

# Druckfehlerverzeichnis

Stand: 7. März 2007

BRAUN, R.; ESSWEIN, W.; GREIFFENBERG, S.: *Einführung in die Programmierung: Grundlagen, Java, UML*. Berlin et al.: Springer, 2006

- **Seite 15, Absatz 1, Satz 1**

- Falsch: „... spielt eine bedeutende Rolle bei der Automatisierung, ...“
- Richtig: „... spielt eine bedeutende Rolle bei der Automatisierung, ...“

- **Seite 24, Absatz 2, Satz 1**

- Falsch: „Demnach können die speziellen Methoden nicht den Objekten, sondern den Klassen zugeordnet werden.“
- Richtig: „Demnach können die speziellen Methoden nicht den Objekten, sondern müssen den Klassen zugeordnet werden.“

- **Seite 31, Abb. 3.1**

```
class Kuh {  
  
    Kuh () {  
        milch = 3;  
    }  
  
    int gibMilch() {  
        return 3;  
    }  
}
```

Quellcode in einer .java Datei  
(z.B. Kuh.java)

(a) Falsch

```
class Kuh {  
  
    Kuh() {  
        milch = 3;  
    }  
  
    int gibMilch() {  
        return 3;  
    }  
}
```

Quellcode in einer .java Datei  
(z.B. Kuh.java)

(b) Richtig

- **Seite 53–54**

- Falsch: „Durch die Vererbung können Zugriffsrechte auf Methoden und Attribute verändert werden. Dabei gilt:
  - \* Eine als `private` deklarierte Methode/Attribut kann als `public` oder `protected` Methode/Attribut und durch das Weglassen der Benennung der Zugriffsart überschrieben werden.

- 
- \* Eine Methode/Attribut, bei der/dem die Benennung der Zugriffsart weggelassen wurde, kann als `public` oder `protected` Methode/Attribut überschrieben werden.
  - \* Eine als `protected` deklarierte Methode/Attribut kann als `public` Methode/Attribut überschrieben werden.
  - \* Eine als `public` deklarierte Methode/Attribut kann nur als `public` Methode/Attribut überschrieben werden.“
- Richtig: „Vererbte Attribute können in der Subklasse neu definiert werden, dabei gelten keine Einschränkungen bezüglich der Festlegung von Zugriffsrechten. Beim Überschreiben von Methoden können deren Zugriffsrechte jedoch nur wie folgt verändert werden:
- \* Eine als `private` deklarierte Methode kann als `private`, `public` oder `protected` Methode oder mit weggelassener Benennung der Zugriffsart überschrieben werden.
  - \* Eine Methode, bei der die Benennung der Zugriffsart weggelassen wurde, kann mit weggelassener Zugriffsart oder als `public` oder `protected` Methode überschrieben werden.
  - \* Eine als `protected` deklarierte Methode kann als `protected` oder `public` Methode überschrieben werden.
  - \* Eine als `public` deklarierte Methode kann nur als `public` Methode überschrieben werden.“
- **Seite 54, Abschnitt 1 nach Quelltextabschnitt 1, ab Satz 2**
    - Falsch: „Private Methoden und Attribute werden nicht vererbt. Damit schränkt das Zugriffsrecht `private` die generelle Vererbung im objektorientierten Paradigma ein (vgl. Abschnitt 2.5.4). Für ein Objekt der Klasse `Cow` bedeutet dies, dass es sich selbst die Nachricht `schlafe()` nicht schicken kann, es sei denn, die entsprechende Methode wird in dieser Klasse neu definiert.“
    - Richtig: „Private Methoden und Attribute werden nicht vererbt. Dies gilt auch für Methoden und Attribute ohne festgelegtem Zugriffsrecht, bei denen die Subklasse nicht im selben Paket wie die Superklasse liegt. Damit schränken diese Zugriffsrechte (`private`, `<ohne>`) die generelle Vererbung im objektorientierten Paradigma ein (vgl. Abschnitt 2.5.4). Für ein Objekt der Klasse `Cow` bedeutet dies, dass es sich selbst die Nachricht `schlafe()` nicht schicken kann, es sei denn, die entsprechende Methode wird in der Klasse `Cow` überschrieben.“

- Seite 70, Quelltextabschnitt 2

– Falsch:

```
b = b * a;
c = c + a;
```

– Richtig:

```
a = a * b;
c = c + a;
```

- Seite 71, Tabelle. 4.5

– Falsch:

Gleitpunkttyp	Wertebereich	Anzahl der Bits
float	$\pm 1.4013 \cdot 10^{-45}$ bis $3.4028 \cdot 10^{38}$	32
double	$\pm 4.9407 \cdot 10^{-234}$ bis $1.7977 \cdot 10^{308}$	64

– Richtig:

Gleitpunkttyp	Wertebereich	Anzahl der Bits
float	$\pm 1.4013 \cdot 10^{-45}$ bis $3.4028 \cdot 10^{38}$	32
double	$\pm 4.9407 \cdot 10^{-324}$ bis $1.7977 \cdot 10^{308}$	64

- Seite 107, Absatz 1, Satz 7

– Falsch: „Dabei kann der Name der aufrufenden Nachricht sowie, durch einen Doppelpunkt getrennt, links von dem Namen, ein etwaiger Rückgabewert und durch ein Gleichheitszeichen getrennt, rechts von dem Namen, das Attribut dem dieser Wert zugeordnet ist, mit an die Antwort angetragen werden.“

– Richtig: „Dabei kann der Name der aufrufenden Nachricht sowie, durch einen Doppelpunkt getrennt, rechts von dem Namen, ein etwaiger Rückgabewert und durch ein Gleichheitszeichen getrennt, links von dem Namen, das Attribut dem dieser Wert zugeordnet ist, mit an die Antwort angetragen werden.“

- Seite 113, Quelltextabschnitt 1

– Falsch:<sup>1</sup>

```
private F() {  
    switch (B) {  
        case W1: F1();  
                break;  
        case W2: F2();  
                break;  
        default: FN();  
    }  
}
```

– Richtig:

```
private F() {  
    switch (B) {  
        case W1: {  
            F1();  
            break;  
        }  
        case W2: {  
            F2();  
            break;  
        }  
        default: {  
            FN();  
        }  
    }  
}
```

---

<sup>1</sup>Anm.: Das Beispiel ist zwar syntaktisch korrekt (wird kompiliert), jedoch sollten generell – der Klarheit des Quelltextes wegen – Anweisungsblöcke in geschweifte Klammern gesetzt werden.