

# Übungsaufgabe – Arithmetik mit Vektoren

---

In dieser Aufgabe soll ein “Taschenrechner” entwickelt werden, welcher arithmetische Operationen (addieren, subtrahieren, multiplizieren) auf Vektoren ausführt. Dividieren gibt es in der Vektorarithmetik nicht.

## Was ist ein Vektor?

Ein Vektor ist sehr einfach ausgedrückt eine Liste an Zahlen. Die Dimension eines Vektors wird über seine Anzahl an Zahlen bestimmt.

Zum Beispiel ist  $\vec{a} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$  ein zweidimensionaler (weil er zwei Werte besitzt) Vektor. Vektoren können aber auch mehr Dimensionen besitzen. Ein eindimensionaler Vektor ist ein Skalar (eine Zahl).

Vektoren lassen sich addieren, subtrahieren, multiplizieren. Jedoch nicht dividieren.

### 1. Addition

Man nimmt die Werte an der jeweils gleichen Position und addiert diese.

Beispiel:

Gegeben seien Vektoren  $\vec{a} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$  und  $\vec{b} = \begin{pmatrix} 6 \\ 7 \end{pmatrix}$ .

Die Summe beider Vektoren ist ein neuer Vektor  $\vec{c}$ :

$$\vec{c} = \vec{a} + \vec{b} = \begin{pmatrix} 9 \\ 11 \end{pmatrix}$$

### 2. Subtraktion

Man nimmt die Werte an der jeweils gleichen Position und subtrahiert diese.

Beispiel:

Gegeben seien Vektoren  $\vec{a} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$  und  $\vec{b} = \begin{pmatrix} 6 \\ 7 \end{pmatrix}$ .

Die Differenz beider Vektoren ist ein neuer Vektor  $\vec{c}$ :

$$\vec{c} = \vec{a} - \vec{b} = \begin{pmatrix} -3 \\ -3 \end{pmatrix}$$

### 3. Multiplikation (Skalarprodukt)

Man nimmt die Werte an der jeweils gleichen Position, multipliziert diese und addiert die Produkte. Das Ergebnis ist also ein skalarer Wert (eine Zahl).

Beispiel:

Gegeben seien Vektoren  $\vec{a} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$  und  $\vec{b} = \begin{pmatrix} 6 \\ 7 \end{pmatrix}$ .

Das Skalarprodukt ist ein reeller Zahlenwert  $c$ :

$$c = \vec{a} \cdot \vec{b} = 3 * 6 + 4 * 7 = 46$$

## Implementierung

Im Code gibt es schon eine abstrakte Klasse `Type`, welches ein `Equatable` Interface nützt. Die Methoden dieses Interfaces werden jedoch nicht in der abstrakten Klasse implementiert.

Das `Equatable` Interface gibt Methoden für das Addieren, Subtrahieren, Multiplizieren und Dividieren von beliebigen Typen vor.

Deine Aufgabe ist es eine Klasse `Vector` zu implementieren, welches von `Type` erbt und die Methoden aus dem `Equatable` Interface entsprechend implementiert.

Als Beispiel wurde schon eine Klasse `Scalar` implementiert um zu zeigen, wie die Methoden des `Equatable` Interfaces implementiert werden müssen.

Vergiss nicht die `toString()` deiner `Vector` Klasse entsprechend zu überschreiben.

## Hinweis

Schau dir zuerst genau an, wie das vorgegebene Programm funktioniert. Überlege dir dann einen Lösungsweg.