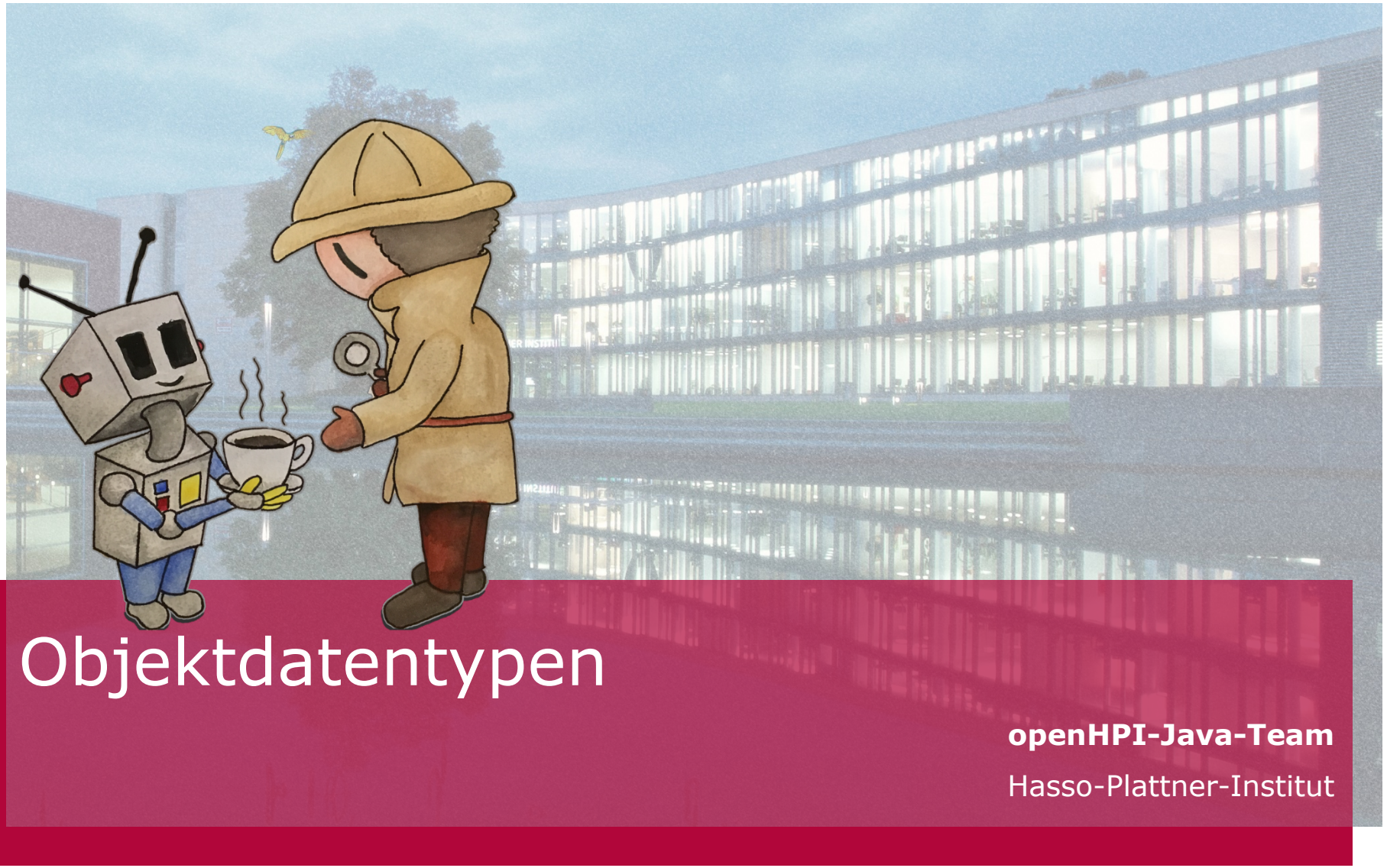


Objektdatentypen

openHPI-Java-Team

Hasso-Plattner-Institut

Primitive Datentypen (Wiederholung Woche 1)

