

Kurs 20022: Objektorientiertes Programmieren

Errata: Fehler im Kursmaterial vom SS 2011 (Stand: 8. August 2011)

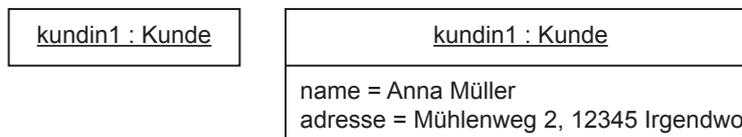
Anbei finden Sie eine Liste bekannter Fehler in den Kursunterlagen (die wir selbstverständlich bei der nächsten Auflage der Materialien beheben werden). Finden Sie weitere, hier noch nicht verzeichnete Fehler, senden Sie uns bitte eine Fehlerbeschreibung an oop.tutor@fernuni-hagen.de zu, so dass wir den Kurs verbessern und auch Ihre Kommilitonen von Ihrer Beobachtung profitieren können. Vielen Dank für Ihre Unterstützung.

Kurseinheit 1, Abschnitt 3.6, S. 33 f.

In einem Objektdiagramm muss der optionale Name und der mit einem Doppelpunkt abgetrennte Typ unterstrichen werden. Der Satz aus S. 33 lautet folglich: *In einem UML-Objektdiagramm wird ein Objekt durch ein Rechteck dargestellt. In dem Rechteck stehen, durch einen Doppelpunkt getrennt und unterstrichen, der Name und die Klasse des Objekts.*

Kurseinheit 1, Abbildung 3.6-1

In Abbildung 3.6-1 müssen die Objektnamen unterstrichen werden:

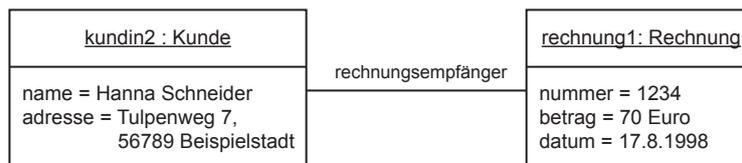


Kurseinheit 1, Lösung zu SA 3.6-1, Abb. 3.6-2

In Abbildung 3.6-2 müssen die Objektnamen unterstrichen werden:

Kurseinheit 2, Kapitel 10, S. 93

Variablen müssen vor dem ersten lesenden Zugriff initialisiert, d. h. erstmalig mit einem Wert belegt werden. Dies kann zum Beispiel bei ihrer Deklaration geschehen, wenn diese mit einer Zuweisung verbunden wird. Eine Initialisierung bei der Deklaration hat die Form: `type varName = expression;`



Kurseinheit 2, Kapitel 14.1, S. 120

Im letzten Quelltextbeispiel auf dieser Seite hat sich ein Syntaxfehler eingeschlichen, hinter condition4 fehlt eine schließende Klammer. Die korrigierte Fassung lautet:

```
while ((condition1 && condition2)
      || (condition3 && condition4)) {
    statement;
}
```

Kurseinheit 3, Selbsttestaufgabe 18.3-8

Hier muss es in der Bedingung in der Lösung neuerRabatt lauten. Die gesamte Methode lautet somit:

```
void legeRabattFest(final double neuerRabatt) {
    // neuer Rabatt ist nur von 0% bis 50% gültig
    if (neuerRabatt >= 0 && neuerRabatt <= 0.5) {
        this.rabatt = neuerRabatt;
    }
}
```

Kurseinheit 3, Abschnitt 20.2, S. 200

Hier muss die Deklaration der Felder chessboard und dreieck jeweils mit einem Semikolon abgeschlossen werden:

```
int[][] chessboard = {{0, 1, 2, 3, 4, 5, 6, 7},
                      {1, 2, 3, 4, 5, 6, 7, 8},
                      {2, 3, 4, 5, 6, 7, 8, 9},
                      {3, 4, 5, 6, 7, 8, 9, 10},
                      {4, 5, 6, 7, 8, 9, 10, 11},
                      {5, 6, 7, 8, 9, 10, 11, 12},
                      {6, 7, 8, 9, 10, 11, 12, 13},
                      {7, 8, 9, 10, 11, 12, 13, 14}};

int[][] dreieck = {{0},
                  {1, 2},
                  {2, 3, 4},
                  {3, 4, 5, 6},
                  {4, 5, 6, 7, 8},
                  {5, 6, 7, 8, 9, 10},
                  {6, 7, 8, 9, 10, 11, 12},
                  {7, 8, 9, 10, 11, 12, 13, 14}};
```

