Serwer

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 checking 5

3.1.1 Opis szczegółowy 5

3.1.2 Dokumentacja funkcji składowych 5

3.1.2.1 checkingdigit() 5

3.1.2.2 CheckParams() 6

3.1.2.3 Help() 6

3.2 Dokumentacja klasy Sck 6

3.2.1 Opis szczegółowy 7

3.3 Dokumentacja klasy TcpListener 7

3.3.1 Opis szczegółowy 8

3.3.2 Dokumentacja konstruktora i destruktora 8

3.3.2.1 TcpListener() 8

3.3.2.2 ~TcpListener() 8

3.3.3 Dokumentacja funkcji składowych 8

3.3.3.1 Cleanup() 9

3.3.3.2 CreateSocket() 9

3.3.3.3 Initialize() 9

3.3.3.4 Run() 9

3.3.3.5 WaitForConnection() 9

3.3.4 Dokumentacja atrybutów składowych 10

3.3.4.1 end 10

3.3.4.2 klienci 10

3.3.4.3 m_IpAdddress 10

3.3.4.4 m_port 10

3.3.4.5 write 10

Indeks **11**

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

checking 5
Sck 6
    TcpListener 7

Rozdział 2

Indeks klas

2.1 Lista klas

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

checking 5
Sck 6
TcpListener 7

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy checking

`#include <checking.h>`

Metody publiczne

- bool checkingdigit (const std::string &adress)
- void Help (const std::string &help)
- bool CheckParams (int argc, char ∗argv[ ])

3.1.1 Opis szczegółowy

Klasa odpowiedzialna za sprawdzenie parametrów wprowadzonych przez użytkownika, w celu zabezpieczenia przed błędami podczas procesu inicjalizacji serwera.

3.1.2 Dokumentacja funkcji składowych

3.1.2.1 checkingdigit()

```
bool checking::checkingdigit (
            const std::string & adress )
```

Metoda sprawdzająca, czy użytkownik dobrze wpisał adres IP.

Parametry

adress	Adres IP wpisany przez użytkownika

Zwraca

Flaga, czy został dobrze wpisany adres.

3.1.2.2 CheckParams()

```
bool checking::CheckParams (
            int argc,
            char * argv[] )
```

Metoda sprawdzająca, czy użytkownik wpisał dobre parametry, aby uruchomić program.

Parametry

argc	Rozmiar tablicy argv.
arg	Parametr wpisany przez użytkownika.

Zwraca

Flaga, czy zostały dobrze wpisane parametry przy starcie.

3.1.2.3 Help()

```
void checking::Help (
            const std::string & help )
```

Metoda informująca użytkownika o autorze programu, jak działa program oraz jak trzeba poprawnie wpisać parametry

Parametry

help	Parametr help.

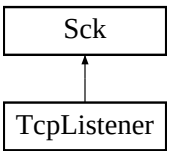
Dokumentacja dla tej klasy została wygenerowana z plików:

- checking.h
- checking.cpp

3.2 Dokumentacja klasy Sck

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

Diagram dziedziczenia dla Sck



Metody publiczne

- virtual SOCKET CreateSocket ()=0
  destruktor wirtualnej klasy Sck
- virtual void Cleanup ()=0
  wirtualna metoda CreateSocket()
- virtual bool Initialize ()=0
  wirtualna metoda Cleanup()

3.2.1 Opis szczegółowy

Wirtualna klasa bazowa klasy TcpListener(serwer) oraz klasy TcpClient(użytkownik clienta)

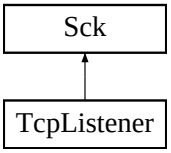
Dokumentacja dla tej klasy została wygenerowana z pliku:

- Sck.h

3.3 Dokumentacja klasy TcpListener

```
#include <TcpListener.h>
```

Diagram dziedziczenia dla TcpListener



Metody publiczne

- TcpListener (const std::string &ipAddress, const int &port)
- ~TcpListener ()
- bool Initialize () override
- void Run ()
- void Cleanup () override

