

Metot Nedir ? Ne için kullanılır ?

Enre Atunbilek Java Dersleri

Metotlar (yordamlar) sayesinde bir kod bloğunun belli bir işi yapması için bir araya gelmesiyle oluşur.

Metotları kullanarak büyük problemleri küçük parçalara ayırabilir, gereksiz kod kalabalığından kurtulabiliriz.

Örneğin size 1 den 10'a kadar olan sayıların toplamını, 30 dan 60 a kadar olan sayıların toplamını sırasıyla yazdıran deseler ne yaparsınız?

Eğer metotları bilmiyorsanız bu problem için iki farklı for döngüsü oluşturup

İşlem yapmanız gerekir. Peki bu problem 2 değil de 10 farklı istek içerirse ne yapacaksınız? 10 tane for döngüsü oluşturmak ne kadar mantıklı ?

Yukarıdaki örnekte farkındaysanız yapılan iş aynıdır. Başlangıç ve bitiş değerleri bilinen bir sayı aralığındaki sayıları toplamak.

Burada metotlar yardımımıza koşuyor. Sadece tek bir metot tanımlayarak farklı parametreler kullanarak istediğiniz kadar işlem yapabilirsiniz.

Tek yapmanız gereken metodu tanımlamak ve gerektiği yerde uygun parametreler ile çağırıp kullanmak olacaktır.

Farklı metot türleri bulunmaktadır. Bu bölümde şunları göreceğiz:

Parametresiz Metotlar : Parametre almayan ve değer döndürmeyen metotlardır.

```
void selamla(){  
    System.out.println("Merhabalar, Nasılsın ?");  
}
```

Parametre Alan Metotlar : Girdi olarak bazı değerler alır ve bunlara bağlı olarak işlem yapar.

```
void topla(int s1, int s2){  
    System.out.println("Parametrelerin toplamı : "+ (s1+s2));  
}
```

Değer Döndüren Metotlar: Çağrıldıkları yere bazı değerleri döndüren metotlar.

```
int toplam = toplamiBul(3,5); //metoda 3 ve 5 parametrelerini  
geçiyoruz.
```

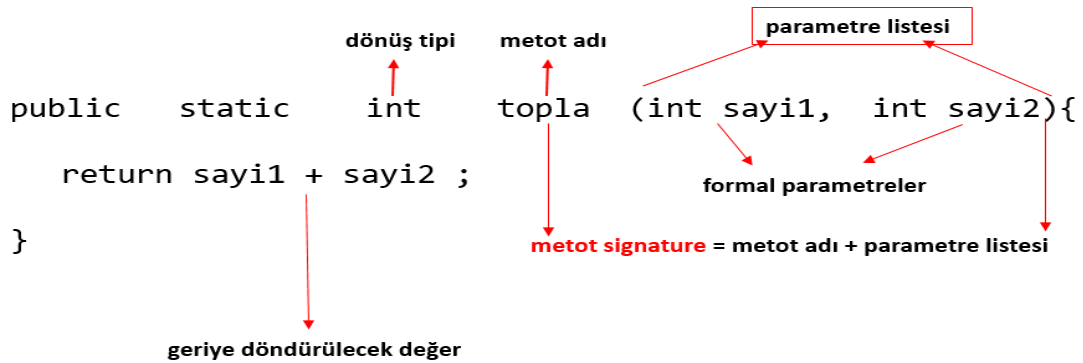
```
//s1 e 3, s2 ye 5 deęerleri atanır.
int toplamiBul(int s1, int s2){
    return s1 + s2;
    //bu deęerlerin toplamını return ile metotun çağrıldığı yere
    göndeririz.
    //bundan dolayı s1+s2 sonucu tanımlanan toplam deęişkenine
    atanır.
}
```

Daha fazla ayrıntıyı video derslerde bulabilirsiniz.

Metotlar Devam

Enre Atunbilek Java Dersleri

Metot Tanımlama



Bu metodu çağırırken adını ve de parametrelerini vermemiz yeterlidir.

```
int toplam = toplama(5,8); //buradaki 5 ve 8 actual parameter veya arguments
```

Eđer metota deęişken gönderirsek bu deęişkenlerin deęerleri metota gönderilir.

```
int toplam = toplama(sayi1, sayi2); //metota sayi1 ve sayi2nin sakladıkları deęerler
gönderilir.
```

Yani metot içerisinde `sayi1` ve `sayi2`nin deęerlerini deęiştirsek bile bunlar asıl olan `sayi1` ve `sayi2` deęişkenlerine yansımayacaktır. (pass by value deęer yollama)

Geriye deęer döndürmeyen metotların dönüş tipi **void** olur.

