Universität Bremen

Praktische Informatik 1
Objektinteraktion 1

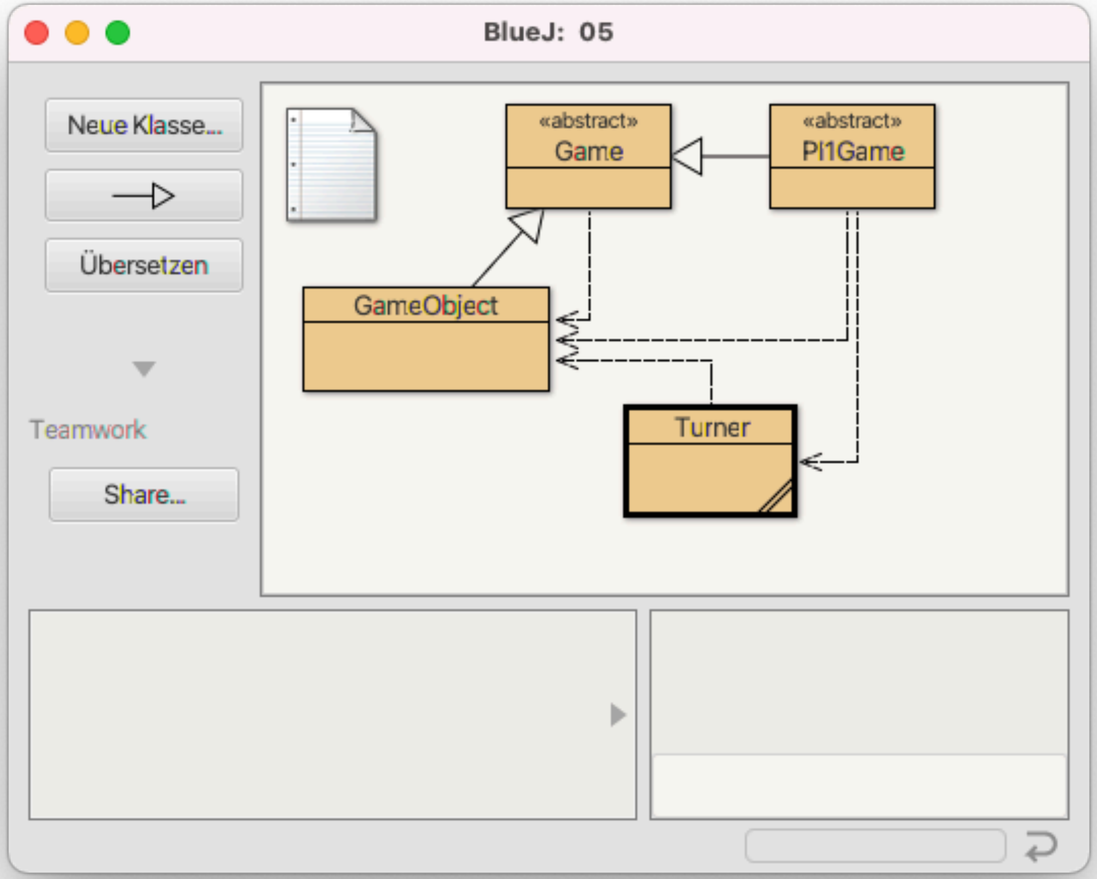
Thomas Röfer

Cyber-Physical Systems
Deutsches Forschungszentrum für Künstliche Intelligenz

Multisensorische Interaktive Systeme
Fachbereich 3, Universität Bremen

Bild: ©Michael Ihle

Kommentare: Demo



Universität Bremen

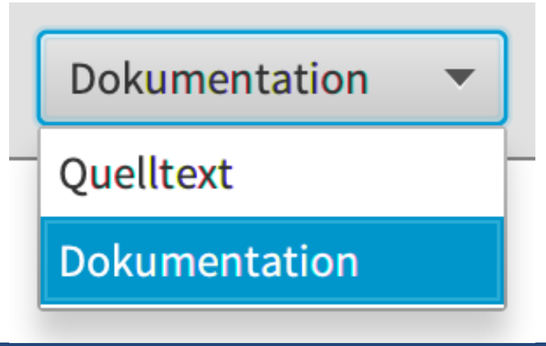
Kommentare

- Methoden sollten dokumentiert sein (Klassen und Attribute auch)
  - Was tut die Methode?
  - Welche Parameter hat sie und was bedeuten sie?
  - Welches Ergebnis liefert sie?
- Der Java-Compiler ignoriert Kommentare, aber der Dokumentations-Compiler wertet **/**** diese Kommentare ***/** aus
  - **/**** … ***/** kann unmittelbar **vor** Klassen, Attributen, Konstruktoren und Methoden stehen

```
/* Kommentar mitten im Code
   (auch mehrzeilig) */
```

```
// Kommentar bis Zeilenende
```

```
/**
 * Diese Methode definiert
 * das Verhalten der Person.
 */
void act()
{
    :
```



Universität Bremen

Kommentare: JavaDoc

- In **/**** JavaDoc-Kommentaren ***/** können spezielle Markierungen verwendet werden, z.B.
  - **@author** *Autor:in*: Die Autor:in dieser Klasse (pro Autor:in)
