**Aufgabe 1)**

Berechnungsserver

Für meinen Berechnungsserver habe ich die abstrakte Klasse “Message” 2 x konkret umgesetzt. Außerdem implementiert sie das Interface Serializable, damit es möglich ist ihre konkreten Klassen durch Streams zu schicken.

Einmal gibt es die Message-Klasse als Protokoll vom Client zum Server(“CalcMsg”) und das zweite mal als Protokoll vom Server zum Client(“ResultMsg”). “CalcMsg” enthält die Berechnung welche an den Server geschickt wird. “ResultMsg” enthält das Ergebnis der Berechnung welches dann wieder zurück an den Client geschickt wird.

Das Zustandsdiagramm für die Protokoll sieht als wie folgt aus:

**Client Server**

CalcMsg ------------------>

<-------------------ResultMsg

Beide Anwendungen fangen alle mir bisher aufgefallenen Exceptions ab und behandeln sie dementsprechend. Außerdem ist der Server in der Lage, mehrere Clients gleichzeitig aufzunehmen und sich um sie zu kümmern. Bei Beendigung des Servers oder eines Clients werden alle Ressourcen wieder freigegeben.