

Client

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 ErrorHandler< T > Class Template Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 ErrorHandler()	6
3.1.3 Member Function Documentation	6
3.1.3.1 GetArg()	6
3.1.3.2 GetArgValue()	6
3.1.3.3 GetError()	7
3.1.3.4 GetFunc()	7
3.1.4 Friends And Related Function Documentation	7
3.1.4.1 operator<<	7
3.2 Interface Class Reference	7
3.2.1 Detailed Description	8
3.2.2 Member Function Documentation	8
3.2.2.1 GetAutFile()	8
3.2.2.2 GetInputFile()	9
3.2.2.3 GetOutputFile()	9
3.2.2.4 GetReference()	9
3.2.2.5 GetServerAddress()	9
3.2.2.6 GetServerPort()	10
3.2.2.7 ReceiveArguments()	10
3.3 TCPClient Class Reference	10
3.3.1 Detailed Description	11
3.3.2 Constructor & Destructor Documentation	11
3.3.2.1 TCPClient()	11
3.3.3 Member Function Documentation	12
3.3.3.1 ReceiveAndGetResponse()	12
3.3.3.2 ReceiveCalcResult()	12
3.3.3.3 SendAutMsg()	12
3.3.3.4 SendVector()	13
3.3.3.5 SendVectorNumber()	13
3.3.3.6 SendVectorSize()	13
3.4 User Class Reference	13
3.4.1 Detailed Description	15
3.4.2 Constructor & Destructor Documentation	15
3.4.2.1 User()	15

3.4.3 Member Function Documentation	15
3.4.3.1 GetAutFile()	15
3.4.3.2 GetCalcNumber()	16
3.4.3.3 GetCalcResult()	16
3.4.3.4 GetHASH()	16
3.4.3.5 GetId()	16
3.4.3.6 GetInputFile()	17
3.4.3.7 GetOutputFile()	17
3.4.3.8 GetPassword()	17
3.4.3.9 GetSALT()	17
3.4.3.10 GetServerAddress()	18
3.4.3.11 GetServerPort()	18
3.4.3.12 GetVector()	18
3.4.3.13 GetVectorNumber()	18
3.4.3.14 GetVectorSize()	19
3.4.3.15 SetCalcNumber()	19
3.4.3.16 SetCalcResult()	19
3.4.3.17 SetHASH()	19
3.4.3.18 SetIdPassword()	20
3.4.3.19 SetSALT()	20
3.4.3.20 SetVector()	20
3.4.3.21 SetVectorNumber()	20
3.5 UserHandler Class Reference	21
3.5.1 Detailed Description	21
3.5.2 Member Function Documentation	21
3.5.2.1 GetAutDataFromFile()	21
3.5.2.2 GetHashFromPassword()	22
3.5.2.3 GetVectorFromFile()	22
3.5.2.4 GetVectorNumberFromFile()	23
3.5.2.5 RecordCalcNumber()	23
3.5.2.6 RecordCalcResult()	24
4 File Documentation	25
4.1 User.h File Reference	25
4.1.1 Detailed Description	26
Index	27

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ErrorHandler< T >	Class for error handling	5
Interface	Interface realisation. Contacts only the User class	7
TCPClient	TCP class for connecting with server, send to server information and receive information from server. Contacts only the User class	10
User	User class for storing large number of attributes	13
UserHandler	UserHandler class for handling user information. Contacts only the User class	21

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

User.h	Header file for client-programm	25
------------------------	---	--------------------

Chapter 3

Class Documentation

3.1 ErrorHandler< T > Class Template Reference

Class for error handling.

```
#include <User.h>
```

Public Member Functions

- [ErrorHandler](#) (string what_error, string what_arg, const T &arg_value, string what_func)
Constructor for initialization.
- string [GetError](#) ()
Getting error name.
- string [GetArg](#) ()
Getting the variable name.
- T [GetArgValue](#) ()
Getting the variable value.
- string [GetFunc](#) ()
Getting The function name.

Friends

- ostream & [operator<<](#) (ostream &os, [ErrorHandler](#) &er)
Overloaded operator<<.

3.1.1 Detailed Description

```
template<typename T>  
class ErrorHandler< T >
```

Class for error handling.

Is a template class. [ErrorHandler](#) have immutable initialization. Contains methods: Constructor for output information about error, getters for returning class attributes and friendly overloaded operator<<.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 ErrorHandler()

```
template<typename T >
ErrorHandler< T >::ErrorHandler (
    string what_error,
    string what_arg,
    const T & arg_value,
    string what_func ) [inline], [explicit]
```

Constuctor for initialization.

