



Vererbung

openHPI-Java-Team

Hasso-Plattner-Institut

Warum Vererbung?

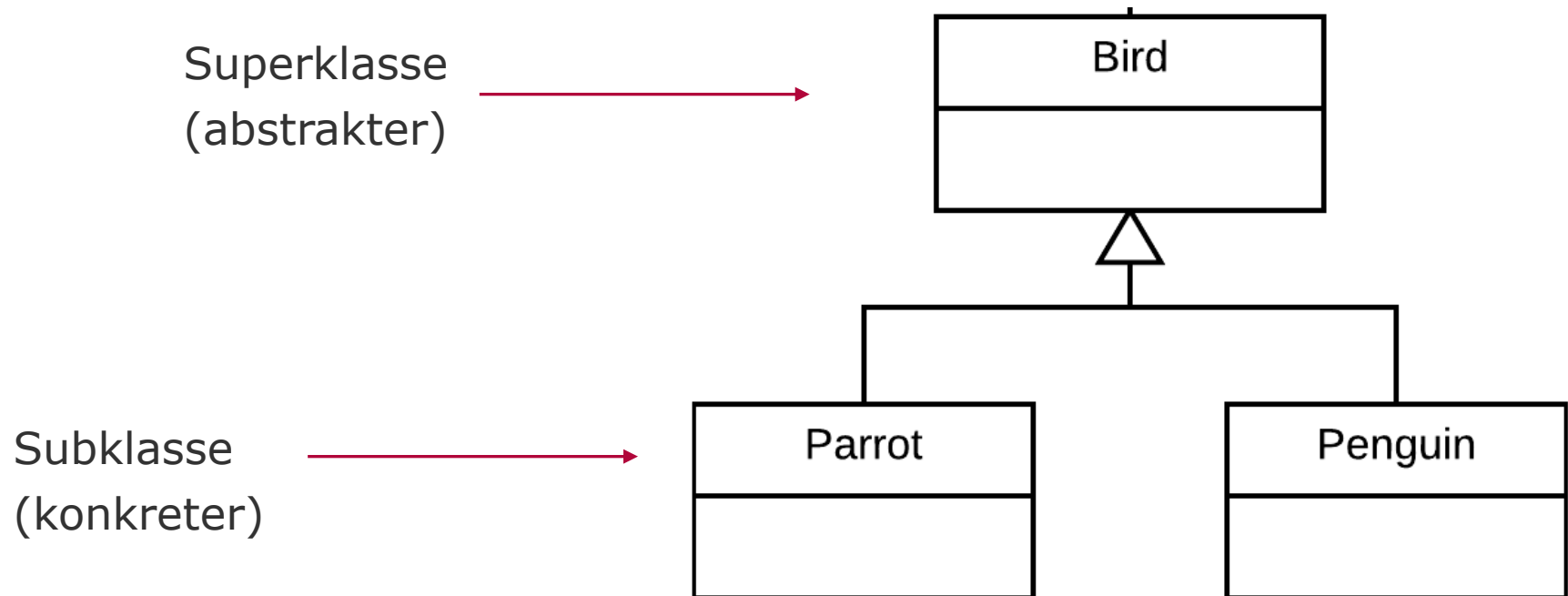


Parrot
name:String species:String dateofBirth: Date featherColors: String
fly() eat() sleep() talk()

Penguin
name:String species:String dateofBirth: Date
swim() eat() sleep()

