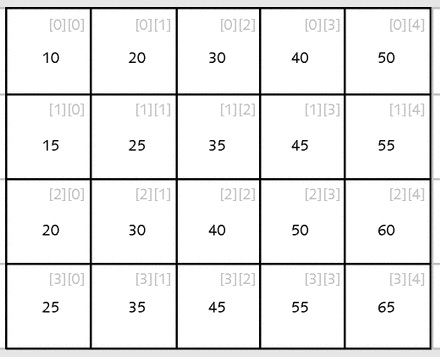
### Eine Matrix ausgeben

**Ziel:**

Deklaration einer Matrix mit sofortiger Initialisierung von Werten. Legen Sie eine Matrix mit den folgenden Werten an und lassen Sie sich die Matrix ausgeben.



**Ausgabe/Darstellung:**



**Programmcode:**

**public** **class** Matrizen01 {

**public** **static** **void** main(String[] args) {

**int**[][] matrix ={{10,20,30,40,50},

{15,25,35,45,55},

{20,30,40,50,60},

{25,35,45,55,65}};

**for** (**int** i=0; i<matrix.length; ++i) {

**for** (**int** j=0; j<matrix[i].length; ++j) {

System.***out***.print(matrix[i][j] +" ");

}

System.***out***.println();

}

}

}

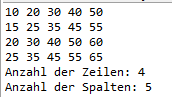
### Eine Matrix ausgeben sowie Anzahl Zeilen/Spalten bestimmen

**Ziel:**

Deklarieren Sie die unten angegebene Matrix (mit dem Namen „matrix“), wobei diese direkt mit den unten angegebenen Werten initialisiert werden soll.

Lassen Sie sich die Matrix und darunter die Anzahl der Zeilen sowie die Anzahl der Spalten ausgeben.

**Ausgabe/Darstellung:**



**Programmcode:**

**public** **class** Matrizen01 {

**public** **static** **void** main(String[] args) {

**int**[][] matrix = { { 10, 20, 30, 40, 50 }, { 15, 25, 35, 45, 55 },

{ 20, 30, 40, 50, 60 }, { 25, 35, 45, 55, 65 } };

**for** (**int** i = 0; i < matrix.length; ++i) {

**for** (**int** j = 0; j < matrix[i].length; ++j) {

System.***out***.print(matrix[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("Anzahl der Zeilen: " + matrix.length);

System.***out***.println("Anzahl der Spalten: " + matrix[0].length);

}

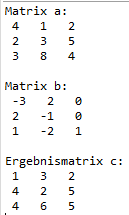
}

### Matrizenaddition (ohne Methode)

**Ziel:**

In einem Programm „MatrizenAddition01“ sollen die Matrizen A =  und B =  zu einer Ergebnismatrix C addiert und ausgegeben werden.

**Ausgabe/Darstellung:**



**Programmcode:**

**public** **class** MatrizenAddition01 {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 4, 1, 2 }, { 2, 3, 5 }, { 3, 8, 4 } };

**int** b[][] = { { -3, 2, 0 }, { 2, -1, 0 }, { 1, -2, 1 } };

System.***out***.println("Matrix a:");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[i].length; ++j) {

System.***out***.print(" " + a[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("\nMatrix b:");

**for** (**int** i = 0; i < b.length; ++i) {

**for** (**int** j = 0; j < a[i].length; ++j) {

System.***out***.print(" " + b[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("\nErgebnismatrix c:");

**int**[][] c = **new** **int**[a.length][a[0].length];

**for** (**int** i = 0; i < a.length; i++)

**for** (**int** j = 0; j < a[0].length; j++)

c[i][j] = a[i][j] + b[i][j];

**for** (**int** i = 0; i < a.length; i++) {

**for** (**int** j = 0; j < a[0].length; j++)

System.***out***.print(" " + c[i][j] + " ");

System.***out***.println();

}

}}

### Matrizenaddition (mit Methode)

Formen Sie das letzte Programm „MatrizenAddition01“ zu einem Programm „MatrizenAddition02“ um, bei dem die eigentliche Addition durch einen Methodenaufruf der Form **int** c[][] = *addMatrix*(a, b); erfolgt.

