
Übung 16: GoF Patterns

Starbucks in Rapperswil

Abgabe 3: per EMail an hrudin@hsr.ch bis zum 6.12.2010

- Abzugeben ist Aufgabe 3 als Java Eclipse Projekt in einer ZIP-Datei **U16-NameVorname.zip**.
-

Aufgabe 1: Design mit GOF Patterns

In Rapperswil soll ein neuer Starbucks Coffee Shop eröffnet werden. Dort gibt es verschiedene Kaffeearten, die man jeweils mit einer grossen Auswahl an Zusätzen bestellen kann.

Sie sollen mehrere Designvarianten entwerfen, um die Bestellmöglichkeiten im Starbucks Coffee Shop zu repräsentieren. Jeweils soll für den Kunden und auf der Kasse der vollständige Name des gewählten Getränks und sein Preis angezeigt werden. Ein bestelltes Getränk setzt sich aus einer Grundbestellung und einer Anzahl nahezu beliebig kombinierbarer Zusätze zusammen. Mögliche Ausschlusskriterien der Zusätze sollen Sie nicht behandeln.

Jedes bestellbare Objekt soll folgende Methoden implementieren:

```
public float getCost()  
public String getDescription()
```

Damit es nicht zu kompliziert wird, sollen Sie nur folgende Basisgetränke berücksichtigen

- Espresso
- Caffé Latte
- Cappuccino
- (Filter-) Kaffee
- Caffé Mocha

Kostenpflichtig können Sie ein Getränk mit folgenden Ingredienzien verfeinern:

- extra shot - extra Espresso
- whipped cream - Schlagrahm

gratis sind die folgenden Variationen

- decaf - entkoffeiniert
- steamed Milk - geschäumte Milch ("nass")
- foamed Milk - Milchschaum ("trocken")

Wie viele Klassen müssten Sie modellieren, um alle möglichen Getränkebestellungen je durch eine Klasse repräsentieren würden?

Das sind wahrscheinlich zu viele um ein gutes Design zu ergeben, ausserdem erhalten Sie sehr viel Code-Duplizierung, was dem DRY Prinzip widerspricht.

Welche **GoF** Patterns könnten sich eignen, um die Anzahl der zu modellierenden Klassen zu reduzieren?

Aufgabe 2: Designvarianten

Designvariante A

Schauen Sie sich das Decorator Pattern genau an, und versuchen Sie das Problem mit diesem Pattern zu lösen. Zeichnen Sie ein entsprechendes Designdiagramm.

Welcher Code wäre notwendig um einen Caffé Mocha mit Schlagrahm und ohne Koffein zu erzeugen?

Designvariante B

Können Sie sich auch eine Lösung vorstellen, die den Einsatz des Decorator Patterns auf die Spitze treibt und ausgehend von einem Kaffeebecher alles was hineinkommt mittels Decorator hinzufügt.

Aufgabe 3: Umsetzung in Java - Abgabe

Codieren Sie nun eine Designvariante nach Aufgabe 2 in Java. Schreiben Sie dazu ein Testprogramm, das Ihren Design/Code verifiziert.