

# TECHNISCHE INFORMATIK 2

---

## OBJEKTLEBENSZYKLUS

- Erstellen Sie eine Klasse „Initializer“ wie folgt:

```
public class Initializer {  
  
    protected String myVal = "leer";  
  
    public Initializer() {  
        initialize();  
    }  
  
    private void initialize() {  
        myVal = "jetzt: " + this.getClass().getName();  
    }  
  
    public void talk(){  
        System.out.println(myVal);  
    }  
}
```

- Legen Sie eine Unterklasse „SubInitializer“ an. Erzeugen Sie je eine Instanz und rufen Sie die Methode talk auf. Wie Interpretieren Sie das Ergebnis?

- Fügen Sie der Klasse SubInitializer einen Konstruktor hinzu, in dem Sie die Methode initialize aufrufen. Was ist das Ergebnis?
- Rufen Sie in den Konstruktoren nach initialize noch talk auf. Was bemerken Sie?
- Geben Sie der Klasse SubInitializer ebenfalls eine Methode „initialize“, in der Sie myVar einen festen Wert zuweisen. Wie verändert sich die Ausgabe?
- Verändern Sie die Sichtbarkeit der initialize Methoden. Welche Kombinationen sind möglich und warum? Wie verändert sich die Ausgabe?

- Fügen Sie der Klasse Initializer eine Methode finalize wie folgt hinzu:

```
protected void finalize(){  
    System.out.println("Beende mich  
    "+this.getClass().getName());  
}
```

- Erzeugen Sie einen Thread, der unheimlich viel Speicher verwendet (z.B. in einer Schleife ein Array anlegen). Starten Sie diesen Thread als letztes in Ihrem main.
- Fügen Sie der Klasse SubInitializer eine Methode finalize hinzu, in der Sie die finalize-Methode der Oberklasse aufrufen und noch eine ergänzende Ausgabe machen.