

VClient

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 ArgsDecodeError	7
4.2 Класс AuthError	7
4.2.1 Подробное описание	7
4.2.2 Конструктор(ы)	7
4.2.2.1 AuthError()	7
4.3 Класс BasicClientError	8
4.3.1 Подробное описание	8
4.3.2 Конструктор(ы)	8
4.3.2.1 BasicClientError()	8
4.3.3 Методы	9
4.3.3.1 what()	9
4.4 Класс CRYPTManager	9
4.4.1 Подробное описание	9
4.4.2 Методы	9
4.4.2.1 get_hash()	9
4.4.2.2 get_salt()	10
4.5 Класс DataDecodeError	10
4.5.1 Подробное описание	10
4.5.2 Конструктор(ы)	10
4.5.2.1 DataDecodeError()	10
4.6 Класс IOError	11
4.6.1 Подробное описание	11
4.6.2 Конструктор(ы)	11
4.6.2.1 IOError()	11
4.7 Класс IOManager	12
4.7.1 Подробное описание	12
4.7.2 Конструктор(ы)	12
4.7.2.1 IOManager()	12
4.7.3 Методы	13
4.7.3.1 conf()	13
4.7.3.2 read()	13
4.7.3.3 write()	14
4.8 Класс NetworkError	14
4.8.1 Подробное описание	14

4.8.2 Конструктор(ы)	14
4.8.2.1 NetworkError()	14
4.9 Класс NetworkManager	15
4.9.1 Подробное описание	15
4.9.2 Конструктор(ы)	15
4.9.2.1 NetworkManager()	15
4.9.3 Методы	16
4.9.3.1 auth()	16
4.9.3.2 calc()	16
4.9.3.3 conn()	17
4.9.3.4 getAddress()	17
4.9.3.5 getPort()	17
4.10 Класс UserInterface	18
4.10.1 Подробное описание	18
4.10.2 Конструктор(ы)	18
4.10.2.1 UserInterface()	18
4.10.3 Методы	19
4.10.3.1 getAddress()	19
4.10.3.2 getConfigPath()	19
4.10.3.3 getInputPath()	19
4.10.3.4 getOutputPath()	20
4.10.3.5 getPort()	20
5 Файлы	21
5.1 Файл /home/student/Документы/kosov/client/source/modules/network.h	21
5.1.1 Подробное описание	21
5.2 network.h	22
5.3 Файл /home/student/Документы/kosov/client/source/modules/ui.h	22
5.3.1 Подробное описание	22
5.4 ui.h	23
5.5 Файл /home/student/Документы/kosov/client/source/modules/errors.h	23
5.5.1 Подробное описание	24
5.6 errors.h	24
5.7 Файл /home/student/Документы/kosov/client/source/modules/crypt.h	25
5.7.1 Подробное описание	25
5.8 crypt.h	26
5.9 Файл /home/student/Документы/kosov/client/source/modules/io.h	26
5.9.1 Подробное описание	26
5.10 io.h	27
5.11 Файл /home/student/Документы/kosov/client/source/main.cpp	27
5.11.1 Подробное описание	27
5.11.2 Функции	28
5.11.2.1 main()	28

Предметный указатель	29
----------------------	----

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

CryptManager	9
std::exception	
BasicClientError	8
ArgsDecodeError	7
AuthError	7
DataDecodeError	10
IOError	11
NetworkError	14
IOManager	12
NetworkManager	15
UserInterface	18

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

ArgsDecodeError	
Исключение для ошибок декодирования аргументов	7
AuthError	
Исключение для ошибок аутентификации	7
BasicClientError	
Базовый класс для клиентских ошибок	8
CryptManager	
Класс для выполнения криптографических операций	9
DataDecodeError	
Исключение для ошибок декодирования данных	10
IOError	
Исключение для ошибок ввода-вывода	11
IOManager	
Класс для управления вводом и выводом данных	12
NetworkError	
Исключение для сетевых ошибок	14
NetworkManager	
Класс для управления сетевым подключением и взаимодействием	15
UserInterface	
Класс для управления пользовательским интерфейсом	18

