

ClientAPP

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
4.3 Класс BasicClientError	7
4.3.1 Подробное описание	8
4.4 Класс Client	8
4.4.1 Подробное описание	8
4.4.2 Конструктор(ы)	9
4.4.2.1 Client()	9
4.4.3 Методы	10
4.4.3.1 auth()	10
4.4.3.2 calc()	10
4.4.3.3 conn()	11
4.5 Класс Data	11
4.5.1 Подробное описание	11
4.5.2 Конструктор(ы)	12
4.5.2.1 Data()	12
4.5.3 Методы	12
4.5.3.1 conf()	12
4.5.3.2 getPathToConf()	12
4.5.3.3 getPathToIn()	13
4.5.3.4 getPathToOut()	13
4.5.3.5 read()	13
4.5.3.6 write()	14
4.6 Класс DataDecodeError	14
4.7 Класс InputOutputError	14
4.8 Класс NetworkError	15
4.9 Класс Terminal	15
4.9.1 Подробное описание	16
4.9.2 Методы	16
4.9.2.1 getAddress()	16
4.9.2.2 getConfigPath()	16
4.9.2.3 getInputPath()	16
4.9.2.4 getOutputPath()	17
4.9.2.5 getPort()	17

4.9.2.6 parseArgs()	17
5 Файлы	19
5.1 client.h	19
5.2 data.h	19
5.3 Файл code/errors.h	20
5.3.1 Подробное описание	20
5.4 errors.h	21
5.5 terminal.h	21
Предметный указатель	23

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Client	8
Data	11
std::exception	
BasicClientError	7
ArgsDecodeError	7
AuthError	7
DataDecodeError	14
InputOutputError	14
NetworkError	15
Terminal	15

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

ArgsDecodeError	7
AuthError	7
BasicClientError	
Базовый класс для клиентских ошибок	7
Client	
Класс для взаимодействия с сервером	8
Data	
Класс для работы с данными, включая конфигурацию и ввод/вывод файлов	11
DataDecodeError	14
InputOutputError	14
NetworkError	15
Terminal	
Класс для работы с терминалом	15

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

code/ client.h	19
code/ data.h	19
code/ errors.h	
Определения классов исключений для клиентской части	20
code/ terminal.h	21

Глава 4

Классы

4.1 Класс ArgsDecodeError

Граф наследования: ArgsDecodeError:

4.2 Класс AuthError

Граф наследования: AuthError:

Граф связей класса AuthError:

Открытые члены

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

Дополнительные унаследованные члены

Объявления и описания членов классов находятся в файлах:

- [code/errors.h](#)
- [code/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`)
- `const char * what () const noexcept override`

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.

Объявления и описания членов классов находятся в файлах:

- `code/errors.h`
- `code/errors.cpp`

4.4 Класс Client

Класс для взаимодействия с сервером.

```
#include <client.h>
```

Открытые члены

- `Client` (`const string &address`, `uint16_t port`)
Конструктор класса `Client`.
- `void conn ()`
Устанавливает соединение с сервером.
- `void auth` (`const string &username`, `const string &password`)
Аутентифицирует пользователя на сервере.
- `vector< uint64_t > calc` (`const vector< vector< uint64_t > > &data`)
Выполняет вычисления на сервере.
- `void close ()`
Закрывает соединение с сервером.

4.4.1 Подробное описание

Класс для взаимодействия с сервером.

4.4.2 Конструктор(ы)

4.4.2.1 Client()

```
Client::Client (
    const string & address,
    uint16_t port )
```

Конструктор класса [Client](#).

Аргументы

address	Адрес сервера.
port	Порт сервера.

4.4.3 Методы

4.4.3.1 auth()

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

Аутентифицирует пользователя на сервере.

Аргументы

username	Логин пользователя.
password	Пароль пользователя.

Исключения

AuthError	В случае ошибки отправки данных или неверной аутентификации.
---------------------------	--

4.4.3.2 calc()

```
vector< uint64_t > Client::calc (
    const vector< vector< uint64_t > > & data )
```

Выполняет вычисления на сервере.

Аргументы

data	Вектор векторов данных для отправки на сервер.
------	--

Возвращает

Вектор результатов вычислений от сервера.

Исключения

NetworkError	В случае ошибки отправки или получения данных.
------------------------------	--

4.4.3.3 conn()

```
void Client::conn ( )
```

Устанавливает соединение с сервером.

