

Kursach

Создано системой Doxygen 1.9.4

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Классы	7
4.1 Класс Authenticator	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 Authenticator()	8
4.1.3 Методы	8
4.1.3.1 authenticate()	8
4.1.3.2 compute_salt()	9
4.1.3.3 generate_hash()	9
4.2 Класс FileManager	9
4.2.1 Подробное описание	10
4.2.2 Методы	10
4.2.2.1 read_vectors()	10
4.2.2.2 write_results()	10
4.3 Класс ServerConnection	11
4.3.1 Подробное описание	11
4.3.2 Конструктор(ы)	11
4.3.2.1 ServerConnection()	11
4.3.3 Методы	12
4.3.3.1 connect()	12
4.3.3.2 get_socket()	12
4.3.3.3 send_vectors()	12
4.4 Класс TestAuthenticator	13
4.4.1 Методы	14
4.4.1.1 Authenticator()	14
4.5 Класс Vector	14
4.5.1 Подробное описание	14
4.5.2 Конструктор(ы)	14
4.5.2.1 Vector()	15
5 Файлы	17
5.1 Файл Authenticator.h	17
5.1.1 Подробное описание	17
5.2 Authenticator.h	18
5.3 Файл FileManager.h	18
5.3.1 Подробное описание	19

5.4 FileManager.h	20
5.5 Файл ServerConnection.h	20
5.5.1 Подробное описание	20
5.6 ServerConnection.h	21
5.7 Файл Vector.h	21
5.7.1 Подробное описание	22
5.8 Vector.h	23
Предметный указатель	25

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Authenticator	7
TestAuthenticator	13
FileManager	9
ServerConnection	11
Vector	14

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Authenticator	A class responsible for handling user authentication	7
FileManager	A utility class for handling file operations related to vectors and results	9
ServerConnection	Manages a connection to a server for sending vectors	11
TestAuthenticator	13
Vector	Represents a mathematical vector with a specified size and values	14

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Authenticator.h	17
FileManager.h	18
ServerConnection.h	20
Vector.h	21

Глава 4

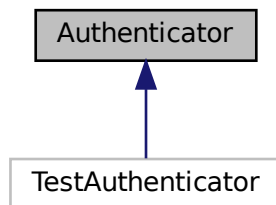
Классы

4.1 Класс Authenticator

A class responsible for handling user authentication.

```
#include <Authenticator.h>
```

Граф наследования: Authenticator:



Открытые члены

- [Authenticator](#) (const std::string &[login](#), const std::string &[password](#))
Constructs an [Authenticator](#) object with provided login and password.
- void [authenticate](#) (int socket)
Performs authentication using a socket.

Закрытые члены

- std::string [compute_salt](#) ()
Computes a random salt.
- std::string [generate_hash](#) (const std::string &salt16)
Generates a hash using the provided salt.

Закрытые данные

- `std::string login`
The user's login.
- `std::string password`
The user's password.

4.1.1 Подробное описание

A class responsible for handling user authentication.

4.1.2 Конструктор(ы)

4.1.2.1 Authenticator()

```
Authenticator::Authenticator (  
    const std::string & login,  
    const std::string & password )
```

Constructs an [Authenticator](#) object with provided login and password.

Аргументы

login	The user's login.
password	The user's password.

4.1.3 Методы

4.1.3.1 authenticate()

```
void Authenticator::authenticate (  
    int socket )
```

Performs authentication using a socket.

Аргументы

socket	The socket descriptor used for communication.
--------	---

Исключения

<code>std::runtime_error</code>	if authentication fails.
---------------------------------	--------------------------

4.1.3.2 `compute_salt()`

```
std::string Authenticator::compute_salt ( ) [private]
```

Computes a random salt.

Возвращает

A 16-byte random string used as a salt.

4.1.3.3 `generate_hash()`

```
std::string Authenticator::generate_hash (
    const std::string & salt16 ) [private]
```

Generates a hash using the provided salt.

Аргументы

<code>salt16</code>	A 16-byte salt used in hash generation.
---------------------	---

Возвращает

A hash of the password concatenated with the salt.

