

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 ArgsDecodeException	7
4.2 Класс AuthException	7
4.2.1 Подробное описание	7
4.2.2 Конструктор(ы)	7
4.2.2.1 AuthException()	7
4.3 Класс BasicClientException	8
4.3.1 Подробное описание	8
4.3.2 Конструктор(ы)	8
4.3.2.1 BasicClientException()	8
4.3.3 Методы	9
4.3.3.1 what()	9
4.4 Класс DataDecodeException	9
4.4.1 Подробное описание	9
4.4.2 Конструктор(ы)	9
4.4.2.1 DataDecodeException()	9
4.5 Класс InputOutputException	10
4.5.1 Подробное описание	10
4.5.2 Конструктор(ы)	10
4.5.2.1 InputOutputException()	10
4.6 Класс InputOutputManager	11
4.6.1 Подробное описание	11
4.6.2 Конструктор(ы)	11
4.6.2.1 InputOutputManager()	11
4.6.3 Методы	12
4.6.3.1 conf()	12
4.6.3.2 read()	12
4.6.3.3 write()	13
4.7 Класс NetworkException	13
4.7.1 Подробное описание	13
4.7.2 Конструктор(ы)	13
4.7.2.1 NetworkException()	13
4.8 Класс NetworkManager	14
4.8.1 Подробное описание	14
4.8.2 Конструктор(ы)	14

4.8.2.1 NetworkManager()	14
4.8.3 Методы	15
4.8.3.1 auth()	15
4.8.3.2 calc()	15
4.8.3.3 conn()	16
4.8.3.4 getAddress()	16
4.8.3.5 getPort()	16
4.9 Класс UserInterface	17
4.9.1 Подробное описание	17
4.9.2 Конструктор(ы)	17
4.9.2.1 UserInterface()	17
4.9.3 Методы	18
4.9.3.1 getAddress()	18
4.9.3.2 getConfigFilePath()	18
4.9.3.3 getInputFilePath()	18
4.9.3.4 getOutputFilePath()	19
4.9.3.5 getPort()	19
5 Файлы	21
5.1 Файл /home/student/Документы/makin/client/source/modules/network.h	21
5.1.1 Подробное описание	21
5.2 network.h	22
5.3 Файл /home/student/Документы/makin/client/source/modules/ui.h	22
5.3.1 Подробное описание	22
5.4 ui.h	23
5.5 Файл /home/student/Документы/makin/client/source/modules/exceptions.h	23
5.5.1 Подробное описание	24
5.6 exceptions.h	24
5.7 Файл /home/student/Документы/makin/client/source/modules/io.h	25
5.7.1 Подробное описание	25
5.8 io.h	26
5.9 Файл /home/student/Документы/makin/client/source/main.cpp	26
5.9.1 Подробное описание	26
5.9.2 Функции	27
5.9.2.1 main()	27
Предметный указатель	29

# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

std::exception	
BasicClientException . . . . .	8
ArgsDecodeException . . . . .	7
AuthException . . . . .	7
DataDecodeException . . . . .	9
InputOutputException . . . . .	10
NetworkException . . . . .	13
InputOutputManager . . . . .	11
NetworkManager . . . . .	14
UserInterface . . . . .	17



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">ArgsDecodeException</a>	
Исключение для ошибок декодирования аргументов . . . . .	7
<a href="#">AuthException</a>	
Исключение для ошибок аутентификации . . . . .	7
<a href="#">BasicClientException</a>	
Базовый класс для клиентских ошибок . . . . .	8
<a href="#">DataDecodeException</a>	
Исключение для ошибок декодирования данных . . . . .	9
<a href="#">InputOutputException</a>	
Исключение для ошибок ввода-вывода . . . . .	10
<a href="#">InputOutputManager</a>	
Класс для управления вводом и выводом данных . . . . .	11
<a href="#">NetworkException</a>	
Исключение для сетевых ошибок . . . . .	13
<a href="#">NetworkManager</a>	
Класс для управления сетевым подключением и взаимодействием . . . . .	14
<a href="#">UserInterface</a>	
Класс для управления пользовательским интерфейсом . . . . .	17





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

/home/student/Документы/makin/client/source/modules/ <a href="#">exceptions.h</a>	
Определения классов исключений для клиентской части . . . . .	23
/home/student/Документы/makin/client/source/modules/ <a href="#">io.h</a>	
Определения классов для управления вводом-выводом . . . . .	25
/home/student/Документы/makin/client/source/modules/ <a href="#">network.h</a>	
Определения классов для управления сетевым взаимодействием . . . . .	21
/home/student/Документы/makin/client/source/modules/ <a href="#">ui.h</a>	
Определения классов для пользовательского интерфейса . . . . .	22
/home/student/Документы/makin/client/source/ <a href="#">main.cpp</a>	
Главный файл программы . . . . .	26



## Глава 4

# Классы

### 4.1 Класс ArgsDecodeException

Исключение для ошибок декодирования аргументов.

