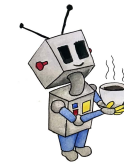


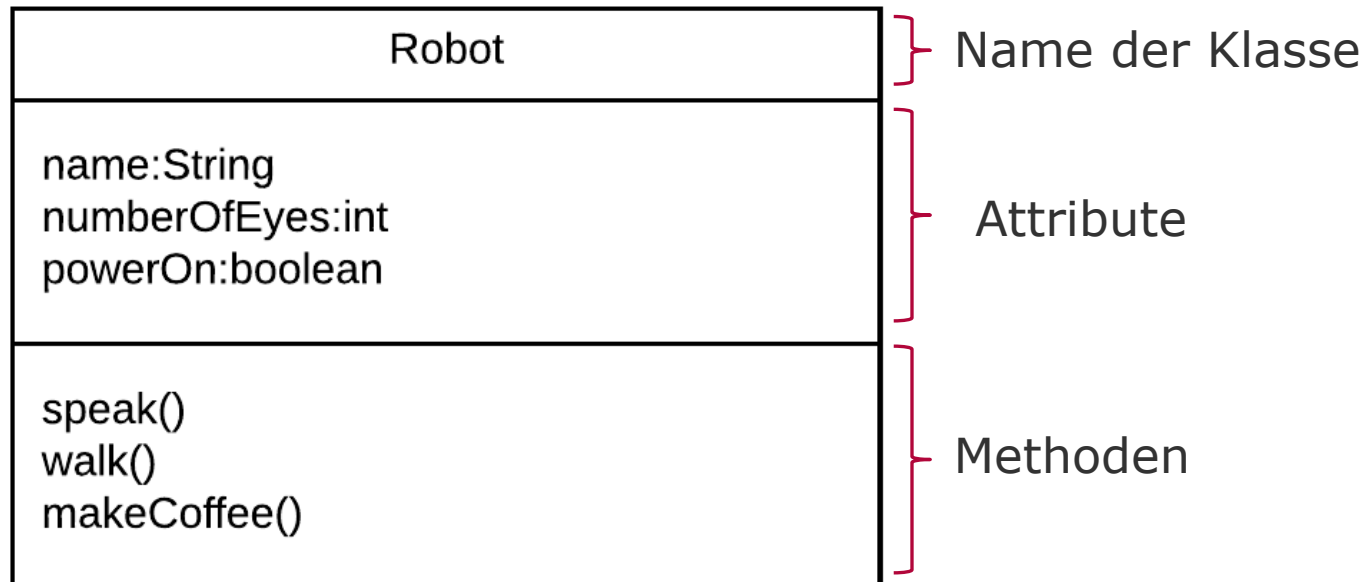


Attribute

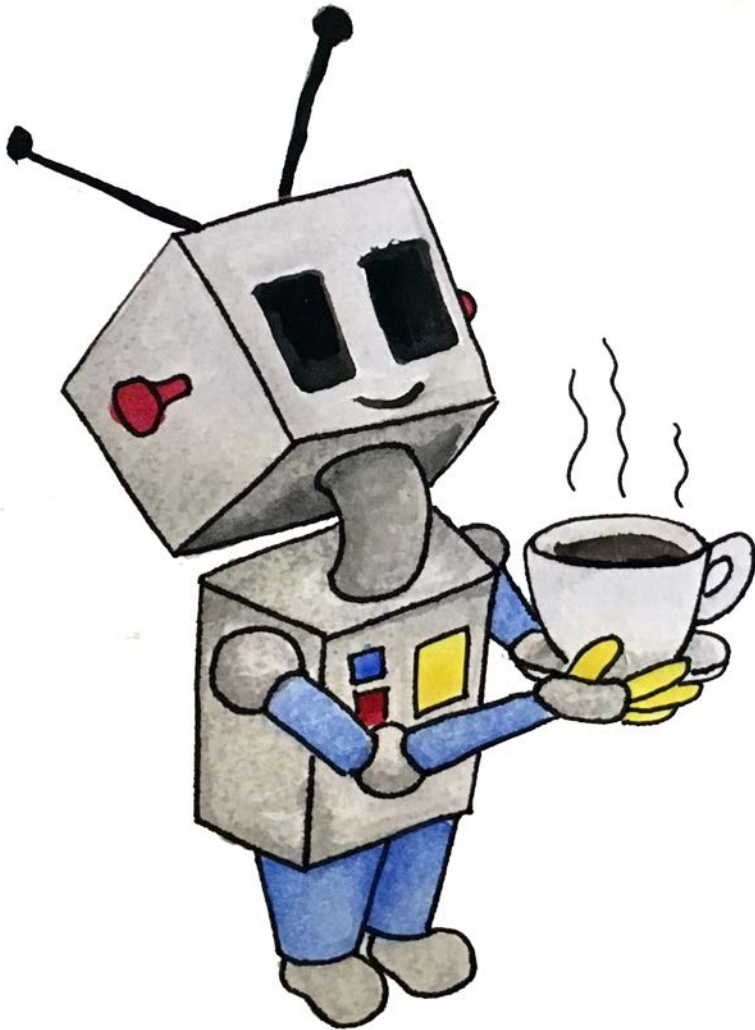
openHPI-Java-Team
Hasso-Plattner-Institut



- Beschreiben Eigenschaften bzw. Merkmale der Klasse
- Format in Java: `<Datentyp> <Bezeichner>;`



Der Name unseres Roboters



Datentyp: String

Bezeichner des Attributs: `name`

Wert: `"Robin"`



```
1 class Robot {  
2     String name;  
3     //more attributes  
4  
5     //methods here  
6 }
```

Attribute

- Werden innerhalb der Klasse definiert
- Alle Objekte dieser Klasse haben jeweils diese Attribute
- Syntax: <Datentyp> <Bezeichner>;



```
1 class Robot {  
2     String name = "Robin";  
3 }
```

Syntax für Attribute

- Mit Defaultwert: <Datentyp> <Bezeichner> = <DefaultWert>;
- Damit wird das Attribut initial für alle Objekte auf den Defaultwert gesetzt
 - Der Wert des Attributes kann für jedes Objekt wieder geändert werden

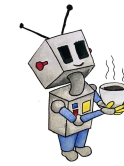


```
1 Robot ourRobot = new Robot();
```

Instanziierung

- Hier wird ein Objekt der Klasse Robot erzeugt
- Der Bezeichner `ourRobot` dient zur Referenzierung des Objektes
- Objekte einer Klasse werden mit
 <Klassenname> <Bezeichner> = `new` <Klassenname>(); instanziiert

Instanziierung von Objekten mit `new`



```
1 class Story{  
2     public static void main(String[] args) {  
3         Robot ourRobot = new Robot();  
4     }  
5 }
```

- Code erzeugt ein Objekt der Klasse Robot
- Durch den Bezeichner `ourRobot` ist dieses Objekt wiederauffindbar



```
1 class Story{
2     public static void main(String[] args){
3         Robot ourRobot = new Robot();
4         ourRobot.name = "Robin";
5         System.out.println(ourRobot.name);
6     }
7 }
```

Ausgabe:

Robin

Punkt-Operator (Dot-Operator)

- Mit dem Punkt-Operator können wir auf Attribute und Methoden zugreifen

Achtung! Zuweisen eines Wertes ist hier notwendig