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

/home/student/Документы/kosov/client/source/modules/ crypt.h	
Определения классов для криптографических операций	25
/home/student/Документы/kosov/client/source/modules/ errors.h	
Определения классов исключений для клиентской части	23
/home/student/Документы/kosov/client/source/modules/ io.h	
Определения классов для управления вводом-выводом	26
/home/student/Документы/kosov/client/source/modules/ network.h	
Определения классов для управления сетевым взаимодействием	21
/home/student/Документы/kosov/client/source/modules/ ui.h	
Определения классов для пользовательского интерфейса	22
/home/student/Документы/kosov/client/source/ main.cpp	
Главный файл программы	27

Глава 4

Классы

4.1 Класс ArgsDecodeError

Исключение для ошибок декодирования аргументов.

```
#include <errors.h>
```

Граф наследования: ArgsDecodeError:

4.2 Класс AuthError

Исключение для ошибок аутентификации.

```
#include <errors.h>
```

Граф наследования: AuthError:

Граф связей класса AuthError:

Открытые члены

- [AuthError](#) (const std::string &[message](#), const std::string &[func](#))
Конструктор [AuthError](#).

Дополнительные унаследованные члены

4.2.1 Подробное описание

Исключение для ошибок аутентификации.

4.2.2 Конструктор(ы)

4.2.2.1 AuthError()

```
AuthError::AuthError (  
    const std::string & message,  
    const std::string & func )
```

Конструктор [AuthError](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникла ошибка.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/kosov/client/source/modules/errors.h`
- `/home/student/Документы/kosov/client/source/modules/errors.cpp`

4.3 Класс BasicClientError

Базовый класс для клиентских ошибок.

```
#include <errors.h>
```

Граф наследования: BasicClientError:

Граф связей класса BasicClientError:

Открытые члены

- [BasicClientError](#) (const std::string &name, const std::string &message, const std::string &func)
Конструктор [BasicClientError](#).
- const char * [what](#) () const noexcept override
Метод для получения сообщения об ошибке.

Защищенные данные

- std::string name
Имя исключения.
- std::string func
Имя функции, в которой возникла ошибка.
- std::string message
Сообщение об ошибке.

4.3.1 Подробное описание

Базовый класс для клиентских ошибок.

4.3.2 Конструктор(ы)

4.3.2.1 BasicClientError()

```
BasicClientError::BasicClientError (
    const std::string & name,
    const std::string & message,
    const std::string & func )
```

Конструктор [BasicClientError](#).

Аргументы

name	Имя исключения.
message	Сообщение об ошибке.
func	Имя функции, в которой возникла ошибка.

4.3.3 Методы

4.3.3.1 what()

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

Метод для получения сообщения об ошибке.

Возвращает

Сообщение об ошибке.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/kosov/client/source/modules/[errors.h](#)
- /home/student/Документы/kosov/client/source/modules/errors.cpp

4.4 Класс CryptManager

Класс для выполнения криптографических операций.

```
#include <crypt.h>
```

Открытые статические члены

- static std::string [get_salt](#) ()
Статический метод для генерации соли.
- static std::string [get_hash](#) (const std::string &salt, const std::string &data)
Статический метод для вычисления хеша.

4.4.1 Подробное описание

Класс для выполнения криптографических операций.

4.4.2 Методы

4.4.2.1 get_hash()

```
std::string CryptManager::get_hash (
    const std::string & salt,
    const std::string & data ) [static]
```

Статический метод для вычисления хеша.

Аргументы

salt	Соль, используемая для хеширования.
data	Данные, которые нужно захешировать.

Возвращает

Вычисленный хеш.

