

Client::TEXT-IN::TEXT-OUT::SHA224::C-SALT::UINT32_T

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() | 11 |
| 4.3 Класс VClient | 12 |
| 4.4 Класс VClientError | 12 |
| 5 Файлы | 13 |
| 5.1 cmd.h | 13 |
| 5.2 error.h | 13 |
| 5.3 rw.h | 14 |
| 5.4 sha224.h | 14 |
| 5.5 vclient.h | 14 |
| Предметный указатель | 17 |

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

| | |
|------------------------|----|
| CMDParser | 7 |
| exception | |
| VClientError | 12 |
| ReadWrite | 9 |
| VClient | 12 |

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

| | |
|--|--------------------|
| CMDParser | |
| Класс для анализа командной строки и получения значений аргументов | 7 |
| ReadWrite | |
| Класс для работы с файлами ввода, вывода и конфигурации | 9 |
| VClient | 12 |
| VClientError | |
| Класс для обработки ошибок клиента | 12 |

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

| | | |
|---------------------------|-------|----|
| cmd.h | | ?? |
| error.h | | ?? |
| rw.h | | ?? |
| sha224.h | | ?? |
| vclient.h | | ?? |

Глава 4

Классы

4.1 Класс CMDParser

Класс для анализа командной строки и получения значений аргументов.

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

Открытые члены

- CMDParser ()=default
Конструктор 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[])
Разобрать аргументы командной строки и задать соответствующие значения.
- void [showHelp](#) () const
Показать сообщение о помощи.

4.1.1 Подробное описание

Класс для анализа командной строки и получения значений аргументов.

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[] )
```

Разобрать аргументы командной строки и задать соответствующие значения.

Аргументы

| | |
|------|------------------------|
| argc | Количество аргументов. |
| argv | Массив аргументов. |

Исключения

| | |
|------------------------------|-------------------------------|
| VClientError | В случае неверного аргумента. |
|------------------------------|-------------------------------|

Объявления и описания членов классов находятся в файлах:

- cmd.h
- cmd.cpp

4.2 Класс ReadWrite

Класс для работы с файлами ввода, вывода и конфигурации.

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

Открытые члены

- [ReadWrite](#) (string &inVec, string &outVec, string &cfg)
Конструктор класса [ReadWrite](#).
- string [getIn](#) ()
Получить путь к файлу входных данных.
- string [getOut](#) ()

- Получить путь к файлу выходных данных.
- `string` `getCfg` ()
- Получить путь к файлу конфигурации.
- `void` `writeVec` (const `vector< uint32_t >` &data)
- Записать данные в файл выходных данных.
- `vector< vector< uint32_t > >` `readVec` ()
- Считать векторы из файла входных данных.
- `array< string, 2 >` `readCfg` ()
- Считать конфигурационные данные из файла конфигурации.

4.2.1 Подробное описание

Класс для работы с файлами ввода, вывода и конфигурации.

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> | Путь к файлу входных данных. |
| <code>outVec</code> | Путь к файлу выходных данных. |
| <code>cfg</code> | Путь к файлу конфигурации. |

4.2.3 Методы

4.2.3.1 getCfg()

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

Получить путь к файлу конфигурации.

Возвращает

Путь к файлу конфигурации.

4.2.3.2 getIn()

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

Получить путь к файлу входных данных.

Возвращает

Путь к файлу входных данных.

4.2.3.3 getOut()

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

Получить путь к файлу выходных данных.

Возвращает

Путь к файлу выходных данных.

4.2.3.4 readCfg()

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

Считать конфигурационные данные из файла конфигурации.

Возвращает

Массив строк, содержащий конфигурационные данные.

4.2.3.5 readVec()

```
vector< vector< uint32_t > > ReadWrite::readVec ( )
```

Считать векторы из файла входных данных.

Возвращает

Вектор векторов, считанных из файла.

4.2.3.6 writeVec()