  - **@param** *Parameter Erklärung*: Erläuterung eines Parameters einer Methode bzw. eines Konstruktors (pro Parameter)
  - **@return** *Erklärung*: Erläuterung der Rückgabe einer Methode
- JavaDoc ignoriert ***** am Anfang von Zeilen und (standardmäßig) private Attribute und Methoden

```
/**
 * Diese Klasse steuert eine
 * sich drehende Person.
 * @author Thomas Röfer
 */
class Turner //...
```

```
/**
 * Setzen der x-Koordinate.
 * @param x Die neue x-
 *          Koordinate.
 */
void setX(final int x) // ...
```

```
/**
 * Abfragen der x-Koordinate.
 * @return Die x-Koordinate.
 */
int getX() // ...
```

Universität Bremen

Uhrenbeispiel

- Anzeige einer Digitaluhr modellieren
  - 24-Stunden-Anzeige
  - Stunden und Minuten durch **:** getrennt

Universität Bremen

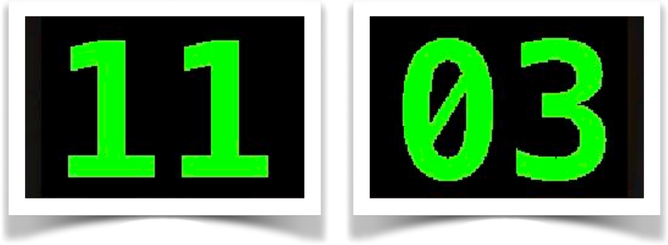
Abstraktion und Modularisierung

- Abstraktion: Details ignorieren, um Gesamtbild erfassen zu können
- Modularisierung: Zerlegung großer Dinge in kleinere Teile (**Teile und herrsche**)

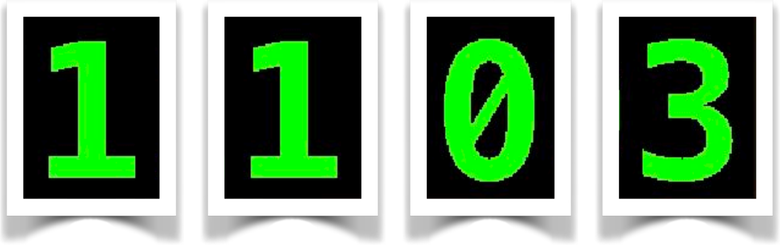
Modularisierung der Uhrenanzeige

- Anzeige mit 4 Ziffern?

- Zwei Anzeigen mit je 2 Ziffern?