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

Граф наследования: ArgsDecodeException:

### 4.2 Класс AuthException

Исключение для ошибок аутентификации.

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

Граф наследования: AuthException:

Граф связей класса AuthException:

Открытые члены

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

Дополнительные унаследованные члены

#### 4.2.1 Подробное описание

Исключение для ошибок аутентификации.

#### 4.2.2 Конструктор(ы)

##### 4.2.2.1 AuthException()

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

Конструктор [AuthException](#).

## Аргументы

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

Объявления и описания членов классов находятся в файлах:

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

## 4.3 Класс BasicClientException

Базовый класс для клиентских ошибок.

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

Граф наследования: BasicClientException:

Граф связей класса BasicClientException:

### Открытые члены

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

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

Конструктор [BasicClientException](#).

## 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 * BasicClientException::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/makin/client/source/modules/exceptions.h`
- `/home/student/Документы/makin/client/source/modules/exceptions.cpp`

## 4.4 Класс DataDecodeException

Исключение для ошибок декодирования данных.

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

Граф наследования: DataDecodeException:

Граф связей класса DataDecodeException:

## Открытые члены

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

## Дополнительные унаследованные члены

## 4.4.1 Подробное описание

Исключение для ошибок декодирования данных.

## 4.4.2 Конструктор(ы)

## 4.4.2.1 DataDecodeException()

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

Конструктор [DataDecodeException](#).

## Аргументы

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

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/makin/client/source/modules/exceptions.h`
- `/home/student/Документы/makin/client/source/modules/exceptions.cpp`

## 4.5 Класс InputOutputException

Исключение для ошибок ввода-вывода.

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

Граф наследования: `InputOutputException`:

Граф связей класса `InputOutputException`:

### Открытые члены

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

### Дополнительные унаследованные члены

#### 4.5.1 Подробное описание

Исключение для ошибок ввода-вывода.

#### 4.5.2 Конструктор(ы)

##### 4.5.2.1 `InputOutputException()`

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

Конструктор `InputOutputException`.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

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

## 4.6 Класс InputOutputManager

Класс для управления вводом и выводом данных.

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

Открытые члены

- [InputOutputManager](#) (const std::string &config, const std::string &input, const std::string &output)  
Конструктор класса [InputOutputManager](#).
- std::array< std::string, 2 > [conf](#) ()  
Метод для чтения конфигурационных данных.
- std::vector< std::vector< int32\_t > > [read](#) ()  
Метод для чтения данных из файла.
- void [write](#) (const std::vector< int32\_t > &data)  
Метод для записи данных в файл.

### 4.6.1 Подробное описание

Класс для управления вводом и выводом данных.

### 4.6.2 Конструктор(ы)

#### 4.6.2.1 InputOutputManager()

```
InputOutputManager::InputOutputManager (  
    const std::string & config,  
    const std::string & input,  
    const std::string & output )
```

Конструктор класса [InputOutputManager](#).

## Аргументы

config	Путь к файлу конфигурации.
input	Путь к входному файлу.
output	Путь к выходному файлу.

## 4.6.3 Методы

## 4.6.3.1 conf()

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

Метод для чтения конфигурационных данных.

Возвращает

Массив строк с конфигурационными данными.

## Исключения

<a href="#">InputOutputException</a>	Если не удалось открыть файл конфигурации.
<a href="#">DataDecodeException</a>	Если отсутствуют логин или пароль.

## 4.6.3.2 read()

```
std::vector< std::vector< int32_t > > InputOutputManager::read ( )
```

Метод для чтения данных из файла.

Возвращает

Двумерный вектор с данными.

## Исключения

<a href="#">InputOutputException</a>	Если не удалось открыть входной файл для чтения.
--------------------------------------	--



## 4.6.3.3 write()

```
void InputOutputManager::write (
    const std::vector< int32_t > & data )
```

Метод для записи данных в файл.