**Programmcode:**

**public** **class** Matrizen03 {

**public** **static** **void** main(String[] args) {

**int** a[][] = {{4, 1, 2}, {2, 3, 5}, {3, 8, 4}};

**int** b[][] = {{-3, 2, 0}, {2, -1, 0}, {1, -2, 1}};

System.***out***.println("Matrix a:");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[i].length; ++j) {

System.***out***.print(" " +a[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("Matrix b:");

**for** (**int** i = 0; i < b.length; ++i) {

**for** (**int** j = 0; j < a[i].length; ++j) {

System.***out***.print(" " +b[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("Ergebnismatrix c:");

**int** c[][] = *addMatrix*(a, b);

**for**(**int** i=0; i<a.length; i++){

**for**(**int** j=0; j<a[0].length; j++)

System.***out***.print(" " + c[i][j] + " ");

System.***out***.println();

}

}

**public** **static** **int**[][] addMatrix(**int**[][] a, **int**[][] b) {

**int**[][] c = **new** **int**[a.length][a[0].length];

**for**(**int** i=0; i<a.length; i++)

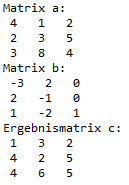
**for**(**int** j=0; j<a.length; j++)

c[i][j] = a[i][j] + b[i][j];

**return** c;

}

}



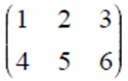
### Transponierte einer Matrix

**Aufgabe:**

Vertauscht man in einer (m,n)-Matrix A die Zeilen mit den Spalten, so erhält man die transponierte Matrix AT vom Format (n,m).

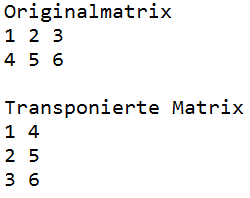
Man schreibt also die erste Zeile als erste Spalte und die zweite Zeile als zweite Spalte usw. Die Matrix wird sozusagen an ihrer Hauptdiagonale gespiegelt.

Gegeben sei die folgende Matrix A:



Ergänzen Sie das Programm „Matrizentransposition“, so dass neben der Originalmatrix A auch die transponierte Matrix AT ausgegeben wird.

**Ausgabe/Darstellung:**



**Programmcode/Teilnehmer:**

**public** **class** Matrizentransposition {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 1, 2, 3 }, { 4, 5, 6 } };

**int** at[][] = **new** **int**[a[0].length][a.length];

System.***out***.println("Originalmatrix");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[i].length; ++j) {

System.***out***.print(a[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println();

System.***out***.println("Transponierte Matrix");

...

**Programmcode (ohne Verwendung einer Methode):**

**public** **class** Matrizentransposition {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 1, 2, 3 }, { 4, 5, 6 } };

**int** at[][] = **new** **int**[a[0].length][a.length];

System.***out***.println("Originalmatrix");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[0].length; ++j) {

System.***out***.print(a[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println();

System.***out***.println("Transponierte Matrix");

**for** (**int** i = 0; i < a[0].length; i++) {

**for** (**int** j = 0; j < a.length; j++) {

at[i][j] = a[j][i];

System.***out***.print(at[i][j] + " ");

}

System.***out***.println();

}

}

}

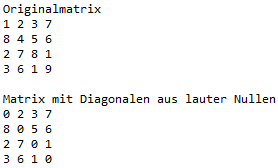
### Nulldiagonale

**Aufgabe:**

Gegeben sei eine Matrix A. Ihre Aufgabe besteht darin, die Matrix auszugeben. Danach soll die gleiche Matrix noch einmal ausgegeben werden, wobei diesmal die Hauptdiagonale aus lauter Nullen bestehen soll.

Ergänzen Sie das Programm „NullDiagonale“, so dass neben der Originalmatrix A auch die Matrix mit der Diagonalen aus lauter Nullen ausgegeben wird.

