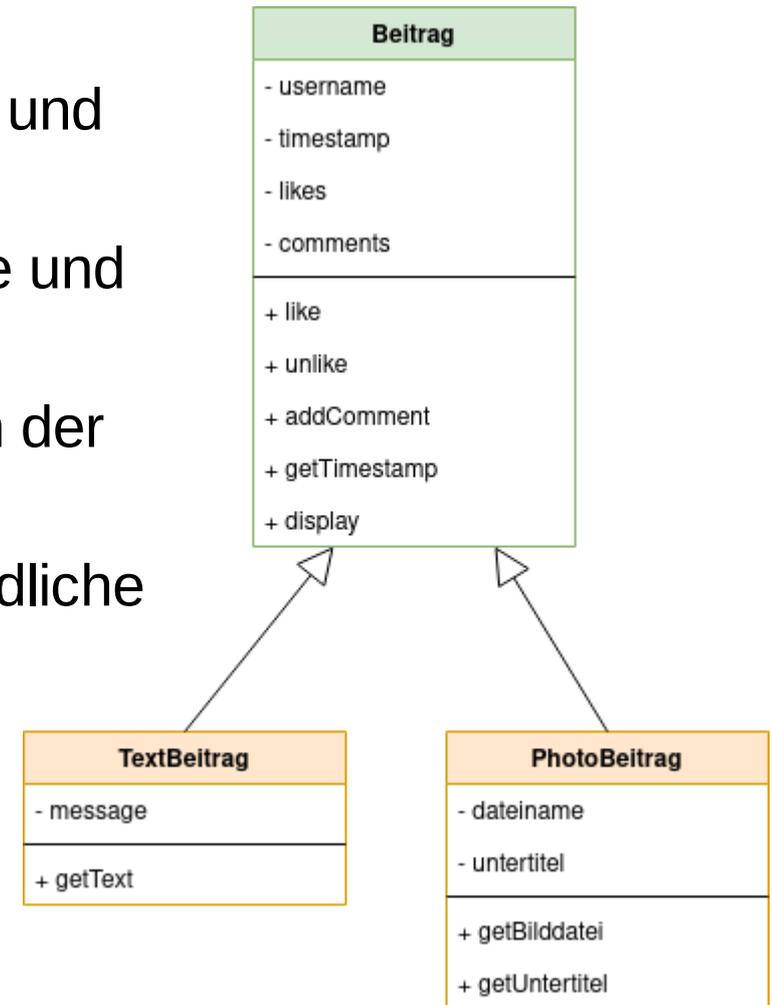
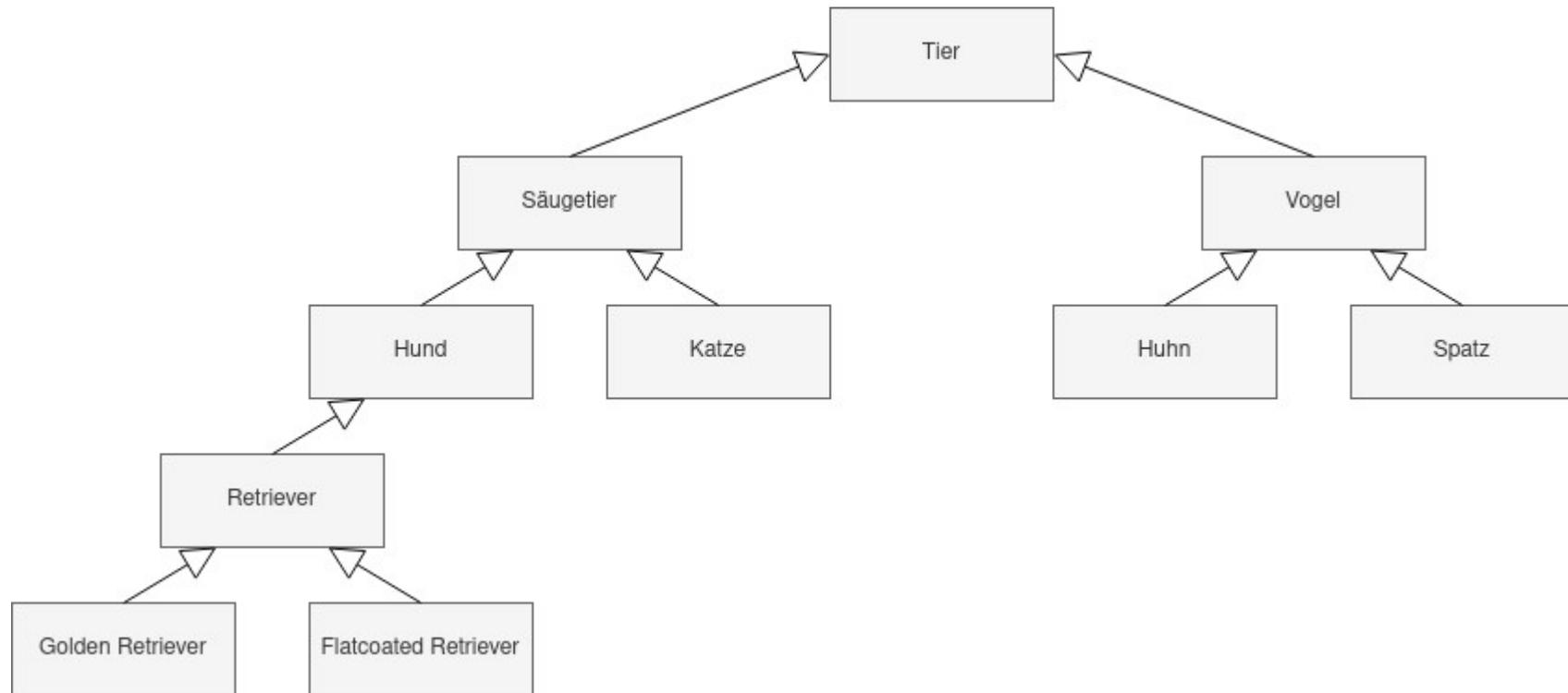