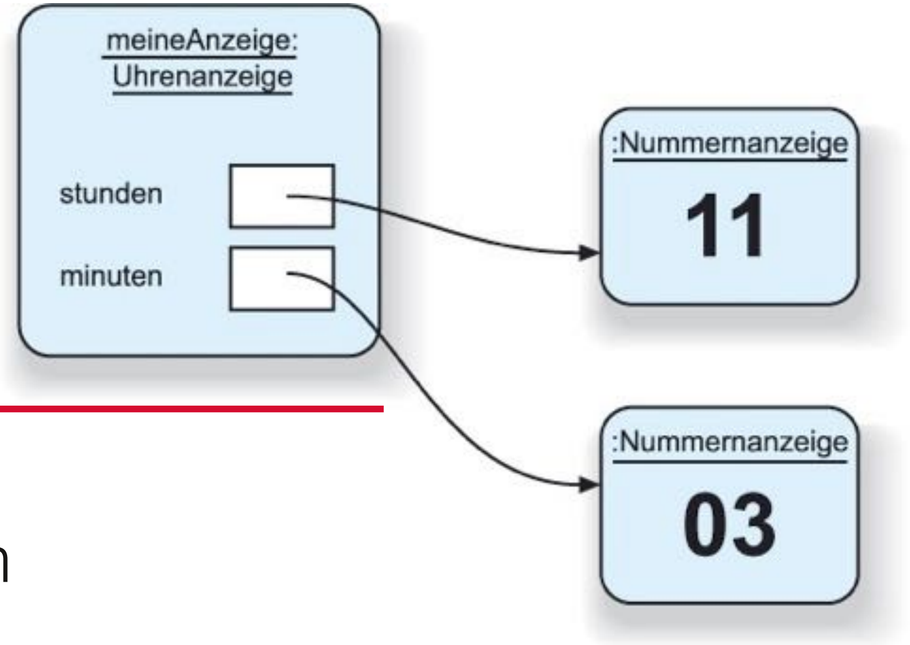


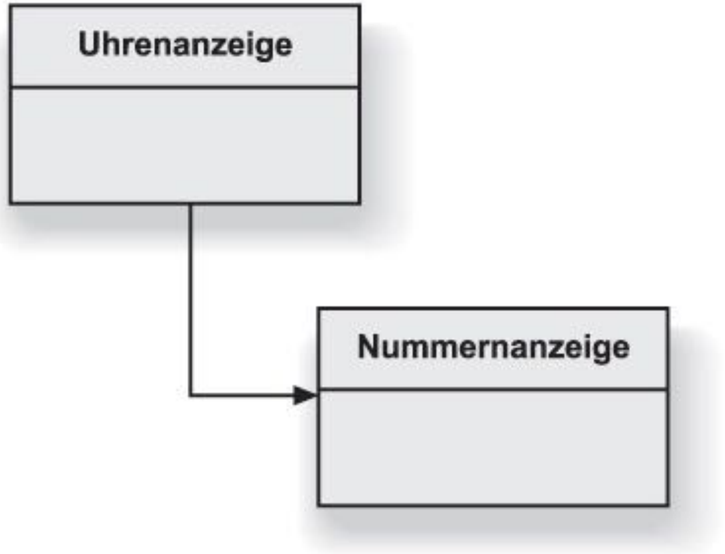
- Vier Anzeigen mit jeweils einer Ziffer?



Klassendiagramme und Objektdiagramme

- Objektdiagramm
  - Dynamische Sicht (zur Laufzeit)





- Klassendiagramm
  - Statische Sicht (über den Quelltext)
  - Beziehung zwischen Klassen einer Anwendung

