

# Belegarbeit I - Handbuch

[Einleitung](#)

[Vector](#)

[get:](#)

[set:](#)

[reinitializeVector:](#)

[getDimension:](#)

[VectorHelper](#)

[getHigherDimension:](#)

[compare:](#)

[ConsoleHelper](#)

[Testfälle](#)

[Hauptprogramm](#)

[Vektorenverwaltung](#)

## Einleitung

Dieses Dokument stellt das Handbuch für die 1. Belegarbeit der Programmierung II Veranstaltung im Wintersemester 2014/15 dar. Die Aufgabe war eine Konsolenanwendung zu entwickeln, mit deren Hilfe verschiedene Rechnungen mit Vektoren möglich sind. Dafür sollten außerdem Testfälle und eine übersichtliche Menüführung erstellt werden.

## Vector

Die 'Vector'-Klasse repräsentiert einen Vektor. Sie definiert dafür zwei Attribute. Ein Double-Array 'vector', welches den eigentlichen Vektor abbildet und ein statisches Attribut 'MaxDimension', um die maximale Dimension aller Vektoren zu limitieren. Letzteres ging aus der Aufgabenstellung hervor und wurde auf 6 gesetzt.

Zur Klasse gehört ein expliziter Konstruktor, der einen Integer-Wert zur Festlegung der Dimension entgegennimmt. Damit wird das 'vector'-Array erzeugt, also dessen Länge festgelegt.

Um den Zugriff auf das 'vector'-Array zu erleichtern, wurden folgende Methoden implementiert:

[get:](#)

Nimmt einen Integer-Indexwert entgegen und gibt den Wert aus dem Array zurück.

[set:](#)

Nimmt einen Integer-Indexwert und einen Double-Wert entgegen. Damit lassen sich die Werte des Arrays leicht an ihrer Indexposition ändern.

#### reinitializeVector:

Um nicht ständig auf 'IndexOutOfBoundsException' zu stoßen, soll diese Methode das 'vector'-Array mit der höheren Dimension neu initialisieren. Das sorgt dafür, dass bei Berechnungen von Vektoren verschiedener Dimensionen ohne Fehler durchgeführt werden können.

#### getDimension:

Gibt die Länge des Arrays zurück, die ja die Dimension repräsentiert.

Um die Ausgabe zu erleichtern, wurde die 'toString'-Methode aus der 'Object'-Klasse überschreiben.

### VectorHelper

Die 'VectorHelper'-Klasse stellt die geforderten Methoden aus der Aufgabenstellung bereit. Da im Quelltext der Anwendung ausschließlich englische Namen verwendet werden, ist hier eine Zuordnung der geforderten Rechenoperationen zu den vorhandenen Java-Methoden:

Operation	Methodenname	Eingabeparamameter	Rückgabotyp
Eingabe	input	-	Vector
Ausgabe	output	Vector	void
Summe	sum	Vector, Vector	Vector
Differenz	sub	Vector, Vector	Vector
Skalarprodukt	getScalarProduct	Vector, Vector	Double
Einheitsvektor	getUnitVector	Vector	Vector
Länge	getLength	Vector	Double
Vektorprodukt	getCrossProduct	Vector, Vector	Vector
Spatprodukt	getDeterminant	Vector, Vector, Vector	Double

Für den leichteren Umgang mit den Vektoren wurden weitere Methoden erstellt:

#### getHigherDimension:

Liefert die höhere Dimension zweier Vektoren zurück. Das ist insbesondere bei Iterationen über zwei Vektoren sehr hilfreich.

#### compare:

Vergleicht die Werte von zwei Vektoren und gibt true zurück, wenn die Vektoren gleich sind. Die Methode wird beim Testen verwendet.

#### getVectorByUser:

Kontrolliert die Benutzereingabe eines Vektors für die Rechenoperationen. Bei einer fehlerhaften Eingabe, wird die Eingabeaufforderung wiederholt.

### **UserInputException**

Die 'UserInputException' wird geworfen, wenn bei der Eingabe des Benutzernamens ein Fehler auftritt. Damit die Anwendung nicht abstürzt wird die Eingabe abgebrochen und zurück ins Hauptmenü gewechselt.

