

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 CMDParser	7
4.1.1 Подробное описание	7
4.1.2 Методы	7
4.1.2.1 getAddr()	8
4.1.2.2 getCfg()	8
4.1.2.3 getIn()	8
4.1.2.4 getOut()	8
4.1.2.5 getPort()	9
4.1.2.6 parse()	9
4.2 Класс ReadWrite	9
4.2.1 Подробное описание	10
4.2.2 Конструктор(ы)	10
4.2.2.1 ReadWrite()	10
4.2.3 Методы	10
4.2.3.1 getCfg()	10
4.2.3.2 getIn()	11
4.2.3.3 getOut()	11
4.2.3.4 readCfg()	11
4.2.3.5 readVec()	11
4.2.3.6 writeVec()	12
4.3 Класс VClient	12
4.3.1 Подробное описание	13
4.3.2 Конструктор(ы)	13
4.3.2.1 VClient()	13
4.3.3 Методы	13
4.3.3.1 auth()	13
4.3.3.2 calc()	14
4.3.3.3 conn()	14
4.3.3.4 getAddr()	14
4.3.3.5 getConf()	15
4.3.3.6 getPort()	15
4.3.3.7 getSock()	15
4.4 Класс VClientError	15
5 Файлы	17

5.1 vclient.h	17
5.2 error.h	17
5.3 rw.h	18
5.4 cmd.h	18
5.5 md5.h	19
Предметный указатель	21

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

CMDParser	7
exception	
VClientError	15
ReadWrite	9
VClient	12

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

CMDParser	Класс для парсинга командной строки	7
ReadWrite	Класс для чтения и записи данных из файлов	9
VClient	Класс для работы с клиентом	12
VClientError	Класс для обработки ошибок в клиенте	15

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

/home/student/Документы/source/doxygen/client/ cmd.h	18
/home/student/Документы/source/doxygen/client/ error.h	17
/home/student/Документы/source/doxygen/client/ md5.h	19
/home/student/Документы/source/doxygen/client/ rw.h	18
/home/student/Документы/source/doxygen/client/ vclient.h	17

Глава 4

Классы

4.1 Класс CMDParser

Класс для парсинга командной строки.

```
#include <cmd.h>
```

Открытые члены

- CMDParser ()=default
Конструктор класса [CMDParser](#) по умолчанию.
- string [getIn](#) () const
Получить путь к входному файлу.
- string [getOut](#) () const
Получить путь к выходному файлу.
- string [getCfg](#) () const
Получить путь к файлу конфигурации.
- string [getAddr](#) () const
Получить адрес сервера.
- uint16_t [getPort](#) () const
Получить порт сервера.
- void [parse](#) (int argc, char *argv[])
Парсить аргументы командной строки.
- void [showHelp](#) () const
Показать справку по использованию.

4.1.1 Подробное описание

Класс для парсинга командной строки.

4.1.2 Методы

4.1.2.1 getAddr()

```
string CMDParser::getAddr ( ) const
```

Получить адрес сервера.

Возвращает

Адрес сервера.

4.1.2.2 getCfg()

```
string CMDParser::getCfg ( ) const
```

Получить путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.1.2.3 getIn()

```
string CMDParser::getIn ( ) const
```

Получить путь к входному файлу.

Возвращает

Путь к входному файлу.

4.1.2.4 getOut()

```
string CMDParser::getOut ( ) const
```

Получить путь к выходному файлу.

Возвращает

Путь к выходному файлу.

4.1.2.5 `getPort()`

```
uint16_t CMDParser::getPort ( ) const
```

Получить порт сервера.

Возвращает

Порт сервера.

4.1.2.6 `parse()`

```
void CMDParser::parse (
    int argc,
    char * argv[] )
```

Парсить аргументы командной строки.

Аргументы

argc	Количество аргументов.
argv	Массив аргументов.

Исключения

VClientError	Если парсинг аргументов завершился ошибкой.
------------------------------	---

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/source/doxygen/client/cmd.h`
- `/home/student/Документы/source/doxygen/client/cmd.cpp`

4.2 Класс ReadWrite

Класс для чтения и записи данных из файлов.

```
#include <rw.h>
```

Открытые члены

- [ReadWrite](#) (string &inVec, string &outVec, string &cfg)
Конструктор класса [ReadWrite](#).
- string [getIn](#) ()
Получить имя входного файла.
- string [getOut](#) ()

- Получить имя выходного файла.
- `string` `getCfg` ()
- Получить имя файла конфигурации.
- `void` `writeVec` (`const vector< int32_t > &data`)
- Записать вектор данных в выходной файл.
- `vector< vector< int32_t > >` `readVec` ()
- Прочитать векторы данных из входного файла.
- `array< string, 2 >` `readCfg` ()
- Прочитать конфигурационные данные из файла конфигурации.

4.2.1 Подробное описание

Класс для чтения и записи данных из файлов.

4.2.2 Конструктор(ы)

4.2.2.1 ReadWrite()

```
ReadWrite::ReadWrite (
    string & inVec,
    string & outVec,
    string & cfg )
```

Конструктор класса `ReadWrite`.

