

GUIClient

Wygenerowano przez Doxygen 1.8.15

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Dokumentacja klas	5
3.1 Dokumentacja klasy ChatBox	5
3.1.1 Dokumentacja konstruktora i destruktora	6
3.1.1.1 ChatBox()	6
3.1.1.2 ~ChatBox()	6
3.1.2 Dokumentacja funkcji składowych	6
3.1.2.1 sendwithline	6
3.1.3 Dokumentacja atrybutów składowych	6
3.1.3.1 cls	6
3.1.3.2 clx	7
3.1.3.3 con	7
3.1.3.4 dis	7
3.1.3.5 list	7
3.1.3.6 ui	7
3.2 Dokumentacja klasy GUIClient	7
3.2.1 Dokumentacja konstruktora i destruktora	8
3.2.1.1 GUIClient()	8
3.2.1.2 ~GUIClient()	8
3.2.2 Dokumentacja funkcji składowych	9
3.2.2.1 ok	9
3.2.3 Dokumentacja atrybutów składowych	9
3.2.3.1 chat	9
3.2.3.2 client	9
3.2.3.3 clt	9
3.2.3.4 ipad	9
3.2.3.5 nick	9
3.2.3.6 port	10
3.2.3.7 ui	10
3.3 Dokumentacja klasy Sck	10
3.3.1 Opis szczegółowy	10
3.4 Dokumentacja klasy TcpClient	11
3.4.1 Dokumentacja konstruktora i destruktora	11
3.4.1.1 TcpClient()	11
3.4.1.2 ~TcpClient()	12
3.4.2 Dokumentacja funkcji składowych	12
3.4.2.1 Cleanup()	12
3.4.2.2 ConnectToSerwer()	12

3.4.2.3 conntoserv()	13
3.4.2.4 CreateSocket()	13
3.4.2.5 Initialize()	13
3.4.2.6 msgrec()	13
3.4.2.7 SendAll()	14
3.4.2.8 showmsgrec()	14
3.4.2.9 ShowName()	14
3.4.2.10 sizerec()	15
3.4.3 Dokumentacja atrybutów składowych	15
3.4.3.1 m_ipAddress	15
3.4.3.2 m_port	15
3.4.3.3 messages	15
3.4.3.4 mtx	15
3.4.3.5 name	15
Indeks	17

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

QDialog	
GUIClient	7
QWidget	
ChatBox	5
Sck	10
TcpClient	11

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

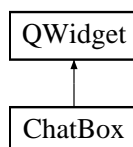
ChatBox	5
GUIClient	7
Sck	10
TcpClient	11

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy ChatBox

Diagram dziedziczenia dla ChatBox



Metody publiczne

- `ChatBox` (`QWidget *parent=Q_NULLPTR`, `TcpClient *cl=nullptr`, `SOCKET add=0`)
- `~ChatBox` ()

Atrybuty publiczne

- `TcpClient * clx`
- `SOCKET con`

Sloty prywatne

- `void sendwithline` ()

Atrybuty prywatne

- `Ui::ChatBox ui`
- `bool cls = true`
- `std::thread list`
- `std::thread dis`

3.1.1 Dokumentacja konstruktora i destruktora

3.1.1.1 ChatBox()

```
ChatBox::ChatBox (
    QWidget * parent = Q_NULLPTR,
    TcpClient * cl = nullptr,
    SOCKET add = 0 )
```

Konstruktor inicjalizujący powstanie okna dialogowego.

Parametry

<i>parent</i>	Wskaźnik na przekazywany widget.
<i>cl</i>	Wskaźnik na klasę użytkownika.
<i>add</i>	Gniazdo połączone z serwerem.

3.1.1.2 ~ChatBox()

```
ChatBox::~~ChatBox ( )
```

Destruktor okna czatowego.

3.1.2 Dokumentacja funkcji składowych

3.1.2.1 sendwithline

```
void ChatBox::sendwithline ( ) [private], [slot]
```

Slot odpowiedzialny za wysyłanie wiadomości.

3.1.3 Dokumentacja atrybutów składowych

3.1.3.1 cls

```
bool ChatBox::cls = true [private]
```

Flaga zarządzająca działaniem wątku.

