

Arbeitsblatt: Composite Pattern 1

Nehmen wir an, dass für den Grafikeditor die beiden Figuren `RectangleFigure` und `OvalFigure` implementiert sind, und dass für Gruppenfiguren die Klasse `GroupFigure` implementiert worden ist. Mit der Methode `addFigure` können zu einer Gruppe neue Figuren hinzugefügt werden.

```
public class GroupFigure implements Figure {
    private String name;
    public GroupFigure(String name, Figure... figures) {
        this.name = name;
        for(Figure f : figures) { addFigure(f); }
    }
    public void addFigure(Figure f) { ... }
    ....
}
```

Betrachten wir folgende Anweisungsfolgen welche mit Gruppenfiguren arbeiten

```
(1) Figure f1 = new RectangleFigure("f1");
    Figure f2 = new RectangleFigure("f2");
    Figure f3 = new OvalFigure("f3");

    GroupFigure g1 = new GroupFigure("g1", f1, f2);
    GroupFigure g2 = new GroupFigure("g2", f3, f1);

(2) Figure f1 = new RectangleFigure("f1");
    Figure f2 = new RectangleFigure("f2");
    Figure f3 = new OvalFigure("f3");

    GroupFigure g1 = new GroupFigure("g1", f1, f2);
    GroupFigure g2 = new GroupFigure("g2", g1, f3);
    g1.addFigure(g2);
```

Aufgaben:

- Diskutieren Sie, warum die Anweisungsfolgen (1) und (2) problematisch sind (zeichnen Sie die Struktur der definierten Gruppenfiguren auf).
- Das Wochenprojekt `06_Composite` enthält obige Definitionen (und die Anweisungsfolgen (1) und (2)) im Paket `patterns.composite`. Ändern Sie den Code so ab, dass problematische Situationen in der Methode `addFigure` erkannt und mit einer `RuntimeException` (z.B. mit einer `IllegalArgumentException`) angezeigt werden. Sie dürfen dazu in allen Klassen weitere Felder (und Methoden) deklarieren, und Sie dürfen, falls das nötig ist, auch die Signaturen der Methoden und Konstruktoren ändern.