



# Dokumentation der SuM-Netzwerkkomponenten

[Verbindung](#)

[ClientVerbindung](#)

[Server](#)



# Dokumentation der Klasse "Verbindung"

Klassenname	Verbindung
Beschreibung	Die Klasse Verbindung beschreibt eine Netzwerkverbindung eines Clients mit einem Server. Es können Nachrichten zum Server geschickt und vom Server empfangen werden.
Protokoll	<p><b>Auftrag</b> init(pIPAdresse : Zeichenkette; pPortNummer : GanzeZahl; pMitProtokollieren : Wahrheitswert)</p> <p><i>nachher</i> Wenn die durch Angabe der IP-Adresse (wahlweise als IP-Nummer oder Domainname) und der Portnummer definierte Verbindung mit dem Server hergestellt werden konnte, ist sie aufgebaut. Wenn der Parameter zum Protokollieren wahr war, wird nun jede Kommunikation über diese Verbindung in einem weiteren Fenster mitprotokolliert.</p> <p><b>Auftrag</b> sende(pNachricht : Zeichenkette)</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>nachher</i> Die Nachricht wurde an den Server geschickt.</p> <p><b>Anfrage</b> empfangeneNachricht : Zeichenkette</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>nachher</i> Diese Anfrage liefert eine Nachricht vom Server und wartet ggf. darauf.</p> <p><b>Anfrage</b> serverAdresse : Zeichenkette</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>Nachher</i> Diese Anfrage liefert die Adresse des Servers.</p> <p><b>Anfrage</b> clientAdresse : Zeichenkette</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>Nachher</i> Diese Anfrage liefert die Adresse des Clients.</p> <p><b>Anfrage</b> serverPort : GanzeZahl</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>Nachher</i> Diese Anfrage liefert die Portnummer des Servers.</p> <p><b>Anfrage</b> clientPort : GanzeZahl</p> <p><i>vorher</i> Die Verbindung ist aufgebaut.</p> <p><i>Nachher</i> Diese Anfrage liefert die Portnummer des Clients.</p> <p><b>Anfrage</b> mitProtokoll : Wahrheitswert</p> <p><i>nachher</i> Diese Anfrage liefert "wahr", genau dann wenn alle Aktivitäten der Verbindung mitprotokolliert werden.</p> <p><b>Auftrag</b> gibFrei</p> <p><i>nachher</i> Die Verbindung wurde geschlossen und steht nicht mehr zur Verfügung.</p>



# Dokumentation der Klasse

## "ClientVerbindung"

Klassenname	ClientVerbindung
Oberklasse	Verbindung
Beschreibung	Die abstrakte Klasse ClientVerbindung beschreibt eine Verbindung, die ihre Nachrichten von einem Server nebenläufig empfängt. Die empfangenen Nachrichten werden wie ein Ereignis bearbeitet.
Bezugsklassen	
Unterklassen	In den Unterklassen muss der Dienst <i>bearbeiteNachricht</i> überschrieben werden.
Protokoll	<p><b>Auftrag</b> <i>init</i></p> <p><i>nachher</i> Die ClientVerbindung ist als Verbindung mit nebenläufigem Nachrichtenempfänger initialisiert.</p> <p><b>Auftrag</b> <i>bearbeiteNachricht(pNachricht : Zeichenkette)</i></p> <p><i>vorher</i> Der Server hat an den Client die angegebene Nachricht geschickt.</p> <p><i>nachher</i> (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.) Der Client hat auf die Nachricht reagiert.</p> <p><b>Auftrag</b> <i>gibFrei</i></p> <p><i>nachher</i> Die ClientVerbindung wurde als Verbindung geschlossen. Der Nachrichtenempfänger existiert nicht mehr.</p>



