

## Übungen zu Softwaretechnik: Programmierung und Software-Entwicklung

### Aufgabe 7-1

### Entwurfsmuster - Allgemeine Fragen

*Präsenz*

- a) Auf Übungsblatt05 haben Sie bereits das Design Pattern *Template Method* kennen gelernt unter dessen Verwendung es möglich ist für ähnliche Subklassen unterschiedliches Verhalten zu modellieren ohne das gesamte Lösungskonzept ändern zu müssen. Mittlerweile haben Sie noch viele weitere Entwurfsmuster in der Vorlesung kennen gelernt. Was sind die wesentlichen Elemente eines Entwurfsmusters?
- b) Welche Vorteile und welche Nachteile können durch die Verwendung von Entwurfsmustern entstehen?
- c) Außerdem werden die durch die „Gang of Four“ beschriebenen Design Patterns klassifiziert in:
- Creational Patterns
  - Structural Patterns
  - Behavioral Patterns

Welche Probleme adressieren die genannten Kategorien und welcher Kategorie wird das bereits bekannte (und verwendete) Template Method Pattern zugeordnet?

- d) Geben Sie für jede Klassifikation ein Beispiel Pattern an und beschreiben Sie es knapp unter der Verwendung der in a) genannten Elemente.

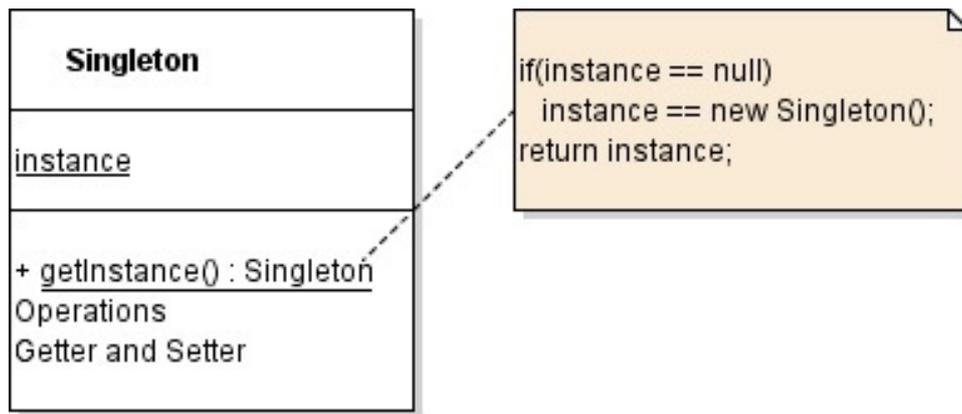


Abbildung 1: Klassendiagramm zu Singleton Design Pattern

- a) In der Vorlesung haben Sie das Singleton Design Pattern kennen gelernt, das laut „Gang of Four“ folgendes machen soll:  
„Ensure a class has only one instance and provide a global point of access to it.“  
In Abbildung 1 ist ein (vereinfachtes) Klassendiagramm für die Umsetzung des Singleton Patterns gegeben. Implementieren Sie eine einfache Klasse **SimpleSingleton**, die das Singleton Pattern umsetzt.
- b) Welche Schwierigkeiten sollten beachtet werden? Kann z.B. mit Ihrer Implementierung für unterschiedliche Anwendungsfälle wirklich immer nur eine Instanz der **SimpleSingleton**-Klasse erzeugt werden?
- c) Implementieren Sie nun die in Teilaufgabe a) beschriebene Klasse **SimpleSingleton** als Enum Type. Beachten Sie, dass Enums in Java vollwertige Klassen sind und sowohl Methoden als auch weitere Felder enthalten können.
- d) Sind die Probleme aus Teilaufgabe b) nun zumindest teilweise behoben?

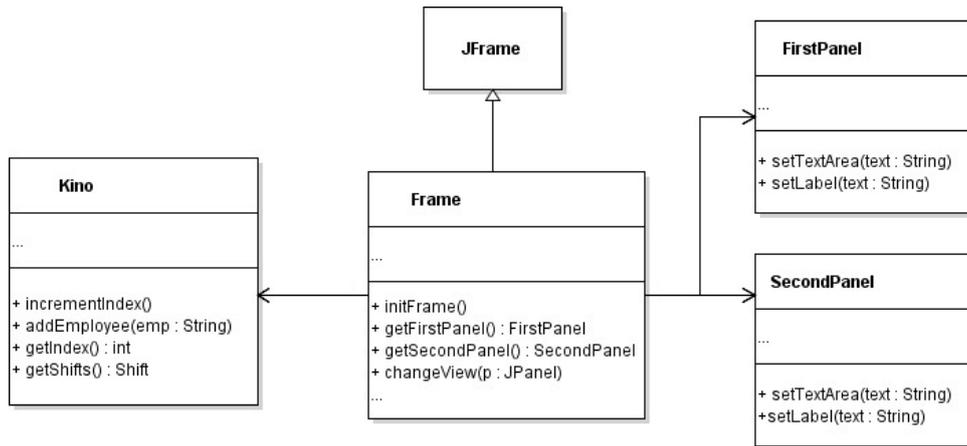


Abbildung 2: Unvollständiges Klassendiagramm zu Observer Pattern

a) Wie Sie sich bestimmt erinnern wurde auf Blatt06 das Architektur Muster MVC anhand von JavaSwing besprochen. Nun soll explizit das *Observer* Entwurfsmuster angewendet werden. Betrachten Sie nur die View und das Model wie in Abbildung 2 dargestellt und erweitern sie das Klassendiagramm entsprechend, so dass das *Observer* Entwurfsmuster sinnvoll umgesetzt wird.

Sollte außerdem noch etwas in der unten abgebildeten *main*-Methode geändert werden?

```

1      public class KinoMain {
2          public static void main(String[] args) {
3              SwingUtilities.invokeLater(new Runnable(){
4                  @Override
5                  public void run(){
6                      Kino kino = new Kino();
7                      Frame frame = new KinoFrame((KinoModel)kino);
8                      frame.initGUI();
9                  }
10             });
11         }
12     }
    
```

b) Geben Sie außerdem die entscheidenden Code-Fragmente aus den betroffenen Klassen an.