

NetClient

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 ArgsDecoder          | 7  |
| 4.1.1 Подробное описание       | 7  |
| 4.1.2 Методы                   | 7  |
| 4.1.2.1 decodeArgs()           | 8  |
| 4.1.2.2 getConfigFilePath()    | 9  |
| 4.1.2.3 getInputFilePath()     | 9  |
| 4.1.2.4 getOutputFilePath()    | 9  |
| 4.1.2.5 getServerAddress()     | 10 |
| 4.1.2.6 getServerPort()        | 10 |
| 4.2 Класс Error                | 10 |
| 4.3 Класс FileManager          | 10 |
| 4.3.1 Подробное описание       | 11 |
| 4.3.2 Конструктор(ы)           | 11 |
| 4.3.2.1 FileManager()          | 11 |
| 4.3.3 Методы                   | 11 |
| 4.3.3.1 getConfigPath()        | 11 |
| 4.3.3.2 getInputPath()         | 12 |
| 4.3.3.3 getOutputPath()        | 12 |
| 4.3.3.4 loadCredentials()      | 12 |
| 4.3.3.5 readVectors()          | 13 |
| 4.3.3.6 writeVectors()         | 13 |
| 4.4 Класс Network              | 13 |
| 4.4.1 Подробное описание       | 14 |
| 4.4.2 Конструктор(ы)           | 14 |
| 4.4.2.1 Network()              | 14 |
| 4.4.3 Методы                   | 14 |
| 4.4.3.1 authenticate()         | 14 |
| 4.4.3.2 calculate()            | 15 |
| 4.4.3.3 connectToServer()      | 15 |
| 4.4.3.4 getServerAddress()     | 16 |
| 4.4.3.5 getServerPort()        | 16 |
| 5 Файлы                        | 17 |
| 5.1 ArgsDecoder.h              | 17 |
| 5.2 CryptoManager.h            | 17 |

|                                |    |
|--------------------------------|----|
| 5.3 Error.h . . . . .          | 18 |
| 5.4 FileManager.h . . . . .    | 18 |
| 5.5 Network.h . . . . .        | 18 |
| Предметный указатель . . . . . | 21 |

# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

|                       |    |
|-----------------------|----|
| ArgsDecoder . . . . . | 7  |
| FileManager . . . . . | 10 |
| Network . . . . .     | 13 |
| runtime_error         |    |
| Error . . . . .       | 10 |



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

|                             |   |                    |
|-----------------------------|---|--------------------|
| <a href="#">ArgsDecoder</a> | Класс для парсинга аргументов командной строки и хранения параметров конфигурации . . . . . | <a href="#">7</a>  |
| <a href="#">Error</a>       | Класс для обработки ошибок, наследуется от <code>std::runtime_error</code> . . . . .        | <a href="#">10</a> |
| <a href="#">FileManager</a> | Класс для управления файлами конфигурации, входных и выходных данных . .                    | <a href="#">10</a> |
| <a href="#">Network</a>     | Класс для управления сетевым подключением к серверу . . . . .                               | <a href="#">13</a> |





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

|                                 |    |
|---------------------------------|----|
| <a href="#">ArgsDecoder.h</a>   | ?? |
| <a href="#">CryptoManager.h</a> | ?? |
| <a href="#">Error.h</a>         | ?? |
| <a href="#">FileManager.h</a>   | ?? |
| <a href="#">Network.h</a>       | ?? |



## Глава 4

# Классы

### 4.1 Класс ArgsDecoder

Класс для парсинга аргументов командной строки и хранения параметров конфигурации.

```
#include <ArgsDecoder.h>
```

#### Открытые члены

- ArgsDecoder ()  
Конструктор по умолчанию, инициализирует параметры с значениями по умолчанию.
- string [getServerAddress](#) () const  
Возвращает адрес сервера.
- int [getServerPort](#) () const  
Возвращает порт сервера.
- string [getInputFilePath](#) () const  
Возвращает путь к входному файлу.
- string [getOutputFilePath](#) () const  
Возвращает путь к выходному файлу.
- string [getConfigFilePath](#) () const  
Возвращает путь к файлу конфигурации.
- void [decodeArgs](#) (int argc, char \*argv[])  
Парсит аргументы командной строки и инициализирует соответствующие параметры.
- void [displayHelp](#) () const  
Отображает справку по использованию программы.

