

Dies ist keine „Musterlösung“, sondern eine gute von vielen möglichen Lösungen. Kommentare, die nicht Teil der Lösung sind, sind kursiv gesetzt.

**Aufgabe 8-1:**

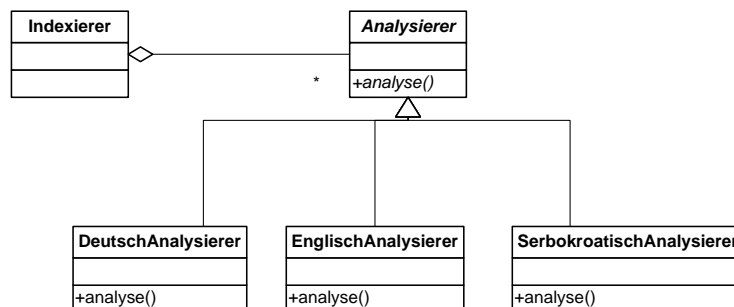
1. Komponenten sind konkrete, ausführbare Programmteile, Entwurfsmuster und Architekturstile nicht. Entwurfsmuster sind im Vergleich zu Architekturstilen „kleiner“. Architekturstile betreffen vor allem die Zusammenarbeit von Komponenten (insbesondere im Hinblick auf nicht-funktionaler Anforderungen), Entwurfsmuster die von Klassen und kleinen Modulen (insbesondere zur Erreichung von Funktionalität). Der Einsatz von Architekturmustern liegt im Entwurfsprozess vor den Entwurfsmustern; der Grobentwurf vor dem Feinentwurf. Man vergleiche z.B. den Interpretierer, den es sowohl als Architekturstil gibt als auch als Entwurfsmuster gibt.

2. Beispiele: Druckerschlange, Protokolldatei, Systemzeit, Systemdesktop als Wurzel für alle Fenster, Standard-out-Stream. Es ist ein (Objekt-) Erzeugungsmuster. Welchen Sinn macht ein Einzelstück? a) Sicherstellen, dass es nur ein Objekt von der Klasse gibt, b) Möglichst spätes Erzeugen dieses Objektes (lazy instantiation), c) Erzeugung auch von nebenläufigen Threads möglich, d.h. es muss nicht festgelegt werden, welcher Thread das Objekt zuerst erzeugt.

3. Dies sind übrigens alles Strukturmuster. Hier nur ein paar Stichworte zur Lösung:  
Stellvertreter: macht einen Zugriff auf eine Implementation über die gleiche Abstraktion mittelbar über ein anderes Objekt (z.B. wegen Remote, Cache, Logging, Locking, Zugriffsrecht, etc.)  
Adapter: passt eine vorgegebene Implementation an eine nicht direkt kompatible Abstraktion an.  
Fassade: fasst mehrere Schnittstellen zu einer Schnittstelle zusammen.  
Brücke: trennt Implementation von Abstraktion (=Interface), um beide variieren zu können.

**Aufgabe 8-2:**

1. Strategie, siehe Klassendiagramm



2. Adapter (auch Wrapper) genannt: Der XAdapter als Unterklasse von Document bildet die Document-Schnittstelle auf die des Typs X ab.

3. (Abstrakte) Fabrik.

4. Stellvertreter (Proxy). Es gibt ein paar ähnliche (Adapter, Fassade), die aber nicht passen.

5. Ablage ist der Index selbst, der bei neu hinzukommenden Dokumenten inkrementell erweitert wird und nicht komplett neu erstellt wird. Ein Datenflussnetz würde in seiner reinen Form immer alle Dokumente neu einlesen und den Index komplett neu erzeugen. Es ist also eine ablagebasierte Architektur angesagt. Aber für Teilkomponenten, z.B. den Analysierer, kann man Datenflussnetz-artig mehrere hintereinander geschaltete Tokenizer/Filter einsetzen, die ein neues Dokument analysieren.

6. Eine schon fertige Komponente suchen und „kaufen“, die die Anforderung erfüllt. Bloß nicht selbst schreiben! In Java ist dies z.B. Lucene (<http://jakarta.apache.org/lucene/>), von dem auch der Entwurf geklaut wurde. ☺

**Aufgabe 8-3: Eine ähnliche Aufgabe war mal Klausuraufgabe.**

Kompositum (composite): JMenu ist hier der Behälter/Kompositum (composite), zu erkennen z.B. an der Beschreibung der add-Methode für JMenuItem, der Oberklasse von Menu.

Beobachter (observer): JMenu ist das (beobachtete) Subjekt (subject), erkennbar an der addMenuListener-Methode

Strategie (strategy): JMenu hat die Strategie für das Austauschen des Look&Feel und ruft sie auf, erkennbar am setUi() in JMenuItem und dem ComponentUI-Argument.

„Halbes“ Kommando (command): Menu hat ein addAction(), wobei ein Action das Kommando (Command) ist. JMenu hat hier die Rolle des Invoker, allerdings wird das Aufrufen des Kommandos an die ActionListener (addActionListener in AbstractButton) übergeben. Dies ist etwas schwer zu erkennen.