Kurseinheit 3, Lösung zu Selbsttestaufgabe 20.2-2

In der Aufgabenstellung wird von einer Matrix vom Typ double gesprochen, deshalb sollte die Lösung folgendermaßen aussehen:

```
double[][] einheitsmatrix = new double[6][6];
for (int i = 0; i < einheitsmatrix.length; i++) {
    einheitsmatrix[i][i] = 1;
}
```

Kurseinheit 3, Kapitel 21, S. 205

Hier muss es lauten:
Die Klasse `String` dient zur Darstellung von Zeichenketten.

Einsendaufgabe zu Kurseinheit 3, Aufgabe 1 a

Da `new` ein Schlüsselwort und kein Operator ist, muss Antwort F lauten:
Mit Hilfe des Schlüsselwortes `new` und eines Konstruktors werden neue Exemplare erzeugt.

Kurseinheit 4, Selbsttestaufgabe 22.2-1

Hier hat sich in den Attributzugriffen und Methodenaufrufen ein Tippfehler eingeschlichen. Statt

```
new Lehrgebiet().inhaber.gebeRaumnummerAus()
```

sollte es

```
new Lehrgebiet().inhaber.gebeRaumnummerAus()
```

lauten.

Kurseinheit 4, Definition 23.1-1

Hier haben sich zwei verschiedene Paketnamen eingeschlichen, die Definition sollte natürlich lauten:
Ein Paket (engl. package) wird wie folgt eingeführt:

```
package paketName;
```

Diese Anweisung muss am Anfang einer Übersetzungseinheit (z. B. einer einzelnen Quelldatei) vor die Klassendeklaration gesetzt werden, die zum Paket `paketName` gehören soll. [JLS: § 7]

Kurseinheit 4, Beispiel 24.2-1: Zugriff auf Klassen

In Klasse `L` ist die Klasse `M` sichtbar, es muss somit lauten:

```
public class L {  
    // Hier sind die Klassen  
    // K, M und N sichtbar  
}
```

Kurseinheit 4, Selbsttestaufgabe 26.2-1

Die Lösung löscht leider einen Buchstaben zu wenig, die Lösung muss somit

```
sb.delete(6, 13).insert(20, "un");
```

lauten.

Kurseinheit 5, Selbsttestaufgabe 30.5-2

Die Lösung enthält Fehler bei den Vergleichsoperatoren in den ersten beiden Äquivalenzklassen. Diese müssten lauten:

- `0 < tage < 14 && 0 <= km <= 200`
- `0 < tage < 14 && km > 200`

Kurseinheit 6, Kapitel 36.4, S. 423

Hier hat sich im ersten Absatz des Kapitels ein Tippfehler eingeschlichen. Es muss Ausgangsgrad statt Ausgangsgraf lauten.

Kurseinheit 7, Kapitel 41.2, S. 482 f.

Hier wird im ersten `catch`-Block die Variable `artikeldatei` verwendet, die aber nicht deklariert wurde. Der Fehler kann behoben werden, indem der Anfang der Methode `leseArtikelEin()` folgendermaßen angepasst wird:

```
private void leseArtikelEin() {
    this.artikeldaten = new HashMap<Long, Artikel>();
    String artikeldatei = "artikelliste.txt";
    File f = new File(artikeldatei);
    // ...
}
```

oder indem im `catch`-Block die Fehlermeldung folgendermaßen angepasst wird:

```
System.err.println("Die Datei " + f.getName() +
    " mit den Artikel Daten konnte nicht gefunden werden.");
```