```
void ReadWrite::writeVec (
    const vector< uint32_t > & data )
```

Записать данные в файл выходных данных.

Аргументы

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

Объявления и описания членов классов находятся в файлах:

- rw.h
- rw.cpp

4.3 Класс VClient

Открытые члены

- VClient (string addr, uint16_t port, array< string, 2 > conf)
- void conn ()
- void stop ()
- void auth ()
- vector< uint32_t > calc (vector< vector< uint32_t > > data)
- int getSock ()
- string getAddr ()
- uint16_t getPort ()
- array< string, 2 > getConf ()

Объявления и описания членов классов находятся в файлах:

- vclient.h
- vclient.cpp

4.4 Класс VClientError

Класс для обработки ошибок клиента.

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

Граф наследования: VClientError:

Глава 5

Файлы

5.1 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
14 class CMDParser {
15 private:
16     string in = "./in.txt";
17     string out = "./out.txt";
18     string cfg = "./cfg.txt";
19     string addr = "127.0.0.1";
20     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 };
```

5.2 error.h

```
1 #pragma once
2
3 #include <string>
4 #include <exception>
5
6 using namespace std;
7
12 class VClientError : public exception
13 {
14 public:
21     VClientError(const string &errorName, const string &errorMsg, const string &funcName);
22
27     const char *what() const noexcept override;
28
33     string getErrorName() const;
34
39     string getErrorMsg() const;
```

```

40
45     string getFuncName() const;
46
47 private:
48     string errorName;
49     string errorMsg;
50     string funcName;
51     mutable string fullMsg;
52 };

```

5.3 rw.h

```

1  #pragma once
2
3  #include <string>
4  #include <vector>
5  #include <array>
6  #include <cstdint>
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<uint32_t> &data);
26
27     vector<vector<uint32_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<uint32_t> &vec);
38
39 void printVectors(const vector<vector<uint32_t>> &vec);

```

5.4 sha224.h

```

1  #pragma once
2
3  #include <string>
4  #include <cryptopp/hex.h>
5  #include <cryptopp/sha.h>
6  #include <cryptopp/osrng.h>
7
8  using namespace std;
9  using namespace CryptoPP;
10
11 string salt();
12
13 string sha224(string &data);

```

5.5 vclient.h

```

1  #pragma once
2
3  #include <cstdint>
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 "sha224.h"
16
17 using namespace std;
18
19 class VClient
20 {
21 private:
22     int sock;
23     string addr;
24     uint16_t port;
25     array<string, 2> conf;
26
27 public:
28     VClient(string addr, uint16_t port, array<string, 2> conf);
29     void conn();
30     void stop();
31     void auth();
32     vector<uint32_t> calc(vector<vector<uint32_t>> data);
33
34     int getSock();
35     string getAddr();
36     uint16_t getPort();
37     array<string, 2> getConf();
38 };
```


Предметный указатель

- CMDParser, [7](#)
 - getAddr, [7](#)
 - getCfg, [8](#)
 - getIn, [8](#)
 - getOut, [8](#)
 - getPort, [8](#)
 - parse, [9](#)
- getAddr
 - CMDParser, [7](#)
- getCfg
 - CMDParser, [8](#)
 - ReadWrite, [10](#)
- getIn
 - CMDParser, [8](#)
 - ReadWrite, [10](#)
- getOut
 - CMDParser, [8](#)
 - ReadWrite, [11](#)
- getPort
 - CMDParser, [8](#)
- parse
 - CMDParser, [9](#)
- readCfg
 - ReadWrite, [11](#)
- readVec
 - ReadWrite, [11](#)
- ReadWrite, [9](#)
 - getCfg, [10](#)
 - getIn, [10](#)
 - getOut, [11](#)
 - readCfg, [11](#)
 - readVec, [11](#)
 - ReadWrite, [10](#)
 - writeVec, [11](#)
- VClient, [12](#)
- VClientError, [12](#)
- writeVec
 - ReadWrite, [11](#)