

Einführung in die Programmierung
WS 2009/10

Wiederholungsblatt

Besprechung: 25./27./28./29.01.2010

Hinweis:

Die Aufgaben auf diesem Blatt dienen der Klausurvorbereitung. Sie decken aber weder den gesamten klausurrelevanten Stoff ab, noch geben sie Aufschluss über den Umfang der Klausuraufgaben. Sie haben in den Übungen vor der Klausur die Gelegenheit, Lösungsideen zu diskutieren. Eventuell kann aber im Rahmen der Übung nicht auf jede Aufgabe ausführlich eingegangen werden. Sie erhalten jedoch zu dem gesamten Wiederholungsblatt Lösungsvorschläge, die wir ab Montag 25.01.2010 auf der Homepage bereitstellen.

Aufgabe W-1 *Syntaxdefinitionen*

In Java gibt es drei Arten von Kommentaren:

- **Einzeilen-Kommentare:** Sie werden durch zwei Schrägstriche (/) eingeleitet und enden durch ein Zeilenendsymbol.
- **Block-Kommentare:** Sie werden durch Schrägstrich (/) und Stern (*) eingeleitet und durch Stern und Schrägstrich beendet. Dazwischen können beliebige¹ Zeichenketten stehen.
- **Javadoc-Kommentare:** Sie werden durch Schrägstrich (/) und zwei Sterne (*) eingeleitet und verhalten sich sonst wie Block-Kommentare².

(a) Modellieren Sie in BNF-Notation die syntaktische Variable $\langle \text{Kommentar} \rangle$, welches alle Arten von Java-Kommentaren akzeptiert ($\langle \text{Kommentar} \rangle$ ist also das sogenannte *Startsymbol*). Gerne dürfen Sie weitere Hilfs-Variablen einführen. Betrachten Sie die syntaktischen Variablen $\langle \text{Buchstabe} \rangle$, $\langle \text{Ziffer} \rangle$, $\langle \text{Symbol} \rangle$, $\langle \text{Leerzeichen} \rangle$ sowie $\langle \text{Zeilenumbruch} \rangle$ als gegeben, d.h. verwenden Sie diese syntaktischen Variablen in Ihren BNF-Regeln.

(b) Geben Sie für den Kommentar `/* Version 2.0 */` die Ableitung an, die Ihre BNF Regeln von oben liefern.

¹Ansich sollte man sagen, dass beliebige Zeichenketten ohne `*/` vorkommen dürfen – denn diese beiden Zeichen beenden ja den Kommentar. Diese Ausnahme braucht allerdings nicht explizit von Ihnen behandelt zu werden.

²Im Grunde genommen gibt es nur 2 (disjunkte) Arten von Kommentaren, wenn man bedenkt, dass Javadoc-Kommentare selber alle auch Blockkommentare sind.

Aufgabe W-2 *Ausdrücke*

Geben Sie für jedes der folgenden Literale an, ob es ein syntaktisch korrekter Java-Ausdruck ist. Falls ja, geben Sie außerdem Wert und Typ des Ausdrucks an, falls nein, eine kurze Begründung, warum der Ausdruck fehlerhaft ist.

- | | |
|-----------------------------|----------------------------|
| (a) <code>-(-1)</code> | (h) <code>7%3=1</code> |
| (b) <code>--1</code> | (i) <code>7%3==1</code> |
| (c) <code>--1</code> | (j) <code>13%19%9%5</code> |
| (d) <code>++1</code> | (k) <code>TRUE</code> |
| (e) <code>'Hello'</code> | (l) <code>4d</code> |
| (f) <code>12*4(-6)</code> | (m) <code>0x2a</code> |
| (g) <code>(12)*(4)-6</code> | (n) <code>052</code> |

Aufgabe W-3 *Arrays*

Implementieren Sie eine Klasse `Taschenrechner`.

Diese enthält eine Methode `double bestimmeMittelwert(int[] zahlen)`, die das arithmetische Mittel von einem Integer-Array `zahlen` berechnet und zurückgibt. Das arithmetische Mittel eines Arrays mit den Elementen a_0, a_1, \dots, a_{n-1} berechnet sich dabei als $\frac{1}{n} \sum_{i=0}^{n-1} a_i$. Kommentieren Sie Ihre Klasse und testen Sie sie für geeignete Werte.

Aufgabe W-4 *Speicherverwaltung*

Betrachten Sie die Klasse Speicherverwaltung.

```
public class Speicherverwaltung {
    public static void swap(int x, int y) {
        int c = x;
        x = y;
        y = c;
        // *2*
    }
    public static void swapInArray(int[] numbers, int i, int j) {
        int c = numbers[i];
        numbers[i] = numbers[j];
        numbers[j] = c;
        // *5*
    }
    public static void swapArrays(int[] numbers1, int[] numbers2) {
        int[] c = numbers1;
        numbers1 = numbers2;
        numbers2 = c;
        // *8*
    }
    public static void main(String[] args) {
        {
            int x = 1;
            // *0*
            int y = 2;
            // *1*
            swap(x, y);
            // *3*
            System.out.print("x = " + x + ", y = " + y);
        }
        {
            System.out.print("\n");
            int[] values = { 4, 5, 6, 7, 8 };
            // *4*
            swapInArray(values, 2, 4);
            for (int i = 0; i < values.length; i++) {
                System.out.print(values[i]);
                if (i != values.length - 1) {
                    System.out.print(", ");
                }
            }
            // *6*
        }
        {
            System.out.print("\n");
            int[] numbers1 = { 1 };
            int[] numbers2 = { 2 };
            // *7*
            System.out.println("numbers1 = " + numbers1 + ", numbers2 = "
                + numbers2);
            swapArrays(numbers1, numbers2);
            // *9*
            System.out.println("numbers1 = " + numbers1 + ", numbers2 = "
                + numbers2);
        }
    }
}
```