Aşağıdaki kodu çalıştırmayı deneyiniz. Neden hata verdiğini yorumlayınız.

```
public static int isaret(int n){
    if(n > 0){
        return 1;
    }else if(n==0){
        return 0;
    }else if(n < 0){
        return -1;
    }
}
```

Metot Overloading :

Metotlar aşırı yüklenerek yani overload edilerek aynı isimli methodlardan birden fazla tanımlanabilir.

Burada önemli olan iki metotun birebir aynı isimde olmaları fakat signature değerlerinin yani parametre sayısının veya parametre veri tiplerinin farklı olmasıdır. Aynı isim altındaki methodları overload ettiğimiz de birbirinden farklı olan durumları parametrelerin sayısı ve bu parametrelerin türleridir.

Bir metotun geri dönüş tipi imza yani signature kavramına dahil değildir.

```
int sayilariTopla(int s1, int s2){
    return s1 + s2;
}

double sayilariTopla(double s1, double s2){
    return s1 + s2;
}
```

Dizi Kavramı

Enre Altunbilek Java Dersleri

Dizi Array Nedir ?

İçerisinde belli sayıda ve aynı türde veri olan yapılara dizi denir. Siz eğer 10 tane integer değeri bir arada tutmak istiyorsanız 10 farklı değişken tanımlamak yerine bir dizi oluşturup kullanabilirsiniz.

Dizilerde primitive veya referans tip veri türlerini barındırabilir. Tek şart bir dizi aynı türde elemanları içermelidir.

Dizi türündeki referanslar dizi nesnelere bağlanır. (Nesne ?). Dizi referans değerleri stack yapısında, dizi nesneleri ise heap alanında tutulur.

Aşağıdaki dizi tanımlamalarının hepsi doğrudur.

```
double[] a = new double[10] --> 10 tane double değer tutacak dizi
int []b = new int[10] --> 10 tane integer değer tutacak dizi
float c[] = new float[10] --> 10 tane double değer tutacak dizi
```

Bu diziler oluşturulduğunda heap alanında 10 adet değer saklayacak bir alan oluşturulur ve bu alana da veri tiplerinin varsayılan değerleri atanır.

Bir diziyi tanımladıktan sonra veri atamak için

```
int []b = new int[10];
```

`b[0] = 5` --> dizinin 0. indeksine 5 değeri atandı

Dizide index değerleri 0'dan başlar. 0 yüzden maksimum index değeri dizinin eleman sayısının 1 eksiği kadardır. Bu sınırı aştığınızda `ArrayIndexOutOfBoundsException` hatası alırsınız.

Ayrıca `double[]` veriler = {3,4, 5,1, 2,3} şeklinde de diziyi oluşturabilirsiniz.

```
double[] myList;
```

```
myList = {1.9, 2.9, 3.4, 3.5}; //bu ifade hata verir
```

Dizideki elemanları gezmek ve kullanmak için genellikle for döngüsü kullanılır

10 elemanlı integer bir dizideki tüm elemanları yazdırırsak;

```
for(int i=0; i<dizi.length ; i++){  
    System.out.println(i+". indeksteki eleman:" + dizi[i]);  
}
```

Dizi bir kere tanımlanır, sonrasında dizideki elemanları değiştirebilirsiniz ancak tutacağı eleman sayısını değiştiremezsiniz.

`int sayilar[] = new int[10];`--> 10 elemanlı integer bir dizi oluşturulur.

`sayilar = new int[15];` --> eski sayilar dizisi silinip yenisi oluşturulur. Burada dizinin genişleyebildiğini düşünmeyin, tamamen yeni bir dizi oluştu...

Diziler hakkında daha fazla ayrıntıya geçmeden önce gelin isterseniz verilerin bellekte nasıl tutulduğunu inceleyelim.

Veriler Nasıl Tutulur ?

Enre Atunbilek Java Dersleri

Tanımladığımız değişkenler, nesneler, diziler aşağıdaki yapılarda tutulur.

Yığın (Stack) : Bellekte yani Ram'de bulunur. Oldukça hızlı çalışan, doğrudan işlemciden donanım desteğine sahip olan alandır. Java derleyicisi program oluşturulmadan önce buraya konulacak verileri ve büyüklüklerini bilmelidir.

İlkel veri tipleri o yüzden burada tutulur. Çünkü hepsinin bellekte tutacağı alan sabit ve bellidir. Ama dizi veya nesnelerin ne kadar yer tutacağı belli olmadığı için burada sadece referansları yer alır. Bu referansların değerleri ise heap alanında bulunan gerçek değerlerin adres bilgisidir.

Heap : Derleyici buradaki değerlerin ne kadar bellek ihtiyacı olduğunu bilmek zorunda değildir. Biz ne zaman new deyip yeni bir dizi veya nesne oluşturmak istersek bu alanda bize yer verilir. Diziler veya nesnelerin gerçek değerleri burada saklanır. Bu özgürlüklerin bize maliyeti ise yığına göre daha yavaş olmasıdır.

