

## Staatsexamen 66116 / 2018 / Frühjahr

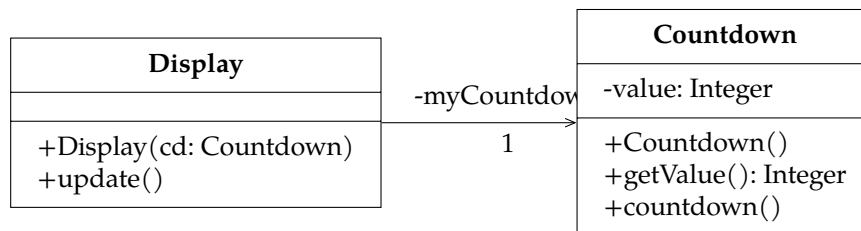
**Thema 2 / Teilaufgabe 2 / Aufgabe 2**

(Countdown und Observer)

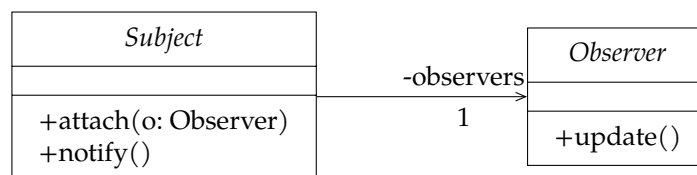
**Stichwörter:** Beobachter (Observer), Klassendiagramm, Implementierung in Java

Es soll eine (kleine) Anwendung entwickelt werden, in der ein Zähler in 1-er Schritten von 5000 bis 0 herunterzählt. Der Zähler soll als Objekt der Klasse `Countdown` realisiert werden, die in UML-Notation dargestellt ist. Das Attribut `value` soll den aktuellen Zählerstand speichern, der mit dem Konstruktor zu initialisieren ist. Die Methode `getValue` soll den aktuellen Zählerstand liefern und die Methode `countdown` soll den Zähler von 5000 bis 0 herunterzählen.

Der jeweilige Zählerstand soll von einem Objekt der in untenstehender Abbildung angegebenen Klasse `Display` am Bildschirm ausgegeben werden. Bei der Konstruktion eines `Display`-Objekts soll es mit einem `Countdown`-Objekt verbunden werden, indem dessen Referenz unter `myCountdown` abgespeichert wird. Die Methode `update` soll den aktuellen Zählerstand vom `Countdown`-Objekt holen und mit `System.out.println` am Bildschirm ausgeben. Dies soll zu Beginn des Zählprozesses und nach jeder Änderung des Zählerstands erfolgen.



Damit das `Display`-Objekt über Zählerstände des `Countdown`-Objekts informiert wird, soll das Observer-Pattern angewendet werden. Untenstehende Abbildung zeigt die im Observer-Pattern vorkommenden abstrakten Klassen. (Kursivschreibweise bedeutet abstrakte Klasse bzw. abstrakte Methode.)



- (a) Welche Wirkung haben die Methoden `attach` und `notify` gemäß der Idee des Observer-Patterns?

Das beobachtete Objekt bietet mit der Methode `attach` einen Mechanismus, um Beobachter anzumelden und diese über Änderungen zu informieren.

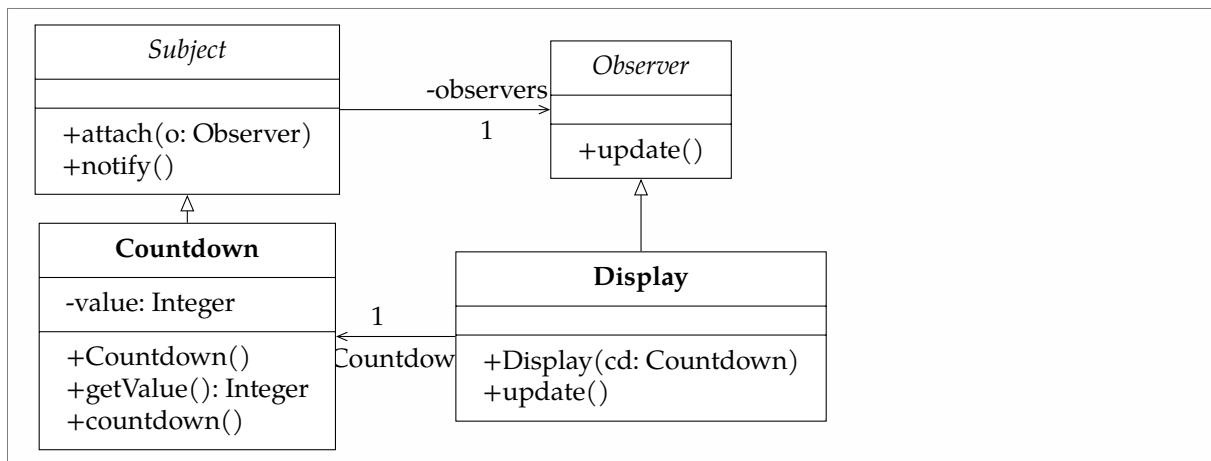
Mit der Methode `notify` werden alle Beobachter benachrichtigt, wenn sich das beobachtete Objekt ändert.

