



Java Einführung

Javas größte Stolpersteine

Groß-/Kleinschreibung

- Java unterscheidet strikt zwischen Groß- und Kleinbuchstaben. Z.B.: in der Datei `HelloWorld.java` muss stehen:
`class HelloWorld { ... }`
- **Achtung:** gilt auch für den den Aufruf von
`javac HelloWorld.java`
`java HelloWorld`
- Wurde ein Fehler mit der Groß-/Kleinschreibung gemacht, wird entweder die `main`-Funktion oder die Klasse nicht gefunden!

Zuweisung oder Vergleich?

- Zuweisungsoperator =
- Vergleichsoperator ==

*Seit Java 1.5 kein Problem mehr!
Typumwandlung zu boolean wurde
abgeschafft.*

Der Zuweisungsoperator wird gerne fälschlich bei Vergleichen verwendet (ein = fehlt).

Ergebnis: das Programm "arbeitet falsch".

Bsp: `if (a=3) { ... }` //ist eine Zuweisung!

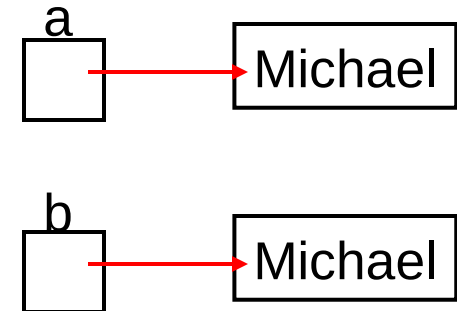
Die Bedingung evaluiert immer true, da eigentlich a der Wert 3 zugewiesen wird!

Der Vergleich wäre `(a==3)`

Vergleich bei String/Objekt-Datentyp

- Die Werte von Strings, Arrays und Objekt-Datentypen (Instanzen) kann man nicht mit dem Vergleichsoperator `==` vergleichen! Dabei würde nur die Referenz (der Zeiger) verglichen werden.

```
String a="Michael";  
String b="Mich" + "ael";  
if (a==b) {...} // vergleicht  
//Zeiger und ergibt daher immer false!
```



Richtig wäre:

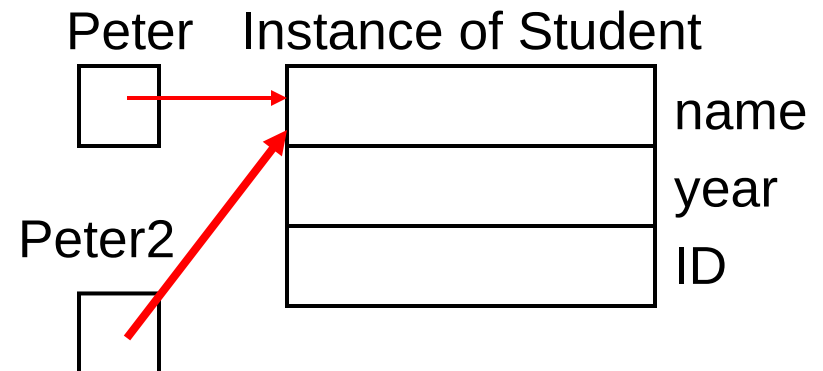
```
if (a.equals(b)) {...} // Methode vergleicht  
// zeichenweise
```

Bei Objekt-Datentypen (Klassen) muss eine Methode `equals()` selbst implementiert werden.

Zuweisung bei Objekt-Datentypen

- Wird bei Instanzen (oder Arrays) der Name einer neuen Variable zugewiesen, wird nicht der Inhalt der Instanz kopiert, sondern nur die Referenz wird kopiert. Die Instanz existiert nur 1x, hat aber dann 2 Namen!

```
Student Peter = new Student();  
/* Annahme: Student ist eine  
bereits deklarierte Klasse */  
Student Peter2;
```



```
Peter2 = Peter; /* Peter und Peter2  
weisen jetzt beide auf die selbe Instant. Wird etwas bei  
Peter2 geändert, wirkt diese Änderung auch bei Peter! */
```

Autoincrement/decrement ++/--

- Achtung diese unären Operatoren verhalten sich bei gleichzeitiger Zuweisung unterschiedlich je nachdem ob sie als Prefix- oder Postfixoperator verwendet werden.
- Prefix: `m=++n;` ist wie `n++; m=n;`
- Postfix: `m=n++;` ist wie `m=n; n++;`
Achtung: Es wird zuerst zugewiesen und dann erst erhöht!