server

Создано системой Doxygen 1.9.4

1 Алфавитный указатель пространств имен		1
1.1 Пространства имен	 	1
2 Иерархический список классов		3
2.1 Иерархия классов	 	3
3 Алфавитный указатель классов		5
3.1 Классы	 	5
4 Список файлов		7
$4.1~\Phi$ айлы	 	7
5 Пространства имен		9
5.1 Пространство имен crutils	 	9
5.1.1 Подробное описание	 	9
5.1.2 Функции	 	9
5.1.2.1 calculateHash()		9
5.1.2.2 generateSalt()		10
6 Классы		11
6.1 Kлаcc ArgsDecodeException	 	11
6.1.1 Подробное описание		12
6.1.2 Конструктор(ы)		12
6.1.2.1 ArgsDecodeException()		12
6.2 Класс AuthException		13
6.2.1 Подробное описание		14
6.2.2 Конструктор(ы)		14
6.2.2.1 AuthException()		14
6.3 Kласс DataDecodeException		14
6.3.1 Подробное описание	 	15
6.3.2 Конструктор(ы)	 	16
6.3.2.1 DataDecodeException()		16
6.4 Kласс Exception		16
6.4.1 Подробное описание		17
6.4.2 Конструктор(ы)		17
6.4.2.1 Exception()		18
6.4.3 Методы		18
6.4.3.1 what()	 	18
6.5 Kласс Interface		18
6.5.1 Подробное описание		20
6.5.2 Конструктор(ы)		20
6.5.2.1 Interface()		20
6.5.3 Методы		20
6.5.3.1 getDbPath()		20
6.5.3.2 getLogPath()		20
the control of the co		

$6.5.3.3 \text{ getNetwork}() \dots \dots$	21
$6.5.3.4 \text{ parseArgs}() \dots \dots$	21
6.5.3.5 readDB()	21
6.5.3.6 run()	21
6.6 Класс IOException	22
6.6.1 Подробное описание	23
$6.6.2 \; { m Kohctpyktop}({ m bi}) \; \dots $	23
6.6.2.1 IOException()	23
6.7 Kласс Network	24
6.7.1 Подробное описание	25
6.7.2 Конструктор(ы)	25
6.7.2.1 Network()	25
6.7.3 Методы	25
6.7.3.1 auth()	25
6.7.3.2 calc()	25
6.7.3.3 getAddress()	27
6.7.3.4 getDatabase()	27
6.7.3.5 getPort()	27
6.7.3.6 wait()	27
6.8 Kлаcc NetworkException	28
6.8.1 Подробное описание	29
6.8.2 Конструктор(ы)	29
6.8.2.1 NetworkException()	29
7 Файлы	31
7.1 Файл crutils.h	31
7.1.1 Подробное описание	32
7.2 crutils.h	33
7.3 Файл exceptions.h	33
7.3.1 Подробное описание	34
7.4 exceptions.h	35
7.5 Файл interface.h	35
7.5.1 Подробное описание	36
7.6 interface.h	37
7.7 Файл main.cpp	37
7.7.1 Подробное описание	38
7.7.2 Функции	39
7.7.2.1 main()	39
7.8 Файл network.h	39
7.8.1 Подробное описание	40
7.9 network.h	40
H	40
Предметный указатель	43

Алфавитный указатель пространств имен

1.1 Пространства имен

Полный спис	ок докуме	нтированных	пространств	имен.
-------------	-----------	-------------	-------------	-------

crutils

A 1			
Алфавитный	указатель	пространств	имен

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

std::exception	
Exception	6
ArgsDecodeException	1
AuthException	3
DataDecodeException	4
IOException	2
NetworkException	8
Interface	8
Network	14

перархический список классов	Иерархический	список	классов
------------------------------	---------------	--------	---------

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

ArgsDecodeException	
Класс для исключений при декодировании аргументов	11
AuthException	
Класс для исключений аутентификации	13
DataDecodeException	
Класс для исключений при декодировании данных	14
Exception	
Базовый класс для исключений	16
Interface	
Класс для управления интерфейсом программы	18
IOException	
Класс для исключений ввода-вывода	22
Network	
Класс для управления сетевым подключением и взаимодействием	24
NetworkException	
Класс для сетевых исключений	28

	Алфавитный	указатель	классо
--	------------	-----------	--------

Список файлов

4.1 Файлы

Полный список документированных файлов.

crutils.h		
Определения вспомогательных функций для криптографических операций	 	. 31
exceptions.h		
Определение классов исключений	 	. 33
interface.h		
Определение класса интерфейса	 	. 35
main.cpp		
Главный файл программы	 	. 37
network.h		
Определения классов для управления сетевым взаимодействием	 	. 39

8 Список файлов

Пространства имен

5.1 Пространство имен crutils

Пространство имен для криптографических утилит.