Исключения

NetworkError	В случае ошибки создания сокета или подключения.
---------------------	--

Объявления и описания членов классов находятся в файлах:

- code/client.h
- code/client.cpp

4.5 Класс Data

Класс для работы с данными, включая конфигурацию и ввод/вывод файлов.

```
#include <data.h>
```

Открытые члены

- **Data** (const string &path_to_conf, const string &path_to_in, const string &path_to_out)
Конструктор класса **Data**.
- array< string, 2 > **conf** ()
Читает конфигурационные данные из файла.
- vector< vector< uint64_t > > **read** ()
Читает данные из входного файла.
- void **write** (const vector< uint64_t > &data)
Записывает числовые данные в выходной файл.
- string **getPathToConf** () const
Возвращает путь к файлу конфигурации.
- string **getPathToIn** () const
Возвращает путь к входному файлу.
- string **getPathToOut** () const
Возвращает путь к выходному файлу.

4.5.1 Подробное описание

Класс для работы с данными, включая конфигурацию и ввод/вывод файлов.

4.5.2 Конструктор(ы)

4.5.2.1 Data()

```
Data::Data (
    const string & path_to_conf,
    const string & path_to_in,
    const string & path_to_out )
```

Конструктор класса [Data](#).

Аргументы

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

4.5.3 Методы

4.5.3.1 conf()

```
array< string, 2 > Data::conf ( )
```

Читает конфигурационные данные из файла.

Возвращает

Массив строк, содержащий логин и пароль.

Исключения

InputOutputError	В случае ошибки открытия файла.
DataDecodeError	В случае отсутствия логина или пароля.

4.5.3.2 getPathToConf()

```
string Data::getPathToConf ( ) const
```

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.5.3.3 getPathToIn()

```
string Data::getPathToIn ( ) const
```

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу.

4.5.3.4 getPathToOut()

```
string Data::getPathToOut ( ) const
```

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу.

4.5.3.5 read()

```
vector< vector< uint64_t > > Data::read ( )
```

Читает данные из входного файла.

Возвращает

Вектор векторов целых чисел.

Исключения

InputOutputError	В случае ошибки открытия файла.
----------------------------------	---------------------------------

4.5.3.6 write()

```
void Data::write (
    const vector< uint64_t > & data )
```

Записывает числовые данные в выходной файл.

Аргументы

data	Вектор целых чисел для записи.
------	--------------------------------

Исключения

InputOutputError	В случае ошибки открытия файла.
----------------------------------	---------------------------------

Объявления и описания членов классов находятся в файлах:

- code/data.h
- code/data.cpp

4.6 Класс DataDecodeError

Граф наследования:DataDecodeError:

Граф связей класса DataDecodeError:

Открытые члены

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

Дополнительные унаследованные члены

Объявления и описания членов классов находятся в файлах:

- code/[errors.h](#)
- code/errors.cpp

4.7 Класс InputOutputError

Граф наследования:InputOutputError:

Граф связей класса InputOutputError:

Открытые члены

- InputOutputError (const std::string &[message](#), const std::string &[func](#))

Дополнительные унаследованные члены

Объявления и описания членов классов находятся в файлах:

- `code/errors.h`
- `code/errors.cpp`

4.8 Класс NetworkError

Граф наследования: NetworkError:

Граф связей класса NetworkError:

Открытые члены

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

Дополнительные унаследованные члены

Объявления и описания членов классов находятся в файлах:

- `code/errors.h`
- `code/errors.cpp`

4.9 Класс Terminal

Класс для работы с терминалом.

```
#include <terminal.h>
```

Открытые члены

- `Terminal ()`
Конструктор класса `Terminal`.
- `string getAddress ()`
Возвращает адрес сервера.
- `int getPort ()`
Возвращает порт сервера.
- `string getInputPath ()`
Возвращает путь к входному файлу.
- `string getOutputPath ()`
Возвращает путь к выходному файлу.
- `string getConfigPath ()`
Возвращает путь к файлу конфигурации.
- `void parseArgs (int argc, char *argv[])`
Разбирает аргументы командной строки.
- `void showHelp ()`
Показывает справку.

4.9.1 Подробное описание

Класс для работы с терминалом.

4.9.2 Методы

4.9.2.1 getAddress()

```
string Terminal::getAddress ( )
```

Возвращает адрес сервера.

Возвращает

Адрес сервера.

4.9.2.2 getConfigPath()

```
string Terminal::getConfigPath ( )
```

Возвращает путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.9.2.3 getInputPath()

```
string Terminal::getInputPath ( )
```

Возвращает путь к входному файлу.

Возвращает

Путь к входному файлу.

4.9.2.4 getOutputPath()

```
string Terminal::getOutputPath ( )
```

Возвращает путь к выходному файлу.

Возвращает

Путь к выходному файлу.

4.9.2.5 getPort()

```
int Terminal::getPort ( )
```

Возвращает порт сервера.

Возвращает

Порт сервера.

4.9.2.6 parseArgs()

```
void Terminal::parseArgs (
    int argc,
    char * argv[] )
```

Разбирает аргументы командной строки.

Аргументы

argc	Количество аргументов.
argv	Массив аргументов.

