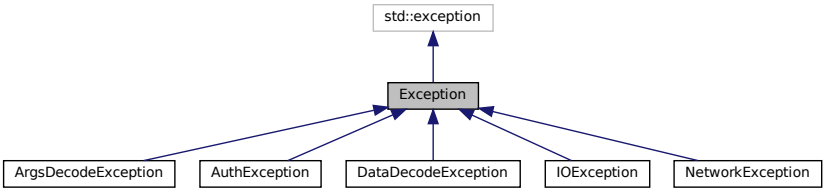
- decoder.h
- decoder.cpp

4.5 Класс Exception

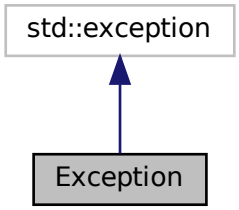
Базовый класс для исключений.

#include <exceptions.h>

Граф наследования:Exception:



Граф связей класса Exception:



Открытые члены

- Exception (const string &name, const string &message, const string &func, const string &log_path, bool critical)

  Конструктор класса Exception.
- const char * what () const noexcept override

  Метод для получения сообщения об ошибке.

Защищенные члены

- void logException () const

  Метод для логирования исключения.

Защищенные данные

- string name

  Имя исключения.
- string func

  Имя функции, в которой возникло исключение.
- string message

  Сообщение об ошибке.
- string log_path

  Путь к файлу журнала.
- bool critical

  Флаг критичности исключения.

4.5.1 Подробное описание

Базовый класс для исключений.

4.5.2 Конструктор(ы)

4.5.2.1 Exception()

```
Exception::Exception (
            const string & name,
            const string & message,
            const string & func,
            const string & log_path,
            bool critical )
```

Конструктор класса Exception.

Аргументы

name	Имя исключения.
message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

4.5.3 Методы

4.5.3.1 what()

const char * Exception::what ( ) const [override], [noexcept]

Метод для получения сообщения об ошибке.

Возвращает

Сообщение об ошибке.

Объявления и описания членов классов находятся в файлах:

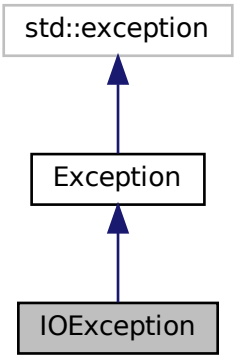
- exceptions.h
- exceptions.cpp

4.6 Класс IOException

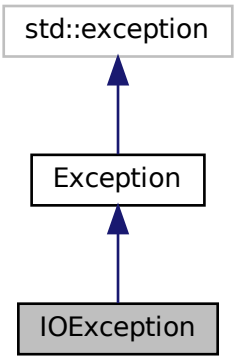
Класс для исключений ввода-вывода.

#include <exceptions.h>

Граф наследования:IOException:



Граф связей класса IOException:



Открытые члены

- IOException (const string &message, const string &func, const string &log_path, bool critical)
  Конструктор класса IOException.

Дополнительные унаследованные члены

4.6.1 Подробное описание

Класс для исключений ввода-вывода.

4.6.2 Конструктор(ы)

4.6.2.1 IOException()

```
IOException::IOException (
            const string & message,
            const string & func,
            const string & log_path,
            bool critical )
```

Конструктор класса IOException.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

- exceptions.h
- exceptions.cpp

4.7 Класс Network

Класс для управления сетевым подключением и взаимодействием.

#include <network.h>

Открытые члены

- Network (const string &address, uint16_t port, const map< string, string > &database, const string &log_path)

  Конструктор класса Network.
- string & getAddress ()

  Метод для получения адреса сервера.
- uint16_t & getPort ()

  Метод для получения порта сервера.
- map< string, string > & getDatabase ()

  Метод для получения базы данных пользователей.
- void start ()

  Метод для старта работы сервера.
- void stop ()

  Метод для закрытия основного соединения.
- void wait ()

  Метод для ожидания соединений.
- void auth ()

  Метод для аутентификации пользователя.
- void sum ()

  Метод для вычисления суммы значений в векторах.

Закрытые данные

- string address

  Адрес сервера.
- uint16_t port

  Порт сервера.
- map< string, string > database

  База данных пользователей.
- int socket

  Сокет сервера.
- int client

  Сокет клиента.
- string log_path

  Путь к файлу логов.

4.7.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

4.7.2 Конструктор(ы)

4.7.2.1 Network()

```
Network::Network (
            const string & address,
            uint16_t port,
            const map< string, string > & database,
            const string & log_path )
```

Конструктор класса Network.

Аргументы

address	Адрес сервера.
port	Порт сервера.
database	База данных пользователей.
log_path	Путь к файлу логов.

