

# Programmieren 1

## Lernteam Semesterwoche 4

### Aufgabe 1: Datentypen und Typecasting

Geben Sie das Ergebnis für folgende mathematischen Ausdrücke im entsprechenden Datentyp in Java an. Beachten Sie hierzu die Folie 14 der Präsentation.

Hinweis: Bei der Auswertung der Ausdrücke gilt die allgemeine "Punkt-vor-Strich" Regel. Die Auswertungsreihenfolge kann, wie in der Mathematik auch, durch Klammern geändert werden.

Beispiel:

```
2.5 * 2 + 3 = 8.0 (Das Ergebnis ist eine Kommazahl)
2.5 * (2 + 3) = 12.5
(int) 2.5 * 2 + 3 = 8
(int) 2.5 * (2 + 3) = 12
(int) (2.5 * 2 + 3) = 8
(int) (2.5 * (2 + 3)) = 12
(int) 2.5 * 2 + (float) 3 = 7.0
```

### Aufgabe 2: Namenskonventionen für Methoden

Auf Folie 19 sehe Sie eine „Konvention“ für die Namen von Zugriffsmethoden.

- Was bedeutet in diesem Zusammenhang „Konvention“?  
→ Konventionen sind soziale Normen. Sie definieren Verhaltensweisen.  
<http://de.wikipedia.org/wiki/Konvention>
- Warum ist eine solche Konvention sinnvoll?  
→ Einheitlichkeit schaffen (Namenskonventionen, Strukturkonventionen)
- Geben Sie Signaturen für Zugriffsmethoden auf die folgenden Attribute gemäss der Konvention an.
  - String vorname; `String getVorname()`
  - float stunden; `float getStunden()`
  - int personalNummer; `int getPersonalNummer()`
  - Object meinObjekt; `Object getMeinObjekt()`
- Kann man für Mutator Methoden auch Namenskonventionen definieren? Begründen Sie.  
→ Ja, zum Beispiel mit dem Präambel "set".  
Beispiel: `void setVorname(String neuerVorname)`

### Aufgabe 3: Fehlermeldungen

Bei den Aufgaben 2.25 und 2.54 mussten Sie Fehlermeldungen notieren. Vergleichen Sie Ihre Notizen. Leiten Sie aus den Fehlermeldungen je eine Regel/ einen Merksatz ab.

→ 2.25: "missing return statement"

→ Merksatz: Wenn eine Methode ein Rückgabewert hat, muss auch ein return vorkommen.

→ 2.54: "unreachable statement"

→ Merksatz: Keine Befehle nach dem letzten return.

### Aufgabe 4: Lernaufgabe zu Switch

Bearbeiten Sie die Lernaufgaben zu switch.

#### Aufgabe 4.1:

Betrachten Sie sich folgende Methode:

```

public void ausgabe (int wert)
{
    System.out.println ();
    System.out.println ("aktueller Parameter: " + wert);
    switch (wert)
    {
        case 1: System.out.println ("eins");
        case 2: System.out.println ("zwei");
        case 3: System.out.println ("drei");
        default: System.out.println ("anderer Wert");
    }
}

```

Lesen Sie die Seiten 439 bis 440 Mitte in Ihrem Buch. Hinweis (zum Buch): expression beim switch bezeichnet einen Wert vom Datentyp byte, char, short oder int.

Stellen Sie eine Vermutung an über die Ausgabe der Methode, wenn Sie beim Methodenaufruf folgende Werte übergeben:

- 1 → eins
- 0 → anderer Wert
- 1 → eins
- 3 → drei

#### Aufgabe 4.2:

Holen Sie sich die Klasse Auswahl von ILIAS. Sie ist in Form eines .jar Files gespeichert.

- Speichern Sie diese Datei ab.
- Verwenden Sie in BlueJ das Kommando Project Open Non BlueJ
- Geben Sie hier das abgespeicherte .jar File an.

Sie erstellen so ein BlueJ Projekt, das Ihnen die Klasse Auswahl zur Verfügung stellt. Überprüfen Sie Ihre Vermutungen aus Aufgabe 1, indem Sie ein Objekt der Klasse Auswahl erzeugen und die Methode "ausgabe" mit oben genannten Werten aufrufen. Was passiert?