Parameters

<i>what_error</i>	Error name.
<i>what_arg</i>	The name variable which contains error.
<i>arg_value</i>	The variable value. It is template attribute.
<i>what_func</i>	The name function which contains error.

Warning

arg_value variable must be a data type that can be force to a string.

3.1.3 Member Function Documentation

3.1.3.1 GetArg()

```
template<typename T >
string ErrorHandler< T >::GetArg ( ) [inline]
```

Getting the variable name.

Returns

The name variable which contains error as string.

3.1.3.2 GetArgValue()

```
template<typename T >
T ErrorHandler< T >::GetArgValue ( ) [inline]
```

Getting the variable value.

Returns

The variable value as string.

3.1.3.3 GetError()

```
template<typename T >
string ErrorHandler< T >::GetError ( ) [inline]
```

Getting error name.

Returns

Error name as string.

3.1.3.4 GetFunc()

```
template<typename T >
string ErrorHandler< T >::GetFunc ( ) [inline]
```

Getting The function name.

Returns

The function name as string.

3.1.4 Friends And Related Function Documentation

3.1.4.1 operator<<

```
template<typename T >
ostream& operator<< (
    ostream & os,
    ErrorHandler< T > & er ) [friend]
```

Overloaded operator<<.

Returns

Output all getters.

The documentation for this class was generated from the following file:

- [User.h](#)

3.2 Interface Class Reference

[Interface](#) realisation. Contacts only the [User](#) class.

```
#include <User.h>
```

Public Member Functions

- int [GetReference](#) ()
Getting the reference.
- int [ReceiveArguments](#) (int argc, char *argv[])
Passing arguments to the program when the executable file is called.
- string [GetServerAddress](#) ()
Getting the server address.
- string [GetServerPort](#) ()
Getting the server port.
- string [GetInputFile](#) ()
Getting the input file.
- string [GetOutputFile](#) ()
Getting the output file.
- string [GetAutFile](#) ()
Getting the autentification file.

3.2.1 Detailed Description

[Interface](#) realisation. Contacts only the [User](#) class.

Attributes are set in method [ReceiveArguments\(\)](#) by passing arguments to the program when the executable file is called. Contains methods: [GetReference](#) for call reference, which set in method [ReceiveArguments\(\)](#); [ReceiveArguments](#) for passing argument in interface and getters for returning class attributes.

Warning

Strictly follow the instructions in the help! Without this class you will not be able to continue work with this program.

3.2.2 Member Function Documentation

3.2.2.1 [GetAutFile\(\)](#)

```
string Interface::GetAutFile ( )
```

Getting the autentification file.

Returns

A absolute path to autentification file as a string.

3.2.2.2 GetInputFile()

```
string Interface::GetInputFile ( )
```

Getting the input file.

Returns

A absolute path to input file as a string.

3.2.2.3 GetOutputFile()

```
string Interface::GetOutputFile ( )
```

Getting the output file.

Returns

A absolute path to output file as a string.

3.2.2.4 GetReference()

```
int Interface::GetReference ( )
```

Getting the reference.

Returns

The reference.

3.2.2.5 GetServerAddress()

```
string Interface::GetServerAddress ( )
```

Getting the server address.

Returns

A server address as a string.

3.2.2.6 GetServerPort()

```
string Interface::GetServerPort ( )
```

Getting the server port.

Returns

A server port as a string.

3.2.2.7 ReceiveArguments()

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

Passing arguments to the program when the executable file is called.

Parameters

<i>argc</i>	Count of arguments which passing to main.
<i>argv</i>	Container with arguments which passing to main.

Warning

Throw ErrorHandler<string> when it finds an argument transfer error or an unsurrected argument format.

Returns

0

The documentation for this class was generated from the following file:

- [User.h](#)

3.3 TCPclient Class Reference

TCP class for connecting with server, send to server information and receive information from server. Contacts only the [User](#) class.

```
#include <User.h>
```

Public Member Functions

- [TCPclient](#) ()=delete
Delete the default constructor.
- [TCPclient](#) ([User](#) user)
Constructor for initialization.
- int [Connection](#) ()
Setup connection with server using server address and server port from [User](#) class(see constructor).
- int [SendAutMsg](#) (string aut_msg)
Send authentication string to server.
- int [SendVectorNumber](#) (uint32_t vector_number)
Send number of vectors.
- int [SendVectorSize](#) (uint32_t vector_size)
Send size of vector.
- int [SendVector](#) (vector< float > vectorvr)
Send vector.
- string [ReceiveAndGetResponse](#) ()
Receive and get response from server.
- float [ReceiveCalcResult](#) ()
Receive calculation result from server.
- int [CloseConnection](#) ()
Close connection with server.