# Dokumentation der Klasse "Server"

Klassenname	Server
Beschreibung	Die abstrakte Klasse Server ermöglicht es, Clients Verbindungen über eine selbst festgelegte Portnummer anzubieten. Dabei werden die Verbindungen automatisch aufgebaut und nebenläufig abgewickelt, der Server muss lediglich auf bestimmte Ereignisse reagieren.
Bezugsklassen	ServerVerbindung, Liste
Unterklassen	In den Unterklassen können die Dienste bearbeiteVerbindungsaufbau, bearbeiteNachricht, bearbeiteVerbindungsverlust und bearbeiteVerbindungsende überschrieben werden.
Protokoll	<p><b>Auftrag</b> init(pPortNummer : GanzeZahl; pMitProtokollieren : Wahrheitswert) <i>nachher</i> Der Server bietet seinen Dienst über die angegebene Portnummer an. Wenn der Parameter zum Protokollieren wahr war, wird nun jede Kommunikation über diese Verbindung in einem weiteren Fenster mitprotokolliert.</p> <p><b>Auftrag</b> sendeAnAlle(pNachricht : Zeichenkette) <i>nachher</i> Die Nachricht wurde an alle Clients geschickt, mit denen der Server verbunden ist.</p> <p><b>Auftrag</b> sendeAnEinen(pClientIP, pNachricht : Zeichenkette) <i>vorher</i> Eine Verbindung zu einem Client mit der angegebenen IP-Nummer besteht. <i>nachher</i> Die Nachricht wurde an den Client mit der angegebenen IP-Nummer verschickt..</p> <p><b>Auftrag</b> melde(pMeldung : Zeichenkette) <i>nachher</i> Im Protokollfenster wurde die angegebene Meldung angehängt..</p> <p><b>Auftrag</b> beendeVerbindung(pClientIP : Zeichenkette) <i>vorher</i> Eine Verbindung zu einem Client mit der angegebenen IP-Nummer besteht. <i>nachher</i> Der Server hat nach Senden der Nachricht "bearbeiteVerbindungsende" die Verbindung zu dem Client mit der angegebenen IP-Nummer beendet.</p> <p><b>Anfrage</b> port : GanzeZahl <i>nachher</i> Diese Anfrage liefert die Portnummer des Servers.</p> <p><b>Anfrage</b> clientListe : Liste <i>nachher</i> Diese Anfrage liefert die Liste mit den Verbindungen des Servers.</p> <p><b>Anfrage</b> serverVerbindungVonIP(pClientIP : Zeichenkette) : ServerVerbindung <i>nachher</i> Diese Anfrage liefert die ServerVerbindung zu dem Client mit der angegebenen IP-Nummer, falls eine Verbindung besteht, sonst nichts.</p> <p><b>Anfrage</b> zahlDerVerbindungen : GanzeZahl <i>nachher</i> Diese Anfrage liefert die Anzahl der Verbindungen, die der Server zurzeit betreibt.</p> <p><b>Anfrage</b> mitProtokoll : Wahrheitswert <i>nachher</i> Diese Anfrage liefert "wahr", genau dann wenn alle Aktivitäten der Verbindung mitprotokolliert werden.</p> <p><b>Auftrag</b> bearbeiteVerbindungsaufbau(pClientIP : Zeichenkette) <i>vorher</i> Eine Verbindung zu dem Client mit der angegebenen IP-Nummer wurde aufgebaut. <i>nachher</i> (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.) Der Server hat auf den Verbindungsaufbau z. B. mit einer Begrüßungsnachricht reagiert.</p> <p><b>Auftrag</b> bearbeiteNachricht(pClientIP, pNachricht : Zeichenkette) <i>vorher</i> Der Client mit der angegebenen IP-Nummer hat dem Server die angegebene Nachricht geschickt. <i>nachher</i> (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.) Der Server hat auf die Nachricht reagiert.</p> <p><b>Auftrag</b> bearbeiteVerbindungsverlust(pClientIP : Zeichenkette) <i>vorher</i> Die Verbindung zu dem Client mit der angegebenen IP-Nummer wurde verloren. <i>nachher</i> (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.) Der Server hat auf den Verbindungsverlust reagiert.</p>



**Auftrag** *bearbeiteVerbindungsende(pClientIP : Zeichenkette)*

**vorher** Die Verbindung zu dem Client mit der angegebenen IP-Nummer wird gerade vom Server beendet.

**nachher** (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.)  
Der Server hat auf das bevorstehende Verbindungsende reagiert.

**Auftrag** *gibFrei*

**nachher** Der Server mit seinen Verbindungen wurde geschlossen und steht nicht mehr zur Verfügung.

Damit es möglich ist, für Testzwecke auch mehrere Clients auf einem Rechner zu starten, muss der Client nicht wie bei den oben aufgeführten Diensten durch die IP-Nummer, sondern durch die Kombination von IP- und Port-Nummer identifiziert werden. Aus diesem Grunde werden die Dienste alternativ auch mit entsprechend erweiterter Parameterliste angeboten:

**Auftrag** *sendeAnEinen(pClientIP : Zeichenkette; pClientPort: GanzeZahl; pNachricht : Zeichenkette)*

**vorher** Eine Verbindung zu einem Client mit der angegebenen IP- und Port-Nummer besteht.

**nachher** Die Nachricht wurde an den Client mit der angegebenen IP- und Port-Nummer verschickt..

**Auftrag** *beendeVerbindung(pClientIP : Zeichenkette; pClientPort: GanzeZahl)*

**vorher** Eine Verbindung zu einem Client mit der angegebenen IP- und Port-Nummer besteht.

**nachher** Der Server hat nach Senden der Nachricht "bearbeiteVerbindungsende" die Verbindung zu dem Client mit der angegebenen IP- und Port-Nummer beendet.

**Anfrage** *serverVerbindungVonIP(pClientIP : Zeichenkette; pClientPort: GanzeZahl) : ServerVerbindung*

**nachher** Diese Anfrage liefert die ServerVerbindung zu dem Client mit der angegebenen IP- und Port-Nummer, falls eine Verbindung besteht, sonst nichts.

**Auftrag** *bearbeiteVerbindungsaufbau(pClientIP : Zeichenkette; pClientPort: GanzeZahl)*

**vorher** Eine Verbindung zu dem Client mit der angegebenen IP- und Port-Nummer wurde aufgebaut.

**nachher** (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.)  
Der Server hat auf den Verbindungsaufbau z. B. mit einer Begrüßungsnachricht reagiert.

**Auftrag** *bearbeiteNachricht(pClientIP : Zeichenkette; pClientPort: GanzeZahl; pNachricht : Zeichenkette)*

**vorher** Der Client mit der angegebenen IP- und Port-Nummer hat dem Server die angegebene Nachricht geschickt.

**nachher** (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.)  
Der Server hat auf die Nachricht reagiert.

**Auftrag** *bearbeiteVerbindungsverlust(pClientIP : Zeichenkette; pClientPort: GanzeZahl)*

**vorher** Die Verbindung zu dem Client mit der angegebenen IP- und Port-Nummer wurde verloren.

**nachher** (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.)  
Der Server hat auf den Verbindungsverlust reagiert.

**Auftrag** *bearbeiteVerbindungsende(pClientIP : Zeichenkette; pClientPort: GanzeZahl)*

**vorher** Die Verbindung zu dem Client mit der angegebenen IP- und Port-Nummer wird gerade vom Server beendet.

**nachher** (Dieser Dienst ist abstrakt und kann in den Unterklassen überschrieben werden.)  
Der Server hat auf das bevorstehende Verbindungsende reagiert.