

# Besser mit Vererbung

Die beiden Arten von Beiträgen sind sich sehr ähnlich und teilen sich zahlreiche Attribute und Methoden. Es existiert also gewissermaßen eine "übergeordnete Klasse" (*Superklasse*, *Super* im Sinne von englisch "oberhalb"), die bereits zahlreiche gemeinsame Aspekte abdeckt. Die Beitragsarten müssen dann nur noch die Unterschiede definieren. Diese Super-Klasse könnte z.B. `Beitrag` heißen, denn die beiden Beitragsarten sind ja auch beides "Beiträge".

---



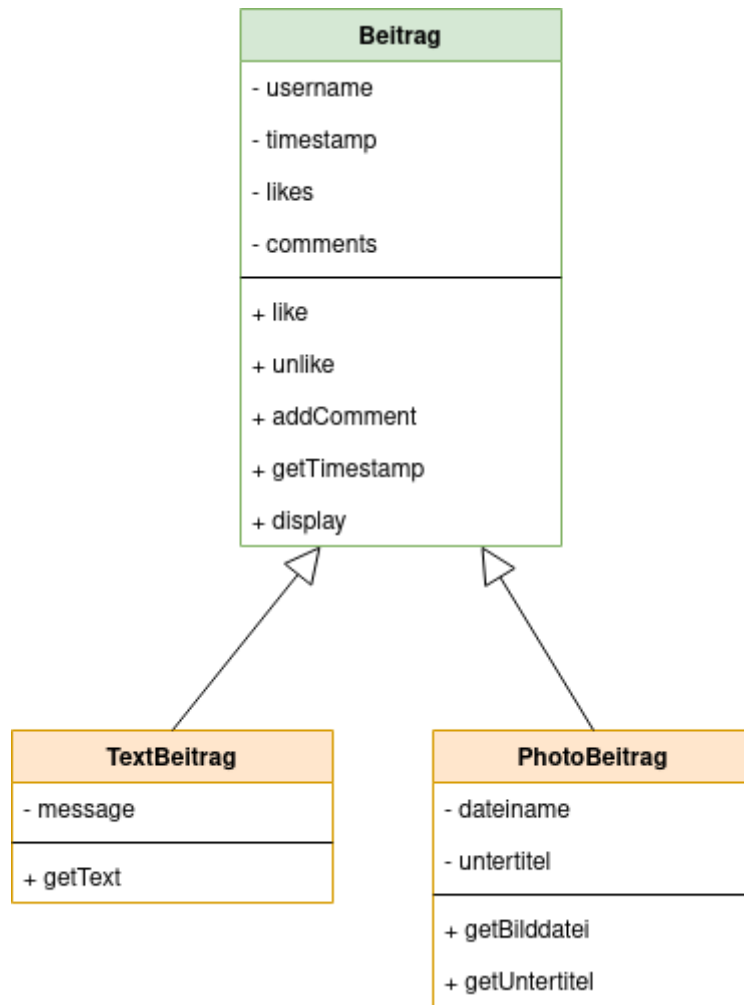
## (A1)

Notiere, welche Attribute/Methoden die Super-Klasse haben sollte und in welchen welchen Attributen und Methoden sich die beiden Arten von Beiträgen unterscheiden:

	<b>Beitrag .....</b>	<b>PhotoBeitrag .....</b>	<b>TextBeitrag .....</b>
Attribute			
Methoden			

## Darstellung im UML-Diagramm

Nun benötigt man noch eine Darstellung, die veranschaulicht, dass die gemeinsamen Attribute und Methoden der Superklasse auch den "abgeleiteten" Klassen zur Verfügung stehen. In einem Entwurfs/Implementationsdiagramm stellt man das durch Pfeile mit leeren Pfeilspitzen von der abgeleiteten Klasse zur Super-Klasse dar:



Das bedeutet, die Klassen **TextBeitrag** und **PhotoBeitrag** **erweitern** die Klasse **Beitrag** - sie **erben** alle Attribute und Methoden der Super-Klasse, können aber eigene Attribute und Methoden hinzufügen.

## Vererbung ganz praktisch

- Definiere die Superklasse: Post
- Definiere Unterklassen für TextBeitrag und PhotoBeitrag
- Die Superklasse legt gemeinsame Attribute und Methoden fest
- Die Unterklassen erben die Eigenschaften der Superklasse.
- Die Unterklassen fügen weitere, unterschiedliche Eigenschaften hinzu.

## Wie sieht das im Java Code aus?

From:  
<https://info-bw.de/> -

Permanent link:  
<https://info-bw.de/faecher:informatik:oberstufe:modellierung:vererbung:besser:start?rev=1636569909>

Last update: 10.11.2021 18:45