Функции

```
    std::string generateSalt ()
    Функция для генерации соли.
    std::string calculateHash (const std::string &data)
```

5.1.1 Подробное описание

Пространство имен для криптографических утилит.

Функция для вычисления хеша.

5.1.2 Функции

5.1.2.1 calculateHash()

```
std::string\ crutils::calculateHash\ ( const\ std::string\ \&\ data\ )
```

Функция для вычисления хеша.

Аргументы

data Данные для хеширования.

10 Пространства имен

Возвращает

Хеш в виде строки.

5.1.2.2 generateSalt()

 $std::string\ crutils::generateSalt\ (\)$

Функция для генерации соли.

Возвращает

Соль в виде строки.

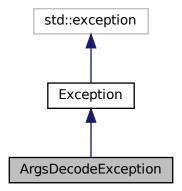
Классы

6.1 Kласс ArgsDecodeException

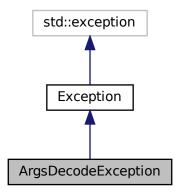
Класс для исключений при декодировании аргументов.

#include <exceptions.h>

Граф наследования:ArgsDecodeException:



Граф связей класса ArgsDecodeException:



Открытые члены

• ArgsDecodeException (const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)

Конструктор класса ArgsDecodeException.

6.1.1 Подробное описание

Класс для исключений при декодировании аргументов.

6.1.2 Конструктор(ы)

6.1.2.1 ArgsDecodeException()

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:

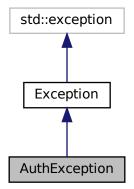
- \bullet exceptions.h
- $\bullet \ \ {\rm exceptions.cpp}$

6.2 Класс AuthException

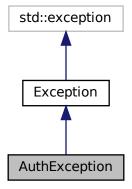
Класс для исключений аутентификации.

#include <exceptions.h>

Граф наследования: Auth
Exception:



Граф связей класса Auth Exception:



Открытые члены

• AuthException (const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)

Конструктор класса AuthException.

6.2.1 Подробное описание

Класс для исключений аутентификации.

6.2.2 Конструктор(ы)

6.2.2.1 AuthException()

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:

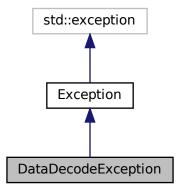
- exceptions.h
- exceptions.cpp

6.3 Класс DataDecodeException

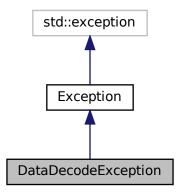
Класс для исключений при декодировании данных.

```
\#include <exceptions.h>
```

Граф наследования: DataDecode
Exception:



Граф связей класса DataDecodeException:



Открытые члены

• DataDecodeException (const std::string &message, const std::string &func, const std::string &log path, bool critical=false)

 ${\bf Kohctpyкtop}\ {\bf knacca}\ {\bf DataDecodeException}.$

6.3.1 Подробное описание

Класс для исключений при декодировании данных.

6.3.2 Конструктор(ы)

6.3.2.1 DataDecodeException()

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:

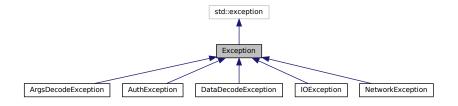
- exceptions.h
- exceptions.cpp

6.4 Класс Exception

Базовый класс для исключений.

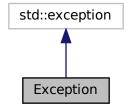
#include <exceptions.h>

Граф наследования:Exception:



6.4 Класс Exception 17

Граф связей класса Exception:



Открытые члены

• Exception (const std::string &name, const std::string &message, const std::string &func, const std::string &log path, bool critical=false)

Конструктор класса Exception.

• const char * what () const no except override

Метод для получения сообщения об ошибке.

• void logException () const

Метод для логирования исключения.

Закрытые данные

• std::string name

Имя исключения.