- (b) Welche der beiden Klassen `Display` und `Countdown` aus obenstehender Abbildung spielt die Rolle eines `Subject` und welche die Rolle eines `Observer`?

Die Klasse **Countdown** spielt die Rolle des **Subjects**, also des Gegenstands, der beobachtet wird.

Die Klasse **Display** spielt die Rolle eines **Observer**, also die Rolle eines Beobachters.

- (c) Erstellen Sie ein Klassendiagramm, das die beiden obenstehenden gegebenen Diagramme in geeigneter Weise, entsprechend der Idee des Observer-Patterns, zusammenfügt. Es reicht die Klassen und deren Beziehungen anzugeben. Eine nochmalige Nennung der Attribute und Methoden ist nicht notwendig.



- (d) Unsere Anwendung soll nun in einer objektorientierten Programmiersprache Ihrer Wahl (z. B. Java oder C++) implementiert werden. Dabei soll von folgenden Annahmen ausgegangen werden:

- Das Programm wird mit einer main-Methode gestartet, die folgenden Rumpf hat:

```

public static void main(String[] args){
    Countdown cd = new Countdown();
    new Display(cd);
    cd.countdown();
}
  
```

- Die beiden Klassen **Subject** und **Observer** sind bereits gemäß der Idee des Observer-Patterns implementiert.

Geben Sie auf dieser Grundlage eine Implementierung der beiden Klassen **Display** und **Countdown** an, so dass das gewünschte Verhalten, d.h. Anzeige der Zählerstände und Herunterzählen des Zählers, realisiert wird. Die Methoden der Klassen **Subject** und **Observer** sind dabei auf geeignete Weise zu verwenden bzw. zu implementieren. Geben Sie die verwendete Programmiersprache an.

```

3 public class Client {
4     public static void main(String[] args){
5         Countdown cd = new Countdown();
6         new Display(cd);
7         cd.countdown();
8         cd.countdown();
  
```

```
9      cd.countdown();
10    }
11  }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2018/fruehjahr/Client.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2018/fruehjahr/Client.java)

```
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public abstract class Subject {
7      private final List<Observer> observers = new ArrayList<Observer>();
8
9      public void attach(Observer o) {
10         observers.add(o);
11     }
12
13     public void notifyObservers() {
14         for (Observer o : observers) {
15             o.update();
16         }
17     }
18 }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2018/fruehjahr/Subject.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2018/fruehjahr/Subject.java)

```
3  public abstract class Observer {
4      public abstract void update();
5  }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2018/fruehjahr/Observer.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2018/fruehjahr/Observer.java)

```
3  public class Countdown extends Subject {
4
5      private int value;
6
7      public Countdown() {
8          value = 5000;
9      }
10
11     public int getValue() {
12         return value;
13     }
14
15     public void countdown() {
16         if (value > 0) {
17             notifyObservers();
18             value--;
19         }
20     }
21 }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2018/fruehjahr/Countdown.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2018/fruehjahr/Countdown.java)

```
3 public class Display extends Observer {  
4     Countdown myCountdown;  
5     public Display(Countdown cd) {  
6         myCountdown = cd;  
7         myCountdown.attach(this);  
8     }  
9  
10    public void update() {  
11        System.out.println(myCountdown.getValue());  
12    }  
13 }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2018/fruehjahr/Display.java](https://github.com/hbschlangaul/examen/examen_66116/jahr_2018/fruehjahr/Display.java)

Hilf mit! Das ist ein Community-Projekt. Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind sehr willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net)  
Der  $\text{\LaTeX}$ -Quelltext dieses PDFs kann unter folgender URL aufgerufen werden:

<https://github.com/hbschlangaul/lehramt-informatik/blob/main/Staatsexamen/66116/2018/03/Thema-2/Teilaufgabe-2/Aufgabe-2.tex>