State(Zustand) – Design Pattern

Das **State** Entwurfsmuster ermöglicht die elegante Modellierung von zustandsabhängigen Verhalten eines Objekts. Je nach internen Zustand ändert sich das Verhalten des Objekts.

Es wird eine einheitliche Schnittstelle für die möglichen Zustände definiert (State). Für jeden Zustand wird eine Klasse erstellt, die diese Schnittstelle realisiert (ConcreteStateX).

Das Objekt, dessen Verhalten in Abhängigkeit vom Zustand geändert werden soll (das Context-Objekt) aggregiert nun ein solches Zustandsobjekt (via Instanzvariable). Dieses Objekt repräsentiert den aktuellen internen Zustand und kapselt das zustandsabhängige Verhalten des Contexts. Der Context delegiert Aufrufe an sein aktuell gesetztes Zustandsobjekt.

Die Zustandswechsel und -übergänge können durch die konkreten Zustände selbst durchgeführt werden, in dem sie dem Context einen Folgezustand zu weisen.

Es muss definiert werden, wo entschieden wird, welcher Zustand als nächstes gesetzt werden soll. Dies kann zu einem in den konkreten Zustandsobjekten, aber auch im Kontextobjekt geschehen.

Zustände bestimmen Folgezustand

Damit jeder Zustand seinen Folgezustand bestimmen kann, muss er diesem beim Contextobjekt setzen. Dazu benötigt er aber

* eine Referenz auf das Contextobjekt.
* Weiterhin muss die Schnittstelle des Contextobjekts um entsprechende Setter erweitert werden.

Damit die Zustandsobjekte nicht immer wieder mit new neu instanziiert werden müssen, kann der Context alle möglichen Zustände als Attribute halten.