

Themen:

Konstruktoren, Referenzen

Aufgabe 3.1

In den Kontodaten bei der Pleite & Geier-Bank werden der Name des Kontoeigentümers, die Kontonummer und der aktuelle Kontostand verwaltet.

- a) Definieren Sie eine Klasse `Konto` zur Verwaltung dieser Daten mit entsprechenden Methoden.

Der Konstruktor erhält die Kontonummer und den Namen des Kontoinhabers als Parameter. Es soll keine Methode geben, um die Kontonummer zu ändern, der Name des Eigentümers kann sich ändern. Änderungen des Kontostands sind nur über Methoden zum Abheben und Einzahlen möglich. Diese beiden Methoden ändern den aktuellen Kontostand um den entsprechenden Betrag und geben den **neuen** Kontostand zurück.

Der Kontostand soll abgefragt werden können, ebenso die Anzahl der Buchungen (Einzahlungen/Abhebungen) seit Kontoeröffnung.

Eine `toString`-Methode soll den Namen des Inhabers, die Kontonummer und den aktuellen Kontostand als String liefern.

- b) Implementieren Sie eine Testklasse, in der Sie verschieden **Testmethoden** (ohne Parameter) definieren.

Jede Testmethode stellt dabei ein eigenes „Szenario“ dar, zB:

- Erstellen eines Kontos, Ausgabe der Kontodaten auf dem Monitor
- Erstellen eines Kontos, Einzahlung von 1000,00 Euro, Abheben von 200,00 Euro (und Ausgabe)
- Erstellen von (mindestens) 2 Konten, mehrere Ein- und Auszahlungen auf jedes der beiden Konten (in „vermischter“ Reihenfolge), Ausgabe aller Kontodaten

In der `main`-Methode werden nur diese Testmethoden aufgerufen.

Aufgabe 3.2

Bei einem Open-Air-Festival werden für verschiedene Bühnen Tickets verkauft.

Die Preise pro Ticket sind für die verschiedenen Bühnen unterschiedlich.

Für jede Bühne können Tickets verkauft werden.

Es soll für jede Bühne die Anzahl der verkauften Tickets abgefragt werden können.

- a) Erstellen Sie eine Klasse `Buehne` mit geeigneten Instanz-Attributen und einem Konstruktor. Der Ticket-Preis für die Bühne wird dem Konstruktor mitgegeben und kann anschließend nicht mehr geändert werden.
- b) Definieren Sie eine Instanz-Methode zum Tickets verkaufen. Dabei soll der Kauf mehrerer Tickets für dieselbe Bühne pro Kauf möglich sein.
- c) Definieren Sie eine zweite Instanz-Methode zur Abfrage der Anzahl der verkauften Tickets für eine Bühne.
- d) Schreiben Sie in einer `Main`-Klasse einige Tests, in denen verschiedene Bühnen erstellt werden, Tickets für die unterschiedlichen Bühnen verkauft werden und am Ende für jede Bühne die Gesamtzahl der verkauften Tickets ausgegeben wird.

Aufgabe 3.3

Betrachten Sie die Klasse Punkt2D aus Kapitel 3.

Nehmen Sie an, die folgende Test-Methode werde ausgeführt:

```
1 public static void test() {  
2     Punkt2D p;  
3     Punkt2D q;  
4     p = new Punkt2D(2.0, -3.0);  
5     q = p;  
6     p = new Punkt2D(7.0, 4.0);  
7     q = p;  
8     q.moveTo(1.0, 2.0);  
9 }
```

Zeichnen Sie Speicherbilder, die die Situation nach Ausführung der Zeilen 2 ... 8 darstellen.

Zeichnen Sie das Bild nach jeder der Codezeilen 2 ... 8 neu!