#### 4.1.1 Подробное описание

Класс для парсинга аргументов командной строки и хранения параметров конфигурации.

#### 4.1.2 Методы

#### 4.1.2.1 decodeArgs()

```
void ArgsDecoder::decodeArgs (  
    int argc,  
    char * argv[] )
```

Парсит аргументы командной строки и инициализирует соответствующие параметры.

## Аргументы

|      |                        |
|------|------------------------|
| argc | Количество аргументов. |
| argv | Массив аргументов.     |

## Исключения

|              |   |
|--------------|---|
| <b>Error</b> | Если переданы неверные или недостаточные аргументы. |
|--------------|---|

## 4.1.2.2 getConfigFilePath()

```
string ArgsDecoder::getConfigFilePath ( ) const
```

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации в виде строки.

## 4.1.2.3 getInputFilePath()

```
string ArgsDecoder::getInputFilePath ( ) const
```

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу в виде строки.

## 4.1.2.4 getOutputFilePath()

```
string ArgsDecoder::getOutputFilePath ( ) const
```

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу в виде строки.

#### 4.1.2.5 `getServerAddress()`

```
string ArgsDecoder::getServerAddress ( ) const
```

Возвращает адрес сервера.

Возвращает

Адрес сервера в виде строки.

#### 4.1.2.6 `getServerPort()`

```
int ArgsDecoder::getServerPort ( ) const
```

Возвращает порт сервера.

Возвращает

Порт сервера в виде целого числа.

Объявления и описания членов классов находятся в файлах:

- `ArgsDecoder.h`
- `ArgsDecoder.cpp`

## 4.2 Класс `Error`

Класс для обработки ошибок, наследуется от `std::runtime_error`.

```
#include <Error.h>
```

Граф наследования: `Error`:

## 4.3 Класс `FileManager`

Класс для управления файлами конфигурации, входных и выходных данных.

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

## Открытые члены

- [FileManager](#) (const string &configPath, const string &inputPath, const string &outputPath)  
Конструктор класса [FileManager](#).
- array< string, 2 > [loadCredentials](#) () const  
Загрузка учетных данных из файла конфигурации.
- vector< vector< double > > [readVectors](#) () const  
Чтение векторов из входного файла.
- void [writeVectors](#) (const vector< double > &data) const  
Запись векторов в выходной файл.
- const string & [getConfigPath](#) () const  
Возвращает путь к файлу конфигурации.
- const string & [getInputPath](#) () const  
Возвращает путь к входному файлу.
- const string & [getOutputPath](#) () const  
Возвращает путь к выходному файлу.

## 4.3.1 Подробное описание

Класс для управления файлами конфигурации, входных и выходных данных.

## 4.3.2 Конструктор(ы)

## 4.3.2.1 FileManager()

```
FileManager::FileManager (
    const string & configPath,
    const string & inputPath,
    const string & outputPath )
```

Конструктор класса [FileManager](#).

Аргументы

|            |                            |
|------------|----------------------------|
| configPath | Путь к файлу конфигурации. |
| inputPath  | Путь к входному файлу.     |
| outputPath | Путь к выходному файлу.    |

## 4.3.3 Методы

## 4.3.3.1 getConfigPath()

```
const string & FileManager::getConfigPath ( ) const
```

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

#### 4.3.3.2 getInputPath()

```
const string & FileManager::getInputPath ( ) const
```

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу.

#### 4.3.3.3 getOutputPath()

```
const string & FileManager::getOutputPath ( ) const
```

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу.

#### 4.3.3.4 loadCredentials()

```
array< string, 2 > FileManager::loadCredentials ( ) const
```

Загрузка учетных данных из файла конфигурации.

Возвращает

Массив строк, содержащий учетные данные (логин и пароль).

Исключения

|              |   |
|--------------|---|
| <b>Error</b> | Если файл конфигурации не найден или содержит ошибки. |
|--------------|---|



## 4.3.3.5 readVectors()

```
vector< vector< double > > FileManager::readVectors ( ) const
```

Чтение векторов из входного файла.

