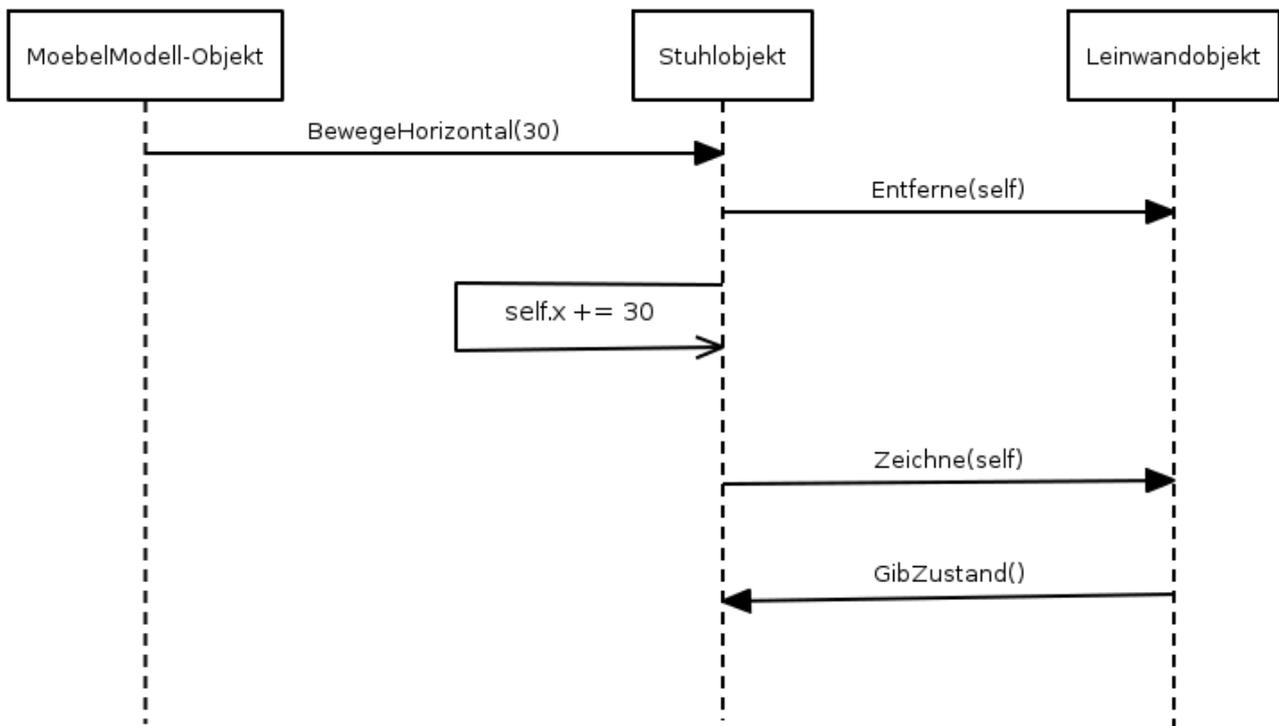


## Beobachter

Die Beziehung zwischen den konkreten Moebel-Objekten<sup>1</sup>, wie beispielsweise einem Stuhl-Objekt, und der Klasse Leinwand erfüllt das Verhaltensmuster **Beobachtermuster**. Bei Gamma e.a. heißt es (leicht bearbeitet) zum Zweck:

Definiere eine 1-zu-n-Abhängigkeit zwischen Objekten, so dass die Änderung des Zustands eines Objektes (des "1"-Objekts) dazu führt, dass alle abhängigen Objekte (die n Objekte) über die Notwendigkeit ihrer Aktualisierung informiert werden.

Im konkreten Fall läuft das (auf den wesentlichen Vorgang vereinfacht) von einem anderen Objekt angestoßen wie im Bild dargestellt ab:



Für die message **Entferne(self)** ist keine Abfrage **GibZustand()** notwendig, da das StuhlObjekt vom LeinwandObjekt ja gerade nicht dargestellt werden muss. Um es mit veränderten Werten aber erneut zeichnen zu können, muss das LeinwandObjekt den aktuellen Zustand beim StuhlObjekt abfragen.

Die Darstellung ist dahingehend vereinfacht, dass hier nur das Senden der Abfrage **GibZustand()** an das Stuhl-Objekt dargestellt ist. Der Programmtext der Leinwandklasse gibt aber vor, dass die Abfrage **GibZustand()** vom (Singleton-) Objekt an alle Moebel-Objekte geschickt werden muss, da für das Bild alle Objekte erneut gezeichnet werden müssen.

1 Beachten Sie, dass im vollständigen Projekt, das dem Modell-View-Controller – Konzept folgt sehr viele weitere Beziehungen das Beobachter-Muster erfüllen. Insbesondere wird (vermutlich) die Beziehung von den konkreten Möbelobjekten auf den Controller umgelenkt.