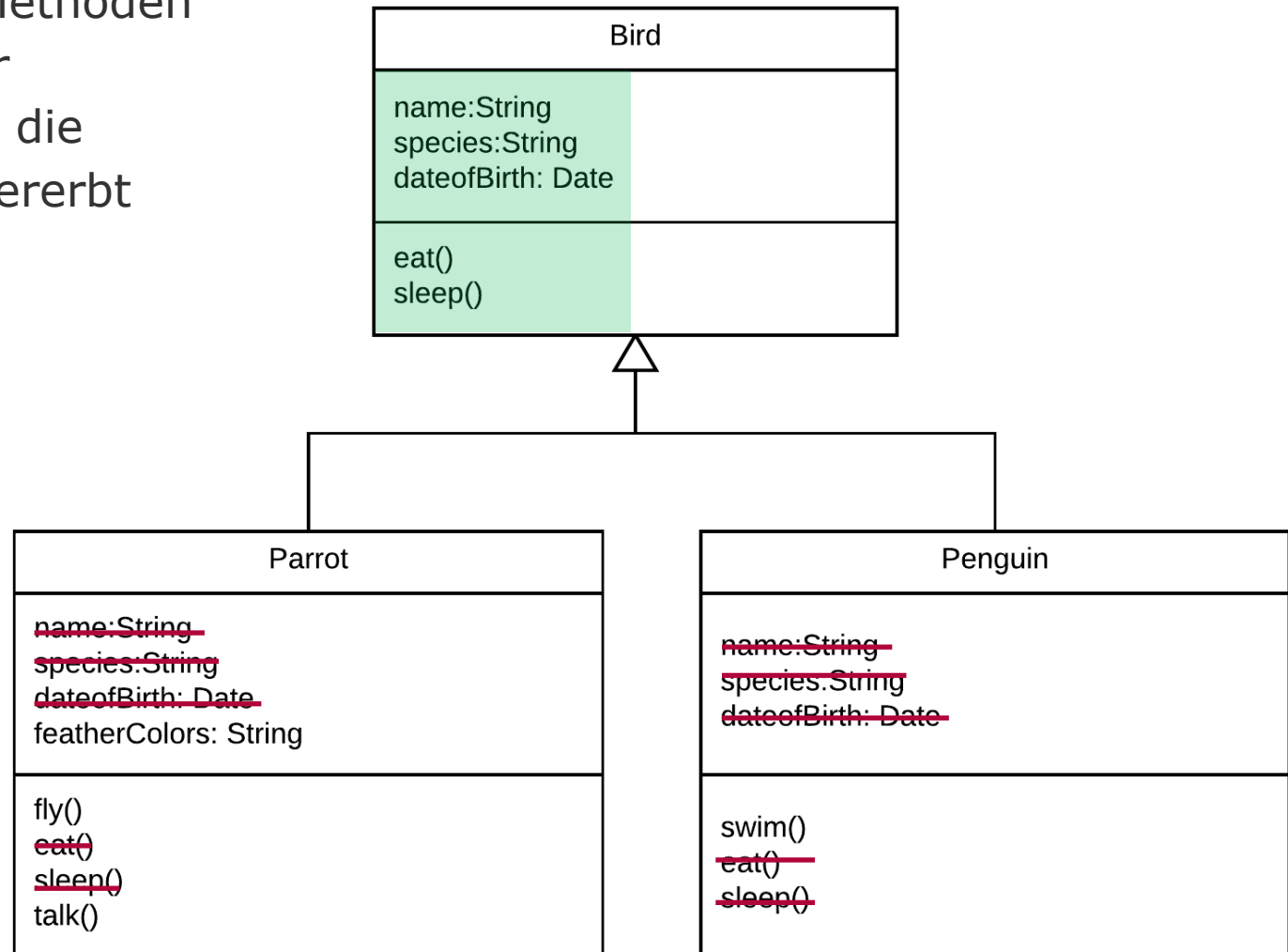


- Ein Papagei ist ein Vogel
- Ein Pinguin ist ein Vogel



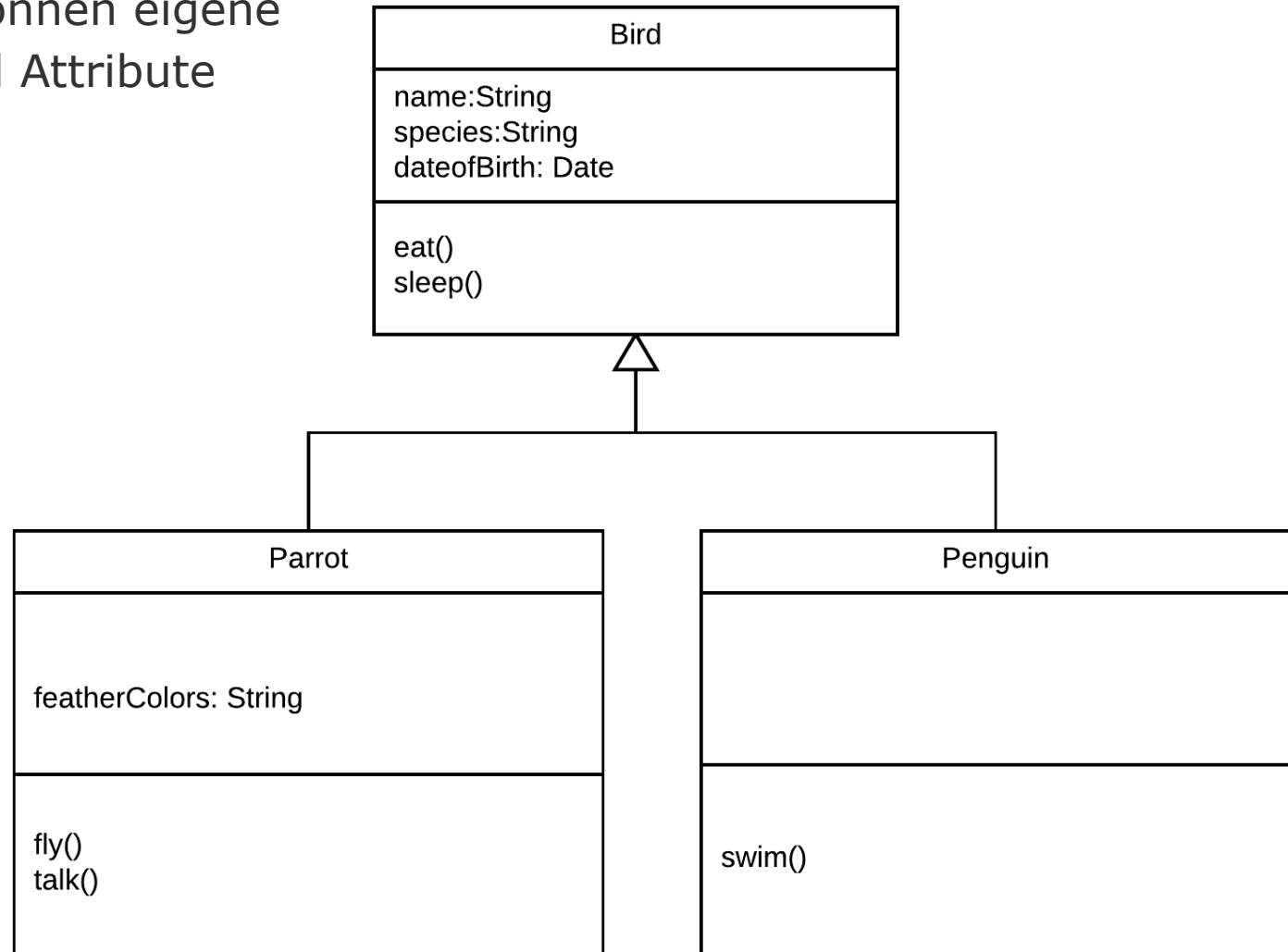


- Attribute und Methoden werden von der Superklasse an die Subklasse(n) vererbt





- Subklassen können eigene Methoden und Attribute ergänzen





```
1 class Bird {  
2     String name;  
3  
4     void sayHello() {  
5     }  
6 }  
1 class Parrot extends Bird {  
2 }
```

Subklassen

- Werden mit **extends** angelegt
- Syntax: **class** <subclass> **extends** <superclass> { }



```
1 class Bird {  
2     String name;  
3  
4     void sayHello() {  
5     }  
6 }  
1 class Parrot extends Bird {  
2 }
```

Subklasse

- Enthält implizit alle Methoden (sayHello()) der Superklasse
- Enthält implizit alle Attribute (name) der Superklasse



Verwendung der Subklasse

```
1 class Story {  
2     public static void main(String[] args) {  
3         Parrot paco = new Parrot();  
4         paco.name = "Paco";  
5     }  
6 }
```

```
1 class Bird {  
2     String name;  
3     // ...  
4 }
```

```
1 class Parrot extends Bird {  
2 }
```

