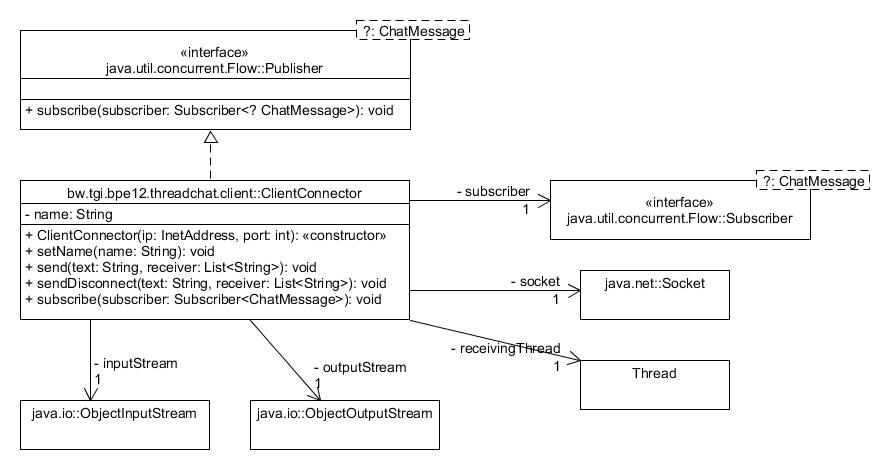
## Erweitern des Chat-Clients

Im Client-Teil ist die GUI vorgefertigt. Es fehlen allerdings Teile der inneren Implementierung, vor allem Teile des Hauptprogramms und Teile des ausgelagerten ClientConnectors.

Neben dem unvollständigen Code erhalten Sie zur Vervollständigung das nachfolgende Klassendiagramm (Hinweis: die gestrichelten Kästchen oben rechts bedeuten, dass die Interfaces typisiert sind: <Type>).



Der Rumpf des Konstruktors ist leer, dafür finden Sie unten die Beschreibung dafür in Pseudocode und Kommentaren

Instanziiere das socket-Objekt

Instanziiere das outputStream-Objekt aus dem socket-Outputstream

Instanziiere das inputStream-Objekt aus dem socket-Inputstream

Die beiden nächsten Anweisungen müssen in einen eigenen Thread eingebaut werden, sonst blockiert der Sender den Empfang

**ChatMessage message = (ChatMessage) inputStream.readObject();**

**subscriber.onNext(message);**

starte den Thread

Am obigen Klassendiagramm erkennen Sie auch, dass wir mit dem Publish-Subscriber-Muster arbeiten, das als Subpackage in java.util.concurrent verfügbar ist.

Ebenfalls ergänzen Sie den Code im Hauptprogramm des Clients nach nachfolgender Beschreibung:

Der Code ist in den Rumpf der Actionhandler-Methode einzufügen. Es muss unterschieden werden, ob schon eine Verbindung besteht oder nicht.

wenn nicht verbunden

hole die IP-Adresse als Text aus dem Textfeld

hole den Port als als Text aus dem Textfeld und wandle in Zahl um

**InetAddress ip = InetAddress.getByName(ipAddress);**

Instanziiere den ClientConnector

Schreibe das Objekt vom BPE12ChatClient ein

Setze connected auf wahr

Enable den Send-Button

Ändere den Text des Connect-Buttons auf "Disconnect"

sonst

schicke disconnect als Nachricht an alle Clients

nachfolgenden Code unverändert einbauen:

**Component[] components = panel.getComponents();**

**for (Component component : components) {**

**if (component.getClass().equals(JCheckBox.class)) {**

**panel.remove((JCheckBox) component); // add this line**

**panel.updateUI();**

**}**

**}**

**frmChatclient.setVisible(true);**

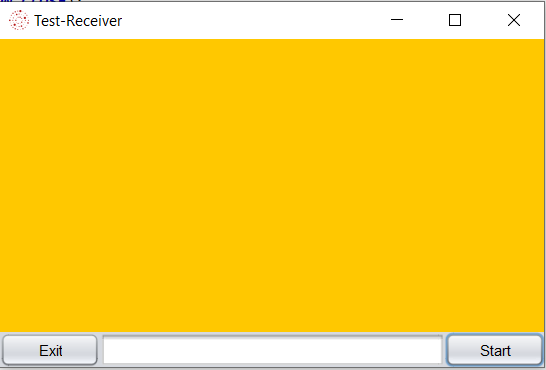
connected auf false setzen

Programm beenden

}

Nach erfolgreicher Umsetzung aller Ergänzungen sollte der Code des Programms fehlerfrei sein.

Im Projekt finden Sie die Klasse TestReceiver, mit der Sie testen können, ob die Verbindung aufgebaut wird. Es kann natürlich nicht gechattet werden, da der TextReceiver nur die Verbindung aufbaut (der TestReceiver ist an den Receiver aus der Serialisierungs-Übung angelehnt – daher: starten nicht vergessen).

Der Test-Receiver wird orange, wenn der Server auf einen Client wartet und grün, wenn die Verbindung steht.

**Optional:**

Ergänzen Sie den TestReceiver so, dass eine Nachricht ChatMessage empfangen werden kann. Senden Sie eine Nachricht vom Client und geben Sie den Inhalt um TestReceiver aus.