4.7.3 Методы

4.7.3.1 auth()

```
void Network::auth ( )
```

Метод для аутентификации пользователя.

Исключения

NetworkException	Если не удалось прочитать логин, найти логин в базе данных, отправить или получить данные от клиента.

4.7.3.2 getAddress()

```
string & Network::getAddress ( )
```

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

4.7.3.3 getDatabase()

map< string, string > & Network::getDatabase ( )

Метод для получения базы данных пользователей.

Возвращает

База данных пользователей.

4.7.3.4 getPort()

uint16_t & Network::getPort ( )

Метод для получения порта сервера.

Возвращает

Порт сервера.

4.7.3.5 sum()

void Network::sum ( )

Метод для вычисления суммы значений в векторах.

Исключения

NetworkException	Если не удалось прочитать данные от клиента или отправить результат.

4.7.3.6 wait()

void Network::wait ( )

Метод для ожидания соединений.

Исключения

NetworkException	Если не удалось создать, привязать или слушать сокет.

Объявления и описания членов классов находятся в файлах:

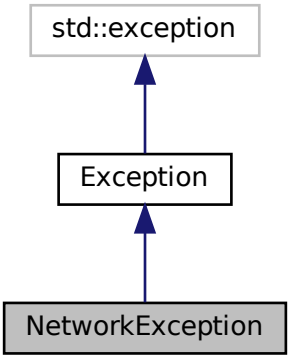
- network.h
- network.cpp

4.8 Класс NetworkException

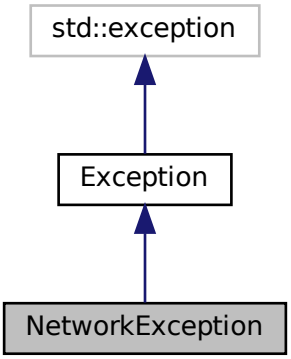
Класс для сетевых исключений.

#include <exceptions.h>

Граф наследования:NetworkException:



Граф связей класса NetworkException:



Открытые члены

- NetworkException (const string &message, const string &func, const string &log_path, bool critical)

  Конструктор класса NetworkException.

Дополнительные унаследованные члены

4.8.1 Подробное описание

Класс для сетевых исключений.

4.8.2 Конструктор(ы)

4.8.2.1 NetworkException()

```
NetworkException::NetworkException (
                const string & message,
                const string & func,
                const string & log_path,
                bool critical )
```

Конструктор класса NetworkException.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

- exceptions.h
- exceptions.cpp

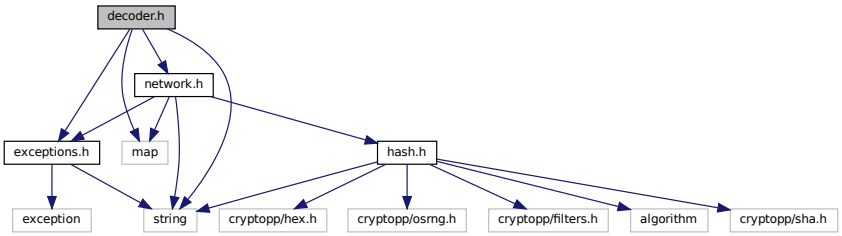
Глава 5

Файлы

5.1 Файл decoder.h

Определение класса интерфейса.

#include <map>
#include <string>
#include "exceptions.h"
#include "network.h"
Граф включаемых заголовочных файлов для decoder.h:



Классы

- class Decoder

  Класс для управления интерфейсом программы.

5.1.1 Подробное описание

Определение класса интерфейса.

Этот файл содержит определения классов для обработки параметров командной строки, чтения базы данных паролей и запуска сервера.

Дата

13.12.2024

Версия

1.0

Автор

Мамелин Д. А.

5.2 decoder.h

См. документацию.

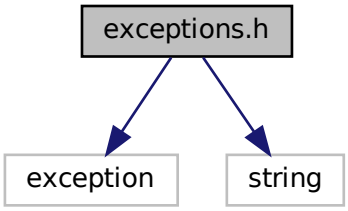
```
1
10 #pragma once
11
12 #include <map>
13 #include <string>
14 #include "exceptions.h"
15 #include "network.h"
16
17 using namespace std;
18
22 class Decoder
23 {
24 public:
25
30     Decoder();
31
38     void parse(int argc, char *argv[]);
39
44     void showHelp() const;
45
50     string &getDataBasePath();
51
56     string &getLogPath();
57
62     string &getAddres();
63
68     int &getPort();
69
70 private:
71     string db_path;
72     string log_path;
73     int port;
74     string address;
75 };
76
77
```

5.3 Файл exceptions.h

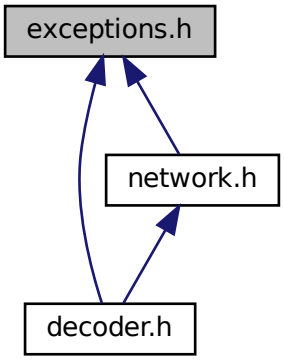
Определение классов исключений.

