

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 22.5.-26.5.2017 besprochen.

Übungsblatt 4

Aufgabe 1

Entwurfsmuster – Dekorierer

Der Telekommunikationsanbieter **Unreachable** hat ein neues Abrechnungsmodell für Internetanschlüsse entwickelt, das beim Anlegen, Anwenden und Abrechnen durch ein Dekorierermuster unterstützt werden soll. Der Nutzer muss zwischen dem Basis- und dem Komfortanschluss wählen und kann verschiedene Zusatzleistungen hinzubuchen, die in der folgenden Tabelle stehen. **Unreachable** erzwingt auf den angeschlossenen Endgeräten Werbeeinblendungen. Diese können durch die Zusatzleistung „Werbereduktion“ um jeweils 50% vom verbleibenden Restwerbeaufkommen reduziert werden: Wird diese Leistung drei Mal gebucht, so wird nur noch ein Achtel der eigentlich vorgesehen Werbung gezeigt ($0,5^3=0,125$):

	Bezeichnung	Internet	Kosten
Anschluss	Basis	4 Mbit/s	€ 7
	Komfort	16 Mbit/s	€ 10
Zusatzleistungen	Bandbreite (klein)	je weitere 2 Mbit/s	€ 1
	Bandbreite (groß)	je weitere 8 Mbit/s	€ 3
	Werbereduktion		€ 2

Skizzieren Sie ein Klassendiagramm für ein Dekorierer-Muster, das das Vertragsmodell von **Unreachable** unterstützt. Skizzieren Sie für jede Klasse des Dekorierermusters die Implementierungen der folgenden Methoden:

- `int gesamtkosten()` berechnet die Gesamtkosten des Anschlusses mit Zusatzleistungen.
- `int bandbreite()` berechnet die gebuchte Bandbreite.
- `double werbereduktion()` liefert den einzublendenden Werbeanteil.

Aufgabe 2

Entwurfsmuster – Kompositum

Skizzieren Sie ein Klassendiagramm für ein Kompositum-Muster, das die im Folgenden beschriebenen Beziehungen zwischen den Objekten verschiedener Klassen geeignet umsetzt:

Ein Film soll aus einer Szene bestehen, die ihrerseits aus mehreren *avi*-, *mp4*- oder *divx*-Dateien oder weiteren Szenen gebildet werden kann.

Für das oben modellierte Kompositum sollen die folgenden Methoden angelegt werden. Skizzieren Sie, wie die Implementierungen dieser Methoden aussehen sollten.

- `void play()` spielt einen Film ab.
- `int scenes()` gibt die Anzahl der Szenen zurück.
- `boolean onlyDivx()` liefert `true`, falls alle Szenen ausschließlich *divx*-Dateien enthalten.

Aufgabe 3

Entwurfsmuster – Beobachter

Das folgende Klassendiagramm zeigt einen Ausschnitt aus dem Klassendiagramm eines Überwachungsprogramms, das Fenster zur graphischen Visualisierung der von Sensoren ermittelten Daten besitzt. Die Anzeige in einem `ReportWindow` soll bei Änderungen der Werte eines Sensors jederzeit aktuell sein.

Erweitern Sie das Klassendiagramm zu dem Entwurfsmuster *Beobachter*, so dass Objekte der Klasse `ReportWindow` ein Objekt der Klasse `Sensor` beobachten können. Geben Sie ein Sequenzdiagramm an, das für den folgenden Ablauf die Aufrufreihenfolge der beteiligten Methoden gemäß des modellierten Entwurfsmusters *Beobachter* beschreibt:

Von einem `Application`-Objekt wird ein `ReportWindow`-Objekt erzeugt und diesem ein `Sensor`-Objekt zugeordnet. Beim Erzeugen des Reports soll der aktuelle Zustand der Sensordaten angezeigt werden. Danach wird das `Sensor`-Objekt durch einen Aufruf der Methode `setValue(2, 5.5)` verändert.