Atrybuty publiczne

- std::thread write
- bool end = false

Metody prywatne

- SOCKET CreateSocket () override
- SOCKET WaitForConnection (const SOCKET &listening, std::unordered_map< SOCKET, std::string > &klienci)

Atrybuty prywatne

- std::unordered_map< SOCKET, std::string > klienci
- std::string m_IpAdddress
- int m_port

3.3.1 Opis szczegółowy

Klasa odpowiedzialna za inicjalizację serwera, czyli postawienie serwera TCP/IP oraz komunikację między użytkownikami. Wykorzystywane

3.3.2 Dokumentacja konstruktora i destruktora

3.3.2.1 TcpListener()

```
TcpListener::TcpListener (
            const std::string & ipAddress,
            const int & port )  [inline]
```

Konstruktor inicjalizujący powstanie klasy TcpListener.

Parametry

ipAddress	Adres IP serwera.
port	Port serwera.

3.3.2.2 ~TcpListener()

```
TcpListener::~TcpListener ( )
```

Destruktor klasy TcpListener.

3.3.3 Dokumentacja funkcji składowych

3.3.3.1 Cleanup()

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

Metoda sprzątająca, czyli usuwanie i czyszczenie zaalokowanych pamięci.

Implementuje Sck.

3.3.3.2 CreateSocket()

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

Metoda inicjalizująca powstanie gniazda nasłuchującego.

Zwraca

Gniazdo stworzone na podstawie parametrów w klasie.

Implementuje Sck.

3.3.3.3 Initialize()

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

Metoda pozwalająca na załadowanie bibliotek WinSocket.

Implementuje Sck.

3.3.3.4 Run()

```
void TcpListener::Run ( )
```

Metoda utrzymująca działanie serwera i komunikację między użytkownikami

3.3.3.5 WaitForConnection()

```
SOCKET TcpListener::WaitForConnection (
            const SOCKET & listening,
            std::unordered_map< SOCKET, std::string > & klienci )  [private]
```

Metoda pozwalająca dołączyć klientowi do serwera.

Parametry

listening	Gniazdo nasłuchujące, które chce dołączyć do serwera
klienci	Tablica mieszająca przechowująca nazwy użytkowników przypisanych do danego gniazda

Zwraca

Gnizado, które zostało przyjęte do serwera

3.3.4 Dokumentacja atrybutów składowych

3.3.4.1 end

```
bool TcpListener::end = false
```

"Flaga" oznajmująca zakończenie pracy serwera, jak i wątku.

3.3.4.2 klienci

```
std::unordered_map<SOCKET, std::string> TcpListener::klienci  [private]
```

Tablica mieszająca przechowująca nazwy użytkowników podłączonych do serwera, który każdej nazwie jest przypisany klucz, czyli numer gniazda.

3.3.4.3 m_IpAdddress

```
std::string TcpListener::m_IpAdddress  [private]
```

Adres IP serwera

3.3.4.4 m_port

```
int TcpListener::m_port  [private]
```

Numer portu serwera

3.3.4.5 write

```
std::thread TcpListener::write
```

Wątek pozwalający wyłączyć serwer.

Dokumentacja dla tej klasy została wygenerowana z plików:

- TcpListener.h
- TcpListener.cpp

Indeks

~TcpListener
    TcpListener, 8

checking, 5
    checkingdigit, 5
    CheckParams, 6
    Help, 6
checkingdigit
    checking, 5
CheckParams
    checking, 6
Cleanup
    TcpListener, 8
CreateSocket
    TcpListener, 9

end
    TcpListener, 10

Help
    checking, 6

Initialize
    TcpListener, 9

klienci
    TcpListener, 10

m_IpAdddress
    TcpListener, 10
m_port
    TcpListener, 10

Run
    TcpListener, 9

Sck, 6

TcpListener, 7
    ~TcpListener, 8
    Cleanup, 8
    CreateSocket, 9
    end, 10
    Initialize, 9
    klienci, 10
    m_IpAdddress, 10
    m_port, 10
    Run, 9
    TcpListener, 8
    WaitForConnection, 9
    write, 10

WaitForConnection
    TcpListener, 9
write
    TcpListener, 10