

Übung 11 zur OOS SS16 Bearbeitung bis 09.06.2016

Aufgabe 40: (Applet)

- a) Machen Sie aus dem Frame aus Übungsaufgabe 27 zusammen mit der Ereignisverarbeitung aus Übungsaufgabe 28 ein Applet und führen Sie dieses in Eclipse aus.
- b) Implementieren Sie eine html-Seite, die auf das Applet aus a) verweist und laden Sie diese Seite in einem Web-Browser. Was passiert dabei?

Aufgabe 45: (WebService)

- a) Machen Sie aus der `MitarbeiterVerwaltung` aus den Aufgaben 31 & 34 einen `WebService` und führen Sie diesen aus. Da die Vererbung bei `WebServices`, wenn überhaupt, dann nur ziemlich kompliziert verwendbar ist, machen Sie hierbei aus der `MitarbeiterVerwaltung` eine `AngestelltenVerwaltung`. Das bedeutet, dass Sie nur `Angestellte` eintragen und löschen können. Ersetzen Sie hierzu an allen Stellen im Programm `Mitarbeiter` durch `Angestellter`.
- b) Sprechen Sie den `WebService` mittels des `Clients` aus Aufgabe 34 an.

Hinweis:

Informationen zur Erstellung von `WebServices` finden Sie in der Praktikumsbeschreibung zu Versuch 5 und in der zusätzlichen Beschreibung auf der Homepage.

Design-Aufgabe 4: (Singleton)

- a) Entwerfen Sie ein Design in UML, so dass eine Klasse implementiert wird, und sichergestellt ist, dass nur Sie selbst Objekte davon erzeugen können und Sie somit die Kontrolle, dass ggf. nur ein einziges Objekt dieser Klasse angelegt wird.
- b) Überlegen Sie, in welchen Zusammenhängen ein solches Problem auftritt.
- c) Geben Sie eine Implementierung Ihres Designs aus a) an.

Design-Aufgabe 5 (Factory Method):

- a) Definieren Sie ein Pattern, das folgenden Zweck erfüllt: Eine (abstrakte) Klasse A muss Objekte einer Klasse B und von allen (möglicherweise noch nicht existenten) Unterklassen von B erzeugen und verarbeiten können.
- b) Beschreiben Sie, warum die nahe liegendste Lösung der Erzeugung mittels Konstruktoren in diesem Zusammenhang nicht möglich ist.

Übung 11 zur OOS SS16 Bearbeitung bis 09.06.2016

- a. Beschreiben Sie, in welchem Zusammenhang das Pattern aus a) verwendet werden kann.
- c) Geben Sie eine Implementierung des Patterns aus a) in Java an, in der die Klasse A eine Methode enthält, die die Anzahl aller erzeugten Objekte von B und deren Unterklasse enthält.

Design-Aufgabe 6: (Strategy)

- a) Definieren Sie ein Pattern, das folgenden Zweck erfüllt: Einem Klienten soll es ermöglicht werden, verschiedene Algorithmen zur Lösung eines Problems zu verwenden. Beispiel: In ADS soll eine Tool-Box zur Untersuchung des Verhaltens von Sortieralgorithmen auf Integer-Zahlen verwendet werden, so dass vom Klienten die Algorithmen ausgewählt werden können und ggf. neue Algorithmen einfach zu integrieren sind.
- b) Implementieren Sie das Pattern aus a), indem Sie mindestens drei Sortieralgorithmen implementieren, die dem Klienten über eine Toolbox zur Verfügung gestellt werden.

Aufgabe 46: (Design-Regeln)

- a) Geben Sie ein Java-Beispielprogramm an, in dem eine Methode bei gleichen Parameterwerten verschiedene Ausgaben liefert.
- b) Gegen welche Design-Regel verstößt Ihr Programm aus a)