Аргументы

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

Исключения

<a href="#">InputOutputException</a>	Если не удалось открыть выходной файл для записи.
--------------------------------------	---

Объявления и описания членов классов находятся в файлах:

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

## 4.7 Класс NetworkException

Исключение для сетевых ошибок.

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

Граф наследования:NetworkException:

Граф связей класса NetworkException:

Открытые члены

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

Дополнительные унаследованные члены

## 4.7.1 Подробное описание

Исключение для сетевых ошибок.

## 4.7.2 Конструктор(ы)

## 4.7.2.1 NetworkException()

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

Конструктор [NetworkException](#).

## Аргументы

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

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/makin/client/source/modules/exceptions.h`
- `/home/student/Документы/makin/client/source/modules/exceptions.cpp`

## 4.8 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< int32_t > calc (const std::vector< std::vector< int32_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 NetworkManager()

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

Конструктор класса `NetworkManager`.

## Аргументы

address	Адрес сервера.
port	Порт сервера.

## 4.8.3 Методы

## 4.8.3.1 auth()

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

Метод для аутентификации пользователя.

## Аргументы

username	Имя пользователя.
password	Пароль.

## Исключения

<a href="#">AuthException</a>	Если не удалось отправить логин, получить соль, отправить хеш или аутентификация не удалась.
-------------------------------	--

## 4.8.3.2 calc()

```
std::vector< int32_t > NetworkManager::calc (
    const std::vector< std::vector< int32_t > > & data )
```

Метод для передачи данных и получения результата.

## Аргументы

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

## Возвращает

Результаты обработки данных.

## Исключения

<a href="#">NetworkException</a>	Если не удалось отправить или получить данные.
----------------------------------	--

## 4.8.3.3 conn()

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

Метод для установления сетевого подключения.

## Исключения

<a href="#">NetworkException</a>	Если не удалось создать сокет, установить соединение или адрес не поддерживается.
----------------------------------	---

## 4.8.3.4 getAddress()

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

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

## 4.8.3.5 getPort()

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

Метод для получения порта сервера.

Возвращает

Порт сервера.

Объявления и описания членов классов находятся в файлах:

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

## 4.9 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 & 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 интерфейса.

### 4.9.1 Подробное описание

Класс для управления пользовательским интерфейсом.

### 4.9.2 Конструктор(ы)

#### 4.9.2.1 `UserInterface()`

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

Конструктор класса `UserInterface`.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

## Исключения

<a href="#">ArgsDecodeException</a>	Если отсутствуют обязательные параметры или их значения.
-------------------------------------	--

## 4.9.3 Методы

## 4.9.3.1 getAddress()

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

Метод для получения адреса.

Возвращает

Адрес.

## 4.9.3.2 getConfigFilePath()

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

Метод для получения пути к конфигурационному файлу.

Возвращает

Путь к конфигурационному файлу.

## 4.9.3.3 getInputFilePath()

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

Метод для получения пути к входному файлу.

Возвращает

Путь к входному файлу.

#### 4.9.3.4 `getOutputFilePath()`

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

Метод для получения пути к выходному файлу.

Возвращает

Путь к выходному файлу.

#### 4.9.3.5 `getPort()`

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

Метод для получения порта.

Возвращает

Порт.

Объявления и описания членов классов находятся в файлах:

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





# Глава 5

## Файлы

### 5.1 Файл

/home/student/Документы/makin/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

23.11.2024

Версия

1.0

Авторы

Макин Д. Ю.

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:
29     NetworkManager(const std::string &address, uint16_t port);
30
35     std::string &getAddress();
36
41     uint16_t &getPort();
42
47     void conn();
48
55     void auth(const std::string &username, const std::string &password);
56
63     std::vector<int32_t> calc(const std::vector<std::vector<int32_t>> &data);
64
68     void close();
69
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/makin/client/source/modules/ui.h

Определения классов для пользовательского интерфейса.

```
#include "io.h"
#include "network.h"
#include "exceptions.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 Макин Д. Ю.

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 "exceptions.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 &getInputFilePath();
56
61     std::string &getOutputFilePath();
62
67     std::string &getConfigFilePath();
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     InputOutputManager *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/makin/client/source/modules/exceptions.h

Определения классов исключений для клиентской части.

```

#include <exception>
#include <string>

```

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

## Классы

- class [BasicClientException](#)  
Базовый класс для клиентских ошибок.
- class [InputOutputException](#)  
Исключение для ошибок ввода-вывода.
- class [ArgsDecodeException](#)  
Исключение для ошибок декодирования аргументов.
- class [DataDecodeException](#)  
Исключение для ошибок декодирования данных.
- class [AuthException](#)  
Исключение для ошибок аутентификации.
- class [NetworkException](#)  
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

23.11.2024

Версия

1.0 @authorsa Макин Д. Ю.

Авторство

ИБСТ ПГУ

## 5.6 exceptions.h

См. документацию.

