

Übung 12 zur OOS SS16 Bearbeitung bis 30.06.2016**Aufgabe 48: (Abstraktion)**

Gegeben sei folgende C-Prozedur zur Implementierung des Quicksort-Algorithmus:

```
quicksort (M a[], int l, int r) {
    M v,t;
    int i,j;
    if (r > l) {
        v = a[r]; i = l-1; j = r;
        for (;;) {
            while (a[++i].kleiner(v)) ;
            while (v.kleiner(a[--j])) ;
            if (i >= j) break;
            t = a[i]; a[i] = a[j]; a[j] = t;
        }
        t = a[i]; a[i] = a[r]; a[r] = t;
        quicksort(a, l, i-1);
        quicksort(a, i+1, r);
    }
}
```

- a) Machen Sie daraus eine Java-Methode.
- b) Wie muss man diese Klasse abändern, damit die Methode nicht nur auf `int`-Werten operieren kann, sondern auf möglichst allgemeinen Klassen (z.B. auf `Strings`)?
Wie sieht die Implementierung der abgeänderten Klasse aus?
- c) Geben Sie eine Definition einer Oberklasse oder eines Interfaces K für die in b) erwähnten Klassen an. Implementieren Sie hierzu die für die Erweiterung in b) unbedingt benötigten Bedingungen.
- d) Implementieren Sie eine Klasse `KString`, die als Ableitung oder Implementierung von K aus c) implementiert ist und die die Klasse `String` um die in K definierten Methoden erweitert. Sie können dabei davon ausgehen, dass zwei `Strings` sich durch $<_{\text{lex}}$ lexikografisch vergleichen lassen und dass `String` nicht als `final` definiert ist.
- e) Implementieren Sie eine Klasse `KStädte`, die als Ableitung oder Implementierung von K aus c) implementiert ist und die ein Attribut eines Aufzählungstyps mit den deutschen Millionenstädten enthält. Zeigen Sie, dass auch dort das verallgemeinerte Quicksort funktioniert, um die Städte von West nach Ost zu sortieren.

Übung 12 zur OOS SS16 Bearbeitung bis 30.06.2016

Aufgabe 49: (Methode extrahieren mit lokalen Variablen)

Gegeben sei folgende Java-Klasse:

```
public class Üb49Methextr {
    String name = "Vektorname";
    Vector<Integer> v = new Vector<Integer>();
    void ausgabe() {
        int ausgabe = 0;
        for (int i=1;i<=5;i++) {
            v.add(i);
        }
        for (int elem : v) {
            ausgabe += elem;
        }
        System.out.print("Der Vektor " + this.name + " hat den Inhalt ");
        System.out.println(ausgabe + ".");
    }
}
```

- a)** Wenden Sie auf die Methode `ausgabe()` dreimal das Refactoring „Methode extrahieren“ an.
- b)** Wo tun sich Probleme auf und wie lösen Sie diese?
- c)** Formulieren Sie allgemein, wie man bei „Methode extrahieren“ mit lokalen Variablen umgehen muss.

Aufgabe 50: (Lokale Erweiterung einführen als Wrapperklasse)

Eine Wrapperklasse ist eine Hülle um eine existierende Klasse, indem sie die existierende Klasse aggregiert, deren Methoden durch Delegation implementiert und neue Methoden hinzufügt.

Implementieren Sie die Erweiterung der Klasse `Date` um eine Methode `nextDay()`, die das Datum des nächsten Tages liefert, als Wrapper-Klasse.

Aufgabe 51: (Bedingte Ausdrücke konsolidieren)

Wenden Sie „bedingte Ausdrücke konsolidieren“ auf folgendes Programmfragment an:

```
if (onVacation()) {
    if (lengthOfService > 10) {
        return 1;
    }
}
return 0.5;
```

Wenden Sie auf die erhaltene Bedingung noch „Methode extrahieren“ an.