

Übungen zu Informatik B

Sommersemester 2018

Blatt 1

Aufgabe 1.1: Sichtbarkeit (20 Punkte)

Betrachten Sie die Datei `Fraglich.java`. Geben Sie an den markierten Stellen 1 bis 15 die Werte der Variablen `a`, `b` und `c` an und begründen Sie Ihre Wahl kurz schriftlich. Denken Sie bei Arrays daran, den Inhalt der einzelnen Einträge anzugeben. Erklären Sie Ihrem Tutor **schriftlich** die Begriffe *call-by-value* und *call-by-reference* in Java.

Aufgabe 1.2: Fraction (30 Punkte)

Implementieren Sie die Klasse `Fraction` mit den beiden Instanzvariablen `numerator` und `denominator` zur Repräsentation eines Bruches.

Es soll zwei Konstruktoren geben, die den Bruch bei der Instanziierung kürzen. Einer, der eine ganze Zahl annimmt und den Nenner auf 1 setzt und ein zweiter, der mit Nenner und Zähler aufgerufen wird. Verketteten Sie die Konstruktoren.

Implementieren Sie zusätzlich die folgenden Instanzmethoden, die bei der Ausführung der entsprechenden numerischen Operation immer eine neue `Fraction` erzeugen.

- `Fraction multiply(int factor)`
- `Fraction multiply(Fraction factor)`
- `Fraction divide(Fraction divisor)`
- `Fraction multiply(Fraction... factors)`

Dabei erwartet die Methode `multiply(Fraction... factors)` eine variable Liste von `Fraction` Instanzen. D.h. sie kann mit beliebig vielen Argumenten vom Typ `Fraction` aufgerufen werden. Machen Sie es möglich, eine Instanz mit der Methode `String toString()` als `String` (z.B. `1/4`) auszugeben.

Testen Sie Ihre neue Klasse mit Hilfe einer separaten Testklasse. Das Testprogramm soll einige Instanzen der Klasse `Fraction` erzeugen und jede der programmierten Operationen mindestens einmal testen. Es soll automatisiert eine Ausgabe erfolgen, ob der jeweilige Test erfolgreich war oder nicht.

Aufgabe 1.3: Copy Constructor (25 Punkte)

Betrachten Sie die Klassen `StringStack` und `StringStackEntry` mit denen eine Datenstruktur vom Typ *Keller* (engl. *util.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 1.4: 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 einen 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.