

## Dokumentation der Methoden der Klasse Connection

### Konstruktor Connection (String pServerIP, int pServerPort)

*nachher* Es ist eine Verbindung zum durch IP-Adresse und Portnummer angegebenen Server eingerichtet, so dass Daten gesendet und empfangen werden können. Anstelle einer IP-Adresse kann *pServerIP* auch ein Rechnername sein.

### Auftrag send (String pMessage)

*nachher* Die angegebene Nachricht wurde, um einen Zeilentrenner erweitert, an den Server versandt.

### Anfrage receive () :String

*nachher* Es wurde auf eine eingehende Nachricht vom Server gewartet und diese Nachricht zurückgegeben, wobei der vom Server angehängte Zeilentrenner entfernt wurde. Während des Wartens war der ausführende Prozess blockiert.

### Auftrag close ()

Die Verbindung wurde getrennt und kann nicht mehr verwendet werden.

## Dokumentation der Methoden der Klasse Client

### Konstruktor Client (String pServerIP, int pServerPort)

*nachher* Es ist eine Verbindung zum durch IP-Adresse und Portnummer angegebenen Server eingerichtet, so dass Zeichenketten gesendet und empfangen werden können. Anstelle einer IP-Adresse kann *pServerIP* auch ein Rechnername sein.

### Auftrag send (String pMessage)

*nachher* Die angegebene Nachricht wurde, um einen Zeilentrenner erweitert, an den Server versandt.

### Auftrag processMessage (String pMessage)

*vorher* Der Server hat die angegebene Nachricht gesendet. Der Zeilentrenner wurde entfernt.

*nachher* Der Client hat auf die Nachricht reagiert. Diese Methode enthält keine Anweisungen und muss in Unterklassen überschrieben werden, damit die Nachricht verarbeitet werden kann.

### Auftrag close ()

Die Verbindung wurde getrennt und kann nicht mehr verwendet werden.

## Dokumentation der Methoden der Klasse Server

### **Konstruktor Server (int pPortNr)**

*nachher* Der Server bietet seinen Dienst über die angegebene Portnummer an. Clients können sich nun mit dem Server verbinden.

### **Auftrag closeConnection (String pClientIP, int pClientPort)**

*vorher* Eine Verbindung mit dem angegebenen Client besteht.

*nachher* Die Verbindung besteht nicht mehr. Der Server hat sich die Nachricht *processClosedConnection* gesendet.

### **Auftrag processClosedConnection (String pClientIP, int pClientPort)**

*vorher* Die Verbindung zu dem angegebenen Client wurde beendet.

*nachher* Der Server hat auf das Beenden der Verbindung reagiert. Diese Methode enthält keine Anweisungen und kann in Unterklassen überschrieben werden.

### **Auftrag processMessage (String pClientIP, int pClientPort, String pMessage)**

*vorher* Der Client mit der angegebenen IP und der angegebenen Portnummer hat dem Server eine Nachricht gesendet.

*nachher* Der Server hat auf die Nachricht reagiert. Diese Methode enthält keine Anweisungen und kann in Unterklassen überschrieben werden.

### **Auftrag processNewConnection (String pClientIP, int pClientPort)**

*vorher* Der Client mit der angegebenen IP-Adresse und der angegebenen Portnummer hat eine Verbindung zum Server aufgebaut.

*nachher* Der Server hat auf den Verbindungsaufbau reagiert. Diese Methode enthält keine Anweisungen und kann in Unterklassen überschrieben werden.

### **Auftrag send(String pClientIP, int pClientPort, String pMessage)**

*vorher* Eine Verbindung zu dem angegebenen Client besteht.

*nachher* Die angegebene Nachricht wurde dem Client, um einen Zeilentrenner erweitert, gesendet.

### **Auftrag sendToAll (String pMessage)**

*nachher* Die angegebene Nachricht wurde an alle verbundenen Clients gesendet.

### **Auftrag close ()**

Alle Verbindungen wurden getrennt. Der Server kann nicht mehr verwendet werden.