

## Übungen zu Einführung in die Software-Entwicklung

*Sommersemester 2021*

### Blatt 2

#### Aufgabe 2.1: Singleton, Garbage Collector (25 Punkte)

In dieser Aufgabe sollen Sie eine Art altertümlichen Börsenticker simulieren, der alle Aktienkurse nacheinander in einer langen Zeichenkette auf dem Bildschirm ausgibt.

Implementieren Sie dazu die Klasse `Ticker` nach dem Singleton - Entwurfsmuster. Ein Objekt der Klasse `Ticker` enthält die Methode `print(String text)`, die den übergebenen `String` auf der Standard-Konsole ausgibt. Dabei werden alle Zeilenumbrüche entfernt und jeder übergebene `String` mit `+++` von den vorherigen und nachfolgenden `String`-Objekten getrennt.

Implementieren Sie zusätzlich die Klasse `Company`, die eine beliebige Firma repräsentieren soll. Jede `Company` enthält einen Namen und die Methode `changeStockPrice(double d)`, die mit dem übergebenen `double` und einer geeigneten Meldung über den auf `d` geänderten Aktienkurs die Methode `print` von `Ticker` aufruft. Außerdem soll die Klasse `Company` um einen Destruktor, der bei Aufruf einen `String` mit einer Nachricht über die Insolvenz der Firma an den `Ticker` schickt, erweitert werden.

Beispielsweise könnte die Ausgabe auf der Kommandozeile folgendermaßen aussehen:

```
+++Weyland Yutani 528.0+++Umbrella 491.0+++Dharma is insolvent
```

Nachdem Sie die oben beschriebene Funktionalität umgesetzt haben, bearbeiten Sie folgende Teilaufgaben:

1. Erklären Sie Ihrem Tutor anhand der implementierten Klassen die Vor- und Nachteile des Singleton - Entwurfsmusters
2. Implementieren Sie zusätzlich eine Testklasse, die durch die Erzeugung von Objekten des Typs `Company` und expliziten Anstoß des *Garbage Collectors* dessen Arbeitsweise im Zusammenspiel mit *Konstruktoren* und *Destrukturen* verdeutlicht. Erklären Sie Ihrem Tutor wie Sie vorgegangen sind und was man aus den Ergebnissen Ihrer Testklasse schließen kann.

#### Aufgabe 2.2: Copy Constructor (30 Punkte)

Betrachten Sie die Klassen `StringStack` und `StringStackEntry` mit denen eine Datenstruktur vom Typ *Stack* implementiert wurde, die auf Verweisen beruht. Ergänzen Sie den Quellcode um einen *Copy-Constructor*, mit dem es ermöglicht wird, eine Kopie des übergebenen `StringStack` zu erzeugen. Beurteilen Sie, ob Ihr Konstruktor eine *deep* oder *shallow-copy* erzeugt und testen Sie sein Verhalten in einer separaten Testklasse auf Korrektheit. Für die Testklasse empfiehlt es sich, eine separate Klasse mit Testmethoden anzulegen, die sie auch in noch folgenden Aufgaben wieder benutzen können.

### Aufgabe 2.3: Testen (25 Punkte)

Bei den jährlich stattfindenden *Hungerspielen* treten 24 *Tribute* aus den 12 Distrikten des fiktiven Landes Panem in einer künstlichen Arena gegeneinander an. Bei der 75. Durchführung der Hungerspiele ist die kreisrunde Arena analog zu einer Uhr in zwölf gleich große Bereiche unterteilt. Die Mitte der Arena habe die Position (0,0) und die Arena einen Durchmesser von 3 Meilen. Je nach Uhrzeit wird ein Bereich aktiv und in ihm werden tödliche Fallen freigeschaltet. Beispielsweise ist von 12 bis 1 Uhr der Bereich im Winkel von  $0^\circ$  bis  $30^\circ$  aktiv, wobei  $0^\circ$  die Linie von der Mitte bis zur Position (0,1.5) anzeigt. Ein Tribut wird fast sicher umkommen, wenn er sich in einem aktiven Bereich aufhält oder die Arena verlässt.

Schreiben sie eine Java-Klasse `Arena`, die die Methode `int getArea(double x, double y)` enthält und zurück gibt, in welchem der zwölf Bereiche sich ein Tribut mit der Position (x,y) befindet. Wenn sich der Tribut außerhalb der Arena befindet, soll die Methode `-1` zurück geben. Die Methoden in der Klasse `java.lang.Math` können dafür nützlich sein. Testen Sie Ihre Implementation ausführlich mit einem automatisierten Testprogramm. Wie viele Testfälle müssen Sie mindestens erzeugen, um alle möglichen Eingaben hinsichtlich einer korrekten Ausgabe geprüft zu haben? Wenn Sie separate Hilfs-Methoden implementieren, sollten Sie diese ebenfalls testen. Für die Testklasse empfiehlt es sich, die separate Testklasse mit den wiederverwendbaren Testmethoden aus Aufgabe 2.2 zu erweitern.

### Aufgabe 2.4: Fragen (20 Punkte)

Beantworten Sie Ihrer Tutorin / Ihrem Tutor Fragen zu den verschiedenen Arten von Kopien die Sie innerhalb der Veranstaltung kennengelernt haben, sowie einige generelle Fragen zum Thema *Java*.