4.4 Netzstrukturen

Die Klasse Connection

Objekte der Klasse `Connection` ermöglichen eine Netzwerkverbindung mit dem TCP/IP-Protokoll. Es können nach Verbindungsaufbau zu einem Server Zeichenketten (Strings) gesendet und empfangen werden. Zur Vereinfachung geschieht dies zeilenweise, d. h., beim Senden einer Zeichenkette wird ein Zeilentrenner ergänzt und beim Empfangen wird er entfernt.
Eine Fehlerbehandlung, z.B. ein Zugriff auf eine bereits geschlossene Verbindung, ist in dieser Klasse aus Gründen der Vereinfachung nicht vorgesehen.

Dokumentation der Klasse Connection

Konstruktor `Connection(String pServerIP, int pServerPort)`
Es wird eine Verbindung zum durch IP-Adresse und Portnummer angegebenen Server aufgebaut, so dass Daten gesendet und empfangen werden können.

Auftrag `void send(String pMessage)`
Die angegebene Nachricht `pMessage` wird - um einen Zeilentrenner erweitert - an den Server versandt.

Anfrage `String receive()`
Es wird auf eine eingehende Nachricht vom Server gewartet und diese Nachricht zurückgegeben, wobei der vom Server angehängte Zeilentrenner entfernt wird. Während des Wartens ist der ausführende Prozess blockiert.

Auftrag `void close()`
Die Verbindung wird getrennt und kann nicht mehr verwendet werden.

Die Klasse Client

Über die Klasse Client werden Netzwerkverbindungen mit dem TCP/IP-Protokoll ermöglicht. Es können - nach Verbindungsaufbau zu einem Server - Zeichenketten (Strings) gesendet und empfangen werden, wobei der Empfang nebenläufig geschieht. Zur Vereinfachung geschieht dies zeilenweise, d. h., beim Senden einer Zeichenkette wird ein Zeilentrenner ergänzt und beim Empfangen wird er entfernt.
Die empfangene Nachricht wird durch eine Ereignisbehandlungsmethode verarbeitet, die in Unterklassen überschrieben werden muss.
Eine Fehlerbehandlung ist in dieser Klasse aus Gründen der Vereinfachung nicht vorgesehen.

Dokumentation der Klasse Client

Konstruktor `Client(String pServerIP, int pServerPort)`
Es wird eine Verbindung zum durch IP-Adresse und Portnummer angegebenen Server aufgebaut, so dass Zeichenketten gesendet und empfangen werden können.

Auftrag `void send(String pMessage)`
Die angegebene Nachricht `pMessage` wird - um einen Zeilentrenner erweitert - an den Server versandt.

Auftrag `void processMessage(String pMessage)`
Nachdem der Server die angegebene Nachricht `pMessage` gesendet hat wurde der Zeilentrenner entfernt. Der Client kann auf die Nachricht `pMessage` in dieser Methode reagieren. Allerdings enthält diese Methode keine Anweisungen und muss in Unterklassen überschrieben werden, damit die Nachricht verarbeitet werden kann.

Auftrag `void close()`
Die Verbindung zum Server wird getrennt und kann nicht mehr verwendet werden.

Die Klasse Server

Über die Klasse `Server` ist es möglich, eigene Serverdienste anzubieten, so dass Clients Verbindungen gemäß dem TCP/IP-Protokoll hierzu aufbauen können. Nachrichten werden grundsätzlich zeilenweise verarbeitet, d. h., beim Senden einer Zeichenkette wird ein Zeilentrenner ergänzt und beim Empfangen wird er entfernt.
Verbindungsaufbau, Nachrichtenempfang und Verbindungsende geschehen nebenläufig. Durch Überschreiben der entsprechenden Methoden kann der Server auf diese Ereignisse reagieren.
Eine Fehlerbehandlung ist in dieser Klasse aus Gründen der Vereinfachung nicht vorgesehen.

Dokumentation der Klasse Server

Konstruktor `Server(int pPortNr)`
Nach dem Aufruf dieses Konstruktors bietet ein Server seinen Dienst über die angegebene Portnummer an. Clients können sich nun mit dem Server verbinden.

Auftrag `void closeConnection(String pClientIP, int pClientPort)`
Unter der Voraussetzung, dass eine Verbindung mit dem angegebenen Client existiert, wird diese beendet. Der Server sendet sich die Nachricht `processClosedConnection`.

Auftrag `void processClosedConnection(String pClientIP, int pClientPort)`
Diese Methode ohne Anweisungen wird aufgerufen, bevor der Server die Verbindung zu dem in der Parameterliste spezifizierten Client schließt. Durch das Überschreiben in Unterklassen kann auf die Schließung der Verbindung zum angegebenen Client reagiert werden.

Auftrag `void processMessage(String pClientIP, int pClientPort, String pMessage)`
Der Client mit der angegebenen IP und der angegebenen Portnummer hat dem Server eine Nachricht gesendet. Dieser ruft daraufhin diese Methode ohne Anweisungen auf. Durch das Überschreiben in Unterklassen kann auf diese Nachricht des angegebenen Client reagiert werden.

Auftrag `void processNewConnection(String pClientIP, int pClientPort)`
Der Client mit der angegebenen IP-Adresse und der angegebenen Portnummer hat eine Verbindung zum Server aufgebaut. Der Server hat daraufhin diese Methode aufgerufen, die in dieser Klasse keine Anweisungen enthält. Durch das Überschreiben in Unterklassen kann auf diesen Neuaufbau einer Verbindung von dem angegebenen Client zum Server reagiert werden.

Auftrag `void send(String pClientIP, int pClientPort, String pMessage)`
Wenn eine Verbindung zum angegebenen Client besteht, dann wird diesem Client die angegebene Nachricht - um einen Zeilentrenner erweitert - gesendet.

Auftrag `void sendToAll(String pMessage)`
Die angegebene Nachricht wird - um einen Zeilentrenner erweitert - an alle verbundenen Clients gesendet.

Auftrag `void close()`
Alle bestehenden Verbindungen werden getrennt. Der Server kann nicht mehr verwendet werden.