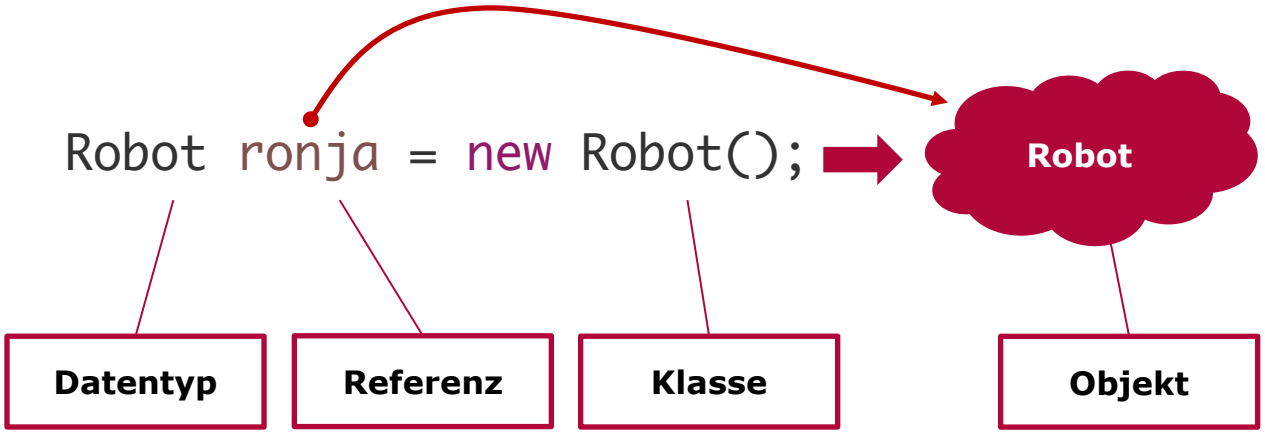
HPI Hasso Plattner Institut

- Erkennbar: kleingeschrieben, Schlüsselwörter
- Keine Methoden oder Attribute

Name	Art	Beispiele
`char`	Buchstabe, Zeichen	`char country = 'd';` `char cedille = 'ç';`
`int`	Ganzzahl	`int age = 2;` `int truth = -42;`
`double`	Kommazahl	`double speed = 98.7;`
`boolean`	Wahrheitswert	`boolean isGreen = true;`

Objektdatentypen

- **Jede Klasse ist auch ein (Objekt-)Datentyp**
  - Bereits bekannt: `String` → ist ein Objektdatentyp in der Java API
  - https://docs.oracle.com/javase/8/docs/api/
- Robot ist ein von uns selbstdefinierter Objektdatentyp

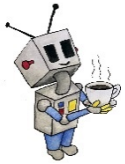


equals und toString (1/3)

- Jede Klasse hat standardmäßig u.a. die zwei Methoden
  - `equals(Object x)`
    - Vergleicht zwei Objekte
    - Standard: Vergleich der Speicheradressen der Objekte
    - Für eigene Klassen können andere Eigenschaften maßgeblich sein
  - `toString()`
    - Repräsentiert ein Objekt als String
    - Standard: *Klassenname@Speicheradresse*
    - Rückgabe sollte an die jeweilige Klasse angepasst werden

➔ **Ändern für eigene Klassen sinnvoll**

equals und toString (2/2)



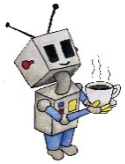
Hasso Plattner Institut

```java
public class Parrot /* extends Object */ {
   private String name;
   private int age;

   @Override
   public boolean equals(Object o) {
       // [...] sicherstellen dass o vom richtigen Typ ist
       // Deep Dive
       return (o != null) &&
              this.name.equals(((Parrot) o).name) &&
              this.age == ((Parrot) o).age;
   }

   [...]
}
```

equals und toString (2/3)



HPI Hasso Plattner Institut

```java
public class Parrot /* extends Object */ {
    private String name;
    private int age;

    [...]

    @Override
    public String toString() {
        String newline = System.getProperty("line.separator");
        String output = "Parrot: " + newline;
        output += "Name: " + this.name + newline;
        output += "Age: " + this.age + newline;
        return output ;
    }
}
```

Wrapper

- Jeder primitive Datentyp hat einen passenden Objektdatentyp
- Bieten zusätzlich:
  - Methoden, die spezifisch für diesen Datentyp sind → z.B. `Integer.parseInt(String s)` zum Umwandeln eines Strings
  - Konstanten, z.B. `Integer.MAX_VALUE` → 2147483647

Primitiver Datentyp	Zugehöriger Objektdatentyp
`int`	`Integer`
`double`	`Double`
`boolean`	`Boolean`
`char`	`Character`