Nummernanzeige: Demo

Universität Bremen

Logische Operatoren

- Verknüpfen boolesche Werte und liefern ein boolesches Ergebnis
- Die wichtigsten logischen Operatoren:
  - **&&** (**und**) *a* **&&** *b* ist **true** gdw. sowohl *a* als auch *b* **true**
  - **||** (**oder**) *a* **||** *b* ist **true** gdw. *a* **oder** *b* oder beide **true**
  - **!** (**nicht**) **!** *a* ist **true** gdw. *a* **false** ist
- **&&** und **||** werten ihren rechten Operanden nur aus, wenn das Ergebnis noch von ihm abhängt

```
if (ratio != 0
        && percent < 100 / ratio) {
```

```
if (ratio == 0
        || percent < 100 / ratio) {
```

Logische Operatoren: Wahrheitstabellen

	false	true
false	false	false
true	false	true

&&

	false	true
false	false	true
true	true	true

||

false	true
true	false

!

	false	true
false	true	false
true	false	true

==

	false	true
false	false	true
true	true	false

!=

Universität Bremen

Logische Operatoren und Modulo-Operator

- **||**, **&&** und **!** verhalten sich für **boolean** analog zu **+**, ***** und **-** (Vorzeichen) für **int**
  - Sie können verkettet werden: **a && b && c && d**
  - Sie können gemischt werden: **a || b && !c**
  - Es gilt **&&**- vor **||**-Rechnung analog zu Punkt- vor Strichrechnung
- **a % b** berechnet den Divisionsrest
  - Es gilt: **(a / b) * b + a % b == a**

9 % 2 == 1

8 % 2 == 0

Universität Bremen

String-Konkatenation

- Strings können mit **+** aneinander gehängt (**konkateniert**) werden
- **+** ist **überladen** und wird von links nach rechts ausgewertet

"Hallo" + "Welt"	→ "HalloWelt"
"Route" + " " + 6 + 6	→ "Route 66"
6 + 6 + " Monkeys"	→ "12 Monkeys"

- Wird **+** auf einen String und einen Wert eines **anderen Typs** angewendet, wird dieser zuerst in einen String umgewandelt und dann konkateniert
- Bei Objekten findet die Umwandlung durch Aufruf der Methode **toString()** statt

Universität Bremen

Methoden mit Rückgabewerten

- Die **return**-Anweisung beendet die Ausführung der aktuellen Methode und gibt einen Wert an den **Aufrufer** zurück

```
int gibMirFünf()
{
    return 5;
}
```

