

# Bilgisayar Bilimlerinde Doğrusal Cebir

## Matris Ödev Raporu

Her şeyden önce projemin çalışma mantığı iki tane class üzerindendir; bir varsayılan class olan “main” diğeri ise benim oluşturduğum “Matrix” classıdır. “Matrix” classının içinde tüm metotler yer almaktadır. Görevleri açıklarken sadece o işlemleri yapan, “Matrix” classının içindeki metotleri açıklayacağım.

### Görev1: Matris Çarpımı

**Görev Tanımı:** İki tane matrisin çarpımı söz konusudur, ilk matrisin 1. satırının; ikinci matrisin de 1. sütununun elemanlarından başlayarak çarpılmasıdır. İlk matriste her satırın çarpımı bittiğinden ikinci matrisin 1. sütununa geri dönmelidir. Algoritma bu mantık üzerinden yazılmıştır.

**Kullanılan Yöntemler:** Bu algoritma çözümü için üç tane (i, j, k) iç içe “for” döngüleri kullanılmıştır.

**Kod Açıklaması:** i, j, k değişkenlerinin artış hareketlerine göre ilk matrisin ve ikinci matrisin elemanları ile işlemler yapılır. Mesela ikinci matrisin sütunları ile olan işlemler bitmeden ilk matrisin bir sonraki satırına geçilemez yani ilk matrisin satırı en son artması gerekecektir. Bu nedenler ilk matrisin satır numarası “i” değişkeniyle belirlenmelidir.

```
First Matrix!
Enter 00 elemen: 1
Enter 01 elemen: 2
Enter 02 elemen: 3
Enter 10 elemen: 4
Enter 11 elemen: 5
Enter 12 elemen: 6
Enter 20 elemen: 7
Enter 21 elemen: 8
Enter 22 elemen: 9

Second Matrix!
Enter 00 elemen: 1
Enter 01 elemen: 2
Enter 02 elemen: 3
Enter 10 elemen: 4
Enter 11 elemen: 5
Enter 12 elemen: 6
Enter 20 elemen: 7
Enter 21 elemen: 8
Enter 22 elemen: 9
```

Şekil 1. Matrisleri oluşturma

```
First Matrix:
1.00 2.00 3.00
4.00 5.00 6.00
7.00 8.00 9.00

Second Matrix:
1.00 2.00 3.00
4.00 5.00 6.00
7.00 8.00 9.00

Product of Matrixes:
30.00 36.00 42.00
66.00 81.00 96.00
102.00 126.00 150.00
```

Şekil 2. Matrisleri görüntüleme ve çarpım sonucunu çıkarma

**Sonuçlar:** Şekil 1.’deki gibi ilk önce iki matrisin de elemanları kullanıcı tarafından alınır. Ardından Şekil 2.’deki gibi önce iki matris de yazdırılır ardından çarpım işlemi yapıp çarpım sonucu ekrana yazılır.

## Görev 2: Determinant Hesaplama

**Görev Tanımı:** Bir matrisin determinantını almak için matris, kare matris olmalıdır. Ana işlemler matrisin ilk satır elemanları ile devam eder. İlk satır elemanlar tek tek seçilir ve seçilen elemanların bulunduğu satır ve sütun kaldırılıp kalan elemanlar üzerinde işlemler yapılır. (Tabi bu işlem 3x3 tipindeki bir matris içindir.) Kalan elemanlar çaprazlamasına şekilde çarpılıp birbirinden çıkarılır ve çıkan sonuç ilk satırdan seçilen eleman ile çarpılır. İlk satırdan seçilen eleman, satırın ikinci elemanı ise çıkan sonuç “-1” ile çarpılır. 2x2’lik matriste direkt olarak elemanlar çaprazlamasına çarpılıp birbirinden çıkarılır.