4.4.2.2 get_salt()

```
std::string CryptManager::get_salt ( ) [static]
```

Статический метод для генерации соли.

Возвращает

Сгенерированная соль.

Объявления и описания членов классов находятся в файлах:

- [/home/student/Документы/kosov/client/source/modules/crypt.h](#)
- [/home/student/Документы/kosov/client/source/modules/crypt.cpp](#)

4.5 Класс DataDecodeError

Исключение для ошибок декодирования данных.

```
#include <errors.h>
```

Граф наследования: DataDecodeError:

Граф связей класса DataDecodeError:

Открытые члены

- [DataDecodeError](#) (const std::string &message, const std::string &func)
Конструктор [DataDecodeError](#).

Дополнительные унаследованные члены

4.5.1 Подробное описание

Исключение для ошибок декодирования данных.

4.5.2 Конструктор(ы)

4.5.2.1 DataDecodeError()

```
DataDecodeError::DataDecodeError (
    const std::string & message,
    const std::string & func )
```

Конструктор [DataDecodeError](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникла ошибка.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/kosov/client/source/modules/errors.h`
- `/home/student/Документы/kosov/client/source/modules/errors.cpp`

4.6 Класс IOError

Исключение для ошибок ввода-вывода.

```
#include <errors.h>
```

Граф наследования:IOError:

Граф связей класса IOError:

Открытые члены

- `IOError` (`const std::string &message`, `const std::string &func`)
Конструктор `IOError`.

Дополнительные унаследованные члены

4.6.1 Подробное описание

Исключение для ошибок ввода-вывода.

4.6.2 Конструктор(ы)

4.6.2.1 IOError()

```
IOError::IOError (  
    const std::string & message,  
    const std::string & func )
```

Конструктор `IOError`.

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникла ошибка.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/kosov/client/source/modules/errors.h`
- `/home/student/Документы/kosov/client/source/modules/errors.cpp`

4.7 Класс IOManager

Класс для управления вводом и выводом данных.

```
#include <io.h>
```

Открытые члены

- `IOManager` (`const std::string &path_to_conf`, `const std::string &path_to_in`, `const std::string &path_to_out`)
Конструктор класса `IOManager`.
- `std::array< std::string, 2 > conf ()`
Метод для чтения конфигурационных данных.
- `std::vector< std::vector< uint32_t > > read ()`
Метод для чтения данных из файла.
- `void write` (`const std::vector< uint32_t > &data`)
Метод для записи данных в файл.

4.7.1 Подробное описание

Класс для управления вводом и выводом данных.

4.7.2 Конструктор(ы)

4.7.2.1 IOManager()

```
IOManager::IOManager (
    const std::string & path_to_conf,
    const std::string & path_to_in,
    const std::string & path_to_out )
```

Конструктор класса `IOManager`.

Аргументы

path_to_conf	Путь к файлу конфигурации.
path_to_in	Путь к входному файлу.
path_to_out	Путь к выходному файлу.

4.7.3 Методы

4.7.3.1 conf()

```
std::array< std::string, 2 > IOManager::conf ( )
```

Метод для чтения конфигурационных данных.

Возвращает

Массив строк с конфигурационными данными.

Исключения

IOError	Если не удалось открыть файл конфигурации.
DataDecodeError	Если отсутствуют логин или пароль.

4.7.3.2 read()

```
std::vector< std::vector< uint32_t > > IOManager::read ( )
```

Метод для чтения данных из файла.

Возвращает

Двумерный вектор с данными.

Исключения

IOError	Если не удалось открыть входной файл для чтения.
-------------------------	--

4.7.3.3 write()

```
void IOManager::write (
    const std::vector< uint32_t > & data )
```

Метод для записи данных в файл.

