# Auftrag 8: Unit-Tests mit Stub- und Mock-Klassen

Im Modul 226-1 haben Sie bereits Unit-Tests mit JUnit implementiert. Sie haben dort gesehen, dass ein Unit-Test dazu dient eine einzelne Klasse isoliert zu testen.

In der Einführung „*R226B-08.1 Einführung in Stub- und Mock-Klassen*“ (siehe Ressourcen zu dieser Lektion) wird nun gezeigt, dass es Fälle gibt, wo es schwierig ist, eine Klasse isoliert zu testen. In solchen Fällen arbeitet man mit sogenannten *Stub-* oder *Mock-Objekten*. Genau diese Technik werden Sie in diesem Arbeitsblatt selber anwenden.

## Aufgabe 1: Klassen über Interface entkoppeln

*Einleitung:*

Bevor man eine Klasse isoliert testen kann, müssen die Abhängigkeiten zu anderen, störenden Klassen aufgebrochen werden. Im Screencast „*person.mp4*“ wird gezeigt, wie man dies durch Einführung eines Interfaces macht.

K 4.1

*Aufgabe:*

In den Ressourcen finden Sie die beiden Klassen FigurParser und FigurFileDAO. Importieren Sie diese beiden Klassen in Ihr Grafikeditor-Projekt. Isolieren Sie sodann die beiden Klassen voneinander durch die Einführung eines geeigneten Interfaces.

Zeichnen Sie für die Situation nach der Isolierung ein UML-Klassendiagramm.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Aufgabe 2: Stub-Klasse schreiben und in JUnit-Test einsetzen

*Einleitung:*

Nach der Isolation der beiden Klassen FigurParser und FigurFileDAO soll nun die parse()-Methode der Klasse FigurParser in einem Unit-Test getestet werden, wie diese in der zweiten Hälfte des Screencasts „*person.mp4*“ gezeigt wird.

K 4.2   
K 4.3

*Aufgabe:*

Schreiben Sie einen JUnit-Testfall für die Klasse FigurParser und eine Stub-Klasse, welche die Schnittstelle aus Aufgabe 1 implementiert. Stellen Sie sicher, dass Ihr Testfall die Methode parse() vollständig abdeckt.

## Vertiefungs- und Ergänzungsaufgaben

Die folgenden Aufgaben dienen Ihrer individuellen Vertiefung. Sie können diese nach Belieben bearbeiten und dadurch auch Zusatzkompetenzen erlangen.

#### Klasse Kreis mit Stub-Klasse für Graphics testen

*Einleitung:*

Im Screencast „*zeichneTest mit Stub.mp4*“ wird gezeigt, wie man die Methode zeichne(Grapghics g) der Klasse Rechteck des Grafikeditors testen kann. Die Schwierigkeit dabei liegt beim Parameter g der Methode. Hier muss beim Aufruf eine Referenz auf ein Graphics-Objekt mitgegeben werden. Im Unit-Test steht ein solches aber nicht zur Verfügung. Deshalb wir eine Stub-Klasse für Graphics eingesetzt.

K 4.4

*Aufgabe:*

Schreiben Sie einen Unit-Test für die Klasse Kreis Ihres Grafikeditors. Implementieren Sie eine eigene Stub-Klasse für die Graphics-Klasse und testen Sie damit die Methode zeichne(Grapghics g).

Ihr Test soll insbesondere folgenden Umstand überprüfen: In der Klasse Kreis entsprechen die beiden von Figur geerbten Instanzvariablen x und y den Koordinaten des Kreismittelpunktes. Die drawOval()-Methode von Graphics erwartet beim Aufruf allerdings einen anderen Punkt als Ort (siehe Java-API-Doc von Graphics). Haben Sie die Anpassung zwischen den beiden Orten korrekt implementiert? Haben auch die weiteren Argumente beim Aufruf von drawOval() den richtigen Wert?

Wenn Sie nicht sicher sind, machen Sie nachfolgend eine Skizze mit einem Kreis. Zeichnen Sie den Mittelpunkt und den Radius ein. Zeichnen Sie anschliessend ein, welchen Punkt und welche Grössen die Methode drawOval() braucht, damit der Kreis richtig gezeichnet wird.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### Klasse Zeichnung mit Mock-Klasse für Graphics testen

*Einleitung:*

Im Screencast „*zeichneTest mit Mock.mp4*“ wird gezeigt, wie mit der Bibliothek *Mockito* sehr schnell Mock-Klassen erstellt und für einen Unit-Test konfiguriert werden können. Insbesondere ist es sehr einfach, zu überprüfen, welche Methoden beim Mock aufgerufen worden sind und mit welchen Argumenten dies geschehen ist.

K 4.5

*Aufgabe:*

Schreiben Sie einen Unit-Test für die Klasse Zeichnung Ihres Grafikeditors. Fügen Sie der Zeichnung für den Test einige Figur-Objekte hinzu. Testen Sie sodann die Methode zeichneFiguren(Graphics g) und verwenden Sie dabei ein Mock-Objekt für den Graphics-Parameter. Überprüfen Sie im Test, ob beim Graphics-Objekt die richtigen draw-Methoden, in der richtigen Reihenfolge und mit den richtigen Argumenten aufgerufen worden sind. Setzen Sie *Mockito* ein. In den Ressourcen finden Sie weitere Hilfe dazu.