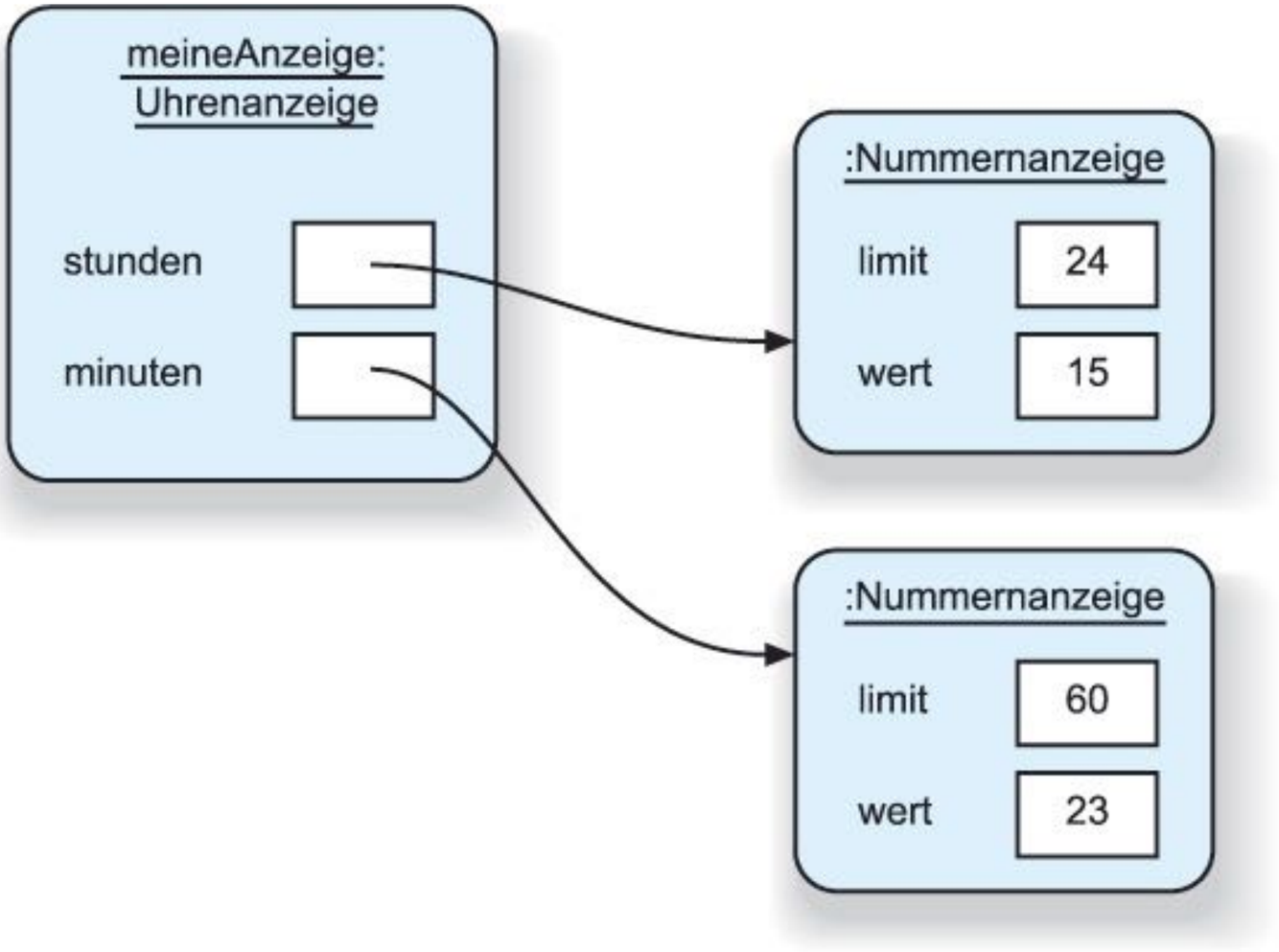
- Der **Typ** des Ausdrucks hinter **return** muss kompatibel zum **Rückgabetyp** der Methode sein
- Jeder mögliche **Ausführungspfad** durch die Methode muss in einer **return**-Anweisung enden
- Methoden ohne Rückgabe können mit **return;** vorzeitig beendet werden

```
int fakultät(final int n)
{
    if (n == 0) {
        return 1;
    }
    else {
        return n * fakultät(n - 1);
    }
}
```