Объявления и описания членов классов находятся в файлах:

- [Authenticator.h](#)
- `Authenticator.cpp`

4.2 Класс FileManager

A utility class for handling file operations related to vectors and results.

```
#include <FileManager.h>
```

Открытые статические члены

- static std::vector< [Vector](#) > [read_vectors](#) (const std::string &filename)
Reads vectors from a specified file.
- static void [write_results](#) (const std::string &filename, const std::vector< uint64_t > &results)
Writes results to a specified file.

4.2.1 Подробное описание

A utility class for handling file operations related to vectors and results.

4.2.2 Методы

4.2.2.1 read_vectors()

```
std::vector< Vector > FileManager::read_vectors (
    const std::string & filename ) [static]
```

Reads vectors from a specified file.

Аргументы

filename	The name of the file to read from.
----------	------------------------------------

Возвращает

A vector of [Vector](#) objects read from the file.

4.2.2.2 write_results()

```
void FileManager::write_results (
    const std::string & filename,
    const std::vector< uint64_t > & results ) [static]
```

Writes results to a specified file.

Аргументы

filename	The name of the file to write to.
results	A vector of uint64_t results to be written to the file.

Объявления и описания членов классов находятся в файлах:

- [FileManager.h](#)
- [FileManager.cpp](#)

4.3 Класс ServerConnection

Manages a connection to a server for sending vectors.

```
#include <ServerConnection.h>
```

Открытые члены

- [ServerConnection](#) (const std::string &ip, uint16_t port)
Constructs a [ServerConnection](#) object with specified IP and port.
- void [connect](#) ()
Establishes a connection to the server.
- void [send_vectors](#) (const std::vector< [Vector](#) > &vectors)
Sends a collection of vectors to the server.
- void [close](#) ()
Closes the connection to the server.
- int [get_socket](#) () const
Retrieves the socket descriptor.

Закрытые данные

- std::string ip
The IP address of the server.
- uint16_t port
The port number to connect to.
- int sock
The socket file descriptor.

4.3.1 Подробное описание

Manages a connection to a server for sending vectors.

4.3.2 Конструктор(ы)

4.3.2.1 ServerConnection()

```
ServerConnection::ServerConnection (  
    const std::string & ip,  
    uint16_t port )
```

Constructs a [ServerConnection](#) object with specified IP and port.

Аргументы

ip	The IP address of the server.
port	The port number to connect to.

4.3.3 Методы

4.3.3.1 connect()

```
void ServerConnection::connect ( )
```

Establishes a connection to the server.

Исключения

std::runtime_error	if the connection fails.
--------------------	--------------------------

4.3.3.2 get_socket()

```
int ServerConnection::get_socket ( ) const [inline]
```

Retrieves the socket descriptor.

Возвращает

The socket file descriptor.

4.3.3.3 send_vectors()

```
void ServerConnection::send_vectors (
    const std::vector< Vector > & vectors )
```

Sends a collection of vectors to the server.

Аргументы

vectors	The vectors to be sent.
---------	-------------------------

Исключения

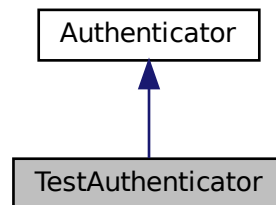
<code>std::runtime_error</code>	if sending fails.
---------------------------------	-------------------

Объявления и описания членов классов находятся в файлах:

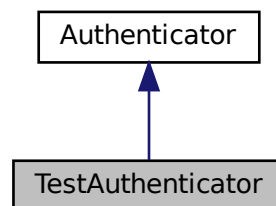
- [ServerConnection.h](#)
- [ServerConnection.cpp](#)

4.4 Класс TestAuthenticator

Граф наследования: TestAuthenticator:



Граф связей класса TestAuthenticator:



Открытые члены

- [Authenticator](#) (const std::string &[login](#), const std::string &[password](#))
Constructs an [Authenticator](#) object with provided login and password.

4.4.1 Методы

4.4.1.1 Authenticator()

```
Authenticator::Authenticator (
    const std::string & login,
    const std::string & password )
```

Constructs an [Authenticator](#) object with provided login and password.

Аргументы

login	The user's login.
password	The user's password.