3.3.1 Detailed Description

TCP class for connecting with server, send to server information and receive information from server. Contacts only the [User](#) class.

Storing server address, server port, socket and connection. [TCPclient](#) does not have a default constructor, and you must pass an [User](#) class object to it to create an object of this class. Contains methods: Constructor for initialization, method for connection setup, 4 methods for sending information to server, 2 methods for receive information from server and method for close connection.

Warning

Without this class you can not contacts with server.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 TCPclient()

```
TCPclient::TCPclient (
    User user )
```

Constructor for initialization.

Parameters

<i>user</i>	Object of class User . In this constructor attributes are initialized through server address and server port from User class(see User class).
-------------	---

3.3.3 Member Function Documentation

3.3.3.1 ReceiveAndGetResponse()

```
string TCPclient::ReceiveAndGetResponse ( )
```

Receive and get response from server.

Returns

Server response as string.

3.3.3.2 ReceiveCalcResult()

```
float TCPclient::ReceiveCalcResult ( )
```

Receive calculation result from server.

Returns

Calculation result as float variable.

3.3.3.3 SendAutMsg()

```
int TCPclient::SendAutMsg (
    string aut_msg )
```

Send authentication string to server.

Parameters

<i>aut_msg</i>	String that will be send to server.
----------------	-------------------------------------

3.3.3.4 SendVector()

```
int TCPclient::SendVector (
    vector< float > vectorvr )
```

Send vector.

Parameters

<i>vectorvr</i>	Data type vector float variable that storing elements which to be sent to the server.
-----------------	---

3.3.3.5 SendVectorNumber()

```
int TCPclient::SendVectorNumber (
    uint32_t vector_number )
```

Send number of vectors.

Parameters

<i>vector_number</i>	Uint32_t data type variable that contains number of vectors to be sent to the server.
----------------------	---

3.3.3.6 SendVectorSize()

```
int TCPclient::SendVectorSize (
    uint32_t vector_size )
```

Send size of vector.

Parameters

<i>vector_size</i>	Uint32_t data type variable that contains number of vector's elements.
--------------------	--

The documentation for this class was generated from the following file:

- [User.h](#)

3.4 User Class Reference

[User](#) class for storing large number of attributes.

```
#include <User.h>
```

Public Member Functions

- [User](#) ()=delete
Delete the default constructor.
- [User](#) ([Interface](#) interface)
Constructor for initialization.
- string [GetInputFile](#) ()
Getting the output file.
- string [GetOutputFile](#) ()
Getting the output file.
- string [GetAutFile](#) ()
Getting the autentification file.
- string [GetServerAddress](#) ()
Getting the server address.
- string [GetServerPort](#) ()
Getting the input file.
- string [GetId](#) ()
Getting the user id.
- string [GetPassword](#) ()
Getting the user password.
- string [GetSALT](#) ()
Getting the salt for md5 hash.
- string [GetHASH](#) ()
Getting the md5 hash.
- uint32_t [GetVectorNumber](#) ()
Getting the vector number.
- uint32_t [GetVectorSize](#) ()
Getting the vector size.
- vector< float > [GetVector](#) ()
Getting the vector.
- uint32_t [GetCalcNumber](#) ()
Getting the calculation number.
- float [GetCalcResult](#) ()
Getting the calculation result.
- int [SetIdPassword](#) (vector< string > aut_data)
Setting user id and user password.
- int [SetSALT](#) (string SALT)
Setting salt.
- int [SetHASH](#) (string HASH)
Setting hash.
- int [SetVectorNumber](#) (uint32_t vector_number)
Setting vector number.
- int [SetVector](#) (vector< float > vector)
Setting vector.
- int [SetCalcNumber](#) (uint32_t calculation_number)
Setting calculation number.
- int [SetCalcResult](#) (float calculation_result)
Setting calculation result.

3.4.1 Detailed Description

[User](#) class for storing large number of attributes.