Universität Bremen

Uhrenanzeige: Demo

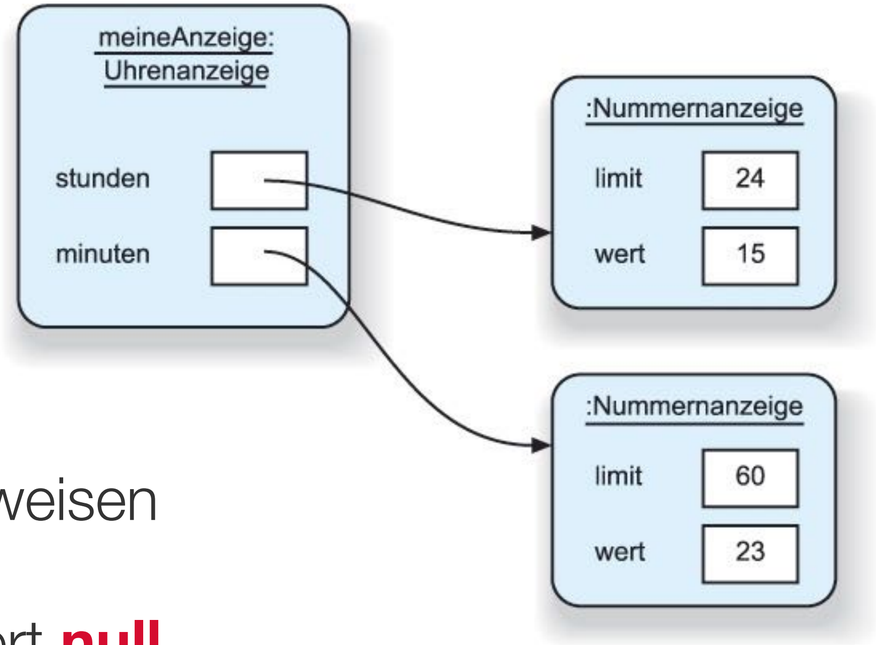
Objekte erzeugen Objekte



Universität Bremen

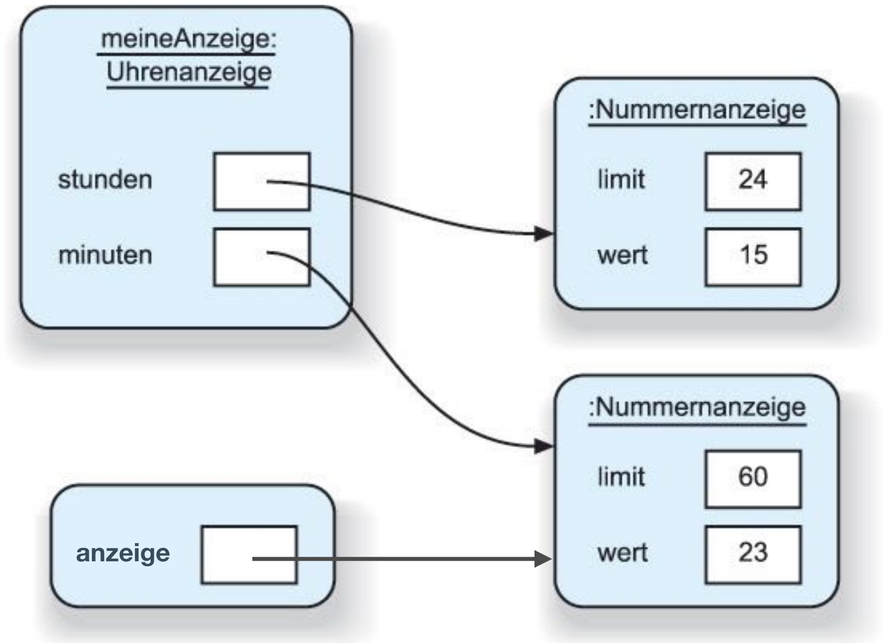
Referenzen

- Werte von primitiven Typen werden direkt in Variablen gespeichert
- Objekte werden **per Referenz** gespeichert
  - Sie werden eigenständig abgelegt
  - In Variablen steht nur, wo sie abgelegt sind
  - Mehrere Variablen können auf **dasselbe Objekt** verweisen
- Referenzen, die auf **kein Objekt** zeigen, haben den Wert **null**
- **==** und **!=** vergleichen die Referenzen, nicht die Objekte



Referenzen: Beispiel

- Während des Aufrufs von **erhöhe(minuten)**:




```
class Uhrenanzeige
{
    // ...
    void taktsignalGeben()
    {
        if (erhöhe(minuten)) {
            erhöhe(stunden);
        }
        anzeigeAktualisieren();
    }

    private boolean erhöhe(final Nummernanzeige anzeige)
    {
        anzeige.erhöhen();
        return anzeige.gibWert() == 0;
    }
}
```

Universität Bremen

Zusammenfassung der Konzepte

- **Kommentare** und **JavaDoc**
- **Abstraktion** und **Modularisierung**
- **Klassendiagramm** und **Objektdiagramm**
- **Logische Operatoren** und **Modulo-Operator**
- **return-Anweisung**
- **Referenzen**