3.1.3.2 clx

```
TcpClient* ChatBox::clx
```

Wskaźnik na klasę [TcpClient](#).

3.1.3.3 con

```
SOCKET ChatBox::con
```

Zmienna przechowująca gniazdo połączone z serwerem.

3.1.3.4 dis

```
std::thread ChatBox::dis [private]
```

Wątek służący do wyświetlania odebranych wiadomości.

3.1.3.5 list

```
std::thread ChatBox::list [private]
```

Wątek służący do odbierania wiadomości.

3.1.3.6 ui

```
Ui::ChatBox ChatBox::ui [private]
```

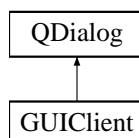
Inicjalizacja obiektu [ChatBox](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- GUIClient/chatbox.hpp
- GUIClient/chatbox.cpp

3.2 Dokumentacja klasy GUIClient

Diagram dziedziczenia dla GUIClient



Metody publiczne

- [GUIClient](#) (QWidget *parent=0)
- [~GUIClient](#) ()

Atrybuty chronione

- SOCKET [clt](#)
- TcpClient * [client](#) = nullptr
- ChatBox * [chat](#) =nullptr

Sloty prywatne

- void [ok](#) ()

Atrybuty prywatne

- Ui::GUIClientClass [ui](#)
- std::string [ipad](#)
- int [port](#)
- std::string [nick](#)

3.2.1 Dokumentacja konstruktora i destruktora

3.2.1.1 GUIClient()

```
GUIClient::GUIClient (
    QWidget * parent = 0 )
```

Konstruktor inicjalizujący powstanie okna dialogowego.

Parametry

<i>parent</i>	Wskaźnik na przekazywany widget.
---------------	----------------------------------

3.2.1.2 ~GUIClient()

```
GUIClient::~~GUIClient ( )
```

Destruktor okna dialogowego.

3.2.2 Dokumentacja funkcji składowych

3.2.2.1 ok

```
void GUIClient::ok ( ) [private], [slot]
```

Slot odpowiedzialny za wywołanie funkcji, aby dołączyć do danego czatu.

3.2.3 Dokumentacja atrybutów składowych

3.2.3.1 chat

```
ChatBox* GUIClient::chat =nullptr [protected]
```

Wskaźnik na obiekt [ChatBox](#).

3.2.3.2 client

```
TcpClient* GUIClient::client = nullptr [protected]
```

Wskaźnik na obiekt [TcpClient](#), czyli użytkownika.

3.2.3.3 clt

```
SOCKET GUIClient::clt [protected]
```

Gniazdo połączone z serwerem.

3.2.3.4 ipad

```
std::string GUIClient::ipad [private]
```

Zmienna przechowująca adres IP serwera.

3.2.3.5 nick

```
std::string GUIClient::nick [private]
```

Zmienna przechowująca nazwę użytkownika.

3.2.3.6 port

```
int GUIClient::port [private]
```

Zmienna przechowująca port serwera.

3.2.3.7 ui

```
Ui::GUIClientClass GUIClient::ui [private]
```

Inicjalizacja obiektu GuiClient.

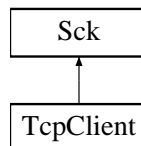
Dokumentacja dla tej klasy została wygenerowana z plików:

- GUIClient/guiclient.h
- GUIClient/guiclient.cpp

3.3 Dokumentacja klasy Sck

```
#include <Sck.h>
```

Diagram dziedziczenia dla Sck



Metody publiczne

- virtual SOCKET [CreateSocket](#) ()=0
destruktor klasy wirtualnej SCK
- virtual void [Cleanup](#) ()=0
wirtualna metoda [CreateSocket\(\)](#)
- virtual bool [Initialize](#) ()=0
wirtualna metoda [Cleanup\(\)](#)

3.3.1 Opis szczegółowy

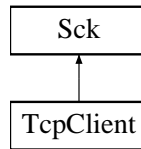
Wirtualna klasa bazowa klasy TcpListener(serwer) oraz klasy [TcpClient](#)(użytkownik klienta)