```
1 #ifndef ERRORS_H
2 #define ERRORS_H
3
4 #include <exception>
5 #include <string>
6
20 class BasicClientException : public std::exception
21 {
22 public:
29     BasicClientException(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 InputOutputException : public BasicClientException
```

```

47 {
48 public:
54     InputOutputException(const std::string &message, const std::string &func);
55 };
56
60 class ArgsDecodeException : public BasicClientException
61 {
62 public:
68     ArgsDecodeException(const std::string &message, const std::string &func);
69 };
70
74 class DataDecodeException : public BasicClientException
75 {
76 public:
82     DataDecodeException(const std::string &message, const std::string &func);
83 };
84
88 class AuthException : public BasicClientException
89 {
90 public:
96     AuthException(const std::string &message, const std::string &func);
97 };
98
102 class NetworkException : public BasicClientException
103 {
104 public:
110     NetworkException(const std::string &message, const std::string &func);
111 };
112
113 #endif // ERRORS_H

```

## 5.7 Файл /home/student/Документы/makin/client/source/modules/io.h

Определения классов для управления вводом-выводом.

```

#include <string>
#include <vector>
#include <array>
#include "exceptions.h"

```

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

### Классы

- class `InputOutputManager`  
Класс для управления вводом и выводом данных.

### 5.7.1 Подробное описание

Определения классов для управления вводом-выводом.

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

Дата

23.11.2024

Версия

1.0

Авторы

Макин Д. Ю.

Авторство

ИБСТ ПГУ

## 5.8 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 "exceptions.h"
8
22 class InputOutputManager {
23 public:
30     InputOutputManager(
31         const std::string& config,
32         const std::string& input,
33         const std::string& output
34     );
35
42     std::array<std::string, 2> conf();
43
49     std::vector<std::vector<int32_t>> read();
50
56     void write(const std::vector<int32_t>& data);
57
58 private:
59     std::string config;
60     std::string input;
61     std::string output;
62 };
63
64 #endif // IO_MANAGER_H
```

## 5.9 Файл /home/student/Документы/makin/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.

### 5.9.1 Подробное описание

Главный файл программы.

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

Дата

23.11.2024

Версия

1.0 @authora Макин Д. Ю.

Авторство

ИБСТ ПГУ

## 5.9.2 Функции

### 5.9.2.1 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/makin/client/source/main.cpp, write, 12  
26  
/home/student/Документы/makin/client/source/modules/exceptions.h,  
23, 24  
/home/student/Документы/makin/client/source/modules/main.cpp,  
25, 26  
/home/student/Документы/makin/client/source/modules/network.h,  
21, 22  
/home/student/Документы/makin/client/source/modules/tui.h,  
22, 23  
ArgsDecodeException, 7  
auth  
    NetworkManager, 15  
AuthException, 7  
    AuthException, 7  
BasicClientException, 8  
    BasicClientException, 8  
    what, 9  
calc  
    NetworkManager, 15  
conf  
    InputOutputManager, 12  
conn  
    NetworkManager, 16  
DataDecodeException, 9  
    DataDecodeException, 9  
getAddress  
    NetworkManager, 16  
    UserInterface, 18  
getConfigFilePath  
    UserInterface, 18  
getInputFilePath  
    UserInterface, 18  
getOutputFilePath  
    UserInterface, 18  
getPort  
    NetworkManager, 16  
    UserInterface, 19  
InputOutputException, 10  
    InputOutputException, 10  
InputOutputManager, 11  
    conf, 12  
    InputOutputManager, 11  
    read, 12  
    write, 12  
    write, 12  
    main, 27  
    main, 27  
    NetworkException, 13  
    NetworkException, 13  
    NetworkManager, 14  
    auth, 15  
    calc, 15  
    conn, 16  
    getAddress, 16  
    getPort, 16  
    NetworkManager, 14  
read  
    InputOutputManager, 12  
UserInterface, 17  
    getAddress, 18  
    getConfigFilePath, 18  
    getInputFilePath, 18  
    getOutputFilePath, 18  
    getPort, 19  
    UserInterface, 17  
what  
    BasicClientException, 9  
write  
    InputOutputManager, 12