**Kullanılan Yöntemler:** Açık konuşmak gerekirse determinant için, çarpma işleminde yazdığım gibi, dinamik bir algoritma yazamadım. Kod çok basit ve ilkel tamamen “if” ve “for” blokları kullanılmıştır.

**Kod Açıklaması:** İlk önce kodda “if” bloğu ile matrisin tipini kontrol ediyoruz ve ona göre işlemler yapıyoruz. 3x3’lük için üç tane sonuç alan değişkenler oluşturuyoruz. Bunlar aslında matrisin ilk satırının elemanları için oluşturulmuş değişkenlerdir. Ardında ilk satırın elemanlarına ulaşabilmek için bir “for” döngüsü oluşturuyoruz, ardından gerekli “if” koşulları ve işlemler ile algoritmamızı yazıyoruz.

```
Enter 00 elemen: 1
Enter 01 elemen: 2
Enter 02 elemen: 3
Enter 10 elemen: 4
Enter 11 elemen: 5
Enter 12 elemen: 6
Enter 20 elemen: 7
Enter 21 elemen: 8
Enter 22 elemen: 9

First Matrix:
1.00 2.00 3.00
4.00 5.00 6.00
7.00 8.00 9.00

Determinant of Matrix: 0.0
```

Şekil 3. Matris elemanlarını kullanıcı tarafından alıp matrisi görüntülüyoruz ardından determinantını yazdırıyoruz

**Sonuçlar:** Şekil 3.'deki gibi ilk önce matrisin elemanları alınır ardından görüntülenir. Daha sonra matris determinant işleminden geçerek matrisin determinantı yazdırılır.

### Görev 3: Matris Tersini Bulma

**Görev Tanımı:** Her şeyden önce bir matrisin tersi olabilmesi için matrisin determinantı “0”dan farklı olması lazım. Ardından matrisin her bir elemanı için işlem uygulanır. Her matris seçildiğinde o matris ile birlikte matrisin bulunduğu sütun ve satırlar silinir ve kalan elemanlar arasında çaprazlama şekilde çarpım işlemi uygulanır ve birbirinden çıkarılır. Ve bu işlem matrisin son elemanına kadar bu şekilde devam eder. Sıra numarası çift olan elemanların seçiminde çıkan sonuç “-1” ile çarpılır ve nihai sonuç seçilen elemanın yerine yazılır. Ve bu sayede “Ek” matris oluşur. Yeni oluşan bu matrisin sütun ve satır elemanları birbirleriyle yer değiştirerek yani transpoz ederek “Adj” matrisi oluşur. En son olarak; oluşan son matrisimizin tüm elemanlarını  $1/determinant$  ile çarpılarak matrisin tersi bulunmuş olur.

**Kullanılan Yöntemler:** Açıkçası bu algoritma da pek dinamik değil. Sırf koşul blokları ve döngüler kullanılarak yazılmış bir algoritmadır.

**Kod Açıklaması:** İlk öncelikle kullanıcıdan 2x2 veya 3x3 tipinde matris alınması istendiği için parametrelili bir metot oluşturmam gerek idi bu yüzden metot içerisinde parametreyi kullanabilmek adına yeni bir matris tanımladım. Kodun daha okunaklı hâle gelebilmesi için “Ek” matris oluşturmada “switch-case” koşul bloğunu kullandım. Ardından transpoz işlemi için ek bir metot yazdım ve oluşan ek matrisimizi o transpoz() metotüne parametre olarak gönderdim. En sonda oluşan yeni matrisin tüm elemanlarını “for” döngüsü kullanarak  $1/determinant$  ile çarparak işlemi sonlandırdım.

```

Enter number of row and column number (2 or 3):
2
Enter 00 elemen: 1
Enter 01 elemen: 2
Enter 10 elemen: 7
Enter 11 elemen: 9

Matrix:
1.00 2.00
7.00 9.00

Cofactor of Matrix
9.00 -2.00
-7.00 1.00

Reversed Matrix:
-1.80 0.40
1.40 -0.20

