

## Software Engineering 2 Übungsblatt 4

### - Entwurfsmuster 3 -

**Ausgabe: 20.10.2022**

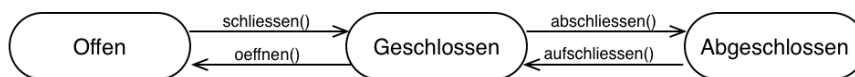
**Besprechung: 27.10.2022**  
**(31.10.2022 keine Besprechung, da**  
**Feiertag (Reformationstag))**

#### 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.
- Die Abgabe erfolgt per Moodle als PDF-Datei vor der Besprechung in der Übung.
- Das PDF-Dokument enthält Namen und Matrikelnummern der an der Lösung beteiligten Personen.
- Das (identische) PDF-Dokument wird von allen an der Lösung beteiligten Personen auf Moodle hochgeladen.
- Das Ergebnis soll in der Übung vorgestellt und diskutiert werden.

#### Aufgabe 4.1: State Pattern

Die Klasse `Tuer`, bspw. eine Haustür, Garagentür, etc. besitzt ein zustandsabhängiges Verhalten. Eine Tür kann offen, geschlossen oder abgeschlossen sein. Eine offene Tür kann man nur schließen, aber nicht öffnen, ab- oder aufschließen. Das Verhalten der Klasse `Tuer` ist durch nachfolgendes Zustandsmodell definiert.



- Modellieren Sie das Verhalten der Klasse `Tuer` mit dem State Pattern.
- Implementieren Sie die Klasse `Tuer` sowie die weiteren notwendigen Klassen. Das Verhalten der vier Methoden soll jeweils durch eine aussagekräftige Meldung beschrieben werden, bspw. der Aufruf der Methode `abschliessen()` im Zustand `Abgeschlossen` ergibt:  

```
System.out.println("Die Tuer ist bereits abgeschlossen und kann kein weiteres Mal abgeschlossen werden.");
```

#### Aufgabe 4.2: Template Pattern vs. Strategy Pattern

In einem Bibliothekssystem gibt es unterschiedliche Ausleihkonditionen für Bücher:

- "Normale" Bücher können für vier Wochen ausgeliehen werden
- Vorgemerkte Bücher haben eine verkürzte Ausleihfrist von zwei Wochen
- Präsenzexemplare können nur für einen Tag ausgeliehen werden

Modellieren Sie eine Klasse `Buch` mit der Methode `leiheAus(Benutzer b)`, die alle Aufgaben für die Ausleihe eines Buches abarbeitet (u.a. Berechtigung prüfen, Rückgabedatum bestimmen, Ausleihe verbuchen, Beleg drucken etc.). Je nach Typ des Buches (normal, vorgemerkt, Präsenz) muss das korrekte Rückgabedatum berechnet werden.

- Skizzieren Sie die Implementierung der Klasse `Buch`, in dem Sie in der Methode `leiheAus()` die unterschiedlichen Ausleihfristen mit einer Fallunterscheidung berechnen.
- Modellieren Sie eine Lösung mit dem Template Pattern. Skizzieren Sie die Implementierung der Klasse `Buch` mit der Methode `leiheAus()`.
- Modellieren Sie eine Lösung mit dem Strategy Pattern. Skizzieren Sie die Implementierung der Klasse `Buch` mit der Methode `leiheAus()`.
- Worin liegt der wesentliche Unterschied zwischen den beiden unterschiedlichen Lösungen?

#### Aufgabe 4.3: Pattern aus Source Code

Gegeben sei folgender Java-Quellcode:

```
interface IOutput {  
    public void sendToOutput(String text);  
}  
  
class OutputChannel implements IOutput {  
    public void sendToOutput(String text) {  
        // write to output channel  
        System.out.println("Ausgabe: "+text);  
    }  
}  
  
class OutputFilter implements IOutput {  
    OutputChannel outChannel = new OutputChannel();  
  
    public void sendToOutput(String text) {  
        if (text.startsWith("ignore")) {  
            // do nothing / ignore text  
        } else {  
            outChannel.sendToOutput(text);  
        }  
    }  
}
```

- Erstellen Sie das Klassendiagramm, das den obigen Quellcode beschreibt.
- Welches Entwurfsmuster wird durch den obigen Quellcode implementiert?
- Für das Entwurfsmuster liefert die GoF-Darstellung bestimmte Rollenbezeichnungen/Klassen-namen (siehe Vorlesungsfolien). Ordnen Sie die GoF-Rollenamen den obigen Klassen zu.