

Einführung in die Programmierung  
WS 2018/19

Übungsblatt 5: Ausdrücke, Substitution, Rekursion

Besprechung: 26.11 - 30.11.2018

**Aufgabe 5-1**      *Ausdrücke*

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

- |                |                     |          |
|----------------|---------------------|----------|
| (a) - -1       | (g) (17)*(11)-16    | (m) 3d   |
| (b) -(-1)      | (h) 5%3=1           | (n) 0x1a |
| (c) ++2        | (i) 8%3==2          | (o) 0x2g |
| (d) --2        | (j) 18%21%7%5       | (p) 057  |
| (e) 'FALSE'    | (k) - - - 2 + - + 5 | (q) 41L  |
| (f) 17*11(-16) | (l) TRUE            | (r) 2/0  |

**Aufgabe 5-2**      *Rekursion und Methoden in Java I*

- (a) Wir benötigen später eine Methode, die die Quadratwurzel einer Gleitkommazahl  $x$  berechnet. Für diese Aufgabe darf keine andere Klasse (insbesondere nicht `Math`) benutzt werden. Um die Wurzel anzunähern, bedienen wir uns einem Annäherungsverfahren aus der Numerik, das auch als Heronverfahren oder babylonisches Wurzelziehen bekannt ist. Die induktiv definierte Folge

$$x_0 = \frac{x+1}{2}$$
$$x_n = \frac{1}{2} \left( x_{n-1} + \frac{x}{x_{n-1}} \right)$$

konvergiert gegen den Wert  $\sqrt{x}$ . Es ist hier nicht nötig, dass sie verstehen, warum dieses Verfahren korrekt arbeitet. Die rekursive Funktion, die die Wurzel berechnet, lässt sich damit wie folgt als Pseudocode programmieren:

```
function:  quadratwurzel(Reelle Zahl x, Ganze Zahl n)
output:   Quadratwurzel von x
pre:      x >= 0, n >= 0
body:
  Wenn n == 0
    Dann (x + 1)/2
  Sonst 0.5*(quadratwurzel(x,n-1)+x/quadratwurzel(x,n-1))
```

Führen Sie formal einen Funktionsaufruf dieser Funktion mit der Variablenbelegung  $\sigma = [x/3.0, n/1]$  durch, wie in der Vorlesung behandelt. Runden Sie gegebenenfalls Zwischenergebnisse auf eine Nachkommastelle.

(b) Implementieren Sie eine rekursive Wurzelfunktion gemäß voriger Aufgabe in Java:

```
public static double quadratwurzel(double x, int n) {...}
```

Nutzen Sie dazu keine Methoden außer den Ihnen bereits bekannten Basisoperatoren (+, -, \*, /). Sie dürfen annehmen, dass alle Eingabewerte positiv sind. Vergessen Sie nicht, Ihre Methode zu testen.

### Aufgabe 5-3 *Rekursion und Methoden in Java II*

Jetzt werden Sie eine Methode implementieren, die die Kreiszahl  $\pi$  annähert. Dazu nutzen Sie die folgende analytische Darstellung, die im 16. Jahrhundert von Vieta entwickelt wurde:

$$\frac{2}{\pi} = \left(\frac{1}{2}\sqrt{2}\right) \cdot \left(\frac{1}{2}\sqrt{2+\sqrt{2}}\right) \cdot \left(\frac{1}{2}\sqrt{2+\sqrt{2+\sqrt{2}}}\right) \cdot \left(\frac{1}{2}\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}\right) \cdot \dots$$

Dafür benötigen Sie die Wurzelfunktion aus der vorherigen Aufgabe. Falls Sie diese nicht gelöst haben, dürfen Sie stattdessen auf die Funktion `Math.sqrt(x)` zurückgreifen.

Die Faktoren der obigen Darstellung lassen sich induktiv definieren:

$$a_0 = \frac{1}{2}\sqrt{2}$$

$$a_{n+1} = \frac{1}{2}\sqrt{2+2a_n}$$

Implementieren Sie in Java eine rekursive Funktion

```
public static double vietaFaktor(int n) {...}
```

Für ein beliebiges  $n \geq 0$  soll `vietaFaktor(n)` das Folgeglied  $a_n$  berechnen.

Implementieren Sie anschließend eine Methode

```
public static double pi(int n) {...}
```

Diese Methode nutzt  $n$  Faktoren, um mit der Formel von Vieta die Kreiszahl  $\pi$  anzunähern und zurückzugeben. Hier ist es Ihnen überlassen, ob Sie dies rekursiv umsetzen.

### Aufgabe 5-4 *Rekursion und Methoden in Java III*

Implementieren Sie eine Methode `berechneDistanz(x1,y1,x2,y2)`, die die (euklidische) Distanz

$$d(x, y) = \sqrt[2]{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

zwischen zwei Punkten im zweidimensionalen Raum berechnet. Hier können Sie Ihre Wurzelfunktion gerne anwenden oder Sie nutzen die bereits implementierte Methode `Math.sqrt(double x)`.