Аргументы

<code>inVec</code>	Имя входного файла.
<code>outVec</code>	Имя выходного файла.
<code>cfg</code>	Имя файла конфигурации.

4.2.3 Методы

4.2.3.1 getCfg()

```
string ReadWrite::getCfg ( )
```

Получить имя файла конфигурации.

Возвращает

Имя файла конфигурации.

4.2.3.2 getIn()

```
string ReadWrite::getIn ( )
```

Получить имя входного файла.

Возвращает

Имя входного файла.

4.2.3.3 getOut()

```
string ReadWrite::getOut ( )
```

Получить имя выходного файла.

Возвращает

Имя выходного файла.

4.2.3.4 readCfg()

```
array< string, 2 > ReadWrite::readCfg ( )
```

Прочитать конфигурационные данные из файла конфигурации.

Возвращает

Массив строк с конфигурационными данными.

Исключения

VClientError	Если файл не может быть открыт или чтение конфигурации завершилось ошибкой.
------------------------------	---

4.2.3.5 readVec()

```
vector< vector< int32_t > > ReadWrite::readVec ( )
```

Прочитать векторы данных из входного файла.

Возвращает

Вектор векторов данных.

Исключения

VClientError	Если файл не может быть открыт или чтение из файла завершилось ошибкой.
------------------------------	---

4.2.3.6 writeVec()

```
void ReadWrite::writeVec (
    const vector< int32_t > & data )
```

Записать вектор данных в выходной файл.

Аргументы

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

Исключения

VClientError	Если файл не может быть открыт или запись в файл завершилась ошибкой.
------------------------------	---

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/source/doxygen/client/rw.h
- /home/student/Документы/source/doxygen/client/rw.cpp

4.3 Класс VClient

Класс для работы с клиентом.

```
#include <vclient.h>
```

Открытые члены

- [VClient](#) (string addr, uint16_t port, array< string, 2 > conf)
Конструктор класса [VClient](#).
- void [conn](#) ()
Установить соединение с сервером.
- void [stop](#) ()
Закрыть соединение с сервером.
- void [auth](#) ()
Выполнить аутентификацию на сервере.
- vector< int32_t > [calc](#) (vector< vector< int32_t > > data)
Выполнить расчет данных на сервере.
- int [getSock](#) ()
Получить сокет клиента.

- string [getAddr](#) ()
Получить адрес сервера.
- uint16_t [getPort](#) ()
Получить порт сервера.
- array< string, 2 > [getConf](#) ()
Получить конфигурационные данные для аутентификации.

4.3.1 Подробное описание

Класс для работы с клиентом.

4.3.2 Конструктор(ы)

4.3.2.1 VClient()

```
VClient::VClient (  
    string addr,  
    uint16_t port,  
    array< string, 2 > conf )
```

Конструктор класса [VClient](#).

Аргументы

addr	Адрес сервера.
port	Порт сервера.
conf	Массив строк с конфигурационными данными для аутентификации.

4.3.3 Методы

4.3.3.1 auth()

```
void VClient::auth ( )
```

Выполнить аутентификацию на сервере.

Исключения

VClientError	Если аутентификация не удалась.
------------------------------	---------------------------------

4.3.3.2 calc()

```
vector< int32_t > VClient::calc (
    vector< vector< int32_t > > data )
```

Выполнить расчет данных на сервере.

Аргументы

data	Вектор векторов данных для расчета.
------	-------------------------------------

Возвращает

Вектор рассчитанных данных.

Исключения

VClientError	Если расчет завершился ошибкой.
------------------------------	---------------------------------

4.3.3.3 conn()

```
void VClient::conn ( )
```

Установить соединение с сервером.

Исключения

VClientError	Если не удалось установить соединение.
------------------------------	--

4.3.3.4 getAddr()

```
string VClient::getAddr ( )
```

Получить адрес сервера.

Возвращает

Адрес сервера.

4.3.3.5 getConf()

```
array< string, 2 > VClient::getConf ( )
```

Получить конфигурационные данные для аутентификации.

Возвращает

Массив строк с конфигурационными данными для аутентификации.

4.3.3.6 getPort()

```
uint16_t VClient::getPort ( )
```

Получить порт сервера.

Возвращает

Порт сервера.

4.3.3.7 getSock()

```
int VClient::getSock ( )
```

Получить сокет клиента.

