

# Freiwilliges Aufgabenblatt #3

Lösen Sie die die erste gestellt Aufgabe und kommen Sie mit den vorbereiteten Blättern in die Übungsgruppe, in die Sie sich angemeldet haben.

Die anderen Beispiele dienen zur Vorbereitung für den Test und sollten deshalb unbedingt vor dem Test gelöst werden. Diese werden in der Übung nur bei Bedarf durchgenommen.

Wenn Sie es nicht bereits gemacht haben, lesen Sie bitte die relevanten Teile des Skriptums und erarbeiten Sie gemeinsam die bereits durchgegangenen Fragen am Ende des Kapitels.

Vollständiges erstes und zweites Kapitel.

## Aufgabe 1: Vorbereitung auf freiwillige Übungsrunde

Setzen Sie sich in der Kleingruppe zusammen und wählen Sie insgesamt 12 Begriffe, die Sie in der Programmierung oder allgemeiner in der Softwareentwicklung für besonders wichtig erachten und die im Skriptum behandelt werden. Wählen Sie vor allem Begriffe, die von anderen Kleingruppen Ihrer Vermutung nach eher nicht gewählt werden. Schreiben Sie jeden der 12 Begriffe groß und deutlich lesbar auf ein eigenes Blatt der Größe DIN A6 und bringen Sie die Blätter zur Übungsrunde mit.

Findet am 5. bis 7. November statt.

Das ist ein DIN A4-Blatt in vier Teile geteilt.

## Programme verstehen

Listing 1: Zahlenmanipulation

```
public static String foo(int n, int b) {
    String output = "";
    while (n > 0) {
        int m = n; // Temp. Variable
        int o = 0; // Zähler
        while (m >= b) {
            m = m - b;
            o++;
        }
        output += String.valueOf(m);
        n = o;
    }
    return new // Gibt String rückwärts gelesen aus
        StringBuffer(output).reverse().toString();
}
```

## Aufgabe 2: Verhalten erklären

Erklären Sie anhand eines Beispiels das Verhalten dieser Methode. Die letzte Zeile gibt den String „output“ rückwärts gelesen aus und dient nur zum besseren Verständnis der Methode, Sie müssen diese nicht nachvollziehen.

## Aufgabe 3: Beispiele durchrechnen

Führen Sie den durch `foo` gegebenen Algorithmus mit folgenden Parametern durch: `foo(20,2)` `foo(20,10)` `foo(20,16)`

## Aufgabe 4: Programm umwandeln

Wandeln Sie diese Methode in eine rekursive Methode um. Lassen Sie dabei die letzte Zeile außer acht.

## Rekursion nachvollziehen

### Aufgabe 5: Werte berechnen

Gegeben ist die folgende rekursive Funktion

$$\begin{aligned}f(0) &= 0 \\f(1) &= 1 \\f(n) &= f(n-1) + f(n-3)\end{aligned}$$

Berechnen Sie die Werte von  $f(3)$ ,  $f(5)$  und  $f(20)$ , und geben Sie den Rechenweg an. Benutzen Sie zur Berechnung ausschließlich die rekursive Definition!

## Sprachkonstrukte umbauen

```
switch(input)
{
case 0: output = 5; break;
case 1: output = 12; break;
5 case 2: output = 4; break;
case 4: output = 16; break;
case 8: output = 1; break;
case 16: output = 0; break;
}
```

**Aufgabe 6: Auf if umbauen**

Schreiben und testen Sie eine Methode welche statt `switch if` verwendet. Welche Lösung ist vorzuziehen?

**Aufgabe 7: Auf array umbauen**

Schreiben und testen Sie eine Methode welche mittels ohne `if` oder `switch` mittels eines Arrays die gleiche Funktionalität hat. Welches Problem können Sie hier erkennen?

**Aufgabe 8: Rekursive Methode schreiben**

Schreiben und testen Sie eine rekursive Methode `fib`, die den Wert der Funktion  $f$  berechnet. Wie oft wird die Methode `fib` bei einem Aufruf von `fib(3)`, `fib(5)` und `fib(20)` aufgerufen?

**Aufgabe 9: Methode iterativ schreiben**

Versuchen Sie, die Methode `fib` so umzuschreiben, dass Sie anstatt der rekursiven Aufrufe eine Schleife benutzen.