- Die Klasse Parrot hat implizit das Attribut `name` der Klasse Bird



Erweiterung der Subklasse

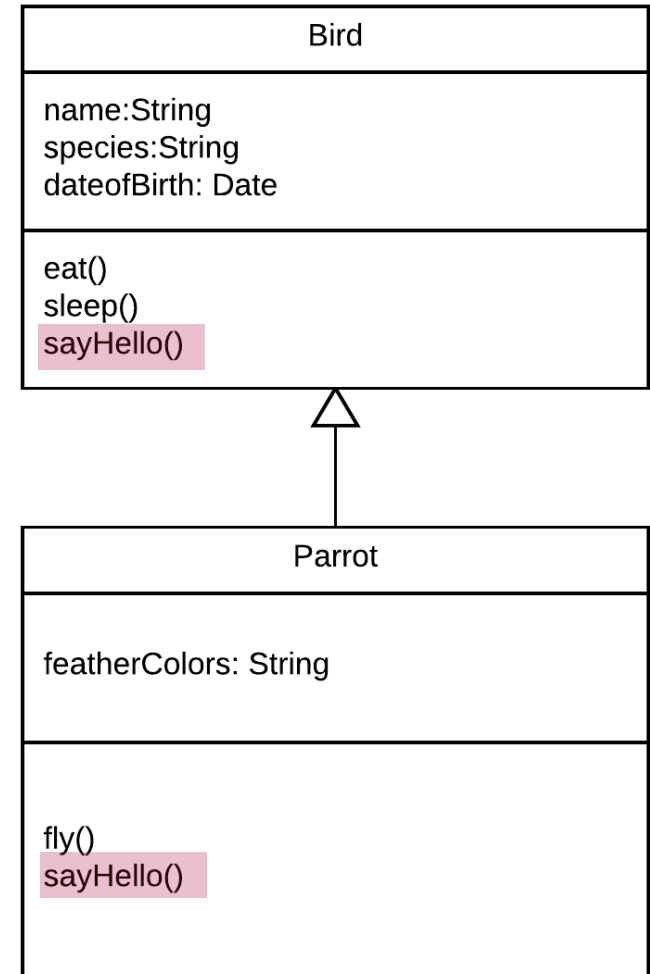
```
1 class Bird {  
2     String name;  
3  
4     void sayHello() {  
5     }  
6 }  
1 class Parrot extends Bird {  
2     String featherColor;  
3 }
```

Subklasse

- Kann Attribute ergänzen (`featherColor`)
- Kann Methoden ergänzen



- Subklassen können Methoden überschreiben und damit geerbtes Verhalten individuell verändern





```
1 class Bird {  
2     String name;  
3  
4     void sayHello() {  
5     }  
6 }
```

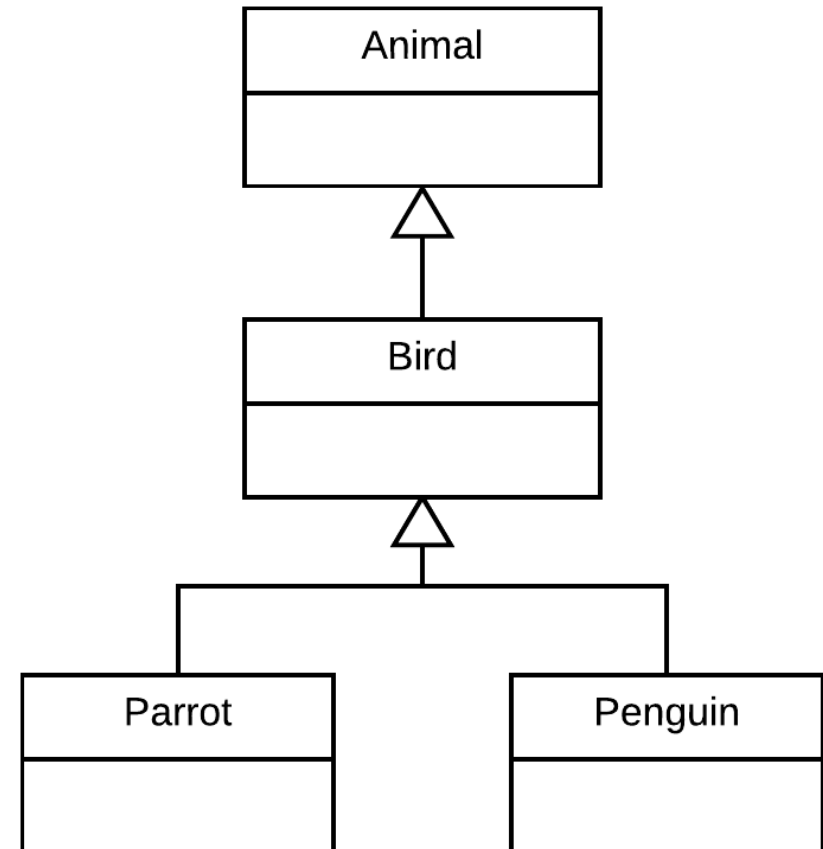
```
1 class Parrot extends Bird {  
2     @Override  
3     void sayHello() {  
4         System.out.println("Hallo Ann Katrin und Tom!");  
5     }  
6 }
```