### **ConsoleHelper**

Stellt statische Methoden bereit, um Werte aus der Konsole auszulesen. Angeboten werden das Auslesen von Double- oder Integer-Werten oder einfach Text.

### **Testfälle**

Um die Vektoroperationen zu testen wurden vier Testmethoden implementiert. Sie werden gleich nach dem Anwendungsstart ausgeführt. Dafür werden drei default Vektoren mit drei Dimensionen erstellt. Getestet werden: Einheitsvektor, Skalarprodukt, Vektorprodukt (Kreuzprodukt) und Determinante (Spatprodukt). Nach jedem Test wird abschließend das jeweilige Ergebnis ausgegeben.

```
Programmieren II - Belegarbeit I
-----

initialize default vectors ['a', 'b', 'c']...

Testing:
-----

Calculate unit vector:
vector 'b': [ 2.5, 6.0, 10.0 ]
result: [ 0.20961090407515925, 0.5030661697803822, 0.838443616300637 ]
Test 'UnitVector' returned : true

Calculate scalar product of 'c' and 'b':
vector 'c': [ 3.0, 4.0, -9.0 ]
vector 'b': [ 2.5, 6.0, 10.0 ]
result: -58.5
Test 'ScalarProduct' returned : true

Calculate cross product of 'c' and 'a':
vector 'c': [ 3.0, 4.0, -9.0 ]
vector 'a': [ -4.0, 5.0, 9.0 ]
result:
[ 81.0, 9.0, 31.0 ]
Test 'CrossProdukt' returned : true

Calculate determinant of 'a', 'b' and 'c':
a: [ -4.0, 5.0, 9.0 ], b: [ 2.5, 6.0, 10.0 ], c: [ 3.0, 4.0, -9.0 ]
result: 566.5
Test 'Determinant' returned : true
```

## Hauptprogramm

Der Hauptteil der Konsolenanwendung findet nach den Tests statt. Der Benutzer befindet sich zunächst im Hauptmenü und hat dort die Möglichkeit jede der vorhandenen Vektoroperationen auszuprobieren. Dafür stehen die drei Vektoren aus den Test weiterhin zur Verfügung.

```
Hauptmenue:
-----
1) Vektoren verwalten
2) Summe
3) Differenz
4) Skalarprodukt
5) Einheitsvektor
6) Laenge eines Vektors
7) Vektorprodukt
8) Determinante (Spatprodukt) (n=3)
10) Beenden
```

## Vektorenverwaltung

Um eigene Vektoren zu erstellen kann mit '1' in das Untermenü "Vektorverwaltung" gewechselt werden. Dort lassen sich Vektoren anlegen / bearbeiten, löschen oder wieder zur Ausgangssituation zurücksetzen. Vor allem werden hier aber auch alle Vektoren aufgelistet

```
Vektorenverwaltung:
-----

aktuell definierte Vektoren:
a: [ -4.0, 5.0, 9.0 ]
b: [ 2.5, 6.0, 10.0 ]
c: [ 3.0, 4.0, -9.0 ]

1) Hinzufuegen / Bearbeiten
2) Loeschen
3) Vektoren zuruecksetzen
10) Abbrechen
```

```
Name des Vektors:newVector
Dimension zwischen 1 und 6 festlegen:5
Wert 1:
7.4
Wert 2:
-14.8
Wert 3:
17
Wert 4:
0
Wert 5:
-3.008
```

Mit '10' kehren wir wieder zurück ins Hauptmenü. Dort können jetzt die gerade erstellten Vektoren für alle Rechenoperationen genutzt werden. Unterschiedliche Dimensionen spielen dabei keine Rolle. Die Anwendung ist intelligent genug die fehlenden Achsenwerte mit Nullen aufzufüllen.

```
Determinante welcher Vektoren? (n=3)
newVector
b
a
Ergebnis: 1575.1
```

Zum Beenden des Programms ins Hauptmenü wechseln und dort dann Option '10' auswählen.