[User](#) does not have a default constructor, and you must pass an [Interface](#) class object to it to create an object of this class. Contains methods: Constructor for initialization and large number of getters and setters.

Warning

Without this class you will not be able to work with [UserHandler](#) and [TCPclient](#) classes.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 User()

```
User::User (
    Interface interface )
```

Constructor for initialization.

Parameters

<i>interface</i>	Object of class Interface . In this constructor attributes are initialized through all getters of the Interface class(see Interface class).
------------------	---

3.4.3 Member Function Documentation

3.4.3.1 GetAutFile()

```
string User::GetAutFile ( )
```

Getting the autentification file.

Returns

A absolute path to autentification file as a string.

3.4.3.2 GetCalcNumber()

```
uint32_t User::GetCalcNumber ( )
```

Getting the calculation number.

Returns

The calculation number as uint32_t variable.

3.4.3.3 GetCalcResult()

```
float User::GetCalcResult ( )
```

Getting the calculation result.

Returns

The calculation result as float variable.

3.4.3.4 GetHASH()

```
string User::GetHASH ( )
```

Getting the md5 hash.

Returns

The hash as a hex numbers string.

3.4.3.5 GetId()

```
string User::GetId ( )
```

Getting the user id.

Returns

The user id as a string.

3.4.3.6 GetInputFile()

```
string User::GetInputFile ( )
```

Getting the output file.

Returns

A absolute path to output file as a string.

3.4.3.7 GetOutputFile()

```
string User::GetOutputFile ( )
```

Getting the output file.

Returns

A absolute path to output file as a string.

3.4.3.8 GetPassword()

```
string User::GetPassword ( )
```

Getting the user password.

Returns

The user password as a string.

3.4.3.9 GetSALT()

```
string User::GetSALT ( )
```

Getting the salt for md5 hash.

Returns

The salt as a hex numbers string.

3.4.3.10 GetServerAddress()

```
string User::GetServerAddress ( )
```

Getting the server address.

Returns

A server address as a string.

3.4.3.11 GetServerPort()

```
string User::GetServerPort ( )
```

Getting the input file.

Returns

A absolute path to input file as a string.

3.4.3.12 GetVector()

```
vector<float> User::GetVector ( )
```

Getting the vector.

Returns

The vector as container float data.

3.4.3.13 GetVectorNumber()

```
uint32_t User::GetVectorNumber ( )
```

Getting the vector number.

Returns

The vector number as uint32_t variable.

3.4.3.14 GetVectorSize()

```
uint32_t User::GetVectorSize ( )
```

Getting the vector size.

Returns

The vector size as uint32_t variable.

3.4.3.15 SetCalcNumber()

```
int User::SetCalcNumber (
    uint32_t calculation_number )
```

Setting calculation number.

Parameters

<i>calculation_number</i>	Uint32_t data type variable that contains number of result to be record in output file.
---------------------------	---

3.4.3.16 SetCalcResult()

```
int User::SetCalcResult (
    float calculation_result )
```

Setting calculation result.

Parameters

<i>calculation_result</i>	Float data type variable that contains the sum of vector elements sent to the server.
---------------------------	---

3.4.3.17 SetHASH()

```
int User::SetHASH (
    string HASH )
```

Setting hash.

Parameters

<i>HASH</i>	Hex number string.
-------------	--------------------

3.4.3.18 SetIdPassword()

```
int User::SetIdPassword (
    vector< string > aut_data )
```

Setting user id and user password.

Parameters

<i>aut_data</i>	Vector which contains id and password for user.
-----------------	---

3.4.3.19 SetSALT()

```
int User::SetSALT (
    string SALT )
```

Setting salt.

Parameters

<i>SALT</i>	Hex number string.
-------------	--------------------

3.4.3.20 SetVector()

```
int User::SetVector (
    vector< float > vector )
```

Setting vector.

Parameters

<i>vector</i>	Data type vector float variable that storing elements which to be sent to the server.
---------------	---

3.4.3.21 SetVectorNumber()

```
int User::SetVectorNumber (
    uint32_t vector_number )
```

Setting vector number.

