

---

**Softwaretechnik**

<http://swt.informatik.uni-freiburg.de/node/94>  
<http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/>

---

**Übungsblatt 4**

2008-05-30

**Aufgabe 1** (Design by Contract; 7 Punkte)

Auf der Vorlesungsseite finden Sie den Code einer Klasse für Stacks. Leider fehlen alle Vor- und Nachbedingungen sowie die Invarianten. Fügen Sie diese hinzu.

**Aufgabe 2** (Design by Contract und Vererbung; (5+4) Punkte)

Betrachten Sie folgendes Interface, welches eine Abbildung von Schlüssel (vom Typ  $K$ ) auf Werte (vom Typ  $V$ ) spezifiziert.

```
interface Map<K,V> {
    boolean containsKey(K key);
    V get(K key);
    void put(K key, V value);
}
```

Neben den üblichen und offensichtlichen Erwartungen an das Verhalten der Methoden von `Map` (siehe <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html> falls Ihnen diese nicht geläufig sind) sollen mit dem Interface weitere Bedingungen verbunden sein:

- Es werden keine `null` Schlüssel unterstützt.
- Die Methode `get` liefert `null` zurück, falls unter dem angegebenen Schlüssel kein Wert gespeichert ist.

- (a) Fügen Sie den Methoden des Interfaces passende Vor- und Nachbedingungen hinzu.
- (b) Erstellen Sie zwei Subinterfaces von `Map`. Beide Subinterfaces sollen Vor-/Nachbedingungen vorhandener Methoden überschreiben. Eines der Subinterfaces soll diese Methodenspezialisierungen korrekt durchführen, das andere Subinterface soll eine inkorrekt Methodenspezialisierung enthalten. Begründen Sie Ihr Ergebnis.

**Aufgabe 3** (Deadlocks; 4 Punkte)

Schreiben Sie ein Java Program, welches ein Deadlock enthält. Erläutern Sie, wie es zu dem Deadlock kommt.

**Abgabe:** 2008-06-06, 12 Uhr **vor** der Saalübung im HS 00-036, Geb. 101.