Объявления и описания членов класса находятся в файле:

- test.cpp

4.5 Класс Vector

Represents a mathematical vector with a specified size and values.

```
#include <Vector.h>
```

Открытые члены

- [Vector](#) (uint32_t size, const std::vector< uint64_t > &values)
Constructs a [Vector](#) with given size and values.

Открытые атрибуты

- uint32_t size
The number of elements in the vector.
- std::vector< uint64_t > values
The values of the vector elements.

4.5.1 Подробное описание

Represents a mathematical vector with a specified size and values.

4.5.2 Конструктор(ы)

4.5.2.1 Vector()

```
Vector::Vector (
    uint32_t size,
    const std::vector< uint64_t > & values ) [inline]
```

Constructs a [Vector](#) with given size and values.

Аргументы

size	The number of elements in the vector.
values	The values stored in the vector.

Объявления и описания членов класса находятся в файле:

- [Vector.h](#)

Глава 5

Файлы

5.1 Файл Authenticator.h

```
#include <cryptopp/hex.h>
#include <cryptopp/osrng.h>
#include <string>
#include <openssl/sha.h>
#include <iostream>
#include <sstream>
#include <iomanip>
#include <cstdlib>
#include <stdexcept>
#include <sys/socket.h>
```

Граф включаемых заголовочных файлов для Authenticator.h:



Классы

- class [Authenticator](#)
A class responsible for handling user authentication.

5.1.1 Подробное описание

Автор

Заболотских З.В.

Версия

1.0

Дата

15.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.2 Authenticator.h

[См. документацию.](#)

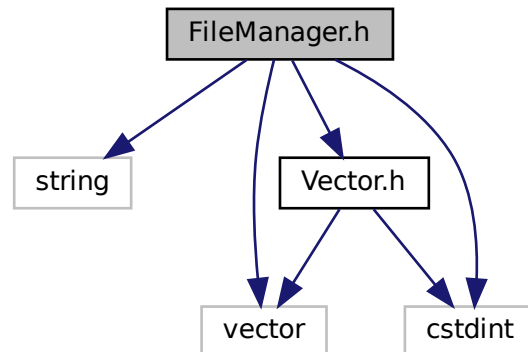
```
1
8 #ifndef AUTHENTICATOR_H
9 #define AUTHENTICATOR_H
10
11 #include <cryptopp/hex.h>
12 #include <cryptopp/osrng.h>
13 #include <string>
14 #include <openssl/sha.h>
15 #include <iostream>
16 #include <sstream>
17 #include <iomanip>
18 #include <cstdlib>
19 #include <stdexcept>
20 #include <sys/socket.h>
21
22 namespace CPP = CryptoPP;
23
24 class Authenticator {
25 public:
26     Authenticator(const std::string& login, const std::string& password);
27
28     void authenticate(int socket);
29
30 private:
31     std::string compute_salt();
32
33     std::string generate_hash(const std::string& salt16);
34
35     std::string login;
36     std::string password;
37 };
38
39 #endif // AUTHENTICATOR_H
```

5.3 Файл FileManager.h

```
#include <string>
#include <vector>
#include <cstdlib>
```

```
#include "Vector.h"
```

Граф включаемых заголовочных файлов для FileManager.h:



Классы

- class [FileManager](#)

A utility class for handling file operations related to vectors and results.

5.3.1 Подробное описание

Автор

Заболотских З.В.

Версия

1.0

Дата

15.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.4 FileManager.h

См. документацию.

```

1
8 #ifndef FILE_MANAGER_H
9 #define FILE_MANAGER_H
10
11 #include <string>
12 #include <vector>
13 #include <cstdint>
14 #include "Vector.h"
15
20 class FileManager {
21 public:
27     static std::vector<Vector> read_vectors(const std::string& filename);
28
34     static void write_results(const std::string& filename, const std::vector<uint64_t>& results);
35 };
36
37 #endif // FILE_MANAGER_H