(Wenn Sie überrascht sind, lesen Sie nochmals die Seiten 439 bis 440 durch und suchen Sie nach einer Erklärung.)

→ Die Methode "ausgabe" hat scheinbar einen Programmfehler. Folgende Ausgabe erschien:



```

BlueJ: Terminal Window - Auswahl
Options
-----
aktueller Parameter: 1
eins
zwei
drei
anderer Wert

aktueller Parameter: 3
drei
anderer Wert
→

```

#### Aufgabe 4.3:

Verändern Sie den Code der Methode "ausgabe" so, dass sie folgendes Verhalten aufweist:

Wert des aktuellen

Parameters

Ausgabe

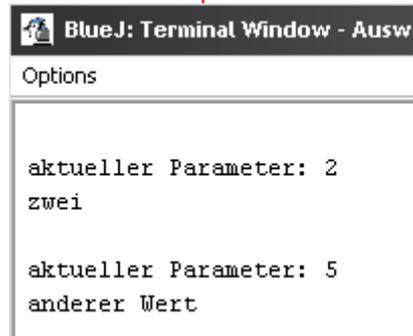
1 → eins

2 → zwei

3 → drei

anderer Wert → anderer Wert

→ Nach dem Einfügen von "break;" Befehlen zwischen den einzelnen Cases funktioniert das Switch-Case Konstrukt perfekt:



```
BlueJ: Terminal Window - Ausw
Options

aktueller Parameter: 2
zwei

aktueller Parameter: 5
anderer Wert
```

```
//Code:
public void ausgabe (int wert)
{
    System.out.println ();
    System.out.println ("aktueller Parameter: " + wert);
    switch (wert)
    {
        case 1: System.out.println ("eins");
                break;
        case 2: System.out.println ("zwei");
                break;
        case 3: System.out.println ("drei");
                break;
        default: System.out.println ("anderer Wert");
                break;
    }
}
```

#### Aufgabe 4.4:

Beschreiben Sie in Ihren eigenen Worten die Funktionsweise von break.

→ Break verhindert, dass die Anweisung durch alle case-Fälle hindurch fällt. Das Break nach dem "default"-Case wird nicht zwingend benötigt – zeugt aber von gutem Programmierstil.

#### Aufgabe 4.5:

Ergänzen Sie die Klasse Auswahl um eine Methode ausgabeTag mit folgender Signatur:  
ausgabeTag (int zahl)

Wenn Zahl einen Wert von 1 bis 7 hat, soll entsprechend dem Wert „Montag“, „Dienstag“, ... „Sonntag“ ausgegeben werden. Im anderen Fall ist „falscher Wert“ auszugeben.

```
public void ausgabeTag (int zahl)
{
    System.out.println ();
    switch (zahl){
        case 1: System.out.println ("Montag"); break;
        case 2: System.out.println ("Dienstag"); break;
        case 3: System.out.println ("Mittwoch"); break;
        case 4: System.out.println ("Donnerstag"); break;
        case 5: System.out.println ("Freitag"); break;
        case 6: System.out.println ("Samstag"); break;
        case 7: System.out.println ("Sonntag"); break;
        default: System.out.println ("falscher Wert"); break;
    }
}
```

Ergänzen Sie die Klasse Auswahl um eine Methode `ausgabeTag2` mit folgender Signatur:  
`ausgabeTag2 (int zahl)`

Wenn Zahl einen Wert von 1 bis 5 hat, soll „Werktag“ ausgegeben werden. Bei einem Wert von 6 oder 7 wird „Wochenende“ ausgegeben. Im anderen Fall ist „falscher Wert“ auszugeben.

```
public void ausgabeTag2 (int zahl)
{
    char wert = '0';
    if(zahl >=1 && zahl <=5)
    {
        wert = 't';
    }
    else
    {
        wert = 'e';
    }
    System.out.println ();

    switch (wert)
    {
        case 't': System.out.println ("Werktag"); break;
        case 'e': System.out.println ("Wochenende"); break;
        default: System.out.println ("falscher Wert"); break;
    }
}
```

### **Aufgabe 5: Weitere Aufgaben aus dem Buch [optional]**

Bearbeiten Sie die Aufgaben ab Exercise 2.59 in Ihrem Buch.