Возвращает

Сокет клиента.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/source/doxygen/client/vclient.h
- /home/student/Документы/source/doxygen/client/vclient.cpp

4.4 Класс VClientError

Класс для обработки ошибок в клиенте.

```
#include <error.h>
```

Граф наследования: VClientError:

Глава 5

Файлы

5.1 vclient.h

```
1 #pragma once
2
3 #include <stdint>
4 #include <string>
5 #include <cstring>
6 #include <array>
7 #include <iostream>
8 #include <vector>
9
10 #include <sys/socket.h>
11 #include <arpa/inet.h>
12 #include <unistd.h>
13
14 #include "error.h"
15 #include "md5.h"
16
17 using namespace std;
18
23 class VClient
24 {
25 private:
26     int sock;
27     string addr;
28     uint16_t port;
29     array<string, 2> conf;
30
31 public:
38     VClient(string addr, uint16_t port, array<string, 2> conf);
39
44     void conn();
45
49     void stop();
50
55     void auth();
56
63     vector<int32_t> calc(vector<vector<int32_t>> data);
64
69     int getSock();
70
75     string getAddr();
76
81     uint16_t getPort();
82
87     array<string, 2> getConf();
88 };
```

5.2 error.h

```
1 #pragma once
2
3 #include <string>
4 #include <exception>
5
6 using namespace std;
```

```

7
12 class VClientError : public exception {
13 public:
20   VClientError(const string& errorName, const string& errorMsg, const string& funcName);
21
26   const char* what() const noexcept override;
27
32   string getErrorName() const;
33
38   string getErrorMsg() const;
39
44   string getFuncName() const;
45
46 private:
47   string errorName;
48   string errorMsg;
49   string funcName;
50   mutable string fullMsg;
51 };

```

5.3 rw.h

```

1 #pragma once
2
3 #include <string>
4 #include <vector>
5 #include <array>
6 #include <stdint>
7 #include <fstream>
8 #include <sstream>
9 #include <iostream>
10 #include "error.h"
11
12 using namespace std;
13
14 class ReadWrite
15 {
16 public:
17   ReadWrite(string &inVec, string &outVec, string &cfg);
18
19   string getIn();
20
21   string getOut();
22
23   string getCfg();
24
25   void writeVec(const vector<int32_t> &data);
26
27   vector<vector<int32_t>> readVec();
28
29   array<string, 2> readCfg();
30
31 private:
32   string in;
33   string out;
34   string cfg;
35 };
36
37 void printVector(const vector<int32_t> &vec);
38
39 void printVectors(const vector<vector<int32_t>> &vec);

```

5.4 cmd.h

```

1 #pragma once
2
3 #include <string>
4 #include <iostream>
5 #include <stdexcept>
6 #include "error.h"
7
8 using namespace std;
9
10 class CMDParser {
11 private:
12   string in = "./in.txt";
13   string out = "./out.bin";
14   string cfg = "./cfg.txt";
15   string addr = "127.0.0.1";
16   uint16_t port = 33333;

```

```
21
22 public:
26     CMDParser() = default;
27
32     string getIn() const;
33
38     string getOut() const;
39
44     string getCfg() const;
45
50     string getAddr() const;
51
56     uint16_t getPort() const;
57
64     void parse(int argc, char* argv[]);
65
69     void showHelp() const;
70 };
```

5.5 md5.h

```
1 #pragma once
2
3 #include <string>
4 #include <cryptopp/hex.h>
5 #include <cryptopp/md5.h>
6 #include <cryptopp/osrng.h>
7
8 using namespace std;
9
14 string salt();
15
21 string md5(string &data);
```


Предметный указатель

[/home/student/Документы/source/doxygen/client/cmd.h](#), [ReadWrite](#), [11](#)
[18](#) [readVec](#)

[/home/student/Документы/source/doxygen/client/error.h](#), [ReadWrite](#), [11](#)
[17](#) [ReadWrite](#), [9](#)

[/home/student/Документы/source/doxygen/client/md5.h](#), [getCfg](#), [10](#)
[19](#) [getIn](#), [10](#)

[/home/student/Документы/source/doxygen/client/rw.h](#), [getOut](#), [11](#)
[18](#) [readCfg](#), [11](#)

[/home/student/Документы/source/doxygen/client/vclient.h](#), [readVec](#), [11](#)
[17](#) [ReadWrite](#), [10](#)
[writeVec](#), [12](#)

[auth](#)
[VClient](#), [13](#)

[calc](#)
[VClient](#), [13](#)

[CMDParser](#), [7](#)
[getAddr](#), [7](#)
[getCfg](#), [8](#)
[getIn](#), [8](#)
[getOut](#), [8](#)
[getPort](#), [8](#)
[parse](#), [9](#)

[conn](#)
[VClient](#), [14](#)

[getAddr](#)
[CMDParser](#), [7](#)
[VClient](#), [14](#)

[getCfg](#)
[CMDParser](#), [8](#)
[ReadWrite](#), [10](#)

[getConf](#)
[VClient](#), [14](#)

[getIn](#)
[CMDParser](#), [8](#)
[ReadWrite](#), [10](#)

[getOut](#)
[CMDParser](#), [8](#)
[ReadWrite](#), [11](#)

[getPort](#)
[CMDParser](#), [8](#)
[VClient](#), [15](#)

[getSock](#)
[VClient](#), [15](#)

[parse](#)
[CMDParser](#), [9](#)

[readCfg](#)

[VClient](#), [12](#)
[auth](#), [13](#)
[calc](#), [13](#)
[conn](#), [14](#)
[getAddr](#), [14](#)
[getConf](#), [14](#)
[getPort](#), [15](#)
[getSock](#), [15](#)
[VClient](#), [13](#)
[VClientError](#), [15](#)

[writeVec](#)
[ReadWrite](#), [12](#)