**Ausgabe/Darstellung:**



**Programmcode/Teilnehmer:**

**public** **class** NullDiagonale {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 1, 2, 3, 7 }, { 8, 4, 5, 6 }, { 2, 7, 8, 1 }, { 3, 6, 1, 9 } };

**int** at[][] = **new** **int**[a[0].length][a.length];

System.***out***.println("Originalmatrix");

**Programmcode/Lösung:**

**public** **class** NullDiagonale {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 1, 2, 3, 7 }, { 8, 4, 5, 6 }, { 2, 7, 8, 1 }, { 3, 6, 1, 9 } };

**int** at[][] = **new** **int**[a[0].length][a.length];

System.***out***.println("Originalmatrix");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[0].length; ++j) {

System.***out***.print(a[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println();

System.***out***.println("Matrix mit Diagonalen aus lauter Nullen");

**for** (**int** i = 0; i < a.length; ++i) {

**for** (**int** j = 0; j < a[0].length; ++j) {

**if** (i == j) {

a[i][j] = 0;

}

System.***out***.print(a[i][j] + " ");

}

System.***out***.println();

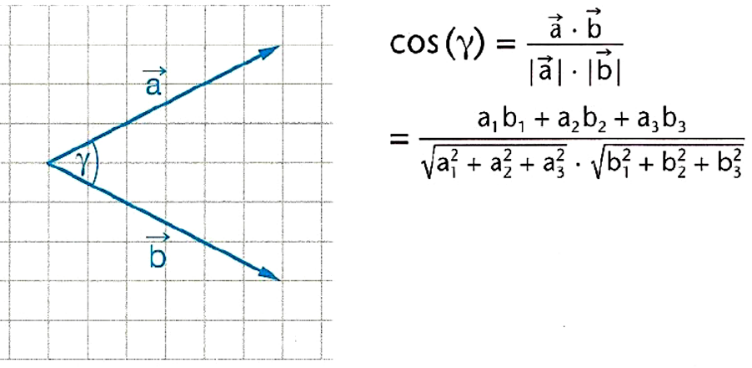
}

}

}

### Skalarprodukt

Das Skalarprodukt (auch inneres Produkt) ist eine mathematische Verknüpfung, die zwei Vektoren eine Zahl (Skalar) ohne Maßeinheit zuordnet. Die Zahl ist im Prinzip ein Maß für den Winkel zwischen den beiden Vektoren.

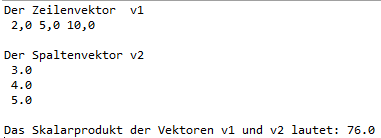


In der Geometrie hat das Skalar eine zentrale Bedeutung, denn es stellt die Grundlage für alle räumlichen Winkelberechnungen dar. Zudem wird es verwendet, um zu prüfen, ob zwei Vektoren senkrecht aufeinander stehen. Dies ist der Fall, wenn ihr Skalarprodukt gleich null ist.

Es ist aus Sicht der Matrizen nichts anderes als die Multiplikation eines Zeilenvektorsa v1 mit dem Spaltenvektor v2.

**Ziel:**

In der Abbildung unten sie die beiden Vektoren v1 und v2 (beide vom Typ „doubke“) angegeben. Schreiben Sie ein Programm „SkalarProdukt“ mit dem Sie mitels einer Methode „skalarproduktBerechnung das Skalarprodukt der beiden Vektoren ausrechnen und ausgeben können.



**Programmcode:**

**public** **class** SkalarProdukt {

**public** **static** **void** main(String[] args) {

**double**[] v1 = { 2, 5, 10 };

**double**[] v2 = { 3, 4, 5 };

// Ausgabe des Zeilenvektors v1

System.***out***.println("Der Zeilenvektor v1");

**for** (**int** i = 0; i < v1.length; i++) {

System.***out***.printf(" %.1f", v1[i]);

}

// Ausgabe des Spaltenvektors v2

System.***out***.println("\n\nDer Spaltenvektor v2");

**for** (**int** i = 0; i < v2.length; i++) {

System.***out***.println(" " + v2[i]);

}

System.***out***.println();

**double** ergebnis = *skalarproduktBerechnung*(v1, v2);

System.***out***.println("Das Skalarprodukt der Vektoren v1 und v2 lautet: " + ergebnis);

