VClient

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 ArgsDecodeError

Класс для обработки ошибок декодирования аргументов.

#include <errors.h>

Граф наследования:ArgsDecodeError:

4.2 Класс AuthError

Класс для обработки ошибок аутентификации.

#include <errors.h>

Граф наследования:AuthError:

Граф связей класса AuthError:

Открытые члены

- AuthError (const std::string &message, const std::string &func)
  Конструктор класса AuthError.

Дополнительные унаследованные члены

4.2.1 Подробное описание

Класс для обработки ошибок аутентификации.

4.2.2 Конструктор(ы)

4.2.2.1 AuthError()

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

Конструктор класса AuthError.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/yagolnicki/client/source/modules/errors.h
- /home/student/Документы/yagolnicki/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 класса BasicClientError.
- const char * what () const noexcept override
  Метод для получения сообщения об ошибке.

Защищенные данные

- std::string name
  Имя ошибки.
- 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 класса BasicClientError.

Аргументы

name	Имя ошибки.
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/yagolnicki/client/source/modules/errors.h
- /home/student/Документы/yagolnicki/client/source/modules/errors.cpp

4.4 Класс CryptMan

Класс для управления криптографическими операциями.

#include <cryptman.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 управления криптографическими операциями.

4.4.2 Методы

4.4.2.1 get_hash()

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

Статический метод для вычисления хеша.

Аргументы

salt	Соль, используемая для хеширования.
data	Данные для хеширования.

Возвращает

Хеш в виде строки.

4.4.2.2 get_salt()

std::string CryptMan::get_salt ( ) [static]

Статический метод для генерации соли.

Возвращает

Соль в виде строки.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/yagolnicki/client/source/modules/cryptman.h
- /home/student/Документы/yagolnicki/client/source/modules/cryptman.cpp

4.5 Класс FileNotFoundError

Класс для обработки ошибок "файл не найден".

#include <errors.h>

Граф наследования:FileNotFoundError:

Граф связей класса FileNotFoundError:

Открытые члены

- FileNotFoundError (const std::string &message, const std::string &func)
  Конструктор класса FileNotFoundError.

Дополнительные унаследованные члены

4.5.1 Подробное описание

Класс для обработки ошибок "файл не найден".

4.5.2 Конструктор(ы)

4.5.2.1 FileNotFoundError()

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

Конструктор класса FileNotFoundError.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

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

4.6 Класс InvalidDataFormatError

Класс для обработки ошибок некорректного формата данных.

#include <errors.h>

Граф наследования:InvalidDataFormatError:

Граф связей класса InvalidDataFormatError:

Открытые члены

- InvalidDataFormatError (const std::string &message, const std::string &func)
  Конструктор класса InvalidDataFormatError.

Дополнительные унаследованные члены

4.6.1 Подробное описание

Класс для обработки ошибок некорректного формата данных.

4.6.2 Конструктор(ы)

4.6.2.1 InvalidDataFormatError()

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

Конструктор класса InvalidDataFormatError.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

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

4.7 Класс IOMan

Класс для управления вводом и выводом данных.

#include <ioman.h>

Открытые члены

- IOMan (const std::string &path_to_conf, const std::string &path_to_in, const std::string &path_to_out)
  Конструктор класса IOMan.
- std::array< std::string, 2 > conf ()
  Метод для чтения конфигурационных данных.
- std::vector< std::vector< int16_t > > read ()
  Метод для чтения данных из файла.
- void write (const std::vector< int16_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 IOMan()

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

Конструктор класса IOMan.

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 > IOMan::conf ( )

Метод для чтения конфигурационных данных.

Возвращает

Массив строк с конфигурационными данными.

Исключения

FileNotFoundError	Если не удалось открыть файл конфигурации.
InvalidDataFormatError	Если отсутствуют логин или пароль.

4.7.3.2 read()

std::vector< std::vector< int16_t > > IOMan::read ( )

Метод для чтения данных из файла.

Возвращает

Двумерный вектор с данными.

Исключения

std::runtime_error	Если не удалось открыть входной файл.

4.7.3.3 write()

void IOMan::write (
const std::vector< int16_t > & data)

Метод для записи данных в файл.

Аргументы

data	Вектор данных для записи.

Исключения

FileNotFoundError	Если не удалось открыть выходной файл.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/yagolnicki/client/source/modules/ioman.h
- /home/student/Документы/yagolnicki/client/source/modules/ioman.cpp

4.8 Класс NetMan

Класс для управления сетевым подключением и взаимодействием.

