

4. Übung zu Informatik I im WS 16/17  
Abgabe: 14. November 2016, 12<sup>00</sup> Uhr.

**Aufgabe 8 - Gruppenabgabe**

(8 Punkte)

Bitte stellen Sie für folgende Rechnungen in Zwischenschritten die Bit-Operationen dar, welche von Java intern durchgeführt werden. Stellen Sie alle Zahlen in ihrer Bit-Repräsentation dar. Falls eine Zahl innerhalb der Rechnung oder das Ergebnis eine negative Zahl sein sollte, zeigen Sie mit Hilfe des Zweierkomplements, um welchen Wert es sich handelt.

**Als vereinfachende Annahme:** Gehen Sie dabei davon aus, dass sämtliche der nachfolgenden Zahlen vom Typ *Byte* sind und auch die Ergebnisse der Rechnungen wieder vom Typ *Byte* sind (Jede Bit-Repräsentation hat genau 8 Stellen).

- |                  |                |                 |                |
|------------------|----------------|-----------------|----------------|
| a) $6 + 3$       | b) $127 \ll 3$ | c) $5 - 7$      | d) $-25 \gg 2$ |
| e) $7 \wedge 42$ | f) $13 \& 6$   | g) $98 \mid 53$ | h) $-3 \ggg 5$ |

**Hinweis:** Zur Subtraktion zweier Zahlen bildet Java intern aus dem Subtrahend eine negative Zahl (Zweierkomplement) und führt anschließend eine Addition durch.

**Hinweis:** Zur Überprüfung von Zwischen- und Endergebnissen kann z.B. der Aufruf

`(byte) Integer.parseInt("11000010",2)`

verwendet werden. Der erste Parameter definiert dabei die Bitfolge, welche Sie in ein Byte umwandeln wollen und der zweite Parameter definiert die Basis der Zahl des ersten Parameters (Im Falle von Dual/Binärzahlen ist dies 2).

**Aufgabe 9 - Gruppenabgabe**

(8 Punkte)

Bitte Klammern Sie folgende Ausdrücke anhand der Operatorpräzedenz, wie auf Vorlesungsfolie 47 demonstriert. Aus der Klammerung sollte die Ausführungsreihenfolge aller Operatoren eindeutig hervorgehen!

- a)  $x++ \ll 2 * 1 \gg = --y$
- b)  $x \gg = y != 2 \gg = z-- \&\& !b$
- c)  $x += z * y + 3 \ggg 2 * 5 - 2$
- d)  $b = x \ll 2 \gg = 2 ? y != 3 : b \mid x > 2$

### Aufgabe 10 - Einzelabgabe

(4+4+6 = 14 Punkte)

Bitte implementieren Sie die beiden Methoden folgender Klasse:

```
public class Schleife{

    // Gibt das groesste Element des Arrays zurueck.
    public int max(int[] array){
        ...
    }

    // Gibt das Skalarprodukt beider Arrays zurueck.
    public int scalar(int[] vektor1, int[] vektor2){
        ...
    }

    // Wenn array mind. eine Zahl doppelt enthalt -> true; andernfalls false.
    public boolean hasDoppelte(int[] array){
        ...
    }
}
```

- a) Die Methode `max(int[] array)` soll das größte Element eines Arrays zurückgeben.
- b) Die Methode `skalarprodukt(int[] vektor1, int[] vektor2)` soll das Skalarprodukt zweier Arrays zurückgeben. Ein Skalarprodukt von zwei Vektoren  $x$  und  $y$  (Arrays) berechnet sich in der Mathematik nach der Formel  $\sum_{i=1}^n x_i * y_i$ . Gehen Sie davon aus, dass beide Vektoren (Arrays) dieselbe Länge haben!
- c) Die Methode `hasDoppelte(int[] array)` gibt `true` zurück, falls das Array mindestens eine Zahl doppelt enthält. Andernfalls wird `false` zurückgegeben.

**Anmerkung:** *Versehen Sie ihre Abgaben bitte mit **Namen, Matrikelnummern, E-Mail-Adressen** und **Studiengängen** der beteiligten Mitarbeiter/innen und laden Sie diese in der entsprechenden Aktivität im Learnweb hoch. Aufgaben, die mit dem Hinweis **Gruppenabgabe** versehen sind, dürfen mit maximal 3 Mitarbeiter/innen gelöst werden. Organisieren Sie sich dafür bitte zu Beginn der Veranstaltung in **Dreiergruppen**. Aufgaben mit dem Hinweis **Einzelabgabe** müssen von jedem Studenten und jeder Studentin eigenständig gelöst und abgegeben werden. Viele der Aufgaben werden über das **EASy** (E-Assessment System) System im Learnweb eingereicht und automatisch vorausgewertet. Bei ihrem ersten Übungstermin werden Sie eine kurze Einführung dazu bekommen. Um Probleme bei der Abgabe wegen Überlastung des EASy-Servers zu vermeiden, würde ich Sie bitten, eine Abgabe auf den letzten Drücker nach Möglichkeit zu vermeiden. Bei Fragen zum Übungsbetrieb wenden Sie sich bitte an ihren/ihre Tutor/in oder an Tobias Reischmann.*