

Übung 1 – JUnit

Die Aufgaben auf diesem Übungsblatt sollen Sie mit der Arbeit mit Unittests vertraut machen. Dabei beziehen sich die Tests, die Sie schreiben sollen, auf Klassen aus dem vorhergehenden Übungsblatt.

Aufgabe 1 (Theorie)

Welche Annotationen für JUnit gibt es, und welche Zwecke erfüllen sie?

Aufgabe 2

Sie haben im vorigen Übungsblatt einen Stapel und eine Schlange modelliert und unter Verwendung eines Feldes (Arrays) implementiert. Zu diesen Implementierungen sollen Sie nun einige passende Tests schreiben. Schreiben Sie hierzu die Klassen **SchlangeMitArrayTest** und **StapelMitArrayTest**. Überlegen Sie sich wo Sie die Dateien am besten in Ihrer IDE ablegen, damit die Tests nicht direkt neben den zu testenden Klassen liegen.

- Schreiben Sie Tests mit JUnit, um die Funktionalität von **StapelMitArray** und **SchlangeMitArray** zu testen. *Welche Sonderfälle können dabei auftreten?*
- Prüfen Sie insbesondere auch die korrekte Erzeugung von **Exceptions**. Verwenden Sie dabei Lambdafunktionen.

Aufgabe 3

Schreiben Sie ein funktionales Interface **Funktion**. Das Interface repräsentiert eine Funktion, die auf einen ganzzahligen Wert angewendet wird (wir vernachlässigen hierbei eventuelle numerische Fehler). Hierzu hat das Interface die Methode **auswerten**, die einen ganzzahligen Wert annimmt und einen ganzzahligen Rückgabewert hat. Sie wendet die Funktion auf das übergebene Argument an.

Implementieren Sie in Ihrer Klasse **StapelMitArray** eine Methode **applyToAll**. Diese hat keinen Rückgabewert und nimmt als Argument eine Referenz vom Typ **Funktion** an. Sie soll die übergebene Funktion auf alle Werte des Stapels anwenden und diese damit verändern (es wird kein neuer Speicher angelegt!). In der Methode soll in einer Schleife über alle Werte des internen Array iteriert werden, der Methode **auswerten** der jeweilige Wert des Arrays übergeben werden und das Ergebnis wieder diesem Wert zugewiesen werden, um diesen zu verändern.

Schreiben Sie nun Tests für die Methode **applyToAll**, in der Sie zunächst einen Stapel mit Werten anlegen. Übergeben Sie verschiedene Lambda-Ausdrücke an die **applyToAll** Methode. Ein Lambda-Ausdruck soll den Wert verdoppeln, ein anderer quadrieren (ohne **Math.pow!**)."



Aufgabe 4

Testen Sie JUnit-Tests für die Klassen `Student` und `Boxer`, um diese Klassen zu testen. Prüfen Sie dabei auch das korrekte Verhalten Ihrer Implementierung der Methode `equals` (siehe Java-API) auf

- Reflexivität;
- Symmetrie;
- Transitivität;
- Konsistenz;
- Korrekter Vergleich mit `null`.