Interfaces

Syntax anhand eines Beispielinterfaces:

```
public interface Kalendereintrag {
      String getStartzeit();
      void verschiebeAuf(String date);
}

public interface AlarmEmpfaenger {
      void alarm();
}
```

Beispielklasse, welche die Interfaces implementiert:

```
public class Termin implements Kalendereintrag, AlarmEmpfaenger {

      private String startzeit;
      private Teilnehmer[] teilnehmerliste;

      public void teilnehmerHinzufuegen(Teilnehmer teilnehmer) {
            ... // teilnehmer in teilnehmerliste aufnehmen
      }

      public String getStartzeit() {
            return startzeit;
      }

      public void verschiebeAuf(String neueStartzeit) {
            startzeit = neueStartzeit; // Termin verschieben
            ...
      }

      public void alarm() {
            ... // zeige Erinnerungspopup am Bildschrim
      }

}
```

Bsp 1: Countdown

Holen Sie sich Beispiel 1 aus der zip-Datei in Ihre Eclipse, bringen Sie das Beispiel zum Laufen und analysieren sie das Programm.

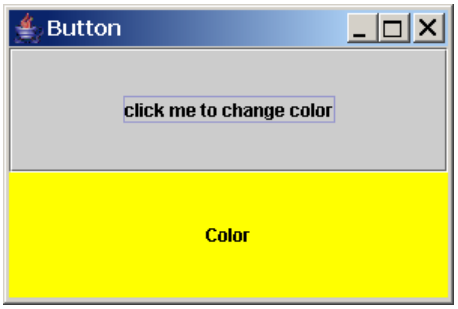
a) Welche Aufgabe hat die Klasse `Countdown` und wozu wird das Interface `CountdownBeobachter` benötigt?
b) Wo gibt es Polymorphismus/dynamisches Binden in diesem Beispiel?
c) Schreiben Sie sich einen eigenen `CountdownBeobachter` und starten Sie ein `Countdown`-Objekt mit Ihrem Beobachter.

Bsp 2: Artikelliste

Holen Sie sich Beispiel 2 aus der zip-Datei in Ihre Eclipse. Schreiben Sie analog der Präsentation des Beispiels eine Implementierung zum Interface `TableContentModel`. Achten Sie darauf, dass Ihre Applikation nicht abstürzt (Ausnahme: Eingabe einer ungültigen Integer/Float-Zahl).
Adaptieren Sie die Klasse `RunArtikelliste`, um das Programm zu starten.

Bsp 3: Button

Holen Sie sich die Angabe aus der Datei „Blatt_09_Angabe.zip“, Verzeichnis Bsp03. Schreiben Sie ein Programm, dass mit Hilfe der Klasse ButtonWindow.createButtonWindow(...) folgendes Fenster darstellt:



Jedes mal wenn der Button gedrückt wird, soll die Farbe der Fläche unter dem Button auf einen Zufallsfarbwert gesetzt werden. Implementieren Sie dazu das Interface `IButtonAction`. Lesen Sie sich die Beschreibung des Interface und der Klasse `Color` durch.