• std::string func

Имя функции, в которой возникло исключение.

• std::string log_path

Путь к файлу журнала.

· bool critical

 Φ лаг критичности исключения.

- std::string message

Сообщение об ошибке.

6.4.1 Подробное описание

Базовый класс для исключений.

6.4.2 Конструктор(ы)

6.4.2.1 Exception()

```
Exception::Exception (

const std::string & name,

const std::string & message,

const std::string & func,

const std::string & log_path,

bool critical = false)
```

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.	

6.4.3 Методы

6.4.3.1 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

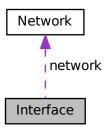
6.5 Класс Interface

Класс для управления интерфейсом программы.

#include <interface.h>

6.5 Класс Interface 19

Граф связей класса Interface:



Открытые члены

• Interface (int argc, char *argv[])

Конструктор принимает параметры командной строки.

• void parseArgs (int argc, char *argv[])

Метод для парсинга аргументов командной строки.

• void readDB ()

Метод для чтения базы данных паролей и записи в атрибут тар.

• void run ()

Метод для запуска сервера.

• void showHelp () const

Метод для вывода справки.

Network * getNetwork ()

Метод для получения объекта Network.

• std::string & getDbPath ()

Метод для получения пути к файлу базы данных.

• std::string & getLogPath ()

Метод для получения пути к файлу логов.

Закрытые данные

• Network * network

Указатель на объект класса Network.

• std::string db path

Путь к файлу базы данных.

• std::string \log_{path}

Путь к файлу логов.

• int port

Порт.

 \bullet std::string address

Адрес.

• std::map< std::string, std::string > database

Словарь для хранения базы данных паролей.

6.5.1 Подробное описание

Класс для управления интерфейсом программы.

6.5.2 Конструктор(ы)

```
6.5.2.1 Interface()
```

```
\begin{split} & \text{Interface::Interface (} \\ & \quad \text{int argc,} \\ & \quad \text{char} * \text{argv[] )} \end{split}
```

Конструктор принимает параметры командной строки.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Исключения

ArgsDecodeException	Если аргументы командной строки некорректны.
---------------------	--

6.5.3 Методы

```
6.5.3.1 getDbPath()
```

```
\operatorname{std}::\operatorname{string}\ \&\ \operatorname{Interface}::\operatorname{getDbPath}\ (\ )
```

Метод для получения пути к файлу базы данных.

Возвращает

Путь к файлу базы данных.

```
6.5.3.2 getLogPath()
```

```
std::string & Interface::getLogPath ( )
```

Метод для получения пути к файлу логов.

Возвращает

Путь к файлу логов.

6.5 Класс Interface 21

6.5.3.3 getNetwork()

```
{\color{red} Network} * Interface::getNetwork ()
```

Метод для получения объекта Network.

Возвращает

Указатель на объект класса Network.

6.5.3.4 parseArgs()

Метод для парсинга аргументов командной строки.

Аргументы

argc	Количество аргументов командной строки	
argv	Аргументы командной строки.	

Исключения

6.5.3.5 readDB()

```
void Interface::readDB ( ) \,
```

Метод для чтения базы данных паролей и записи в атрибут тар.

Исключения

IOException	Если не удалось открыть или прочитать файл базы данных.
DataDecodeException	Если строки в базе данных имеют некорректный формат или
	произошла ошибка при чтении файла.

6.5.3.6 run()

void ${\it Interface::run}$ ()

Метод для запуска сервера.

Исключения

NetworkException | Если возникли ошибки при запуске сервера.

Объявления и описания членов классов находятся в файлах:

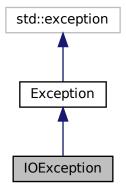
- \bullet interface.h
- interface.cpp

6.6 Класс IOException

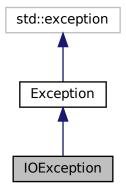
Класс для исключений ввода-вывода.

#include <exceptions.h>

Граф наследования: IOException:



Граф связей класса IOException:



Открытые члены

• IOException (const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)

Конструктор класса IOException.

6.6.1 Подробное описание

Класс для исключений ввода-вывода.

6.6.2 Конструктор(ы)

6.6.2.1 IOException()

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.	