Parameters

<code>vector_number</code>	UInt32_t data type variable that contains number of vectors to be sent to the server.
----------------------------	---

The documentation for this class was generated from the following file:

- [User.h](#)

3.5 UserHandler Class Reference

[UserHandler](#) class for handling user information. Contacts only the [User](#) class.

```
#include <User.h>
```

Public Member Functions

- `uint32_t` [GetVectorNumberFromFile](#) (string input_data_file)
Get number of vector from file.
- `vector< float >` [GetVectorFromFile](#) (string input_data_file, int string_number)
Get vector from file.
- `vector< string >` [GetAutDataFromFile](#) (string aut_data_file)
Get autentification data from file. This is id and password.
- `string` [GetHashFromPassword](#) (string Salt, string user_password)
Create and get hash from salt + password.
- `int` [RecordCalcNumber](#) (uint32_t calculations_number, string output_data_file)
Record calculations number in binary file.
- `int` [RecordCalcResult](#) (float calculation_result, string output_data_file)
Record calculations result in binary file.

3.5.1 Detailed Description

[UserHandler](#) class for handling user information. Contacts only the [User](#) class.

Abstract class. Methods handling information and return it. Contains methods: 3 getters for handling information from files, 1 getters for creating hash and 2 record methods for record data in binary file.

Warning

Without this class [User](#) class does not matter.

3.5.2 Member Function Documentation

3.5.2.1 GetAutDataFromFile()

```
vector<string> UserHandler::GetAutDataFromFile (
    string aut_data_file )
```

Get autentification data from file. This is id and password.

Parameters

<i>aut_data_file</i>	File with contents id and password.
----------------------	-------------------------------------

Warning

throw ErrorHandler<string> when the autentification file is incorrect or its contents are incorrect.

Returns

String vector with 2 cells, first id, second password.

3.5.2.2 GetHashFromPassword()

```
string UserHandler::GetHashFromPassword (
    string Salt,
    string user_password )
```

Create and get hash from salt + password.

Parameters

<i>Salt</i>	Hex string number. Must contains only hex numbers.
<i>user_password</i>	String with password.

Warning

throw ErrorHandler<string> when salt string having not hex numbers, salt is empty or password is empty.

Returns

Hash as string of hex numbers.

3.5.2.3 GetVectorFromFile()

```
vector<float> UserHandler::GetVectorFromFile (
    string input_data_file,
    int string_number )
```

Get vector from file.

Parameters

<i>input_data_file</i>	File with number of vector.
<i>string_number</i>	String number for reading vector.

Warning

throw ErrorHandler<string> when the input file is incorrect or its contents are incorrect.

Returns

Data type vector float.

3.5.2.4 GetVectorNumberFromFile()

```
uint32_t UserHandler::GetVectorNumberFromFile (
    string input_data_file )
```

Get number of vector from file.

Parameters

<i>input_data_file</i>	File with number of vector.
------------------------	-----------------------------

Warning

throw ErrorHandler<string> when the input file is incorrect or its contents are incorrect.

Returns

A number of vector.

3.5.2.5 RecordCalcNumber()

```
int UserHandler::RecordCalcNumber (
    uint32_t calculations_number,
    string output_data_file )
```

Record calculations number in binary file.

Parameters

<i>calculations_number</i>	UInt32_t variable that contains number of calculations.
<i>output_data_file</i>	File for record.

Warning

throw ErrorHandler<string> when the output file is incorrect.

3.5.2.6 RecordCalcResult()

```
int UserHandler::RecordCalcResult (
    float calculation_result,
    string output_data_file )
```

Record calculations result in binary file.

Parameters

<i>calculations_result</i>	Float variable that contains the sum of vector elements sent to the server.
<i>output_data_file</i>	File for record.

Warning

throw ErrorHandler<string> when the output file is incorrect.

The documentation for this class was generated from the following file:

- [User.h](#)

Chapter 4

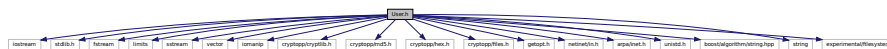
File Documentation

4.1 User.h File Reference

Header file for client-programm.

```
#include <iostream>
#include <stdlib.h>
#include <fstream>
#include <limits>
#include <sstream>
#include <vector>
#include <iomanip>
#include <cryptopp/cryptlib.h>
#include <cryptopp/md5.h>
#include <cryptopp/hex.h>
#include <cryptopp/files.h>
#include <getopt.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <boost/algorithm/string.hpp>
#include <string>
#include <experimental/filesystem>
```

Include dependency graph for User.h:



Classes

- class [Interface](#)
Interface realisation. Contacts only the [User](#) class.
- class [ErrorHandler< T >](#)
Class for error handling.
- class [User](#)
User class for storing large number of attributes.
- class [UserHandler](#)
UserHandler class for handling user information. Contacts only the [User](#) class.
- class [TCPclient](#)
TCP class for connecting with server, send to server information and receive information from server. Contacts only the [User](#) class.