Аргументы

data	Вектор данных для записи.
------	---------------------------

Исключения

IOError	Если не удалось открыть выходной файл для записи.
-------------------------	---

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/kosov/client/source/modules/io.h`
- `/home/student/Документы/kosov/client/source/modules/io.cpp`

4.8 Класс NetworkError

Исключение для сетевых ошибок.

```
#include <errors.h>
```

Граф наследования: `NetworkError`:

Граф связей класса `NetworkError`:

Открытые члены

- [NetworkError](#) (const std::string &message, const std::string &func)
Конструктор [NetworkError](#).

Дополнительные унаследованные члены

4.8.1 Подробное описание

Исключение для сетевых ошибок.

4.8.2 Конструктор(ы)

4.8.2.1 NetworkError()

```
NetworkError::NetworkError (
    const std::string & message,
    const std::string & func )
```

Конструктор [NetworkError](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникла ошибка.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/kosov/client/source/modules/[errors.h](#)
- /home/student/Документы/kosov/client/source/modules/errors.cpp

4.9 Класс NetworkManager

Класс для управления сетевым подключением и взаимодействием.

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

Открытые члены

- [NetworkManager](#) (const std::string &address, uint16_t port)
Конструктор класса [NetworkManager](#).
- std::string & [getAddress](#) ()
Метод для получения адреса сервера.
- uint16_t & [getPort](#) ()
Метод для получения порта сервера.
- void [conn](#) ()
Метод для установления сетевого подключения.
- void [auth](#) (const std::string &username, const std::string &password)
Метод для аутентификации пользователя.
- std::vector< uint32_t > [calc](#) (const std::vector< std::vector< uint32_t > > &data)
Метод для передачи данных и получения результата.
- void close ()
Метод для закрытия сетевого подключения.

4.9.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

4.9.2 Конструктор(ы)

4.9.2.1 NetworkManager()

```
NetworkManager::NetworkManager (
    const std::string & address,
    uint16_t port )
```

Конструктор класса [NetworkManager](#).

Аргументы

address	Адрес сервера.
port	Порт сервера.

4.9.3 Методы

4.9.3.1 auth()

```
void NetworkManager::auth (
    const std::string & username,
    const std::string & password )
```

Метод для аутентификации пользователя.

Аргументы

username	Имя пользователя.
password	Пароль.

Исключения

AuthError	Если не удалось отправить сообщение об аутентификации или аутентификация не удалась.
---------------------------	--

4.9.3.2 calc()

```
std::vector< uint32_t > NetworkManager::calc (
    const std::vector< std::vector< uint32_t > > & data )
```

Метод для передачи данных и получения результата.

Аргументы

data	Данные для обработки.
------	-----------------------

Возвращает

Результаты обработки данных.

Исключения

NetworkError	Если не удалось отправить или получить данные.
------------------------------	--

4.9.3.3 conn()

```
void NetworkManager::conn ( )
```

Метод для установления сетевого подключения.

Исключения

NetworkError	Если не удалось создать сокет или установить соединение.
------------------------------	--

4.9.3.4 getAddress()

```
std::string & NetworkManager::getAddress ( )
```

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

4.9.3.5 getPort()

```
uint16_t & NetworkManager::getPort ( )
```

Метод для получения порта сервера.

Возвращает

Порт сервера.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/kosov/client/source/modules/[network.h](#)
- /home/student/Документы/kosov/client/source/modules/network.cpp

4.10 Класс `UserInterface`

Класс для управления пользовательским интерфейсом.

```
#include <ui.h>
```

Открытые члены

- `UserInterface` (int argc, char *argv[])
Конструктор класса `UserInterface`.
- `~UserInterface` ()
Деструктор класса `UserInterface`.
- `std::string & getAddress` ()
Метод для получения адреса.
- `uint16_t & getPort` ()
Метод для получения порта.
- `std::string & getInputPath` ()
Метод для получения пути к входному файлу.
- `std::string & getOutputPath` ()
Метод для получения пути к выходному файлу.
- `std::string & getConfigPath` ()
Метод для получения пути к конфигурационному файлу.
- `void run` ()
Метод для запуска интерфейса.