Создано системой Doxygen

Объявления и описания членов классов находятся в файлах:

- exceptions.h
- \bullet exceptions.cpp

6.7 Kласс Network

Класс для управления сетевым подключением и взаимодействием.

```
#include <network.h>
```

Открытые члены

• Network (const std::string &address, uint16_t port, const std::map< std::string, std::string > &database, const std::string &log path)

Конструктор класса Network.

• std::string & getAddress ()

Метод для получения адреса сервера.

• uint16 t & getPort ()

Метод для получения порта сервера.

• std::map< std::string, std::string > & getDatabase ()

Метод для получения базы данных пользователей.

• void wait ()

Метод для ожидания соединений.

• void auth ()

Метод для аутентификации пользователя.

• void calc ()

Метод для вычисления суммы значений в векторах.

• void close ()

Метод для закрытия клиентского соединения.

• void quit ()

Метод для закрытия основного соединения.

Закрытые данные

• std::string address

Адрес сервера.

• uint16_t port

Порт сервера.

- std::map< std::string, std::string > database

База данных пользователей.

• int socket

Сокет сервера.

• int client socket

Сокет клиента.

• std::string log path

Путь к файлу логов.

6.7 Класс Network 25

6.7.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

6.7.2 Конструктор(ы)

6.7.2.1 Network()

```
Network::Network (  const\ std::string\ \&\ address, \\ uint16\_t\ port, \\ const\ std::map<\ std::string,\ std::string>\ \&\ database, \\ const\ std::string\ \&\ log\_path\ )
```

Конструктор класса Network.

Аргументы

address	Адрес сервера.
port	Порт сервера.
database	База данных пользователей.
\log_{path}	Путь к файлу логов.

6.7.3 Методы

6.7.3.1 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.

6.7.3.2 calc()

void Network::calc () $\,$

26 Классы Метод для вычисления суммы значений в векторах.

6.7 Класс Network 27

Исключения

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

```
6.7.3.3 getAddress()
std::string & Network::getAddress ( )
Метод для получения адреса сервера.
Возвращает
     Адрес сервера.
6.7.3.4 getDatabase()
std::map < std::string, \ std::string > \& \ Network::getDatabase \ (\ )
Метод для получения базы данных пользователей.
Возвращает
     База данных пользователей.
6.7.3.5 getPort()
uint16_t & Network::getPort ( )
Метод для получения порта сервера.
Возвращает
     Порт сервера.
6.7.3.6 wait()
void Network::wait ( )
```

Метод для ожидания соединений.

Исключения

Объявления и описания членов классов находятся в файлах:

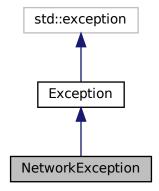
- network.h
- · network.cpp

6.8 Класс NetworkException

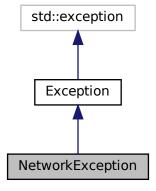
Класс для сетевых исключений.

#include <exceptions.h>

Граф наследования:NetworkException:



Граф связей класса NetworkException:



Открытые члены

• NetworkException (const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)

Конструктор класса NetworkException.

6.8.1 Подробное описание

Класс для сетевых исключений.

6.8.2 Конструктор(ы)

6.8.2.1 NetworkException()

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

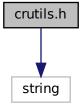
Файлы

7.1 Файл crutils.h

Определения вспомогательных функций для криптографических операций.

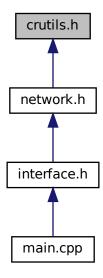
#include <string>

Граф включаемых заголовочных файлов для crutils.h:



32

Граф файлов, в которые включается этот файл:



Пространства имен

• namespace crutils

Пространство имен для криптографических утилит.

Функции

- std::string crutils::generateSalt ()
 - Функция для генерации соли.
- std::string crutils::calculateHash (const std::string &data)

Функция для вычисления хеша.

7.1.1 Подробное описание

Определения вспомогательных функций для криптографических операций.

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

Дата

23.11.2024

Версия

1.0 @authorsa Мамелин Д. А.

7.2 crutils.h 33

7.2 crutils.h

```
Cм. документацию.

1 #ifndef CRUTILS_H

2 #define CRUTILS_H

3

4 #include <string>

5

18 namespace crutils {

23  std::string generateSalt();

24

30  std::string calculateHash(const std::string &data);

31 }

32

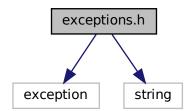
33 #endif // CRUTILS_H
```

7.3 Файл exceptions.h

Определение классов исключений.

