

## Arbeitsblatt: Composite Pattern 2

In der Vorlesung haben Sie gesehen, dass die Composite-spezifischen Methoden entweder im Interface *Composite* (SAFE) oder im Interface *Component* (TRANSPARENT) definiert werden können.

Als Composite-spezifische Methoden haben wir folgende Methoden vorgeschlagen:

```
add(Component c)
remove(Component c)
getComponents()
```

Wir gehen in diesem Arbeitsblatt davon aus, dass wir den TRANSPARENT-Ansatz verfolgen wollen, d.h. die Composite-spezifischen Methoden werden im Interface *Component* definiert und müssen sowohl im Composite als auch in den Leaf-Klassen realisiert werden. Für die Leaf-Objekte sind diese Methoden aber in gewisser Hinsicht unnötig.

Auch wenn diese Methoden "unnötig" sind, so müssen sie trotzdem implementiert werden. Geben Sie an, wie die oben aufgeführten Methoden in den Leaf-Klassen (also jene Klassen, die keine Composites sind) konform implementiert werden können.

Sie können die Signaturen dazu auch ändern und/oder neue Methoden definieren, oder die Sichtbarkeit einzelner Methoden einschränken.