4.10.1 Подробное описание

Класс для управления пользовательским интерфейсом.

4.10.2 Конструктор(ы)

4.10.2.1 `UserInterface()`

```
UserInterface::UserInterface (
    int argc,
    char * argv[] )
```

Конструктор класса `UserInterface`.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Исключения

ArgsDecodeError	Если отсутствуют обязательные параметры или их значения.
---------------------------------	--

4.10.3 Методы

4.10.3.1 `getAddress()`

```
std::string & UI::UserInterface::getAddress ( )
```

Метод для получения адреса.

Возвращает

Адрес.

4.10.3.2 `getConfigPath()`

```
std::string & UI::UserInterface::getConfigPath ( )
```

Метод для получения пути к конфигурационному файлу.

Возвращает

Путь к конфигурационному файлу.

4.10.3.3 `getInputPath()`

```
std::string & UI::UserInterface::getInputPath ( )
```

Метод для получения пути к входному файлу.

Возвращает

Путь к входному файлу.

4.10.3.4 `getOutputPath()`

```
std::string & UserInterface::getOutputPath ( )
```

Метод для получения пути к выходному файлу.

Возвращает

Путь к выходному файлу.

4.10.3.5 `getPort()`

```
uint16_t & UserInterface::getPort ( )
```

Метод для получения порта.

Возвращает

Порт.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/kosov/client/source/modules/ui.h`
- `/home/student/Документы/kosov/client/source/modules/ui.cpp`

Глава 5

Файлы

5.1 Файл

/home/student/Документы/kosov/client/source/modules/network.h

Определения классов для управления сетевым взаимодействием.

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

Граф включаемых заголовочных файлов для network.h: Граф файлов, в которые включается этот файл:

Классы

- class [NetworkManager](#)

Класс для управления сетевым подключением и взаимодействием.

5.1.1 Подробное описание

Определения классов для управления сетевым взаимодействием.

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

Дата

20.11.2024

Версия

1.0

Авторы

Косов Р. С.

Авторство

ИБСТ ПГУ

5.2 network.h

См. документацию.

```
1 #ifndef NETWORK_MANAGER_H
2 #define NETWORK_MANAGER_H
3
4 #include <string>
5 #include <vector>
6 #include <cstdint>
7
21 class NetworkManager
22 {
23 public:
24     NetworkManager(const std::string &address, uint16_t port);
25
30     std::string &getAddress();
31
36     uint16_t &getPort();
37
42     void conn();
43
48     void auth(const std::string &username, const std::string &password);
49
56     std::vector<uint32_t> calc(const std::vector<std::vector<uint32_t>> &data);
57
64     void close();
65
70 private:
71     int socket;
72     std::string address;
73     uint16_t port;
74 };
75
76 #endif // NETWORK_MANAGER_H
```

5.3 Файл /home/student/Документы/kosov/client/source/modules/ui.h

Определения классов для пользовательского интерфейса.

```
#include "io.h"
#include "network.h"
#include "errors.h"
#include <string>
#include <vector>
```

Граф включаемых заголовочных файлов для ui.h: Граф файлов, в которые включается этот файл:

Классы

- class `UserInterface`
Класс для управления пользовательским интерфейсом.

5.3.1 Подробное описание

Определения классов для пользовательского интерфейса.

Этот файл содержит определения классов для обработки пользовательского ввода и управления интерфейсом.

Дата

23.11.2024

Версия

1.0 @authorsa Косов Р. С.

