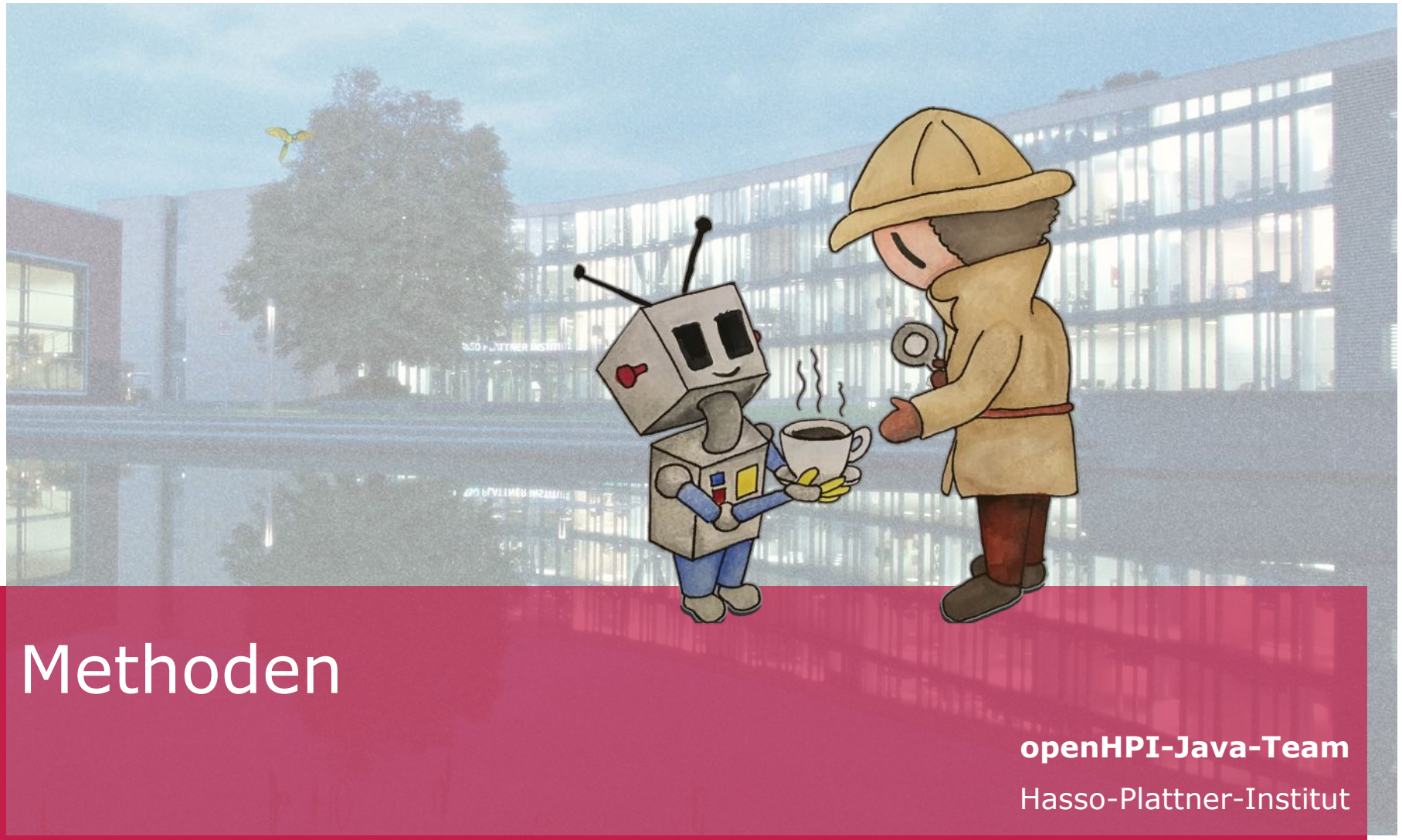


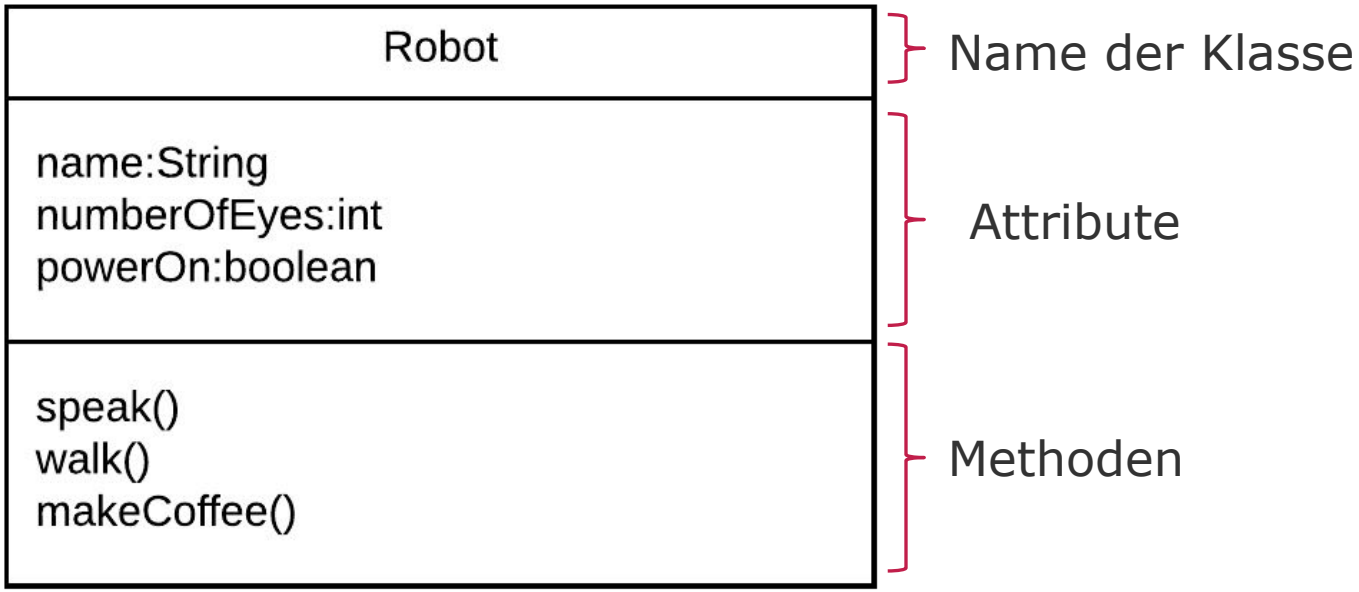
Methoden

openHPI-Java-Team

Hasso-Plattner-Institut

Methoden

- Spezifizieren Operationen für alle Objekte einer Klasse
- Methodendefintionen erfolgen **immer** innerhalb von Klassen
- Methoden definieren Verhalten



Implementierung

```
1 class Robot{
2    void speak(){
3        System.out.println("Hallo");
4    }
5 }
```

Methoden

- Syntax: <Rückgabetyp> <methodenName>(){}
- In den Codeblock zwischen { und } schreiben wir Anweisungen
  - Diese werden ausgeführt, wenn die Methode aufgerufen wird

Beispiel: leere Rückgabe

```
1 class Robot{
2    void speak(){
3       System.out.println("Hallo");
4    }
5 }
```

void

- Bedeutet, dass die Methode nichts zurück gibt
- “leere Rückgabe”

Methodenaufrufe außerhalb der Klasse

Hasso Plattner Institut

```
1 class Robot{
2    void speak() {
3        System.out.println("Hallo");
4    }
5 }
6 class Story{
7    static void main(String[] args){
8        Robot robin = new Robot();
9        robin.speak();
10   }
11 }
```

1. Programm startet in main()
2. Initialisieren Objekt robin
3. Aufruf von speak() auf dem Objekt robin (der Klasse Robot)

Methodenaufrufe außerhalb der Klasse

Hasso Plattner Institut

```
1 class Robot{
2    void speak() {
3        System.out.println("Hallo");
4    }
5 }
6 class Story{
7    static void main(String[] args){