#include <netman.h>

Открытые члены

- NetMan (const std::string &address, uint16_t port)
  Конструктор класса NetMan.
- 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< int16_t > calc (const std::vector< std::vector< int16_t > > &data)
  Метод для передачи данных и получения результата.
- void close ()
  Метод для закрытия сетевого подключения.

4.8.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

4.8.2 Конструктор(ы)

4.8.2.1 NetMan()

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

Конструктор класса NetMan.

Аргументы

address	Адрес сервера.
port	Порт сервера.

4.8.3 Методы

4.8.3.1 auth()

```
void NetMan::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 логин, получить соль, отправить хеш или аутентификация не удалась.

4.8.3.2 calc()

```
std::vector< int16_t > NetMan::calc (
            const std::vector< std::vector< int16_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.

4.8.3.3 conn()

void NetMan::conn ( )

Метод для установления сетевого подключения.

Исключения

NetworkError	Если не удалось создать сокет, установить соединение или адрес не поддерживается.

4.8.3.4 getAddress()

std::string & NetMan::getAddress ( )

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

4.8.3.5 getPort()

uint16_t & NetMan::getPort ( )

Метод для получения порта сервера.

Возвращает

Порт сервера.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/yagolnicki/client/source/modules/netman.h
- /home/student/Документы/yagolnicki/client/source/modules/netman.cpp

4.9 Класс NetworkError

Класс для обработки сетевых ошибок.

#include <errors.h>

Граф наследования:NetworkError:

Граф связей класса NetworkError:

Открытые члены

- NetworkError (const std::string &message, const std::string &func)
  Конструктор класса NetworkError.

Дополнительные унаследованные члены

4.9.1 Подробное описание

Класс для обработки сетевых ошибок.

4.9.2 Конструктор(ы)

4.9.2.1 NetworkError()

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

Конструктор класса NetworkError.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/yagolnicki/client/source/modules/errors.h
- /home/student/Документы/yagolnicki/client/source/modules/errors.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 сервера.
- uint16_t & getPort ()
  Метод для получения порта сервера.
- std::string & getInputFilePath ()
  Метод для получения пути к входному файлу.
- std::string & getOutputFilePath ()
  Метод для получения пути к выходному файлу.
- std::string & getConfigFilePath ()
  Метод для получения пути к конфигурационному файлу.
- void run ()
  Метод для запуска программы.

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 & UserInterface::getAddress ( )

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

4.10.3.2 getConfigFilePath()

std::string & UserInterface::getConfigFilePath ( )

Метод для получения пути к конфигурационному файлу.

Возвращает

Путь к конфигурационному файлу.

4.10.3.3 getInputFilePath()

std::string & UserInterface::getInputFilePath ( )

Метод для получения пути к входному файлу.

Возвращает

Путь к входному файлу.

4.10.3.4 getOutputFilePath()

std::string & UserInterface::getOutputFilePath ( )

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 сервера.

Возвращает

Порт сервера.

Объявления и описания членов классов находятся в файлах:

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

Глава 5

Файлы

5.1 Файл /home/student/Документы/yagolnicki/client/source/modules/ioman.h

Определение класса для управления вводом и выводом данных.

#include <string>
#include <vector>
#include <array>
#include "errors.h"
Граф включаемых заголовочных файлов для ioman.h: Граф файлов, в которые включается этот файл:

Классы

- class IOMan

  Класс для управления вводом и выводом данных.

5.1.1 Подробное описание

Определение класса для управления вводом и выводом данных.

Этот файл содержит определения методов для чтения и записи данных, а также для чтения конфигурационных файлов.

Дата

23.11.2024

Версия

1.0 @authorsa Ягольницкий Р. С.

5.2 ioman.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
21 class IOMan {
22 public:
29     IOMan(
30         const std::string& path_to_conf,
31         const std::string& path_to_in,
32         const std::string& path_to_out
33     );
34
41     std::array<std::string, 2> conf();
42
48     std::vector<std::vector<int16_t» read();
49
55     void write(const std::vector<int16_t>& data);
56
57 private:
58     std::string path_to_conf;
59     std::string path_to_in;
60     std::string path_to_out;
61 };
62
63 #endif // IO_MANAGER_H
```

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

Определение класса для пользовательского интерфейса.

#include "ioman.h"
#include "netman.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

Определение класса для пользовательского интерфейса.

Этот файл содержит определения методов для обработки аргументов командной строки, показа справки, запуска программы, и управления вводом/выводом и сетевым взаимодействием.

Дата

23.11.2024

Версия

1.0 @authorsa Ягольницкий Р. С.

5.4 ui.h

См. документацию.

