

Übungsaufgaben 1

1.1 Überschreiben der `toString()`-Methode

Bringen Sie die Klasse `Vektor2D` aus der Vorlesung zum Laufen.

Überschreiben Sie für die Klasse `Vektor2D` und `Punkt` die Methode `toString()`. Die `toString()`-Methode soll in der Klasse `Punkt` die Koordinaten des Punktes liefern (Beispiel: `(x, y)`) und in der Klasse `Vektor2D` die Attribute zurückliefern (Beispiel: `(delX, delY), anker: (x, y)`). Testen Sie die Klasse und ihre Instanzmethoden in einer separaten Testklasse `Vektor2DTest`.

Beispiele für Vektoren:

Δx	Δy	Anker	Betrag
1	0	(1, 1)	1
0	1	(2, 2)	1
1	4	(3, 4)	4,12
3	2	(2, 3)	3,61
3	5	(3, 3)	5,83

1.2 Instanz- in Klassenmethoden umwandeln

Wandeln Sie dann die Instanzmethoden `betrag()` in eine Klassenmethode um. Testen Sie diese Klassenmethode in der Testklasse `Vektor2DTest` und vergleichen Sie die Methodenaufrufe der Instanz- und Klassenmethoden miteinander.

Implementieren Sie eine neue Methode `multipliziereMitSkalar()` einmal als eine Instanz- und einmal als eine Klassenmethode. Testen Sie beide Methode durch geeignete Aufrufe in der Testklasse `Vektor2DTest`.

1.3 Objektarray

Erstellen Sie in der Testklasse `Vektor2DTest` ein Array aus Objekten vom Typ `Vektor2D`. Implementieren Sie eine Methode (in welcher Klasse wäre das sinnvoll?), die ein solches Array (d.h. alle seinen Elemente) in gut lesbarer Form auf der Konsole ausgibt. Verwenden Sie hierzu die `for-each`-Schleife.

1.4 Matrizen ausgeben

Implementieren Sie (in einer neuen Klasse) eine Java-Methode, die eine gegebene Matrix aus float-Zahlen (d.h. ein 2-dimensionales float-Array) auf dem Bildschirm ausgibt.

1.5 Häufigstes Element im Array suchen

Implementieren Sie in einer Klasse `Arrayelement` eine Java-Methode `haeufigstesElem()`, die ein beliebiges Array `arr` aus `byte`-Werten als Parameter bekommt und ein Array mit Elementen aus `arr` zurückliefert, die in `arr` *am häufigsten* vorkommen. Je nachdem, ob es nur ein Element in `arr` gibt, das am häufigsten vorkommt, oder mehrere solche Elemente, kann das Rückgabearray aus einem oder mehreren Elementen bestehen.

Zur Lösung dieser Aufgabe können weitere Hilfsmethoden und/oder vordefinierte Java-Methoden z.B. aus der Klasse `Arrays` verwendet werden.

Kommentieren Sie die wichtigsten Schritte jeder Methode. Achten Sie auf die Namens- und Formatierungskonventionen. Testen Sie Ihr Programm in einer separaten Testklasse.