}

**public** **static** **double** skalarproduktBerechnung(**double**[] v1, **double**[] v2) {

**double** skalarprodukt = 0;

**for** (**int** i = 0; i < v1.length; i++) {

skalarprodukt = skalarprodukt + v1[i] \* v2[i];

}

**return** skalarprodukt;

}

}

### Matrizenmultiplikation

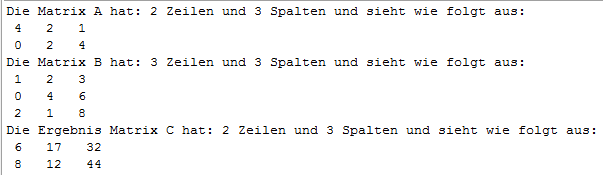
**Aufgabe:**

Gegeben seien die beiden Matrizen A und B.

A =  B = 

Ergänzen Sie das Programm "Matrizenmultiplikation" in Ihrem Workspace, so dass es beide Matrizen und das Produkt von A\*B ausgibt.

**Ausgabe/Darstellung:**

****

**Programmcode/Teilnehmer:**

**public** **class** Matrizenmultiplikation {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 4, 2, 1 }, { 0, 2, 4 } };

**int** b[][] = { { 1, 2, 3 }, { 0, 4, 6 }, { 2, 1, 8 } };

**int**[][] c = **new** **int**[a.length][b[0].length];

// Ausgabe der Matrix A

}

// Ausgabe der Matrix B

// Kern der Matrizenmultiplikation

// Ausgabe der Ergebnismatrix

}

}

}

**Programmcode:**

**public** **class** Matrizenmultiplikation {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 4, 2, 1 }, { 0, 2, 4 } };

**int** b[][] = { { 1, 2, 3 }, { 0, 4, 6 }, { 2, 1, 8 } };

**int**[][] c = **new** **int**[a.length][b[0].length];

// Ausgabe der Matrix A

System.***out***.print("Die Matrix A hat: " + a.length + " Zeilen");

System.***out***.println(" und " + a[0].length + " Spalten und sieht wie folgt aus:");

**for** (**int** u = 0; u < a.length; u++) {

**for** (**int** v = 0; v < a[0].length; v++)

System.***out***.print(" " + a[u][v] + " ");

System.***out***.println();

}

// Ausgabe der Matrix B

System.***out***.print("Die Matrix B hat: " + b.length + " Zeilen");

System.***out***.println(" und " + b[0].length + " Spalten und sieht wie folgt aus:");

**for** (**int** u = 0; u < b.length; u++) {

**for** (**int** v = 0; v < b[0].length; v++)

System.***out***.print(" " + b[u][v] + " ");

System.***out***.println();

}

// Kern der Matrizenmultiplikation

**for** (**int** i = 0; i < a.length; i++)

**for** (**int** j = 0; j < b[0].length; j++)

**for** (**int** k = 0; k < a[0].length; k++)

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

System.***out***.print("Die Ergebnis Matrix C hat: " + a.length + " Zeilen");

System.***out***.println(" und " + b[0].length + " Spalten und sieht wie folgt aus:");

// Ausgabe der Ergebnismatrix

**for** (**int** i = 0; i < a.length; i++) {

**for** (**int** j = 0; j < b[0].length; j++)

System.***out***.print(" " + c[i][j] + " ");

System.***out***.println();

}

}

}

**Aufgabe:**

Gegeben seien die Rohstoff-Zwischenprodukt-Matrix A sowie die Zwischenprodukt-Endprodukt-Matrix B.

Matrix A:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Z1 | Z2 | Z3 |
| R1 | 3 | 1 | 4 |
| R2 | 2 | 5 | 1 |

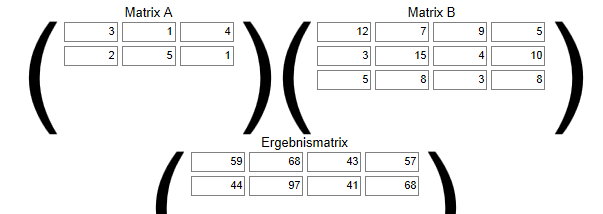
**Matrix B:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | E1 | E2 | E3 | E4 |
| Z1 | 12 | 7 | 9 | 5 |
| Z2 | 3 | 15 | 4 | 10 |
| Z3 | 5 | 8 | 3 | 8 |

