Einführung in die Informatik: Übung 3

Alexander Waldenmaier

27. November 2020

Aufgabe 1: Teiler

```
1: procedure BERECHNETEILERZAHL(start, end)
2:     for n ← start to end do
3:         teiler ← 0
4:         for t ← 2 to n do
5:             if n mod t == 0 then
6:                 teiler ← teiler + 1
7:         print teiler
```

Aufgabe 2: Präzedenzregeln

```
42  +  5  *  -  2  !  =  4  %  2  +  2
42  +  5  * (-2)   !=  4  %  2  +  2
42  +  (5 * (-2))  !=  (4  %  2)  +  2
(42 +  (5 * (-2))) !=  ((4  %  2)  +  2)
```

Aufgabe 3: Ausgaben

Die Ausgaben lauten:

```
0.8999999999999999
198!
3
Rechnung: 3-30
```

Erklärungen zu jeder Zeile:

1. Da nur floats addiert werden, findet innerhalb des *println* statements eine arithmetische Addition der drei Zahlen statt. Erst am Ende wird die resultierende Zahl in einen String verwandelt und dann ausgegeben. Der Grund warum nicht wie erwartet die 0.9, sondern 0.8999999999999999 ausgegeben wird, ist die endliche Genauigkeit, mit der floats gespeichert werden können. Bei der Verrechnung können daher kleine Ungenauigkeiten auftreten.

2. Da die Buchstaben A, B und C als *chars* deklariert wurden (erkennbar an der Verwendung der Hochkommas statt Anführungszeichen) werden diese in einen *int* gemäß der ASCII-Konvention umgewandelt. Diese sind 65, 66 und 67. Die Summe dieser drei Zahlen ist 198. Da erst im Anschluss ein tatsächlicher *string* folgt, findet erst dann eine Umwandlung des *int* 198 in einen *string* statt. Dieser wird dann um das Ausrufezeichen ergänzt.

3. Da sowohl die 11 als auch die 3 ein *int* sind, entsteht bei der Division ebenfalls eine Ganzzahl (Kommastellen werden abgeschnitten). Das Ergebnis ist der *int* 3, der dann ausgegeben wird.

4. Da zu Beginn der *string* `"Rechnung "` steht, werden die nachfolgenden „Additionen“ als *string*-Zusammenfügungen verstanden. Folglich werden die *ints* 3, -3 und 0 jeweils in *strings* umgewandelt und dann hintereinander ausgegeben.

Aufgabe 4: Grumpy cats

```java
public class aufgabe4 {
    public static void main(String[] args){
        String prefix, suffix;
        for (int i = 2; i <= 100; i++) {
            if (i % 5 == 0) {
                prefix = "look! ";
            } else {
                prefix = "";
            }
            if (i % 3 == 0) {
                suffix = " grumpy";
            } else {
                suffix = "";
            }
            System.out.println(prefix + i + suffix + " cats");
        }
    }
}
```

Zusatzaufgabe (1/6): Primteiler

```java
import java.util.ArrayList;

public class zusatzaufgabe {
    public static void main(String[] args){
        // Zahl n von der Konsole einlesen
        Long n = Long.parseLong(args[0]);

        // Beliebig erweiterbare ArrayList verwenden um alle Faktoren zu sammeln
        ArrayList<Long> factors = new ArrayList<Long>(0);

        // Alle Teiler von 2 bis maximal sqrt(n) durchprobieren
        long t = 2;
        while (t * t < n) {
            // Wenn t ein Teiler ist, diesen zur Liste hinzufuegen. Gleichzeitig n
                durch t teilen
            if (n % t == 0) {
                factors.add(t);
                n /= t;
            // Wenn kein Teiler, naechsthoehere Zahl probieren
            } else {
                t++;
            }
        }
        // Das finale n ist eine Primzahl und wird auch zu den Faktoren ergaenzt
        factors.add(n);

        // Ausgabe erstellen durch Ergaenzung aller Primfaktoren zum Outputstring
        String output = "Primfaktorzerlegung von " + n + " = ";
        for (Long factor : factors) {
            output += factor + "*";
        }
        output = output.substring(0, output.length()-1);

        // Finale Ausgabe
        System.out.println(output);
    }
}
```