```

Şekil 4. Matrisin tipini 2x2 seçtiğimizde çıkan görüntü

```

Enter number of row and column number (2 or 3):
3
Enter 00 elemen: 4
Enter 01 elemen: 7
Enter 02 elemen: 0
Enter 10 elemen: 2
Enter 11 elemen: 3
Enter 12 elemen: 5
Enter 20 elemen: 6
Enter 21 elemen: 9
Enter 22 elemen: 8

```

Şekil 5. Matris tipini 3x3 seçtikten sonra elemanların belirlenmesi

```

Cofactor of Matrix
-21.00 14.00 0.00
-56.00 32.00 6.00
35.00 -20.00 -2.00

Transpose of Matrix:
-21.00 -56.00 35.00
14.00 32.00 -20.00
0.00 6.00 -2.00

Reversed Matrix:
-1.50 -4.00 2.50
1.00 2.29 -1.43
0.00 0.43 -0.14

```

Şekil 6. 3x3 matris tipinin gerekli işlemler sonucu çıkan çıktılar

**Sonuçlar:** Şekil 4.’deki gibi ilk öncelikle matrisin tipini belirlememiz gerekiyor. 2x2 seçildiğinde transpoza gerek duymadan direkt kofaktörünü alıp tüm elemanlarını  $1/determinant$  ile çarpılır. Şekil 5.’deki gibi matrisin tipi 3x3 seçildiğinde tüm elemanlar kullanıcı tarafından belirlenir. Ardından Şekil 6.’daki gibi belirli işlemlerden geçtikten sonra sonuçlar çıktı olarak verilir.

#### Görev 4: Vektör Uzakında Doğrusal Bağımsızlık Kontrolü

**Görev Tanımı:** Burada yalnızca determinant işlemi ile kontrol edilen algoritmayı yazdım. Eğer determinant “0” ise vektör doğrusal bağımlı; değil ise olmuyordu.

**Kullanılan Yöntemler:** İlk önce 3 tane vektör oluşturup tek tek “for” döngüsü ile elemanları kullanıcıdan alıyoruz. Ardından determinant hesabı için yeni 3x3’lük bir matris oluşturup vektörlerin elemanlarını bu matrise atıyoruz. Ardından basit bir koşul bloğu ile önceden matris işlemleri için yazılan determinant alma işlemini kullanıp determinantın “0” olup olmadığını kontrol ediyoruz.

**Kod Açıklaması:** Java dilinin Vector kütüphanesi var, yapılan değer atamaları (vector.add(var);) vesaire bu kütüphanenin içinde olan hazır metotlar kullanılmıştır. Fakat şöyle bir sıkıntı ile karşılaştım; vektörün elemanları herhangi bir “primitive” veri tipinde değil “Vector” tipinde tanımlanıyordu bu da benim bu elemanlar ile işlem yapmamı kısıtlıyordu. Bunu çözmek içinde ((double) vector.get(indexNumber);) yazdım. Yani vector’ün o indeksinde bulunan elemanı “double”e döndür.

```
First Vector
Enter 0. elemen:
1
Enter 1. elemen:
2
Enter 2. elemen:
3

Second Vector
Enter 0. elemen:
4
Enter 1. elemen:
5
Enter 2. elemen:
6

Third Vector
Enter 0. elemen:
7
Enter 1. elemen:
8
Enter 2. elemen:
9

Matris determinantı '0' olduğundan bu vektörler doğrusal bağımlıdır.
```

Şekil 7. Üç vektöründe tek tek elemanlarının belirlenip sonucun ekrana yazdırılması

**Sonuçlar:** Şekil 7.’deki gibi üç vektörün de “for” döngüsü ile ve hazır metot olan “add();” metodu kullanılarak elemanlar belirlendi. Ardından gerekli işlemlerden geçerek sonuçlar ekrana yansdı.

