

Software Engineering 2 Übungsblatt 1

- Analyse-Pattern -

Ausgabe: 29.09.2021

Besprechung: 06.10.2021 und 10.10.2021

Ablauf der Übungen

- Erfolgreiches Bearbeiten von 2/3 der Pflichtübungsaufgaben und persönliche Vorstellung mindestens einer Aufgabenlösung (auf Nachfrage) berechtigen zur Klausurteilnahme.
- Eine Aufgabe gilt als erfolgreich bearbeitet, wenn eine korrekte Lösung bzw. ein nachvollziehbarer Lösungsversuch termingerecht abgegeben wurden.
- Die Aufgaben sollen in 2er Gruppen bearbeitet werden. Jede Gruppe soll die Ergebnisse dieses Übungsblattes in elektronischer Form festhalten, um diese bei der Diskussion der Lösungen in der Übung per Beamer vorstellen zu können.
- Die Abgabe erfolgt per Moodle vor der Besprechung in der Übung (bitte als PDF-Datei).

Aufgabe 1: Organisationshierarchie-Pattern

In Kapitel 2.3 wurde das Organisationshierarchie-Pattern anhand der Modellierung der Organisationsstruktur eines Unternehmens eingeführt.

Zeichnen Sie ein UML-Objektdiagramm (also eine Instanz des Klassendiagramms aus **Folie 20** mit Objekt-Instanzen und deren Beziehungen zueinander), das die folgenden Beziehungen darstellt:

- Business Unit Europa_Afrika (BU_EA) *ist weisungsbefugt für* Geschäftsbereich Europa (GB_EU).
- Business Unit Europa_Afrika (BU_EA) *ist weisungsbefugt für* Geschäftsbereich Afrika (GB_AF).
- Geschäftsbereich Europa (GB_EU) *ist weisungsbefugt für* Abteilung Deutschland (Abt_DEU).
- Gruppe Hannover (G_HAN) *gehört vertrieblisch zu* Abteilung (Abt_DEU).
- Gruppe Bremen (G_BRE) *gehört vertrieblisch zu* Abteilung (Abt_DEU).

Aufgabe 2: Wissens-Ebene-Pattern (Pflichtaufgabe)

Entwerfen Sie ein Klassendiagramm für ein Software-System zur Bestellung von Notebooks, das den folgenden Sachverhalt abbildet:

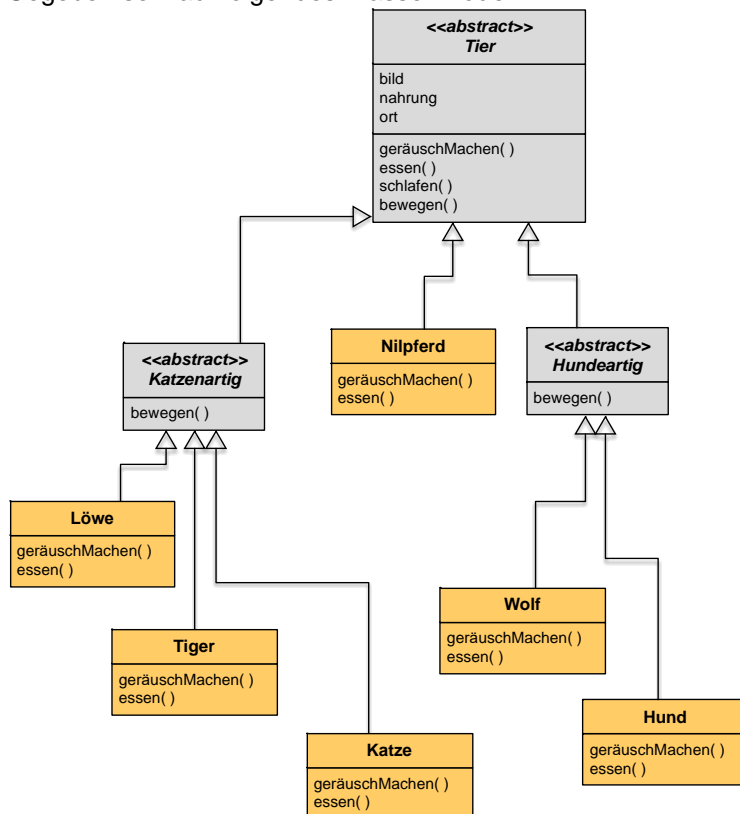
- Ein Kunde kann ein oder mehrere Notebooks bestellen.
 - Jedes Notebook ist von einem bestimmten Typ (z.B. ThinkPad T490, Apple MacBook Pro, Acer Swift etc.)
 - Für jedes Notebook kann eine Menge von Ausstattungsteilen ausgewählt werden (z.B. Grafikkarte, Hauptspeichererweiterung, Stromkabel etc.)
 - Jedes Ausstattungsteil ist von einem bestimmten Typ.
 - Neue Typen für Notebooks als auch für Ausstattungsteile sollen dynamisch zur Laufzeit definiert werden können.
- (a) Entwerfen Sie ein Klassenmodell mit allen Beziehungen und Kardinalitäten und benennen Sie jede Beziehung mit einem geeigneten Namen.
- (b) Erweitern Sie das Klassenmodell um eine Metaebene, so dass Ausstattungsteile eines bestimmten Typs nur für bestimmte Notebooks des richtigen Notebook-Typs gewählt werden können (z.B. Lenovo-ThinkPad Pro Docking Station für ThinkPad T490, aber nicht für Apple MacBooks).

Aufgabe 3: Implementierung des Wissens-Ebene-Pattern (Pflichtaufgabe)

Erweitern Sie Ihr Klassenmodell aus Aufgabe 2 um die notwendigen Methoden zum Pflegen der Beziehungen. Programmieren Sie die Klassen aus und schreiben Sie ein Hauptprogramm, das Ihre Klassen testet und unterschiedlichen Notebooks unterschiedliche Ausstattungsteile zuordnet.

Aufgabe 4: Erweiterung von Vererbungshierarchien (Pflichtaufgabe)

Gegeben sei nachfolgendes Klassenmodell:



Erweitern Sie die Klassenhierarchie um die beiden "Haustier"-Methoden (1) `spielen()` und (2) `kuscheln()`. Katzen und Hunde verhalten sich unterschiedlich beim Spielen und Kuscheln. Mit den anderen Tieren macht Spielen und Kuscheln keinen Sinn, da es sich nicht um Haustiere handelt.

Welche unterschiedlichen Alternativen gibt es, diese Erweiterung zu modellieren? Eine Alternative wäre, die beiden Methoden in der Oberklasse `Tier` zu implementieren. Welches sind die Vor- und Nachteile der jeweiligen Alternativen?