Statik Alan: İlerleyen derslerden sonra anlatılacak.

Sabit disk: İlerleyen derslerden sonra anlatılacak.

Dizi Örnek Sorular

Emre Atunbilek Java Dersleri

Soru 1

10 elemanlı bir diziye eleman olarak dizi indeksinin karesini atayan uygulama

Soru 2

Kullanıcıdan hangi ayda olduğunu bilgisini alıp ilgili ay değerini yazdırın. Ay adlarını bir dizide saklayın.

Soru 3

Kullanıcıdan kaç tane sayının ortalamasını bulmasını istediğini sorun ve buna bağlı olarak aldığınız değerleri bir dizide saklayın. Veri alma işlemi bittikten sonra tüm sayıları ve ortalamasını ekrana yazdırın.

Dizi Örnek Soru Çözümleri

Emre Atunbilek Java Dersleri

```
public class DiziArrayKavrami {  
  
    public static void main(String[] args) {  
        //ÇÖZÜM 1  
        int sayilar[]=new int[10];  
        for(int index=0; index < sayilar.length ; index++){  
            //sayilar[index] = index * index;  
            sayilar[index] = (int) Math.pow(index, 2);  
        }  
  
        for(int index = 0; index < sayilar.length; index++){
```

```

        System.out.println("Eleman : " + sayilar[index]);
    }

    //ÇÖZÜM2
    //String[] aylar=new String[12];
    String[] aylar =
{"Ocak","Şubat","Mart","Nisan","Mayıs","Haziran","Temmuz","Ağustos","Eylül","Ekim","Kasım","Aralık"};
    Scanner tara=new Scanner(System.in);
    System.out.println("Ay değerini rakamla yazınız");
    int ay = tara.nextInt();

    System.out.println("Seçtiğiniz ay : " + aylar[ay-1]);

    //ÇÖZÜM 3
    System.out.println("Kaç tane sayının ortalamasını bulmak istiyorsun ?");
    int sayi = tara.nextInt();

    int kullanıcıSayilari[] = new int[sayi];

    double ortalama = 0;
    for (int i=0; i<sayi; i++){
        System.out.println("Sayı giriniz : \n");
        kullanıcıSayilari[i] = tara.nextInt();
        ortalama = ortalama + kullanıcıSayilari[i];
    }

    System.out.println("Girdiğiniz sayıların ortalaması : " + (ortalama/kullanıcıSayilari.length));
}
}

```

Dizi Kavramı Devam

Emre Atunbilek Java Dersleri

Foreach Döngüsü

Daha önce dizi elemanlarını for döngüsü ile gezmiştik. Dizi elemanlarını okumak için çok daha kullanışlı ve foreach isimli bir yapı da vardır. Bu yapıyı eğer index ile ilginiz yoksa ve sadece dizi elemanlarını göstermek istiyorsanız kullanabilirsiniz.

```

for(int oankiSayi : diziAdi){
    System.out.println(oankiSayi);
}

```

Dizilerin Metotlara Gönderilmesi

Metotlarımıza primitive verileri parametre olarak geçebildiğimiz gibi dizileri de parametre olarak geçebiliriz.

Burda dikkat edilmesi gereken metotlara dizideki değerlerin kopyası değil referansları gider. O yüzden metot içinde yapılacak değişiklikler dizinin kendisinde yapılmış sayılır. Primitive tiplerde ise metotlara sadece değer kopyası gönderiliyordu.

Metotlar geriye dizi de döndürebilir. Aşağıdaki metot parametre olarak aldığı bir dizinin elemanlarını ters yönde içeren bir dizi oluşturur.

```
public static int[] tersiniAl(int[] dizi) {
    int[] sonuc= new int[dizi.length];

    for (int i = 0, j = sonuc.length - 1; i < dizi.length; i++, j--) {
        sonuc[j] = dizi[i];
    }

    return sonuc;
}
```

Dizilerin Kopyalanması

Bazen bir dizinin kopyasının alınması gerekebilir. Burda yapmanız gereken her elemanı tek tek okuyup yeni bir diziye atamaktır.

İsterseniz System sınıfının arraycopy() metoduyla da bu işlemi yapabilirsiniz.

```
arraycopy(kaynakDizi, hangiElemandanBaslanacak, hedefDizi,
hangiElemandanBaslanacak,kacElemanKopyalanacak);
Bu işlem için hedefDizi önceden oluşturulmalıdır.
```

```
int[] sayilar = new int[10];
int[] kopyaSayilar = new int[10];
```

```
kopyaSayilar = sayilar; // Bu ifade dizileri kopyalar mı ?
```

İki Boyutlu Diziler

Enre Atunbilek Java Dersleri

Dizileri tanımlarken aynı türde veri tutan yapılar demiştik. İki boyutlu dizi kavramı da aslında bir dizi tutan dizi olarak düşünülebilir.