Возвращает

Вектор векторов, содержащий данные.

Исключения

|       |  |
|-------|--|
| Error | Если входной файл не найден или содержит ошибки. |
|-------|--|

## 4.3.3.6 writeVectors()

```
void FileManager::writeVectors (
    const vector< double > & data ) const
```

Запись векторов в выходной файл.

Аргументы

|      |                                       |
|------|---------------------------------------|
| data | Вектор, содержащий данные для записи. |
|------|---------------------------------------|

Исключения

|       |   |
|-------|---|
| Error | Если выходной файл недоступен для записи. |
|-------|---|

Объявления и описания членов классов находятся в файлах:

- FileManager.h
- FileManager.cpp

## 4.4 Класс Network

Класс для управления сетевым подключением к серверу.

```
#include <Network.h>
```

## Открытые члены

- `Network` (const string &serverAddress, uint16\_t serverPort)  
Конструктор класса `Network`.
- void `connectToServer` ()  
Устанавливает соединение с сервером.
- void `authenticate` (const string &username, const string &password)  
Аутентифицирует пользователя на сервере.
- vector< double > `calculate` (const vector< vector< double > > &data)  
Отправляет данные на сервер и получает результат вычислений.
- void `closeConnection` ()  
Закрывает соединение с сервером.
- const string & `getServerAddress` () const  
Возвращает адрес сервера.
- uint16\_t `getServerPort` () const  
Возвращает порт сервера.

### 4.4.1 Подробное описание

Класс для управления сетевым подключением к серверу.

### 4.4.2 Конструктор(ы)

#### 4.4.2.1 Network()

```
Network::Network (
    const string & serverAddress,
    uint16_t serverPort )
```

Конструктор класса `Network`.

Аргументы

|               |                |
|---------------|----------------|
| serverAddress | Адрес сервера. |
| serverPort    | Порт сервера.  |

### 4.4.3 Методы

#### 4.4.3.1 authenticate()

```
void Network::authenticate (
    const string & username,
    const string & password )
```

Аутентифицирует пользователя на сервере.

Аргументы

|          |                      |
|----------|----------------------|
| username | Имя пользователя.    |
| password | Пароль пользователя. |

Исключения

|       |                                 |
|-------|---------------------------------|
| Error | Если аутентификация не удалась. |
|-------|---------------------------------|

#### 4.4.3.2 calculate()

```
vector< double > Network::calculate (
    const vector< vector< double > > & data )
```

Отправляет данные на сервер и получает результат вычислений.

Аргументы

|      |  |
|------|--|
| data | Вектор векторов, содержащий данные для отправки. |
|------|--|

Возвращает

Вектор, содержащий результаты вычислений.

Исключения

|       |  |
|-------|--|
| Error | Если произошла ошибка при отправке данных или получении результатов. |
|-------|--|

#### 4.4.3.3 connectToServer()

```
void Network::connectToServer ( )
```

Устанавливает соединение с сервером.

Исключения

|       |  |
|-------|--|
| Error | Если не удалось установить соединение. |
|-------|--|

#### 4.4.3.4 `getServerAddress()`

```
const string & Network::getServerAddress ( ) const
```

Возвращает адрес сервера.

Возвращает

Адрес сервера в виде строки.

#### 4.4.3.5 `getServerPort()`

```
uint16_t Network::getServerPort ( ) const
```

Возвращает порт сервера.

Возвращает

Порт сервера в виде целого числа.

Объявления и описания членов классов находятся в файлах:

- `Network.h`
- `Network.cpp`

## Глава 5

# Файлы

### 5.1 ArgsDecoder.h

```
1 #pragma once
2
3 #include "Error.h"
4 #include <string>
5 #include <vector>
6 #include <iostream>
7 #include <cstring>
8
9 using namespace std;
10
11 class ArgsDecoder
12 {
13 public:
14     ArgsDecoder();
15
16     string getServerAddress() const;
17
18     int getServerPort() const;
19
20     string getInputFilePath() const;
21
22     string getOutputFilePath() const;
23
24     string getConfigFilePath() const;
25
26     void decodeArgs(int argc, char *argv[]);
27
28     void displayHelp() const;
29
30 private:
31     string serverAddress;
32     uint16_t serverPort;
33     string inputFilePath;
34     string outputFilePath;
35     string configFilePath;
36 };
37
```

