

## Konzepte der Programmierung

### Übung 6 – Abgabe: 07.12.2017

#### Aufgabe 6.1 (Operatoren: Prioritäten)

Ziel dieser Aufgabe ist es, die Auswertungsreihenfolge der Operatoren in Java darzustellen. Entsprechend ist dieser Ausdruck auch ohne Klammern syntaktisch korrekt und ausführbar. Fügen Sie dem folgenden Ausdruck Klammern hinzu, die die Auswertungsreihenfolge in Java verdeutlichen. Die Auswertung des geklammerten Ausdrucks soll äquivalent zu der des nicht geklammerten Ausdrucks sein.

$$a/b * c + d - e >= f - g \parallel h * c + x == f/y * z \&\& x - o + -s < a/d + c$$

(2 Punkte)

#### Aufgabe 6.2 (Ableitungsbaum und Typbestimmung)

Gegeben Sei der folgende Quellcode:

```
1 public void berechneZahl() {  
2     long u = 5468794;  
3     char v = 'x';  
4     byte w = 3;  
5     double x = 1.3;  
6     int y = 1326;  
7     short z = 7;  
8     System.out.println(u + v * w + x / y - z);  
9 }
```

- (a) Geben Sie für den Ausdruck, der in Zeile 8 auf der Konsole ausgegeben wird, einen **Ableitungsbaum** an. Verwenden Sie dazu die EBNF für einfache Ausdrücke (Kap. Ausdrücke, S.22).
- (b) Führen Sie für die Belegung des Ausdrucks eine **Typbestimmung** durch. Erstellen Sie dazu einen Auswertungsbaum und geben Sie dabei für jeden Knoten des Auswertungsbaums dessen Typ an und kennzeichnen Sie, an welchen Stellen eine implizite **Typumwandlung** erfolgt. Geben Sie auch an, ob und wenn ja, wo Genauigkeitsverlust auftritt.

(2+2 = 4 Punkte)

#### Aufgabe 6.3 (Studentenverwaltung, Aufzählungswerte und Typkonversionen)

Wir setzen die Studentenverwaltung von Blatt 4 fort und behandeln dabei Aufzählungstypen und die Typkonversion.

- (a) Implementieren Sie zunächst einen Aufzählungstyp **Studienstatus**. Der Status kann variieren zwischen immatrikuliert, aktiv, unterbrochen, bestanden und nicht bestanden.

- (b) Versehen Sie jeden Student mit einem Attribut, das seinen Studienstatus speichert. Entscheiden Sie sich für ein Objekt- oder Klassenfeld und achten Sie auf die Sichtbarkeit. Standardmäßig ist der Zustand zunächst auf immatrikuliert gesetzt (sonst wäre der Student nicht im System vorhanden).
- (c) Speichern Sie außerdem die Noten eines Studenten in einem Array. Es sollen darin maximal die letzten 15 Noten gespeichert werden. Noten sind an dieser Uni immer ganzzahlig.
- (d) Implementieren Sie eine Methode `fuegeEin()`, der eine Note übergeben wird. Das Notenarray soll zyklisch befüllt werden, d.h. ist es voll werden die Noten von vorne beginnend überschrieben. Bedenken Sie, dass dafür ggf. ein weiteres Attribut in der Klasse Student nötig ist.

Unterscheiden Sie dabei die folgenden Fälle bzgl. des Status. Benutzen Sie für deren Implementierung eine switch-Anweisung.

- **IMMATRIKULIERT:** Der Status wird auf aktiv gesetzt und die Note an der ersten (freien) Position im Notenarray gespeichert.
- **AKTIV:** Die Note wird an der nächsten freien Position im Array gespeichert.
- **UNTERBROCHEN:** Es wird die Meldung ausgegeben, dass der Studienstatus verändert wird und dieser zurück auf aktiv gesetzt. Die Note wird an der nächsten freien Position gespeichert.
- **BESTANDEN:** Es wird eine Meldung auf der Konsole ausgegeben, dass der Student schon bestanden hat und keine Noten mehr speichern kann.
- **NICHT\_BESTANDEN:** Es wird eine Meldung auf der Konsole ausgegeben, dass der Student nicht bestanden hat und demnächst exmatrikuliert wird.

- (e) Implementieren Sie eine Methode `berechneDurchschnitt()`, die den Durchschnitt aller Noten im Array berechnet und zurückgibt. **Vorsicht:** Der Durchschnitt aller Noten muss NICHT ganzzahlig sein. Hier ist auch eine Kommazahl möglich.

Überprüfen Sie das Verhalten Ihrer Klasse und Methoden, indem Sie mit BlueJ Objekte anlegen, Noten zuweisen und die Objekte inspizieren.

**(1+2+2+5+3 = 13 Punkte)**

#### **Aufgabe 6.4 (Codeverständnis)**

Gegeben ist folgende, schlecht verständliche Anweisung:

```
k = (j=5) * (i=2) + --j - ++i * j--;
```

Formulieren Sie die Anweisung in eine Sequenz von einfachen Anweisungen um. Jede Zeile darf nur eine Operation enthalten (Beispiel: `k = k + i;`). Die Reihenfolge der Auswertung soll dabei unverändert bleiben. Benutzen Sie keinesfalls Inkrement- und Dekrementoperatoren. Geben Sie die Werte der Variablen `i`, `j` und `k` während der Ausführung der Sequenz an.

Sie können davon ausgehen, dass alle Variablen ganzzahlig sind.

**(4 Punkte)**