Macros

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

4.1.1 Detailed Description

Header file for client-programm.

Author

Galkin Kirill

Version

Release 1.0.0

Date

26.12.2022

Copyright

K.A.

Index

- ErrorHandler
 - ErrorHandler< T >, 6
- ErrorHandler< T >, 5
 - ErrorHandler, 6
 - GetArg, 6
 - GetArgValue, 6
 - GetError, 6
 - GetFunc, 7
 - operator<<, 7
- GetArg
 - ErrorHandler< T >, 6
- GetArgValue
 - ErrorHandler< T >, 6
- GetAutDataFromFile
 - UserHandler, 21
- GetAutFile
 - Interface, 8
 - User, 15
- GetCalcNumber
 - User, 15
- GetCalcResult
 - User, 16
- GetError
 - ErrorHandler< T >, 6
- GetFunc
 - ErrorHandler< T >, 7
- GetHASH
 - User, 16
- GetHashFromPassword
 - UserHandler, 22
- GetId
 - User, 16
- GetInputFile
 - Interface, 8
 - User, 16
- GetOutputFile
 - Interface, 9
 - User, 17
- GetPassword
 - User, 17
- GetReference
 - Interface, 9
- GetSALT
 - User, 17
- GetServerAddress
 - Interface, 9
 - User, 17
- GetServerPort
 - Interface, 9
 - User, 18
- GetVector
 - User, 18
- GetVectorFromFile
 - UserHandler, 22
- GetVectorNumber
 - User, 18
- GetVectorNumberFromFile
 - UserHandler, 23
- GetVectorSize
 - User, 18
- Interface, 7
 - GetAutFile, 8
 - GetInputFile, 8
 - GetOutputFile, 9
 - GetReference, 9
 - GetServerAddress, 9
 - GetServerPort, 9
 - ReceiveArguments, 10
- operator<<
 - ErrorHandler< T >, 7
- ReceiveAndGetResponse
 - TCPclient, 12
- ReceiveArguments
 - Interface, 10
- ReceiveCalcResult
 - TCPclient, 12
- RecordCalcNumber
 - UserHandler, 23
- RecordCalcResult
 - UserHandler, 23
- SendAutMsg
 - TCPclient, 12
- SendVector
 - TCPclient, 12
- SendVectorNumber
 - TCPclient, 13
- SendVectorSize
 - TCPclient, 13
- SetCalcNumber
 - User, 19
- SetCalcResult
 - User, 19
- SetHASH
 - User, 19
- SetIdPassword

- User, [20](#)
- SetSALT
 - User, [20](#)
- SetVector
 - User, [20](#)
- SetVectorNumber
 - User, [20](#)
- TCPclient, [10](#)
 - ReceiveAndGetResponce, [12](#)
 - ReceiveCalcResult, [12](#)
 - SendAutMsg, [12](#)
 - SendVector, [12](#)
 - SendVectorNumber, [13](#)
 - SendVectorSize, [13](#)
 - TCPclient, [11](#)
- User, [13](#)
 - GetAutFile, [15](#)
 - GetCalcNumber, [15](#)
 - GetCalcResult, [16](#)
 - GetHASH, [16](#)
 - GetId, [16](#)
 - GetInputFile, [16](#)
 - GetOutputFile, [17](#)
 - GetPassword, [17](#)
 - GetSALT, [17](#)
 - GetServerAddress, [17](#)
 - GetServerPort, [18](#)
 - GetVector, [18](#)
 - GetVectorNumber, [18](#)
 - GetVectorSize, [18](#)
 - SetCalcNumber, [19](#)
 - SetCalcResult, [19](#)
 - SetHASH, [19](#)
 - SetIdPassword, [20](#)
 - SetSALT, [20](#)
 - SetVector, [20](#)
 - SetVectorNumber, [20](#)
 - User, [15](#)
- User.h, [25](#)
- UserHandler, [21](#)
 - GetAutDataFromFile, [21](#)
 - GetHashFromPassword, [22](#)
 - GetVectorFromFile, [22](#)
 - GetVectorNumberFromFile, [23](#)
 - RecordCalcNumber, [23](#)
 - RecordCalcResult, [23](#)