

PK

Von Object abwärts

Object

alle Klassen direkt oder indirekt von Object abgeleitet
und alle Klassen erben Methoden von Object:

<code>String toString()</code>	Umwandlung in Zeichenkette
<code>boolean equals(Object)</code>	Vergleich auf Gleichheit
<code>int hashCode()</code>	gleiche Objekte \Rightarrow gleiche Hashwerte
<code>Class getClass()</code>	Klasse des Objekts (= dynamischer Typ)
<code>Object clone()</code>	Erzeugen einer Kopie des Objekts
<code>void finalize()</code>	Aufräumen vor Speicherfreigabe, kaum verwendet
<code>wait, notify, notifyAll</code>	Unterstützung für nebenläufige Programmierung

Implizite Umwandlung in Zeichenkette

ist Operand von + Zeichenkette, anderer Operand in Zeichenkette umgewandelt

bei Referenztypen Umwandlung mittels toString:

```
String s = "Prüfung " + b + "! ";
```

entspricht

```
String s = "Prüfung " + b.toString() + "! ";
```

pragmatisch: Überschreiben von toString nötig

equals, hashCode und toString

folgende Bedingungen müssen gelten:

$$a.equals(a)$$
$$a.equals(b) \Leftrightarrow b.equals(a)$$
$$a.equals(b) \Rightarrow a.hashCode() == b.hashCode()$$
$$a.equals(b) \not\Leftarrow a.hashCode() == b.hashCode()$$

Ebenso:

$$a.equals(b) \not\Leftarrow a.toString() == b.toString()$$

Daher hashCode und toString nicht für Objektvergleiche geeignet

Aufgabe: Warum Verantwortlich?

Warum müssen wir beim Programmieren `toString`, `equals` und `hashCode` selbst implementieren? Warum macht das System das nicht automatisch?

- A: Stimmt nicht, diese Methoden sind ohnehin vorgegeben.
- B: Weil man diese Methoden meist ohnehin nicht braucht.
- C: Weil das System die Semantik der Objekte nicht kennt.
- D: Weil eine automatische Lösung viel zu ineffizient wäre.

Aufgabe: Parametertyp?

Warum hat der formale Parameter von `equals` den Typ `Object` und nicht denselben Typ wie die Klasse, in der `equals` steht?

- A: `equals` steht in `Object`, Parameter hat daher denselben Typ.
- B: Damit man Objekte aller Typen miteinander vergleichen kann.
- C: Weil man beim Überschreiben den Parametertyp nicht ändern kann.
- D: Damit Programmierer(innen) zum Nachdenken angeregt werden.