Авторство

ИБСТ ПГУ

5.4 ui.h

См. документацию.

```

1 #ifndef UI_H
2 #define UI_H
3
4 #include "io.h"
5 #include "network.h"
6 #include "errors.h"
7 #include <string>
8 #include <vector>
9
23 class UserInterface
24 {
25 public:
32     UserInterface(int argc, char *argv[]);
33
37     ~UserInterface();
38
43     std::string &getAddress();
44
49     uint16_t &getPort();
50
55     std::string &getInputPath();
56
61     std::string &getOutputPath();
62
67     std::string &getConfigPath();
68
72     void run();
73
74 private:
75     std::string address;
76     uint16_t port;
77     std::string input_path;
78     std::string output_path;
79     std::string config_path;
80
81     IOManager *io_man;
82     NetworkManager *net_man;
83
84     bool help_flag;
85
92     void parseArgs(int argc, char *argv[]);
93
97     void showHelp();
98 };
99
100 #endif // UI_H

```

5.5 Файл

/home/student/Документы/kosov/client/source/modules/errors.h

Определения классов исключений для клиентской части.

```

#include <exception>
#include <string>

```

Граф включаемых заголовочных файлов для errors.h: Граф файлов, в которые включается этот файл:

Классы

- class [BasicClientError](#)
Базовый класс для клиентских ошибок.
- class [IOError](#)
Исключение для ошибок ввода-вывода.
- class [ArgsDecodeError](#)
Исключение для ошибок декодирования аргументов.

- class [DataDecodeError](#)
Исключение для ошибок декодирования данных.
- class [AuthError](#)
Исключение для ошибок аутентификации.
- class [NetworkError](#)
Исключение для сетевых ошибок.

5.5.1 Подробное описание

Определения классов исключений для клиентской части.

Этот файл содержит определения классов для обработки различных исключений, возникающих в клиентской части программы.

Дата

20.11.2024

Версия

1.0

Авторы

Косов Р. С.

Авторство

ИБСТ ПГУ

5.6 errors.h

[См. документацию.](#)

```
1 #ifndef ERRORS_H
2 #define ERRORS_H
3
4 #include <exception>
5 #include <string>
6
20 class BasicClientError : public std::exception
21 {
22 public:
29     BasicClientError(const std::string &name, const std::string &message, const std::string &func);
30
35     const char *what() const noexcept override;
36
37 protected:
38     std::string name;
39     std::string func;
40     mutable std::string message;
41 };
42
46 class IOError : public BasicClientError
47 {
48 public:
54     IOError(const std::string &message, const std::string &func);
55 };
56
60 class ArgsDecodeError : public BasicClientError
61 {
```

```
62 public:
63     ArgsDecodeError(const std::string &message, const std::string &func);
64 };
65
66 class DataDecodeError : public BasicClientError
67 {
68 public:
69     DataDecodeError(const std::string &message, const std::string &func);
70 };
71
72 class AuthError : public BasicClientError
73 {
74 public:
75     AuthError(const std::string &message, const std::string &func);
76 };
77
78 class NetworkError : public BasicClientError
79 {
80 public:
81     NetworkError(const std::string &message, const std::string &func);
82 };
83
84 #endif // ERRORS_H
```

5.7 Файл

/home/student/Документы/kosov/client/source/modules/crypt.h

Определения классов для криптографических операций.

```
#include <string>
```

Граф включаемых заголовочных файлов для crypt.h:

Классы

- class `CryptManager`

Класс для выполнения криптографических операций.

5.7.1 Подробное описание

Определения классов для криптографических операций.

Этот файл содержит определения классов для генерации соли и вычисления хеша данных.

Дата

20.11.2024

Версия

1.0

Авторы

Косов Р. С.

Авторство

ИБСТ ПГУ

5.8 crypt.h

См. документацию.

```
1 #ifndef CRYPT_MANAGER_H
2 #define CRYPT_MANAGER_H
3
4 #include <string>
5
19 class CryptManager
20 {
21 public:
26     static std::string get_salt();
27
34     static std::string get_hash(const std::string &salt, const std::string &data);
35 };
36
37 #endif // CRYPT_MANAGER_H
```

5.9 Файл /home/student/Документы/kosov/client/source/modules/io.h

Определения классов для управления вводом-выводом.

```
#include <string>
#include <vector>
#include <array>
#include "errors.h"
```

Граф включаемых заголовочных файлов для io.h: Граф файлов, в которые включается этот файл:

Классы

- class [IOManager](#)

Класс для управления вводом и выводом данных.

