

Arbeitsblatt Erzeugen von Objekten 1

Ziele

Wenn in der objektorientierten Programmierung ein Objekt einer Klasse A Methoden in einem Objekt einer Klasse B aufrufen will, so muss Klasse A nur den allgemeinsten Typ kennen, in dem diese Methoden deklariert sind. Das kann ein von der Klasse B implementiertes Interfaces oder eine Basisklasse von B sein.

Um jedoch eine Instanz der Klasse B zu erzeugen, muss bei der `new`-Anweisung der konkrete Klassenname der Klasse B angegeben werden. Damit wird die Klasse A von der Klasse B abhängig, man sagt auch, dass die Klasse A damit stark an die Klasse B gekoppelt wird.

Anhand der folgenden Aufgaben sollen Sie sich Gedanken machen, wie man diese starke Kopplung reduzieren kann.

Ausgangslage

Wir haben Ihnen zu Beginn zusammen mit *JDraw* auch Testklassen abgegeben (im Verzeichnis `src/test/java`). Eine davon, `RectangleTest`, testet das Verhalten der Klasse `Rect` bzgl. seiner Rolle als *Observable*.

Zwischenzeitlich haben Sie (hoffentlich) selbst weitere Figuren dazu programmiert, evtl. eine Gruppenfigur und Dekoratoren. Für alle diese Figuren wäre es eigentlich interessant, die gleichen Tests machen zu können, wie jene in `RectangleTest`.

Aufgaben

- a) Studieren Sie die Klasse `RectangleTest`. Dabei interessiert uns heute weniger der Inhalt der einzelnen Tests, als die Abhängigkeit der Klasse `RectangleTest` von der getesteten Klasse `Rect`.

Um die gleichen Tests auf eine zweite Klasse (z.B. `Line`) anzuwenden, könnten Sie einfach `RectangleTest.java` kopieren zu `LineTest.java`. Wo überall müssten Sie in dieser kopierten Klasse dann etwas ändern, damit sie tatsächlich `Line` und nicht mehr `Rect` testet?

- b) Das Kopieren von Source Code, wie in a) beschrieben, ist eigentlich kein guter Lösungsansatz. Soll z.B. später ein weiterer Test hinzugefügt werden, muss man dies für alle Kopien separat tun. Besser wäre, gemeinsam benutzten Code der Testklassen in eine gemeinsame (abstrakte) Super-Klasse auszufaktorisieren.

Realisieren Sie dies in Ihrem *JDraw*-Projekt für Ihre selbstprogrammierten Figuren (oder – falls Sie keine solchen haben – z.B. für ein mit dem `GreenDecorator` dekoriertes Rechteck. Den `GreenDecorator` finden Sie in den Unterrichtsmaterialien zum Kapitel *09_Decorator1*).

Versuchen Sie, eine Lösung zu finden, die auch noch funktionieren würde, wenn nicht nur an einem sondern an mehreren Orten in `RectangleTest` eine Figur instanziiert werden müsste.

- c) Stellen Sie Ihre Lösung von Aufgabe b) (zusammen mit den getesteten Figurenklassen) schematisch in einem Klassendiagramm dar. Erkennen Sie in diesem Diagramm eine gewisse Struktur?
- d) Wenn Sie die Aufgaben a) bis c) gelöst haben, so testen Sie neben den Figuren die Sie implementiert haben auch eine Decorator-Figur und die Gruppen-Figur. Laufen die Tests erfolgreich durch? Falls nicht, dann können Sie jetzt die Zeit nutzen, um diesen Problemen nachzugehen.