

## Entwurfsmuster

### Aufgabe 12: Zustandsmuster

Gegeben sei ein Getränkeautomat, der die Auswahl zwischen 3 Getränken bietet: Kaffee, Suppe und Limonade. Die Getränke kosten unterschiedlich viel Geld, evtl. ist eine Getränkeart nicht mehr verfügbar. Der Automat akzeptiert Münzen im Wert von 5c, 10c, 20c, 50c, 1e und 2e sowie Geldscheine im Wert von 5e, 10e und 20e, Rückgeld wird erstattet. Beim Rückgeld können Sie davon ausgehen, dass immer genügend Münzen vorhanden sind.

Implementieren Sie ein interaktives Programm, mit dem der Kauf eines Getränks simuliert werden kann. Der Getränkeautomat soll mittels des Zustandsmusters implementiert werden. Die Benutzeroberfläche kann wie folgt aussehen:

Zustand: bereit

- (1) Kaffee: 60c
- (2) Suppe: 78c
- (3) Limonade: 45c

-----  
Ihre Auswahl?

Nach der Auswahl eines Getränks wird der zu zahlende Betrag angezeigt:

Zustand: geldeinwurf

noch zu zahlen: 37c  
Ihre Eingabe?

Auf jeder Ebene soll ein Abbruch möglich sein.

Erstellen Sie zu Ihrer Implementierung ein UML-Klassendiagramm sowie ein Sequenzdiagramm, aus dem der Ablauf beim Zustandswechsel hervorgeht.

### Aufgabe 13: Decorator-Muster<sup>1</sup>

Eine Kaffeehaus-Kette, nennen wir sie Sternback-Kaffee, verkauft in vielen Filialen unterschiedliche Kaffeearten wie Hausmischung, dunkel geröstet, entkoffeiniert oder Espresso sowie verschiedene Zutaten wie heiße Milch, Soja, Schokolade oder Milchschaum. Je nach Kaffeeart und Zutaten ist für das Getränk ein Preis zu berechnen.

Zur Modellierung verwenden wir das Decorator-Muster. Wir beginnen also mit einem Getränk und dekorieren es zur Laufzeit mit Zutaten. Wenn der Kunde eine dunkle Röstung mit Schoko und Milchschaum möchte, geht das beispielsweise so:

1. Wir nehmen ein `DunkleRöstung`-Objekt,
2. dekorieren es mit einem `Schoko`-Objekt,
3. dekorieren es mit einem `Milchschaum`-Objekt,
4. rufen die Methode `preis()` auf und stützen uns auf Delegation, um den Preis für die Zutaten hinzuzufügen.

Implementieren sie obige Klassen unter Verwendung der angegebenen Modellierung. Erstellen Sie ein Beispielprogramm. Wie sieht das UML-Klassendiagramm aus?

### Aufgabe 14: Observer-Muster

Erweitern Sie eine Klasse Ihrer Wahl um Beobachterfunktionen. Die Klasse soll der Speicherung beliebiger Daten dienen (z.B. der Heap aus Aufgabe 4, Übungsblatt 2.). Es soll möglich sein, Beobachter der gewählten Klasse an- und abzumelden und die Besucher über Änderungen an der Speicherstruktur (Einfügen, Löschen, Sortieren etc.) zu unterrichten.

Schreiben Sie eine Klasse `Logger`, die Änderungen (z.B. `insert` und `extractMin` beim Heap) in einer Datei protokolliert. Erstellen Sie ferner eine Klasse `Counter`, der die Anzahl der Änderungen zählt. Beide Klassen sollen als Beobachter angemeldet werden.

---

<sup>1</sup>Dieses Beispiel ist dem Buch „Entwurfsmuster von Kopf bis Fuß“ von Eric Freeman und Elisabeth Freeman entnommen. Das Probekapitel über das Entwurfsmuster „Decorator“ kann beim Verlag O’Reilly heruntergeladen werden.