#include <exception>
#include <string>

Граф включаемых заголовочных файлов для exceptions.h:



Граф файлов, в которые включается этот файл:



Классы

- class Exception

  Базовый класс для исключений.
- class IOException

  Класс для исключений ввода-вывода.
- class ArgsDecodeException

  Класс для исключений при декодировании аргументов.
- class DataDecodeException

  Класс для исключений при декодировании данных.
- class AuthException

  Класс для исключений аутентификации.
- class NetworkException

  Класс для сетевых исключений.

5.3.1 Подробное описание

Определение классов исключений.

Этот файл содержит определения классов исключений для обработки различных ошибок в программе.

Дата

13.12.2024

Версия

1.0

Автор

Мамелин Д. А.

5.4 exceptions.h

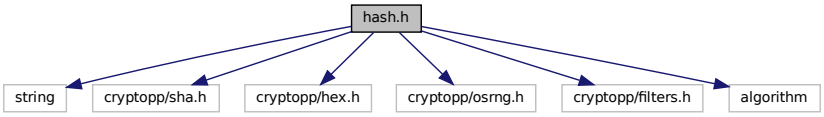
См. документацию.

```
1
10 #pragma once
11
12 #include <exception>
13 #include <string>
14
15 using namespace std;
16
20 class Exception : public std::exception {
21 public:
30     Exception(const string &name, const string &message, const string &func, const string &log_path, bool critical);
31
36     const char *what() const noexcept override;
37
38 protected:
39
43     void logException() const;
44
45     string name;
46     string func;
47     mutable string message;
48     string log_path;
49     bool critical;
50 };
51
55 class IOException : public Exception {
56 public:
64     IOException(const string &message, const string &func, const string &log_path, bool critical);
65 };
66
70 class ArgsDecodeException : public Exception {
71 public:
79     ArgsDecodeException(const string &message, const string &func, const string &log_path, bool critical);
80 };
81
85 class DataDecodeException : public Exception {
86 public:
94     DataDecodeException(const string &message, const string &func, const string &log_path, bool critical);
95 };
96
100 class AuthException : public Exception {
101 public:
109     AuthException(const string &message, const string &func, const string &log_path, bool critical);
110 };
111
115 class NetworkException : public Exception {
116 public:
124     NetworkException(const string &message, const string &func, const string &log_path, bool critical);
125 };
126
```

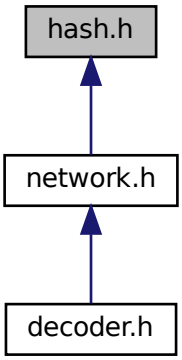
5.5 Файл hash.h

Определения вспомогательных функций для криптографических операций.

#include <string>
#include <cryptopp/sha.h>
#include <cryptopp/hex.h>
#include <cryptopp/osrng.h>
#include <cryptopp/filters.h>
#include <algorithm>
Граф включаемых заголовочных файлов для hash.h:



Граф файлов, в которые включается этот файл:



Функции

- string getSalt ()

  Функция для генерации соли.
- string getHash (const string &data)

  Функция для вычисления хеша.

5.5.1 Подробное описание

Определения вспомогательных функций для криптографических операций.

Этот файл содержит определения функций для генерации соли и вычисления хеша.

Дата

13.12.2024

Версия

1.0

Автор

Мамелин Д. А.

5.5.2 Функции

5.5.2.1 getHash()

```
string getHash (
            const string & data )
```

Функция для вычисления хеша.

Аргументы

data	Данные для хеширования.

Возвращает

Хеш в виде строки.

5.5.2.2 getSalt()

```
string getSalt ( )
```

Функция для генерации соли.

Возвращает

Соль в виде строки.

5.6 hash.h

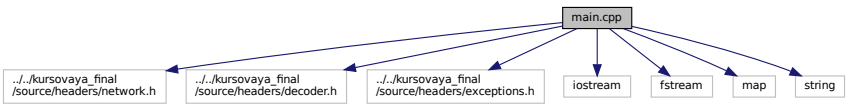
См. документацию.

```
1
10 #pragma once
11
12 #include <string>
13
14 #include <cryptopp/sha.h>
15 #include <cryptopp/hex.h>
16 #include <cryptopp/osrng.h>
17 #include <cryptopp/filters.h>
18 #include <algorithm>
19
20 using namespace std;
21
26 string getSalt();
27
33 string getHash(const string &data);
```

5.7 Файл main.cpp

Главный файл программы.

