



Abstrakte Klassen

openHPI-Java-Team
Hasso-Plattner-Institut

Motivation abstrakte Klassen



■ Parrot paco = new Parrot();



■ Robot ronja = new Robot();



■ Animal animal = new Animal();



■ Manche Klassen sollten nicht instanziiert werden können!



```
1 abstract public class Animal {  
2     String name;  
3  
4     public void move() {  
5     }  
6 }
```

Abstrakte Klassen

- Werden durch das Schlüsselwort **abstract** angelegt
- Können nicht instanziiert werden
- Müssen "gesubclasst" werden um von diesen Subklassen Instanzen erzeugen zu können



```
1 public abstract class Animal {  
2     String name;  
3  
4     public void move() {  
5     }  
6 }
```

Abstrakte Klassen

- Reihenfolge ob `public abstract` oder `abstract public` ist nicht entscheidend

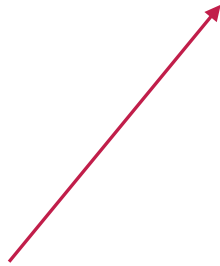


```
1 abstract public class Bird extends Animal {  
2  
3     public void fly() {  
4     }  
5 }
```

- Auch Subklassen von abstrakten Klassen können abstrakt sein



```
1 abstract public class Animal {  
2     String name;  
3  
4     public abstract void eat();  
5 }
```



Abstrakte Methoden

- Werden durch das Schlüsselwort **abstract** angelegt
- Haben keinen Methodenrumpf und enden mit einem Semikolon
- Müssen von der ersten konkreten Subklasse überschrieben werden (und damit definiert)
- Abstrakte Methoden können nur in abstrakten Klassen existieren



```
1 public class Parrot extends Bird {  
2     String name;  
3  
4     @Override  
5     public void eat() {  
6         System.out.println("Ich esse");  
7     }  
8 }
```

Abstrakte Methoden

- Methode `eat()` muss von der ersten konkreten Subklasse überschrieben werden