



Abgabe für Rückmeldungen bis zum 30.11.2020 (08:00 Uhr), Besprechung ab dem 30.11.2020.

**Aufgabe 03-0** (Typumwandlungen und Ausdrücke)

Gegeben seien die folgenden Deklarationen

- `int a = 3;`
- `int b = 5;`
- `int c = 6;`
- `double x = 1.5`
- `double y = 2.3;`
- `int ires;`
- `double dres;`

und Ausdrücke.

- 0. `ires = c - a * 3;`
- 1. `ires = c / b;`
- 2. `ires = a + b / c;`
- 3. `dres = c / b;`
- 4. `dres = c + a / b;`
- 5. `dres = x + y * b;`

Beantworten Sie folgende Fragen für jeden der Ausdrücke.

0. In welcher Reihenfolge werden die Operationen ausgeführt? Setzen Sie Klammern nach Präzedenz und Assoziativität.
1. Welchen Wert hat der Ausdruck auf der rechten Seite des Zuweisungsoperators?
2. Welchen Typ hat der Wert des Ausdrucks auf der rechten Seite des Zuweisungsoperators?
3. Ist der Ausdruck kompilierbar?

**Aufgabe 03-1** (Wahrheitstabelle)

Seien `a` und `b` Wahrheitswerte. Legen Sie eine Wahrheitstabelle folgender Ausdrücke an.

- 0. `!(a == b)`
- 1. `!a == b`
- 2. `!a != b`
- 3. `(a == b) || a`
- 4. `(a == b) && a && b`
- 5. `(a != a) || !(b != b) && b`

**Aufgabe 03-2** (Münzrechner)

Schreiben Sie ein Programm, welches für einen gegebenen Geldbetrag `betrag` (z.B. 112.50, 520.15, etc.) die optimale Anzahl an Scheinen und Münzen zurückgibt, die zusammengenommen im Wert dem Geldbetrag entsprechen. Beispielsweise ist der Geldbetrag 12.70€ optimal durch einen 10€-Schein, eine 2€, eine 50-Cent und eine 20-Cent-Münze zerlegt.

Die Währung Euro hat 7 Scheine (500€, 200€, 100€, 50€, 20€, 10€, 5€) und 8 Münzen (2€, 1€, 0.5€, 0.2€, 0.1€, 0.05€, 0.02€, 0.01€), also insgesamt 15 Einheiten. Zur Abbildung der Währung im Programm definieren Sie einen `double`-Array `wert`, in dem Sie die Werte der jeweiligen Einheiten absteigend sortiert eintragen (z.B. `wert[0]` ist 500 und `wert[14]` ist 0.01). Speichern Sie die Lösung in einem `int`-Array `anzahl` der Länge 15, wobei jeder Eintrag einer Einheit (nach Wert absteigend sortiert) entspricht (z.B. `anzahl[0]` der Anzahl der 500€-Scheine

und `anzahl[14]` der Anzahl der 1-Cent-Stücke). Geben Sie den ursprünglichen Geldbetrag und die Zerlegung verständlich in der Kommandozeile aus.

Erweitern Sie Ihr Programm, indem Sie die erhaltene Zerlegung `anzahl` wieder in den ursprünglichen Geldbetrag umrechnen und diesen in der Variable `betrag_neu` speichern. Prüfen Sie am Ende, ob `betrag_neu` dem ursprünglichen Geldbetrag `betrag` entspricht und geben Sie das Ergebnis des Vergleichs in der Kommandozeile aus. Was fällt Ihnen auf?

### Präsenzaufgabe 03-3 (Klammersetzung)

Setzen Sie in den folgenden Ausdrücken mit der Präzedenztabelle in Folie 9 aus Vorlesung 03 die impliziten Klammern, die sich aus Assoziativität und Präzedenz ergeben. Die Variablen `a`, `b` und `c` sind vom Typ `int` und mit `0` initialisiert.

0. `a += c + -b++ / -a++;`
1. `b -= a == c || b != c && b > 0 ? c - a : b / a;`
2. `a = -a++ > 0 || ++c != 1 ? b + c / a : ++c;`

### Präsenzaufgabe 03-4 (Übungen zu Arrays)

Betrachten Sie folgendes Listing.

```
1 int[] arr = { 5, 6, 7, 10, 13, 24, 233 };
2
3 ...
4
5 /* Ausgabe des Arrays */
6 for (int i = 0; i < arr.length; i++) {
7     System.out.printf("%d ", arr[i]);
8 }
9 System.out.printf("\n");
```

Nutzen Sie dieses Gerüst, um bei den Auslassungszeichen am Array `arr` folgende Modifikationen/Prüfungen/Aktionen vorzunehmen.

0. Drehen Sie die Reihenfolge der Werte um (z.B. soll `{5, 6, 7, 10}` zu `{10, 7, 6, 5}` werden).
1. Bilden Sie die Summe aller Elemente und geben Sie diese in der Kommandozeile aus.
2. Bestimmen Sie den Index (i.e. Offset) und den Wert des Minimums des Arrays.
3. Prüfen Sie, ob alle Werte im Array paarweise verschieden sind.
4. Bilden Sie die Summe der Beträge aller Elemente und geben Sie diese in der Kommandozeile aus.
5. Bestimmen Sie den Index (i.e. Offset) und den Wert des Maximums des Arrays.

Variieren Sie `arr` bezogen auf Inhalt und Länge, um Ihr Programm zu testen.