

# 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".

---



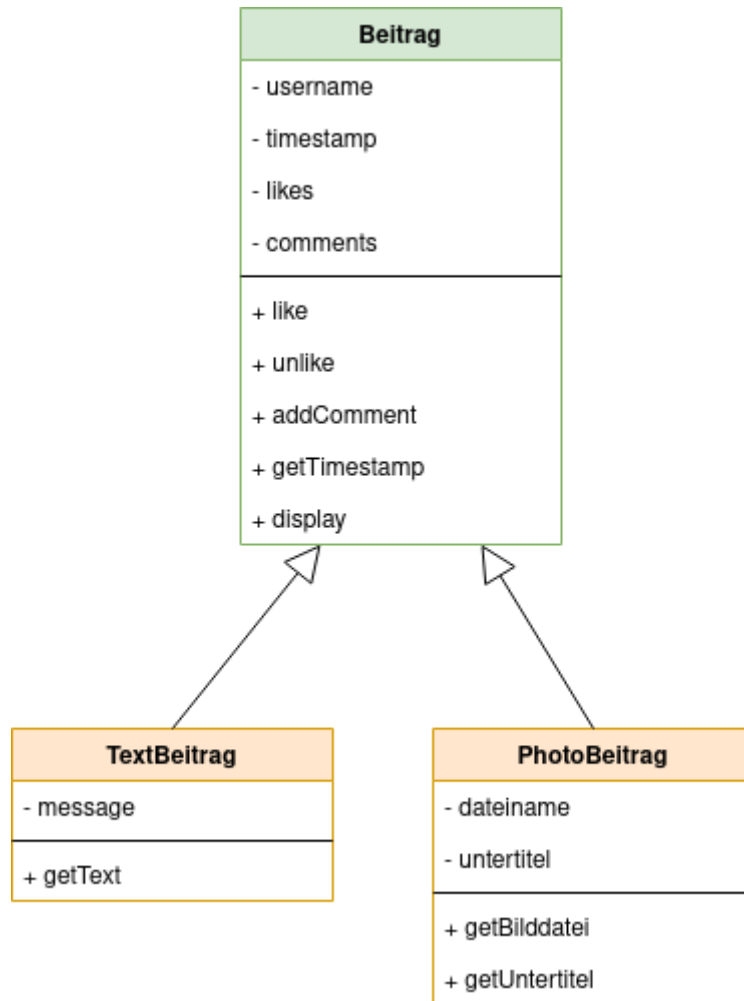
## (A1)

Notiere, welche Attribute/Methoden die Super-Klasse haben sollte und in welchen welche 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.

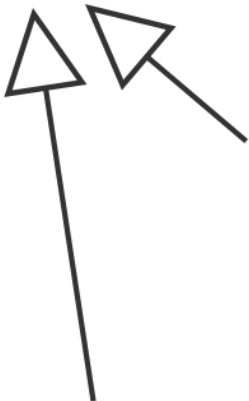
## Vererbung ganz praktisch

- Definiere die **Superklasse**: **Beitrag**
- 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?

```
public class Beitrag  
{  
    ...  
}
```

Die Super-Klasse wird ganz „normal“ implementiert.



```
public class PhotoBeitrag extends Beitrag  
{  
    ...  
}
```

Die abgeleiteten Klassen legen im Klassenkopf fest, dass sie abgeleitet sind – und von welcher Klasse:  
→ **extends Superklasse**

```
public class TextBeitrag extends Beitrag  
{  
    ...  
}
```



(A2)

Implementiere die Klassendefinitionen der Klassen Beitrag, PhotoBeitrag und TextBeitrag mit ihren Attributen, aber zunächst ohne Konstruktoren und Methoden.

### Material

<a href="#">02_vererbung_soz_netz.odp</a>	84.0 KiB	10.11.2021 19:09
<a href="#">02_vererbung_soz_netz.pdf</a>	129.2 KiB	10.11.2021 19:09
<a href="#">auswahl_089.png</a>	59.3 KiB	10.11.2021 18:50
<a href="#">vererbung_soz_net.drawio.png</a>	23.1 KiB	10.11.2021 18:43

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

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

Last update: **12.10.2023 09:56**

