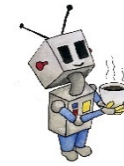




Konstruktoren und „this“

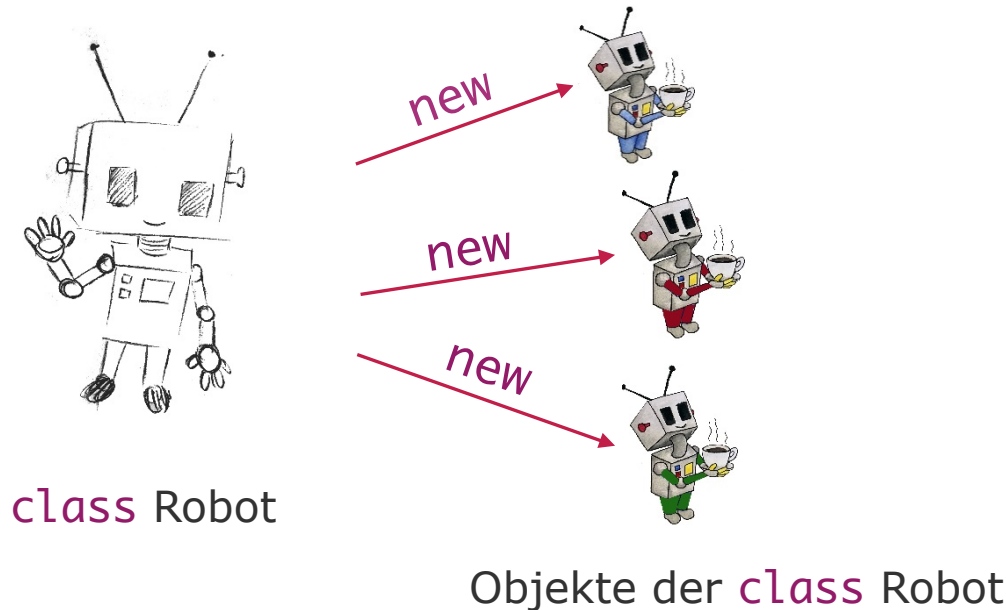
openHPI-Java-Team
Hasso-Plattner-Institut

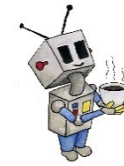


- Was macht eigentlich **new** in:

Robot **robin** **new** Robot(); ?

- **new** ruft **Konstruktor** der Klasse auf, die danach steht
- Konstrukturen **erzeugen** Objekte einer Klasse

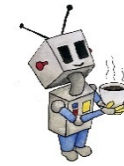




Syntax eines Konstruktors (1/4)

```
1  class Robot{  
2      String name;  
3      int numberOfEyes;  
4  
5      Robot(){  
6          this.name = "Dies ist ein Standard-Name";  
7      }  
8  }
```

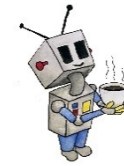
- Kein Rückgabewert
- Gleicher Bezeichner wie Klasse
- **this** wird verwendet, um auf Attribute dieser Klasse zuzugreifen
- Kann **keinen**, einen oder mehrere Parameter haben
- Beispielaufruf: Robot **robin** = new Robot();



Syntax eines Konstruktors (2/4)

```
1  class Robot{  
2      String name;  
3      int numberOfEyes;  
4  
5      Robot(String userDefinedName){  
6          this.name = userDefinedName;  
7      }  
8  }
```

- Kein Rückgabewert
- Gleicher Bezeichner wie Klasse
- **this** wird verwendet, um auf Attribute dieser Klasse zuzugreifen
- Kann keinen, **einen** oder mehrere Parameter haben
- Beispielaufruf: Robot **robert** = new Robot("Robert");

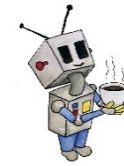


Syntax eines Konstruktors (3/4)

```
1  class Robot{  
2      String name;  
3      int numberOfEyes;  
4  
5      Robot(String name){  
6          this.name = name;  
7      }  
8  }
```

← Gültige Bezeichnung!

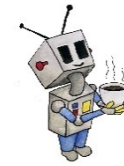
- Kein Rückgabewert
- Gleicher Bezeichner wie Klasse
- **this** wird verwendet, um auf Attribute dieser Klasse zuzugreifen
- Kann keinen, **einen** oder mehrere Parameter haben
- Beispielaufruf: Robot **robert** = new Robot("Robert");



Syntax eines Konstruktors (3/4)

```
1  class Robot{  
2      String name;  
3      int numberOfEyes;  
4  
5      Robot(String userDefinedName, int aNumber){  
6          this.name = userDefinedName;  
7          this.numberOfEyes = aNumber;  
8      }  
9  }
```

- Kein Rückgabewert
- Gleicher Bezeichner wie Klasse
- **this** wird verwendet, um auf Attribute dieser Klasse zuzugreifen
- Kann keinen, einen oder **mehrere** Parameter haben
- Beispielaufruf: Robot **ronja** = new Robot("Ronja", 2);



- Es gibt Default-Konstrukturen
 - Existiert, wenn kein Konstruktor definiert wurde
 - Keine Parameter
 - Alle Attribute haben Default-Werte (int auf 0, String auf null)
 - Wird ungültig, wenn mindestens ein Konstruktor selbst definiert

- **Achtung!** Default-Konstrukturen stehen nicht im Code

- Impliziter Code:

```
Robot(){  
    }  
}
```