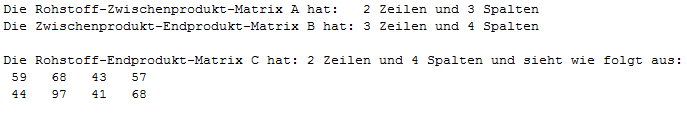
Sie sollen ermitteln, wie viele Rohstoffe in die Endprodukte eingehen. Beantworten Sie vorab die Fragen:

* Können beide Matrizen überhaupt miteinander multipliziert werden?
* Falls ja, welches Aussehen hat die Ergebnismatrix.

Multiplizieren Sie die Matrizen.

****

**Die Ausgabe soll wie folgt aussehen:**

****

**Programmcode (ohne Verwendung einer Methode):**

**public** **class** Matrizen06a {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 3, 1, 4 }, { 2, 5, 1 } };

**int** b[][] = { { 12, 7, 9, 5 }, { 3, 15, 4, 10 }, { 5, 8, 3, 8 } };

System.***out***.print("Die Rohstoff-Zwischenprodukt-Matrix A hat: " + a.length + " Zeilen");

System.***out***.println(" und " + a[0].length + " Spalten");

System.***out***.print("Die Zwischenprodukt-Endprodukt-Matrix B hat: " + b.length + " Zeilen");

System.***out***.println(" und " + b[0].length + " Spalten\n");

// Bestimmen der Dimensionen der Ergebnismatrix

**int**[][] c = **new** **int**[a.length][b[0].length];

// Kern der Matrizenmultiplikation

**for** (**int** i = 0; i < a.length; i++)

**for** (**int** j = 0; j < b[0].length; j++)

**for** (**int** k = 0; k < a[0].length; k++)

c[i][j] += a[i][k] \* b[k][j];

System.***out***.print("Die Rohstoff-Endprodukt-Matrix C hat: " + a.length + " Zeilen");

System.***out***.println(" und " + b[0].length + " Spalten und sieht wie folgt aus:");

// Ausgabe der Ergebnismatrix

**for** (**int** i = 0; i < a.length; i++) {

**for** (**int** j = 0; j < b[0].length; j++)

System.***out***.print(" " + c[i][j] + " ");

System.***out***.println();

}

}

}

**Programmcode (mit Verwendung einer Methode):**

**public** **class** Matrizen06 {

**public** **static** **void** main(String[] args) {

**int** a[][] = { { 3, 1, 4 }, { 2, 5, 1 } };

**int** b[][] = { { 12, 7, 9, 5 }, { 3, 15, 4, 10 }, { 5, 8, 3, 8 } };

System.***out***.print("Die Rohstoff-Zwischenprodukt-Matrix A hat: "

+ a.length + " Zeilen");

System.***out***.println(" und " + a[0].length + " Spalten");

System.***out***.print("Die Zwischenprodukt-Endprodukt-Matrix B hat: " + b.length

+ " Zeilen");

System.***out***.println(" und " + b[0].length + " Spalten\n");

System.***out***.print("Die Rohstoff-Endprodukt-Matrix C hat: " + a.length

+ " Zeilen");

System.***out***.println(" und " + b[0].length

+ " Spalten und sieht wie folgt aus:");

**int** c[][] = *multiMatrix*(a, b);

**for** (**int** i = 0; i < a.length; i++) {

**for** (**int** j = 0; j < b[0].length; j++)

System.***out***.print(" " + c[i][j] + " ");

System.***out***.println();

}

}

**public** **static** **int**[][] multiMatrix(**int**[][] a, **int**[][] b) {

**int**[][] c = **new** **int**[a.length][b[0].length];

**for** (**int** i = 0; i < a.length; i++)

**for** (**int** j = 0; j < b[0].length; j++)

**for** (**int** k = 0; k < a[0].length; k++)

c[i][j] += a[i][k] \* b[k][j];

**return** c;

}

}