Dokumentacja dla tej klasy została wygenerowana z pliku:

- GUIClient/Sck.h

3.4 Dokumentacja klasy TcpClient

Diagram dziedziczenia dla TcpClient



Metody publiczne

- `TcpClient` (const std::string &ipAD, const int &port, const std::string &nick)
- `~TcpClient` ()
- int `SendAll` (int client, const char *buf, int *len)
- bool `Initialize` () override
- SOCKET `conntoserv` ()
- std::string `ShowName` ()
- int `sizerec` ()
- std::string `showmsgrec` ()
- void `msgrec` (const SOCKET &client)
- void `Cleanup` () override

Atrybuty publiczne

- std::mutex `mtx`

Metody prywatne

- SOCKET `CreateSocket` () override
- SOCKET `ConnectToSerwer` (const SOCKET &conn)

Atrybuty prywatne

- std::string `name`
- std::string `m_ipAddress`
- int `m_port`
- std::deque< std::string > `messages`

3.4.1 Dokumentacja konstruktora i destruktor

3.4.1.1 TcpClient()

```
TcpClient::TcpClient (  
    const std::string & ipAD,  
    const int & port,  
    const std::string & nick ) [inline]
```

Konstruktor inicjalizujący powstanie klasy `TcpClient`.

Parametry

<i>ipAddress</i>	Adres IP serwera.
<i>port</i>	Port serwera.
<i>nick</i>	Nazwa użytkownika.

3.4.1.2 ~TcpClient()

```
TcpClient::~~TcpClient ( ) [inline]
```

Destruktor klasy [TcpClient](#).

3.4.2 Dokumentacja funkcji składowych**3.4.2.1 Cleanup()**

```
void TcpClient::Cleanup ( ) [override], [virtual]
```

Metoda wirtualna sprzątająca, czyli usuwanie i czyszczenie zaalokowanych pamięci.

Implementuje [Sck](#).

3.4.2.2 ConnectToSerwer()

```
SOCKET TcpClient::ConnectToSerwer (
    const SOCKET & conn ) [private]
```

Metoda umożliwiająca połączenie z serwerem.

Parametry

<i>conn</i>	Gniazdo utworzone przez użytkownika.
-------------	--------------------------------------

Zwraca

Gniazdo, które zostało przyjęte przez serwer.

3.4.2.3 conntoserv()

```
SOCKET TcpClient::conntoserv ( )
```

Metoda służąca do uruchomienia klienta i łączenie go z klientem.

Zwraca

Gniazdo połączone z serwerem.

3.4.2.4 CreateSocket()

```
SOCKET TcpClient::CreateSocket ( ) [override], [private], [virtual]
```

Metoda wirtualna inicjalizująca powstanie gniazda nasłuchującego.

Zwraca

Gniazdo stworzone na podstawie parametrów w klasie.

Implementuje [Sck](#).

3.4.2.5 Initialize()

```
bool TcpClient::Initialize ( ) [override], [virtual]
```

Metoda utrzymująca działanie serwera i komunikację między użytkownikami

Zwraca

Zwraca true jeśli się powiedzie, w przeciwnym false

Implementuje [Sck](#).

3.4.2.6 msgrec()

```
void TcpClient::msgrec (
    const SOCKET & client )
```

Metoda służąca do odbierania wiadomości przez użytkownika.

Parametry

<i>client</i>	Gniazdo użytkownika połączone z serwerem.
---------------	---

3.4.2.7 SendAll()

```
int TcpClient::SendAll (
    int client,
    const char * buf,
    int * len )
```

Metoda służąca do wysyłania wiadomości.

Parametry

<i>client</i>	Gniazdo użytkownika.
<i>buf</i>	Treść wiadomości wysyłanej.
<i>len</i>	Długość wiadomości.

Zwraca

Wartość, która jest flagą poprawnego wysłania(0) lub nie(-1).

3.4.2.8 showmsgrec()

```
std::string TcpClient::showmsgrec ( )
```

Metoda zwracającą wiadomość odebraną.

Zwraca

Treść wiadomości odebranej od serwera.