```

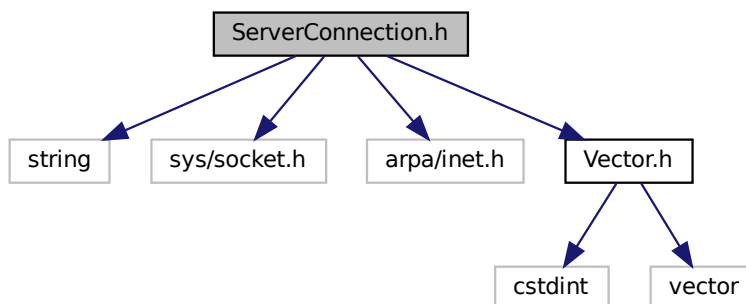
5.5 Файл ServerConnection.h

```

#include <string>
#include <sys/socket.h>
#include <arpa/inet.h>
#include "Vector.h"

```

Граф включаемых заголовочных файлов для ServerConnection.h:



Классы

- class [ServerConnection](#)
Manages a connection to a server for sending vectors.

5.5.1 Подробное описание

Автор

Заболотских З.В.

Версия

1.0

Дата

15.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.6 ServerConnection.h

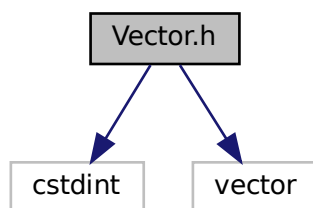
[См. документацию.](#)

```
1
2
3 8 #ifndef SERVER_CONNECTION_H
4 9 #define SERVER_CONNECTION_H
5 10
6 11 #include <string>
7 12 #include <sys/socket.h>
8 13 #include <arpa/inet.h>
9 14 #include "Vector.h"
10 15
11 20 class ServerConnection {
12 21 public:
13 27     ServerConnection(const std::string& ip, uint16_t port);
14 28
15 33     void connect();
16 34
17 40     void send_vectors(const std::vector<Vector>& vectors);
18 41
19 45     void close();
20 46
21 51     int get_socket() const { return sock; }
22 52
23 53 private:
24 54     std::string ip;
25 55     uint16_t port;
26 56     int sock;
27 57 };
28 58
29 59 #endif // SERVER_CONNECTION_H
```

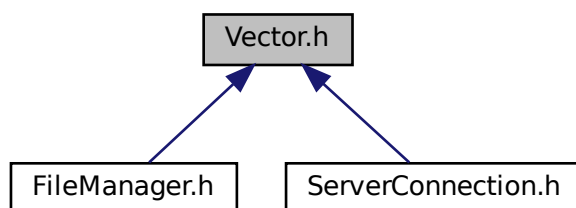
5.7 Файл Vector.h

```
#include <cstdint>
#include <vector>
```

Граф включаемых заголовочных файлов для Vector.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Vector](#)

Represents a mathematical vector with a specified size and values.

5.7.1 Подробное описание

Автор

Заболотских З.В.

Версия

1.0

Дата

15.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.8 Vector.h

[См. документацию.](#)

```
1
8 #ifndef VECTOR_H
9 #define VECTOR_H
10
11 #include <stdint>
12 #include <vector>
13
14 class Vector {
15 public:
16     Vector(uint32_t size, const std::vector<uint64_t>& values)
17         : size(size), values(values) {}
18
19     uint32_t size;
20     std::vector<uint64_t> values;
21 };
22
23 #endif // VECTOR_H
```


Предметный указатель

- authenticate
 - Authenticator, [8](#)
- Authenticator, [7](#)
 - authenticate, [8](#)
 - Authenticator, [8](#)
 - compute_salt, [9](#)
 - generate_hash, [9](#)
 - TestAuthenticator, [14](#)
- Authenticator.h, [17](#)
- compute_salt
 - Authenticator, [9](#)
- connect
 - ServerConnection, [12](#)
- FileManager, [9](#)
 - read_vectors, [10](#)
 - write_results, [10](#)
- FileManager.h, [18](#)
- generate_hash
 - Authenticator, [9](#)
- get_socket
 - ServerConnection, [12](#)
- read_vectors
 - FileManager, [10](#)
- send_vectors
 - ServerConnection, [12](#)
- ServerConnection, [11](#)
 - connect, [12](#)
 - get_socket, [12](#)
 - send_vectors, [12](#)
 - ServerConnection, [11](#)
- ServerConnection.h, [20](#)
- TestAuthenticator, [13](#)
 - Authenticator, [14](#)
- Vector, [14](#)
 - Vector, [14](#)
- Vector.h, [21](#)
- write_results
 - FileManager, [10](#)