

## Matriisitulo

Matriisitulo  $\mathbf{Z} = \mathbf{XY}$  lasketaan pistetulojen avulla.

1. Otetaan ensimmäisestä matriisista ensimmäinen rivi  $x_{1\cdot}$ .
2. Otetaan toisesta matriisista ensimmäinen sarake  $y_{\cdot 1}$ .
3. Lasketaan pistetulo  $x_{1\cdot} \cdot y_{\cdot 1} = c_{11}$  ja sijoitetaan se tulomatriisin soluun (1, 1).
4. Toistetaan kohdat 1–3 kaikille rivi–sarake-kombinaatioille.

Kaavana:

$$(1) \quad c_{jk} = \sum_{l=1}^n x_{jl} y_{lk},$$

jossa  $j = 1, \dots, m$ ,  $k = 1, \dots, p$ , kun  $m$  on  $\mathbf{X}$ -matriisin rivien määrä ja  $p$  on  $\mathbf{Y}$ -matriisin sarakkeiden määrä. Lisäksi täytyy päteä, että sekä  $\mathbf{X}$ :n sarakkeiden määrä että  $\mathbf{Y}$ :n rivien määrä on  $n$ . Kaavassa (1) summalauseke vastaa pistetuloa. Tulomatriisi  $\mathbf{Z}$ :ssa on  $m$  riviä ja  $p$  saraketta.

Toteutetaan Javalla. Oleellista on siis lasketa pistetulo kaikilla rivi–sarake-kombinaatioilla. Lisäksi on hyvä tarkistaa, että matriisien dimensiot ovat laillisia.

```
/** Janne Koljonen
Vaasan yliopisto
AUTO1030 */

public class Matriisitulo
{
    // Pääohjelmametodi: testaa matriisitulo-metodia
    public static void main(String args[]) throws Exception
    {
        // Kirjoittaa
        System.out.println("Matriisitulon testausta");

        // Sovitaan, että taulukon ensimmäinen indeksi kertoo rivin ja toinen sarakkeen.
        // Nyt matriisil sisältää kaksi riviä {1, 4, 3} ja {0, 5, 2}.
        // Matriisi2 sisältää kolme riviä.
        int[][] matriisil={{1, 4, 3}, {0, 5, 2}};
        int[][] matriisi2={{4, 2}, {6, 6}, {7, 0}};

        // Lasketaan ja tulostetaan matriisitulo.
        print(matriisil);
        System.out.println();
        print(matriisi2);
        System.out.println();
        System.out.println("m1*m2=");
        print(matriisitulo(matriisil, matriisi2));
    }

    // Laskee annetuista matriiseista = (kaksiulotteisista int[][] taulukoista) tulon
    public static int[][] matriisitulo(int[][] x1, int[][] x2) throws Exception
    {
        // Tarkistetaan dimensiot: ensimmäisen sarakkeiden määrä == toisen rivien määrä
        // (Tarkistetaan vain ensimmäisen rivin sarakkeiden määrä.
        // Javassa kussakin taulukossa voisi kyllä olla eripituinen taulukko eli
        // nyt kullakin rivillä voisi olla eri lukumäärä sarakkeita!)
        if(x1[0].length != x2.length)
        {
            throw new Exception("Vektorit olivat eri pituisia!");
        }
    }
}
```

```
// Alustetaan tulomatriisi
int[][] temp=new int[x1.length][x2[0].length];

// Lasketaan pistetulot
for(int sarake=0; sarake<x2[0].length; sarake++)
{
    // Nyt rivit on helppo erottaa matriisista, esim.: x1[0] on 1. rivi.
    // Sarakkeet pitää lukea soluiittain.
    int[] temp_sarake = new int[x2.length];
    for(int rivi=0; rivi<x2.length; rivi++)
    {
        temp_sarake[rivi]=x2[rivi][sarake];
    }
    // Sitten pistetulot tällä sarakeella
    for(int rivi=0; rivi<x1.length; rivi++)
    {
        temp[rivi][sarake]=Pistetulo.pistetulo(x1[rivi], temp_sarake);
    }
}
return temp;
}

// Tulostaa kaksiulotteisen taulukon
public static void print(int[][] matriisi)
{
    for(int rivi=0; rivi<matriisi.length; rivi++)
    {
        for(int sarake=0; sarake<matriisi[rivi].length; sarake++)
        {
            System.out.print(matriisi[rivi][sarake]+", ");
        }
        System.out.println();
    }
    System.out.println();
}
}
```