- Definiere die **Superklasse**: Beitrag
- Definiere **Unterklassen** für TextBeitrag und PhotoBeitrag
- Die **Superklasse** legt gemeinsame Attribute und Methoden fest
- Die **Unterklassen** erben die Eigenschaften der Superklasse.
- Die Unterklassen fügen weitere, unterschiedliche Eigenschaften hinzu.



Vererbung: Starkes Hierarchiekonzept zur Modellierung der Miniwelt



Vererbung: Java Code

```
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  
{  
    ...  
}
```

Vererbung: Aufgabe

```
public class Beitrag
{
    private String username;
    private long timestamp;
    private int likes;
    private ArrayList<String> comments;

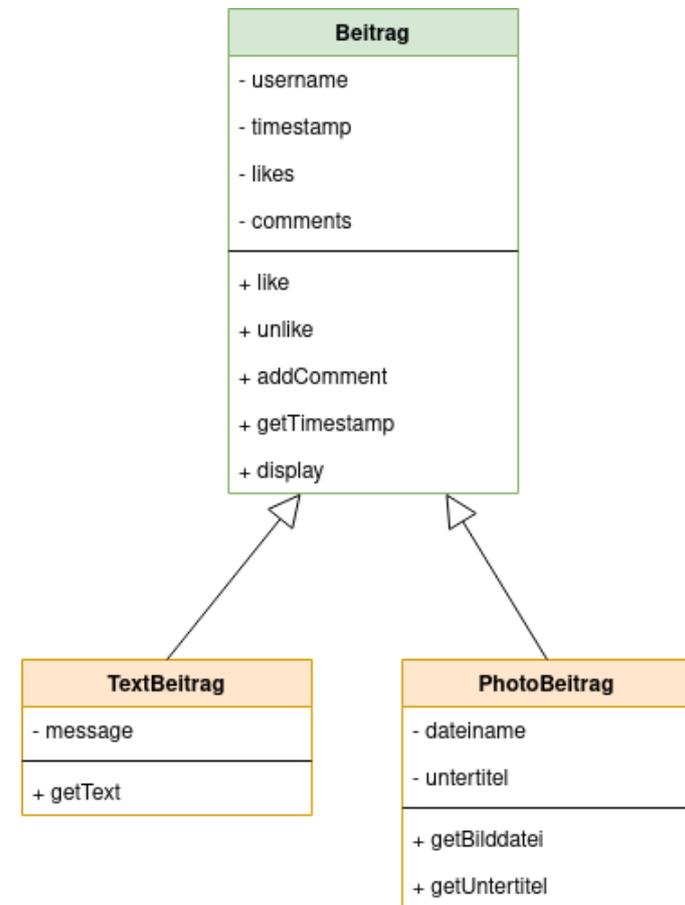
    // Konstruktor und Methoden
}
```

```
public class TextBeitrag extends Beitrag
{
    private String message;

    // Konstruktor und Methoden
}
```

```
public class PhotoBeitrag extends Beitrag
{
    private String dateiname;
    private String undertitel;

    // Konstruktor und Methoden
}
```



Vererbung: Frage

Wo kommt denn da der Konstruktor hin?

```
public class Beitrag
{
    private String username;
    private long timestamp;
    private int likes;
    private ArrayList<String> comments;

    // Konstruktor und Methoden
}
```

```
new Beitrag(...parameter...)
new TextBeitrag(...parameter...)
New PhotoBeitrag(...parameter...)
```

```
public class TextBeitrag extends Beitrag
{
    private String message;

    // Konstruktor und Methoden
}
```

```
public class PhotoBeitrag extends Beitrag
{
    private String dateiname;
    private String undertitel;

    // Konstruktor und Methoden
}
```