3.4.2.9 ShowName()

```
std::string TcpClient::ShowName ( )
```

Metoda pozwalająca przekazanie nazwy użytkownika

Zwraca

Zwraca nazwę użytkownika.

3.4.2.10 sizerec()

```
int TcpClient::sizerec ( )
```

Metoda zwracająca wielkość kolejki.

Zwraca

Wielkość kolejki priorytetowej.

3.4.3 Dokumentacja atrybutów składowych

3.4.3.1 m_ipAddress

```
std::string TcpClient::m_ipAddress [private]
```

Zmienna przechowująca adres IP serwera.

3.4.3.2 m_port

```
int TcpClient::m_port [private]
```

Zmienna przechowująca port serwera.

3.4.3.3 messages

```
std::deque<std::string> TcpClient::messages [private]
```

Kolejka priorytetowa przechowująca wiadomości odebrane.

3.4.3.4 mtx

```
std::mutex TcpClient::mtx
```

Zatrzymuje pracę wątku.

3.4.3.5 name

```
std::string TcpClient::name [private]
```

Zmienna przechowująca nazwę użytkownika.

Dokumentacja dla tej klasy została wygenerowana z plików:

- GUIClient/TcpClient.h
- GUIClient/TcpClient.cpp

Indeks

- ~ChatBox
 - ChatBox, [6](#)
- ~GUIClient
 - GUIClient, [8](#)
- ~TcpClient
 - TcpClient, [12](#)
- chat
 - GUIClient, [9](#)
- ChatBox, [5](#)
 - ~ChatBox, [6](#)
 - ChatBox, [6](#)
 - cls, [6](#)
 - clx, [6](#)
 - con, [7](#)
 - dis, [7](#)
 - list, [7](#)
 - sendwithline, [6](#)
 - ui, [7](#)
- Cleanup
 - TcpClient, [12](#)
- client
 - GUIClient, [9](#)
- cls
 - ChatBox, [6](#)
- clt
 - GUIClient, [9](#)
- clx
 - ChatBox, [6](#)
- con
 - ChatBox, [7](#)
- ConnectToSerwer
 - TcpClient, [12](#)
- conntoserv
 - TcpClient, [12](#)
- CreateSocket
 - TcpClient, [13](#)
- dis
 - ChatBox, [7](#)
- GUIClient, [7](#)
 - ~GUIClient, [8](#)
 - chat, [9](#)
 - client, [9](#)
 - clt, [9](#)
 - GUIClient, [8](#)
 - ipad, [9](#)
 - nick, [9](#)
 - ok, [9](#)
 - port, [9](#)
 - ui, [10](#)
- Initialize
 - TcpClient, [13](#)
- ipad
 - GUIClient, [9](#)
- list
 - ChatBox, [7](#)
- m_ipAddress
 - TcpClient, [15](#)
- m_port
 - TcpClient, [15](#)
- messages
 - TcpClient, [15](#)
- msgrec
 - TcpClient, [13](#)
- mtx
 - TcpClient, [15](#)
- name
 - TcpClient, [15](#)
- nick
 - GUIClient, [9](#)
- ok
 - GUIClient, [9](#)
- port
 - GUIClient, [9](#)
- Sck, [10](#)
- SendAll
 - TcpClient, [14](#)
- sendwithline
 - ChatBox, [6](#)
- showmsgrec
 - TcpClient, [14](#)
- ShowName
 - TcpClient, [14](#)
- sizerec
 - TcpClient, [14](#)
- TcpClient, [11](#)
 - ~TcpClient, [12](#)
 - Cleanup, [12](#)
 - ConnectToSerwer, [12](#)
 - conntoserv, [12](#)
 - CreateSocket, [13](#)

Initialize, [13](#)
m_ipAddress, [15](#)
m_port, [15](#)
messages, [15](#)
msgrec, [13](#)
mtx, [15](#)
name, [15](#)
SendAll, [14](#)
showmsgrec, [14](#)
ShowName, [14](#)
sizerec, [14](#)
TcpClient, [11](#)

ui

ChatBox, [7](#)
GUIClient, [10](#)