- (a) Geben Sie für jede der mit *1*, *2*, ..., *9* gekennzeichneten Zeilen den Inhalt aller sichtbaren Frames, sowie den Inhalt des Heaps an.
- (b) Welche Ausgabe erzeugt dieses Programm auf der Konsole? Wie erklären Sie sich diesen Effekt?

Aufgabe W-5 *Hoare-Kalkül*

Die Methode ggT ermittelt den größten gemeinsamen Teiler zweier ganzer Zahlen $0 < a$ und $0 < b$ nach dem klassischen Algorithmus von Euklid. Beweisen Sie mit den Mitteln des Hoare-Kalküls die partielle Korrektheit des folgenden Programmstücks.

```
// Vorbedingung PRE: (0<a && 0<b)
{
  x = a;
  y = b;
  while (x != y) {
    if(x < y) { y = y-x; }
    else      { x = x-y; }
  }
}
// Nachbedingung POST: (ggT(a,b)==x)
```

Hinweise:

- Sie können voraussetzen, dass für $x < y$ gilt: $\text{ggT}(x, y) == \text{ggT}(x, y-x)$ (und entsprechend für $y < x$: $\text{ggT}(x, y) == \text{ggT}(x-y, y)$).
- Eine mögliche Schleifeninvariante lautet: $0 < x \ \&\& \ 0 < y \ \&\& \ \text{ggT}(a, b) == \text{ggT}(x, y)$

Aufgabe W-6 *Bruchrechnen mit Java*

In dieser Aufgabe soll eine Klasse `Bruch` implementiert werden, die Brüche (rationale Zahlen) und die Grundrechenarten für Brüche implementiert.

- Definieren Sie eine Klasse `Bruch`, die (gewöhnliche) Brüche mit Zähler und Nenner vom Typ `int` repräsentiert. Deklarieren Sie dazu entsprechende Attribute.
- Erweitern Sie die Klasse `Bruch` um einen Konstruktor `Bruch(int n)`, der eine ganze Zahl n als Bruch erzeugt.
- Erweitern Sie die Klasse `Bruch` um eine Methode `String toString()`, die ein `String`-Objekt zurückgibt, das einen entsprechenden Bruch geeignet textuell repräsentiert.
- Erweitern Sie die Klasse `Bruch` um eine Methode `Bruch negiere()`, die einen neuen Bruch erzeugt, so dass gilt: Ist b ein Bruch mit Wert z/n , so ist $b.negiere()$ ein Bruch mit Wert $-z/n$.
- Erweitern Sie die Klasse `Bruch` um folgende Methoden:
 - `Bruch addiere(Bruch b)`
 - `Bruch subtrahiere(Bruch b)`
 - `Bruch multipliziere(Bruch b)`
 - `Bruch dividiere(Bruch b)`

Jede dieser Methoden soll einen neuen Bruch zurückgeben, der den Wert der Berechnung repräsentiert. Es gilt:

$$\frac{z_1}{n_1} + \frac{z_2}{n_2} = \frac{z_1 n_2 + z_2 n_1}{n_1 n_2}, \quad \frac{z_1}{n_1} - \frac{z_2}{n_2} = \frac{z_1 n_2 - z_2 n_1}{n_1 n_2}, \quad \frac{z_1}{n_1} \times \frac{z_2}{n_2} = \frac{z_1 z_2}{n_1 n_2}, \quad \frac{z_1}{n_1} / \frac{z_2}{n_2} = \frac{z_1 n_2}{n_1 z_2}$$

Testen Sie diese Methoden für geeignete Eingaben.

- Erweitern Sie die Klasse `Bruch` um eine Methode `boolean kleinerAls(Bruch b)`, die folgende Werte zurückgibt: `true`, wenn b grösser ist als der Wert des aufrufenden Objekts vom Typ `Bruch`, `false` sonst.
- Erweitern Sie die Klasse `Bruch` um eine Methode `double alsDouble()`, die das aufrufende Objekt vom Typ `Bruch` in eine Gleitkommazahl umwandelt.

Aufgabe W-7 *Vererbungspolymorphie*

Eine einfache Vererbungshierarchie besteht aus den vier Klassen `A`, `B`, `C` und `D`. Die Klasse `Vererbung` enthält einige Tests. Alle Klassen sind auf der Vorlesungs-Homepage verfügbar.

- Laden Sie die Klassen von unserer Vorlesungs-Homepage herunter und führen Sie das Programm aus. Welches Ergebnis erhalten Sie? Wie können Sie sich dieses Ergebnis erklären?
- Was würde passieren, wenn Sie die auskommentierten Zeilen wieder einkommentieren? Warum sind diese Zeilen falsch?