### 5.2 CryptoManager.h

```
1 #pragma once
2
3 #include <string>
4 #include <cryptopp/hex.h>
5 #include <cryptopp/sha.h>
6 #include <cryptopp/osrng.h>
7
8 using namespace std;
9 using namespace CryptoPP;
10
11 string salt();
12
13 string sha224(string &data);
14
```

## 5.3 Error.h

```

1 #pragma once
2
3 #include <stdexcept>
4 #include <string>
5
6 using namespace std;
7
12 class Error : public runtime_error
13 {
14 public:
21     Error(const string &message, const string &func);
22 };

```

## 5.4 FileManager.h

```

1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <array>
6 #include <fstream>
7 #include <iostream>
8 #include <iomanip>
9 #include <sstream>
10 #include "Error.h"
11
12 using namespace std;
13
18 class FileManager
19 {
20 public:
28     FileManager(const string &configPath, const string &inputPath, const string &outputPath);
29
36     array<string, 2> loadCredentials() const;
37
44     vector<vector<double> > readVectors() const;
45
52     void writeVectors(const vector<double> &data) const;
53
59     const string &getConfigPath() const;
60
66     const string &getInputPath() const;
67
73     const string &getOutputPath() const;
74
75 private:
76     string configPath;
77     string inputPath;
78     string outputPath;
79 };
80
86 void printVector(const vector<double> &data);
87
93 void printVectors(const vector<vector<double> > &data);

```

## 5.5 Network.h

```

1 #pragma once
2
3 #include "Error.h"
4 #include "CryptoManager.h"
5 #include <string>
6 #include <vector>
7 #include <cstdlib>
8 #include <cstring>
9 #include <unistd.h>
10 #include <iostream>
11
12 #include <sys/types.h>
13 #include <sys/socket.h>
14 #include <arpa/inet.h>
15
16 using namespace std;
17
22 class Network
23 {
24 public:

```

```
31  Network(const string &serverAddress, uint16_t serverPort);
32
38  void connectToServer();
39
47  void authenticate(const string &username, const string &password);
48
56  vector<double> calculate(const vector<vector<double> &data);
57
61  void closeConnection();
62
68  const string &getServerAddress() const;
69
75  uint16_t getServerPort() const;
76
77 private:
78     string serverAddress;
79     uint16_t serverPort;
80     int serverSocket;
81 };
```





# Предметный указатель

- ArgsDecoder, [7](#)
  - decodeArgs, [7](#)
  - getConfigFilePath, [9](#)
  - getInputFilePath, [9](#)
  - getOutputFilePath, [9](#)
  - getServerAddress, [9](#)
  - getServerPort, [10](#)
- authenticate
  - Network, [14](#)
- calculate
  - Network, [15](#)
- connectToServer
  - Network, [15](#)
- decodeArgs
  - ArgsDecoder, [7](#)
- Error, [10](#)
- FileManager, [10](#)
  - FileManager, [11](#)
  - getConfigPath, [11](#)
  - getInputPath, [12](#)
  - getOutputPath, [12](#)
  - loadCredentials, [12](#)
  - readVectors, [12](#)
  - writeVectors, [13](#)
- getConfigFilePath
  - ArgsDecoder, [9](#)
- getConfigPath
  - FileManager, [11](#)
- getInputFilePath
  - ArgsDecoder, [9](#)
- getInputPath
  - FileManager, [12](#)
- getOutputFilePath
  - ArgsDecoder, [9](#)
- getOutputPath
  - FileManager, [12](#)
- getServerAddress
  - ArgsDecoder, [9](#)
  - Network, [15](#)
- getServerPort
  - ArgsDecoder, [10](#)
  - Network, [16](#)
- loadCredentials
  - FileManager, [12](#)
- Network, [13](#)
  - authenticate, [14](#)
  - calculate, [15](#)
  - connectToServer, [15](#)
  - getServerAddress, [15](#)
  - getServerPort, [16](#)
  - Network, [14](#)
- readVectors
  - FileManager, [12](#)
- writeVectors
  - FileManager, [13](#)