Исключения

ArgsDecodeError	В случае ошибки разбора аргументов.
---------------------------------	-------------------------------------

Объявления и описания членов классов находятся в файлах:

- code/terminal.h
- code/terminal.cpp

Глава 5

Файлы

5.1 client.h

```
1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <stdint>
6
7 using namespace std;
8
9 class Client
10 {
11 public:
12     Client(const string &address, uint16_t port);
13     void conn();
14     void auth(const string &username, const string &password);
15     vector<uint64_t> calc(const vector<vector<uint64_t>> &data);
16     void close();
17 private:
18     int socket;
19     string address;
20     uint16_t port;
21 };
```

5.2 data.h

```
1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <array>
6 #include "errors.h"
7
8 using namespace std;
9
10 class Data
11 {
12 public:
13     Data(
14         const string &path_to_conf,
15         const string &path_to_in,
16         const string &path_to_out);
17     array<string, 2> conf();
18     vector<vector<uint64_t>> read();
19     void write(const vector<uint64_t> &data);
20     string getPathToConf() const;
21 }
```

```
60  string getPathToIn() const;
61
66  string getPathToOut() const;
67
68 private:
69  string path_to_conf;
70  string path_to_in;
71  string path_to_out;
72 };
```

5.3 Файл code/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 [InputOutputError](#)
- class [ArgsDecodeError](#)
- class [DataDecodeError](#)
- class [AuthError](#)
- class [NetworkError](#)

5.3.1 Подробное описание

Определения классов исключений для клиентской части.

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

Версия

1.0

Авторы

Белов А. Р.

Авторство

ИБСТ ПГУ

5.4 errors.h

См. документацию.

```

1
10 #pragma once
11
12 #include <exception>
13 #include <string>
14
15 class BasicClientError : public std::exception
16 {
17 public:
18     BasicClientError(const std::string &name, const std::string &message, const std::string &func);
19
20     const char *what() const noexcept override;
21
22 protected:
23     std::string name;
24     std::string func;
25     mutable std::string message;
26 };
27
28 class InputOutputError : public BasicClientError
29 {
30 public:
31     InputOutputError(const std::string &message, const std::string &func);
32 };
33
34 class ArgsDecodeError : public BasicClientError
35 {
36 public:
37     ArgsDecodeError(const std::string &message, const std::string &func);
38 };
39
40 class DataDecodeError : public BasicClientError
41 {
42 public:
43     DataDecodeError(const std::string &message, const std::string &func);
44 };
45
46 class AuthError : public BasicClientError
47 {
48 public:
49     AuthError(const std::string &message, const std::string &func);
50 };
51
52 class NetworkError : public BasicClientError
53 {
54 public:
55     NetworkError(const std::string &message, const std::string &func);
56 };
57

```

5.5 terminal.h

```

1 #pragma once
2
3 #include "errors.h"
4 #include <string>
5 #include <vector>
6
7 using namespace std;
8
9 class Terminal
10 {
11 public:
12     Terminal();
13
14     string getAddress();
15
16     int getPort();
17
18     string getInputPath();
19
20     string getOutputPath();
21
22     string getConfigPath();
23
24     void parseArgs(int argc, char *argv[]);
25
26     void showHelp();
27

```

```
64 private:
65     string address;
66     uint16_t port;
67     string input_path;
68     string output_path;
69     string config_path;
70     bool help_flag;
71 };
```

Предметный указатель

ArgsDecodeError, [7](#)

auth

Client, [10](#)

AuthError, [7](#)

BasicClientError, [7](#)

calc

Client, [10](#)

Client, [8](#)

auth, [10](#)

calc, [10](#)

Client, [9](#)

conn, [11](#)

code/client.h, [19](#)

code/data.h, [19](#)

code/errors.h, [20](#), [21](#)

code/terminal.h, [21](#)

conf

Data, [12](#)

conn

Client, [11](#)

Data, [11](#)

conf, [12](#)

Data, [12](#)

getPathToConf, [12](#)

getPathToIn, [13](#)

getPathToOut, [13](#)

read, [13](#)

write, [13](#)

DataDecodeError, [14](#)

getAddress

Terminal, [16](#)

getConfigPath

Terminal, [16](#)

getInputPath

Terminal, [16](#)

getOutputPath

Terminal, [16](#)

getPathToConf

Data, [12](#)

getPathToIn

Data, [13](#)

getPathToOut

Data, [13](#)

getPort

Terminal, [17](#)

InputOutputError, [14](#)

NetworkError, [15](#)

parseArgs

Terminal, [17](#)

read

Data, [13](#)

Terminal, [15](#)

getAddress, [16](#)

getConfigPath, [16](#)

getInputPath, [16](#)

getOutputPath, [16](#)

getPort, [17](#)

parseArgs, [17](#)

write

Data, [13](#)