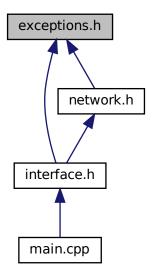
```
#include <exception>
#include <string>
```

Граф включаемых заголовочных файлов для exceptions.h:



34 Файлы

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.

7.3.1 Подробное описание

Определение классов исключений.

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

Дата

23.11.2024

Версия

1.0 @authorsa Мамелин Д. А.

7.4 exceptions.h 35

7.4 exceptions.h

```
См. документацию.
1 #ifndef EXCEPTIONS H
2 #define EXCEPTIONS H
4~\# include < \! exception \! >
5 #include <string>
19 class Exception : public std::exception {
20 public:
      Exception(const std::string &name,
30
               const std::string & message,
31
               const std::string &func,
               const std::string & log_path,
32
33
               bool critical = false);
34
      const char* what() const noexcept override;
40
44
      void logException() const;
45
46 private:
^{47}
      std::string name;
48
      std::string func;
^{49}
      std::string log_path;
50
      bool critical;
      {\it mutable \ std::string \ message;}
51
52 };
53
57 class IOException : public Exception {
58 public:
      IOException(const std::string &message,
                const std::string &func,
const std::string &log_path,
67
68
69
                bool critical = false);
70 };
75 class ArgsDecodeException: public Exception {
76 public:
      ArgsDecodeException(const std::string &message,
const std::string &func,
const std::string &log_path,
84
85
86
                       bool critical = false);
88 };
89
93 class DataDecodeException : public Exception {
94 public:
102
       DataDecodeException(const std::string &message,
103
                        const std::string &func,
104
                        const std::string & log path,
105
                        bool critical = false);
106 };
107
111 class AuthException : public Exception {
112 public:
120
       AuthException(const std::string &message,
121
                   const std::string &func,
122
                   const\ std::string\ \&\log\_path,
123
                   bool critical = false);
124 };
125
129 class NetworkException : public Exception {
130 public:
        NetworkException(const std::string &message,
138
139
                      const std::string &func,
                     const std::string &log_path,
bool critical = false);
140
141
142 };
144 #endif // EXCEPTIONS_H
```

7.5 Файл interface.h

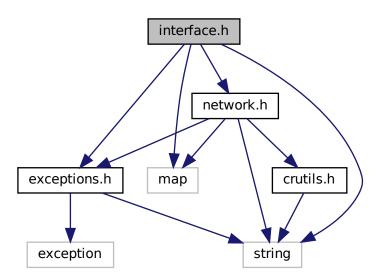
Определение класса интерфейса.

```
#include <map>
#include <string>
```

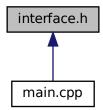
36

```
#include "exceptions.h"
#include "network.h"
```

Граф включаемых заголовочных файлов для interface.h:



Граф файлов, в которые включается этот файл:



Классы

• class Interface

Класс для управления интерфейсом программы.

7.5.1 Подробное описание

Определение класса интерфейса.

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

7.6 interface.h

Дата

23.11.2024

Версия

1.0 @authorsa Мамелин Д. А.

7.6 interface.h

```
Cм. документацию.
1 #ifndef INTERFACE_H
2 #define INTERFACE_H
4 \text{ \#include } < \text{map} >
5 #include <string>
6 #include "exceptions.h"
7 #include "network.h"
21 \ {
m class \ Interface}
22 {
23 public:
30
       Interface(int argc, char *argv[]);
31
       void parseArgs(int argc, char *argv[]);
39
       void readDB();
45
\frac{46}{51}
       void run();
52
56
       void showHelp() const;
62
63
       Network *getNetwork();
       std::string \ \& getDbPath();
68
69
       std::string &getLogPath();
75
76 private:
       Network *network;
std::string db_path;
std::string log_path;
77
78
79
       int port;
       std::string address;
       std::map<std::string, std::string> database;
83 };
85 #endif // INTERFACE_H
```

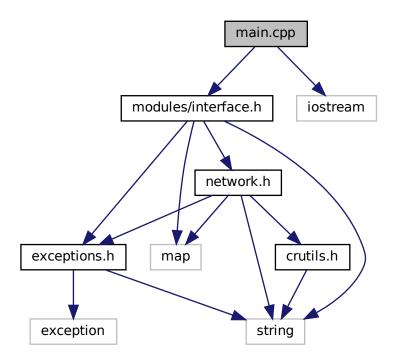
7.7 Файл main.cpp

Главный файл программы.

```
#include "modules/interface.h" #include <iostream>
```

38 Файлы

Граф включаемых заголовочных файлов для main.cpp:



Функции

int main (int argc, char *argv[])
 Главная функция программы.