```
1 #ifndef UI_H
2 #define UI_H
3
4 #include "ioman.h"
5 #include "netman.h"
6 #include "errors.h"
7 #include <string>
8 #include <vector>
9
22 class UserInterface
23 {
24 public:
31     UserInterface(int argc, char *argv[]);
32
36     ~UserInterface();
37
42     std::string &getAddress();
43
48     uint16_t &getPort();
49
54     std::string &getInputFilePath();
55
60     std::string &getOutputFilePath();
61
66     std::string &getConfigFilePath();
67
71     void run();
72
73 private:
74     std::string address;
75     uint16_t port;
76     std::string input_path;
77     std::string output_path;
78     std::string config_path;
79
80     IOMan *io_man;
81     NetMan *net_man;
82
83     bool help_flag;
84
91     void parseArgs(int argc, char *argv[]);
92
96     void showHelp();
97 };
98
99 #endif // UI_H
```

5.5 Файл /home/student/Документы/yagolnicki/client/source/modules/errors.h

Определение классов для обработки клиентских ошибок.

#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 FileNotFoundError

  Класс для обработки ошибок "файл не найден".
- class ArgsDecodeError

  Класс для обработки ошибок декодирования аргументов.

- class InvalidDataFormatError

  Класс для обработки ошибок некорректного формата данных.
- class AuthError

  Класс для обработки ошибок аутентификации.
- class NetworkError

  Класс для обработки сетевых ошибок.

5.5.1 Подробное описание

Определение классов для обработки клиентских ошибок.

Этот файл содержит определения классов исключений для обработки различных клиентских ошибок.

Дата

23.11.2024

Версия

1.0 @authorsa Ягольницкий Р. С.

5.6 errors.h

См. документацию.

```
1 #ifndef ERRORS_H
2 #define ERRORS_H
3
4 #include <exception>
5 #include <string>
6
19 class BasicClientError : public std::exception
20 {
21 public:
28     BasicClientError(const std::string &name, const std::string &message, const std::string &func);
29
34     const char *what() const noexcept override;
35
36 protected:
37     std::string name;
38     std::string func;
39     mutable std::string message;
40 };
41
45 class FileNotFoundError : public BasicClientError
46 {
47 public:
53     FileNotFoundError(const std::string &message, const std::string &func);
54 };
55
59 class ArgsDecodeError : public BasicClientError
60 {
61 public:
67     ArgsDecodeError(const std::string &message, const std::string &func);
68 };
69
73 class InvalidDataFormatError : public BasicClientError
74 {
75 public:
81     InvalidDataFormatError(const std::string &message, const std::string &func);
82 };
83
87 class AuthError : public BasicClientError
88 {
89 public:
95     AuthError(const std::string &message, const std::string &func);
96 };
97
101 class NetworkError : public BasicClientError
102 {
103 public:
109     NetworkError(const std::string &message, const std::string &func);
110 };
111
112 #endif // ERRORS_H
```

5.7 cryptman.h

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

5.8 Файл /home/student/Документы/yagolnicki/client/source/modules/netman.h

Определение класса для управления сетевым взаимодействием.

#include <string>
#include <vector>
#include <cstdint>
Граф включаемых заголовочных файлов для netman.h: Граф файлов, в которые включается этот файл:

Классы

- class NetMan

  Класс для управления сетевым подключением и взаимодействием.

5.8.1 Подробное описание

Определение класса для управления сетевым взаимодействием.

Этот файл содержит определения методов для установки соединения, аутентификации, передачи данных и закрытия соединения.

Дата

23.11.2024

Версия

1.0 @authorsa Ягольницкий Р. С.

5.9 netman.h

См. документацию.

```
1 #ifndef NETWORK_MANAGER_H
2 #define NETWORK_MANAGER_H
3
4 #include <string>
5 #include <vector>
6 #include <cstdint>
7
20 class NetMan
21 {
22 public:
28     NetMan(const std::string &address, uint16_t port);
29
34     std::string &getAddress();
35
40     uint16_t &getPort();
41
46     void conn();
47
54     void auth(const std::string &username, const std::string &password);
55
62     std::vector<int16_t> calc(const std::vector<std::vector<int16_t» &data);
63
67     void close();
68
69 private:
70     int socket;
71     std::string address;
72     uint16_t port;
73 };
74
75 #endif // NETWORK_MANAGER_H
```

5.10 Файл /home/student/Документы/yagolnicki/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.

5.10.1 Подробное описание

Главный файл программы.

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

Дата

23.11.2024

Версия

1.0 @authorsa Ягольницкий Р. С.

5.10.2 Функции

5.10.2.1 main()

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

Главная функция программы.

Инициализирует объект UserInterface 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.

Предметный указатель