Subklasse

- Kann Attribute überschreiben
- Kann Methoden überschreiben (sayHello())
- Auch Override genannt



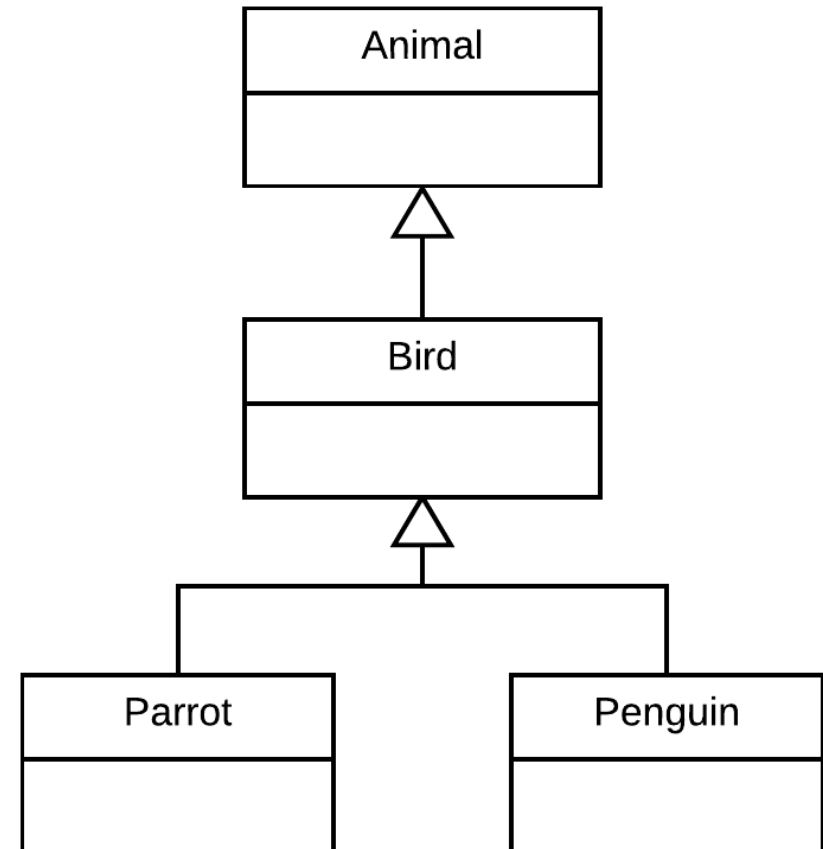
- Ein Papagei **ist ein** Vogel
- Ein Pinguin **ist ein** Vogel
- Ein Vogel **ist ein** Tier
- *Is-a-Beziehung* in einem UML-Klassendiagramm
 - A parrot **is a** bird
 - A parrot **is an** animal
- Die Klasse Parrot erhält transitiv alle Methoden und Attribute der Klasse Animal



In welche Richtung wird vererbt?



Achtung! Vererbung funktioniert nur in eine Richtung. Ein Papagei ist ein Tier. Aber ein Tier ist **kein** Papagei.





Vererbung

openHPI-Java-Team

Hasso-Plattner-Institut