7.7.1 Подробное описание

Главный файл программы.

Этот файл содержит функцию main, которая инициализирует интерфейс и запускает сервер.

Дата

23.11.2024

Версия

1.0

Автор

Мамелин Д. А.

7.8 Файл network.h 39

7.7.2 Функции

```
7.7.2.1 \operatorname{main}() int \operatorname{main}() \operatorname{int argc}, \operatorname{char}*\operatorname{argv}[])
```

Главная функция программы.

Инициализирует объект Interface и запускает его. Обрабатывает все исключения, возникающие во время выполнения программы.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Возвращает

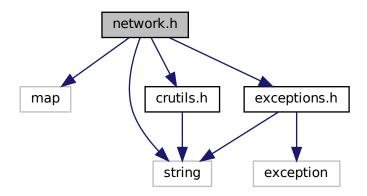
Код завершения программы. 0 - успешное завершение, 1 - ошибка.

7.8 Файл network.h

Определения классов для управления сетевым взаимодействием.

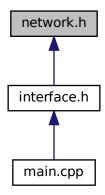
```
#include <map>
#include <string>
#include "crutils.h"
#include "exceptions.h"
```

Граф включаемых заголовочных файлов для network.h:



40 Файлы

Граф файлов, в которые включается этот файл:



Классы

• class Network

Класс для управления сетевым подключением и взаимодействием.

7.8.1 Подробное описание

Определения классов для управления сетевым взаимодействием.

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

Дата

23.11.2024

Версия

1.0 @authorsa Мамелин Д. А.

7.9 network.h

```
Cм. документацию.

1 #ifndef NETWORK_H

2 #define NETWORK_H

3 #include <map>
5 #include <string>
6 #include "crutils.h"

7 #include "exceptions.h"

8

21 class Network

22 {

23 public:
```

7.9 network.h

```
31
        Network(
           const std::string &address,
uint16_t port,
const std::map<std::string, std::string> &database,
const std::string &log_path);
 32
33
34
35
36
 41
42
        std::string \&getAddress();
        uint16_t &getPort();
 ^{47}
 48
53
        std::map{<}std::string,\ std::string{>}\ \&getDatabase();\\
 54
 59
        void wait();
 60
 65
        void auth();
 66
71
72
76
        void calc();
        void close();
 77
81
        void quit();
88
89
90 };
 92 #endif // NETWORK_H
```

42Файлы

Предметный указатель

Avea Dogo do Evrantion 11	main
ArgsDecodeException, 11	main
ArgsDecodeException, 12	main.cpp, 39
auth	main.cpp, 37
Network, 25	main, 39
AuthException, 13	N 4 1 04
AuthException, 14	Network, 24
	auth, 25
calc	$\operatorname{calc}, \frac{25}{}$
Network, 25	getAddress, 27
calculateHash	getDatabase, 27
crutils, 9	getPort, 27
crutils, 9	Network, 25
calculateHash, 9	wait, 27
generateSalt, 10	network.h, 39
crutils.h, 31	NetworkException, 28
•	NetworkException, 29
DataDecodeException, 14	• ,
DataDecodeException, 16	$\operatorname{parseArgs}$
	Interface, 21
Exception, 16	
Exception, 17	readDB
what, 18	Interface, 21
exceptions.h, 33	run
, , , ,	Interface, 21
generateSalt	,
crutils, 10	wait
getAddress	Network, 27
Network, 27	what
getDatabase	Exception, 18
Network, 27	Encoperon, 10
getDbPath	
Interface, 20	
getLogPath	
Interface, 20	
getNetwork	
Interface, 20	
getPort	
Network, 27	
Interface, 18	
getDbPath, 20	
,	
getLogPath, 20	
getNetwork, 20	
Interface, 20	
parseArgs, 21	
readDB, 21	
$\operatorname{run}, 21$	
interface.h, 35	
IOException 22	

IOException, 23