5.9.1 Подробное описание

Определения классов для управления вводом-выводом.

Этот файл содержит определения классов для работы с файлами ввода-вывода и конфигурационными данными.

Дата

20.11.2024

Версия

1.0

Авторы

Косов Р. С.

Авторство

ИБСТ ПГУ

5.10 io.h

См. документацию.

```
1 #ifndef IO_MANAGER_H
2 #define IO_MANAGER_H
3
4 #include <string>
5 #include <vector>
6 #include <array>
7 #include "errors.h"
8
22 class IOManager {
23 public:
30     IOManager(
31         const std::string& path_to_conf,
32         const std::string& path_to_in,
33         const std::string& path_to_out
34     );
35
42     std::array<std::string, 2> conf();
43
49     std::vector<std::vector<uint32_t>> read();
50
56     void write(const std::vector<uint32_t>& data);
57
58 private:
59     std::string path_to_conf;
60     std::string path_to_in;
61     std::string path_to_out;
62 };
63
64 #endif // IO_MANAGER_H
```

5.11 Файл /home/student/Документы/kosov/client/source/main.cpp

Главный файл программы.

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

Граф включаемых заголовочных файлов для main.cpp:

Функции

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

Главная функция программы.

5.11.1 Подробное описание

Главный файл программы.

Этот файл содержит функцию `main`, которая инициализирует интерфейс пользователя и запускает программу.

Дата

20.11.2024

Версия

1.0

Авторы

Косов Р. С.

Авторство

ИБСТ ПГУ

5.11.2 Функции

5.11.2.1 main()

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

Главная функция программы.

Инициализирует объект [UserInterface](#) и запускает его. Обработывает 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.

Предметный указатель

/home/student/Документы/kosov/client/source/main.cpp, 27
NetworkManager, 17
UserInterface, 20
/home/student/Документы/kosov/client/source/modules/crypt.h, 25, 26
IOError, 11
/home/student/Документы/kosov/client/source/modules/crypt.h, 23, 24
IOError, 11
IOManager, 12
/home/student/Документы/kosov/client/source/modules/conf.h, 26, 27
IOManager, 12
/home/student/Документы/kosov/client/source/modules/network.h, 21, 22
read, 13
write, 13
/home/student/Документы/kosov/client/source/modules/ui.h, 22, 23
main
main.cpp, 28
main.cpp
main, 28
ArgsDecodeError, 7
auth
NetworkManager, 16
AuthError, 7
AuthError, 7
BasicClientError, 8
BasicClientError, 8
what, 9
calc
NetworkManager, 16
conf
IOManager, 13
conn
NetworkManager, 17
CryptManager, 9
get_hash, 9
get_salt, 10
DataDecodeError, 10
DataDecodeError, 10
get_hash
CryptManager, 9
get_salt
CryptManager, 10
getAddress
NetworkManager, 17
UserInterface, 19
getConfigPath
UserInterface, 19
getInputPath
UserInterface, 19
getOutputPath
UserInterface, 19
getPort
NetworkManager, 17
UserInterface, 20
IOError, 11
IOManager, 12
read, 13
write, 13
main
main.cpp, 28
main.cpp
main, 28
NetworkError, 14
NetworkError, 14
NetworkManager, 15
auth, 16
calc, 16
conn, 17
getAddress, 17
getPort, 17
NetworkManager, 15
read
IOManager, 13
UserInterface, 18
getAddress, 19
getConfigPath, 19
getInputPath, 19
getOutputPath, 19
getPort, 20
UserInterface, 18
what
BasicClientError, 9
write
IOManager, 13