8        Robot robin = new Robot();
9        robin.speak();
10   }
11 }
```

Allgemeine Form: `objectName.methodName();`

Ausgabe:

```
Hallo
```

Methodenaufrufe außerhalb der Klasse

Hasso Plattner Institut

```
1 class Robot{
2    void speak() {
3        System.out.println("Hallo");
4    }
5 }
6 class Story{
7    static void main(String[] args){
8        Robot robin = new Robot();
9        robin.speak();
10   }
11 }
```

- Der Methodenaufruf von speak() erfolgt aus der Klasse Story auf dem Objekt der Klasse Robot
  - also außerhalb der Klasse in der speak() definiert wurde

Methodenaufrufe innerhalb der Klasse

Hasso Plattner Institut

```
1 class Robot {
2    void sayHelloWorld() {
3        sayHello();
4        System.out.println("Welt");
5    }
6    void sayHello() {
7        System.out.println("Hallo");
8    }
9 }
```

1. Aufruf von sayHelloWorld() von außen
2. sayHelloWorld() ruft sayHello() auf
3. Nach Ausführen von sayHello() wird der Rest von sayHelloWorld() ausgeführt

Methodenaufrufe innerhalb der Klasse

HPI Hasso Plattner Institut

```
1 class Robot {
2    void sayHelloWorld() {
3        sayHello();
4        System.out.println("Welt");
5    }
6    void sayHello() {
7        System.out.println("Hallo");
8    }
9 }
```

Allgemein: `methodenName();`
ruft Methoden innerhalb der selben Kasse auf

Ausgabe:

```
Hallo
Welt
```