

## Übung 2: Arbeiten mit Code

### Aufgabe A: Lesbarer Quelltext

Was macht folgender Quelltext? Schreiben Sie den Quelltext so um dass er leicht verstanden und wiederverwendet werden kann. Achten Sie dabei auf die in der Vorlesung vorgestellten Hinweise.

```

package Math;
public class Sort {
    private int[] N1;
    private int N0;
    public void Execute(int[] v) {
        if (v == null || v.length == 0) {
            return;
        }
        this.N1 = v;
        N0 = v.length;
        s(0, N0 - 1);
    }
    private void s(int h, int low) {
        int i=h,j=low;
        int p=N1[h + (low-h)/2];
        while (i <= j) {
            while(N1[i] < p) i++;
            while (N1[j] > p) { j--; }
            if (i <= j) {
                tmp_numberi_numberj(i, j);
                i++;
                j--;
            }
        }
        if (h < j) s(h, j);
        if (i < low) s(i, low);
    }
    private void tmp_numberi_numberj(int i, int j) { int
        tom = N1[i];
        N1[i] =
        N1[j]; N1[j] = tom; }
}

```

**Aufgabe B: Refactorings**

- a) Erläutern Sie das Konzept Refactoring. Was ist der Anlass dafür ein Refactoring durchzuführen?
- b) Kann folgende Quelltextänderung als Refactoring betrachtet werden? Begründen Sie ihre Antwort.

```

public class Math { //vorher
    public static void main(String[] args) {
        System.out.println(fak(10));
        debug = true;
        System.out.println(fak(10));
    }

    public static boolean debug = false;

    private static long fak(long i) {
        long result;
        if (i == 0)
            result = 1;
        else
            result = i * fak(i - 1);
        if (debug)
            System.out.println("DEBUG: " + result);
        return result;
    }
    ...
}

```

---

```

public class Math { //danach
    public static void main(String[] args) {
        System.out.println(fak(10, false));
        System.out.println(fak(10, true));
    }

    private static long fak(long i, boolean debug) {
        long result;
        if (i == 0)
            result = 1;
        else
            result = i * fak(i - 1, debug);
        if (debug)
            System.out.println("DEBUG: " + result);
        return result;
    }
    ...
}

```

### **Aufgabe C: Inline Method Refactoring**

In der Vorlesung wurde das Extract-Method-Refactoring beschrieben. Invers dazu kann man auch ein Inline-Method-Refactoring definieren, welches eine Methode entfernt.

Welche Motivation kann es dafür geben ein Inline-Method-Refactoring durchzuführen? Beschreiben Sie das Vorgehen und die notwendigen Bedingungen.

### **Abgabe des Übungszettels**

*Bitte beachten Sie: Die Abgabe des Übungszettels erfolgt getrennt zu der Abgabe der Hausaufgabe 1. Für die Bearbeitung der Hausaufgabe stehen Ihnen zwei Wochen zur Verfügung.*

*Ihre Lösung dieses Übungszettels geben Sie bitte wie gewohnt bis zum 28.10 23:59 Uhr per Email an [selecture@mathematik.uni-marburg.de](mailto:selecture@mathematik.uni-marburg.de) ab.*