#include "../../kursovaya_final/source/headers/network.h"
#include "../../kursovaya_final/source/headers/decoder.h"
#include "../../kursovaya_final/source/headers/exceptions.h"
#include <iostream>
#include <fstream>
#include <map>
#include <string>
Граф включаемых заголовочных файлов для main.cpp:



Функции

- map< string, string > getDataBase (string &path)

  Читает файл базы данных и заполняет карту парами "имя пользователя - пароль".
- void loop (Network *network)

  Основной цикл для обработки сетевых операций. Эта функция запускает сетевой сервис и входит в бесконечный цикл, в котором она ожидает входящих соединений, аутентифицирует пользователей и выполняет операции. Любые сетевые исключения обрабатываются и регистрируются. Если возникает неожиданное исключение, сетевой сервис останавливается.
- int main (int argc, char *argv[])

  Главная точка входа в приложение.

Переменные

- string LOG_PATH

5.7.1 Подробное описание

Главный файл программы.

Этот файл содержит функцию main, которая запускает сервер.

Дата

13.12.2024

Версия

1.0

Автор

Мамелин Д. А.

5.7.2 Функции

5.7.2.1 getDataBase()

```
map< string, string > getDataBase (
                string & path )
```

Читает файл базы данных и заполняет карту парами "имя пользователя - пароль".

Аргументы

path	Ссылка на строку, представляющую путь к файлу базы данных.

Возвращает

Карта, содержащая пары "имя пользователя - пароль"

Исключения

IOException	Если файл не может быть открыт.
DataDecodeException	Если строка в файле не соответствует ожидаемому формату или если произошла ошибка при чтении из файла.

5.7.2.2 loop()

```
void loop (
            Network * network )
```

Основной цикл для обработки сетевых операций. Эта функция запускает сетевой сервис и входит в бесконечный цикл, в котором она ожидает входящих соединений, аутентифицирует пользователей и выполняет операции. Любые сетевые исключения обрабатываются и регистрируются. Если возникает неожиданное исключение, сетевой сервис останавливается.

Аргументы

network	Указатель на объект Network, который обрабатывает сетевые операции.

5.7.2.3 main()

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

Главная точка входа в приложение.

Эта функция инициализирует объект Decoder для парсинга аргументов командной строки, получает путь к журналу и базу данных из указанной конфигурации, инициализирует объект Network с этими настройками и входит в цикл для обработки сетевых операций. Также включает обработку ошибок для сетевых исключений.

Аргументы

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

Возвращает

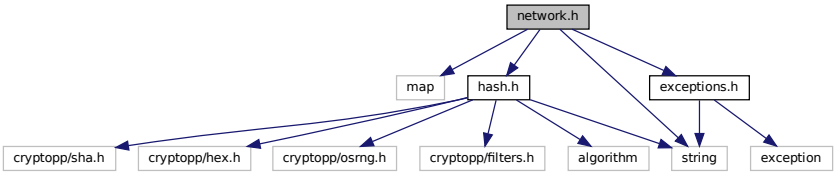
Целое число, представляющее статус завершения программы (0 для успешного завершения).

5.8 Файл network.h

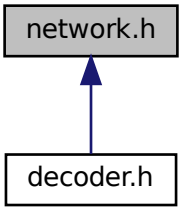
Определения классов для управления сетевым взаимодействием.

```
#include <map>
#include <string>
#include "hash.h"
```

#include "exceptions.h"
Граф включаемых заголовочных файлов для network.h:



Граф файлов, в которые включается этот файл:



Классы

- class Network

  Класс для управления сетевым подключением и взаимодействием.

5.8.1 Подробное описание

Определения классов для управления сетевым взаимодействием.

Этот файл содержит определения классов для управления сетевыми подключениями и передачей данных.

Дата

13.12.2024

Версия

1.0

Автор

Мамелин Д. А.

5.9 network.h

См. документацию.

```
1
10 #pragma once
11
12 #include <map>
13 #include <string>
14 #include "hash.h"
15 #include "exceptions.h"
16
17 using namespace std;
18
23 class Network
24 {
25 public:
33     Network(
34         const string &address,
35         uint16_t port,
36         const map<string, string> &database,
37         const string &log_path);
38
43     string &getAddress();
44
49     uint16_t &getPort();
50
55     map<string, string> &getDatabase();
56
60     void start();
61
65     void stop();
66
71     void wait();
72
77     void auth();
78
83     void sum();
84
85 private:
86     string address;
87     uint16_t port;
88     map<string, string> database;
89     int socket;
90     int client;
91     string log_path;
92 };
93
94
```

Предметный указатель