Matematikteki matris sistemini İki boyutlu diziler kullanarak oluşturabiliriz.

Ayrıca bir tablodaki verileri belirtmek için de iki boyutlu diziler kullanılabilir.

BorçTablosu	Emre	Hasan	Ali
Emre	0	100	200
Hasan	350	0	250
Ali	150	500	0

For mül:

veritipi[][] diziAdi = new veritipi[satir_sayisi][sutun_sayisi]
 int[][] sayilar = new int[2][3] ifadesi 2 satırlı 3 sütunlu bir matris demektir

int[][] borclar = { {0,100,200}, {350,0,250}, {150,500,0}}
 Hasanın aliye borcu için borclar[1,2] demek yeterli

İki boyutlu bir dizide her bir satır aslında başka bir diziyi tutan dizidir.

borclar[0].length ==> 3 elemanlı bir dizidir

0 yüzden her bir satır farklı elemanlı diziler de tutabilir.

```
int[][] degisikElemanSayiliDizi = new int[5][];
degisikElemanSayiliDizi[0] = new int[5];
degisikElemanSayiliDizi[1] = new int[4];
degisikElemanSayiliDizi[2] = new int[3];
degisikElemanSayiliDizi[3] = new int[2];
degisikElemanSayiliDizi[4] = new int[1];
```

Bu dizilerin elemanlarını gezerken iç içe for döngüleri kullanılabilir.

```
for (int satir = 0; satir < dizi.length; satir++) {
    for (int sutun= 0; sutun< dizi[satir].length; sutun++) {
        System.out.println("Eleman:" + dizi[satir][sutun]);
    }
}
```

dizi.length ==> satır sayısını dizi[0].length ==>0. satırdaki sütun sayısı

Dizilerle ilgili işlemler

Emre Atunbilek Java Dersleri

Dizi elemanlarının yer değiştirilmesi

Dizi elemanlarının karılması, yer değiştirilmesi için aşağıdaki kodu kullanabilirsiniz.

```
for(int i = dizi.lenght -1; i>0 ; i--){  
  
    int j = (int)(Math.random() * (i+1));  
    double gecici = dizi[i];  
    dizi[i] = dizi[j];  
    dizi[j] = gecici;  
}
```

Dizi de Eleman Arama Binary Search

Dizide istediğimiz bir elemanın olup olmadığını linear olarak arayabiliriz.

Bu işlemde sırası ile aranılan elemanın dizide olup olmadığını kontrol ederiz.

Küçük boyutlu dizilerde kolayca yapılırken, eleman sayısı çok olan dizilerde verimsizdir. 5000 elemanlı bir dizide olmayan bir eleman için 5000 kere kontrol yapmak gerekir.

Binary search ile daha etkin aramalar yapılabilir. Ama binary search yapmak için dizi elemanlarının sıralı olarak dizide yer alması gerekir.

Dizi Elemanların Sıralama

Bazen elemanlarımızı sıralı olarak saklamak isteriz, bazen de bazı işlemler için bu zorunlu olur, örneğin binary search.

Farklı sıralama algoritmaları bulunur. Aşağıda selectionsort algoritması tanımlanmıştır.

```
public static void selectionSortSiralama(int[] dizi){  
  
    for(int i=0; i<dizi.length - 1 ; i++){  
        //en küçük elemanın ve indexinin bulunması  
        int oankiEnKucukSayi = dizi[i];  
        int oankiEnKucukElemaninIndex = i ;  
  
        for(int j = i+1 ; j < dizi.length ; j++){  
            if(oankiEnKucukSayi > dizi[j]){  
                oankiEnKucukSayi = dizi[j];  
                oankiEnKucukElemaninIndex = j;  
            }  
        }  
        //eğer gerekli ise değerler yer değiştirir  
        if(oankiEnKucukElemaninIndex != i){  
            dizi[oankiEnKucukElemaninIndex] = dizi[i];  
            dizi[i] = oankiEnKucukSayi;  
        }  
    }  
}
```

```
}
```

Arrays Sınıfı Kullanımı

Enre Atunbilek Java Dersleri

Bazı önemli kullanımlar

`Arrays.sort(dizi);` // tüm diziyi sıralar
`Arrays.parallelSort(dizi)` // tüm diziyi daha etkili bir şekilde sıralar. Eğer bilgisayarınızda birden fazla işlemci varsa tercih edilebilir.

`Arrays.binarySearch(dizi, aranilacakEleman);` eleman bulunursa indexini döndürür. Eleman yoksa negatif bir değer döndürür.

`Arrays.equals(dizi1, dizi2);` //elemanları aynı ise true değilse false

`Arrays.fill(dizi1, 14);` //tüm elemanlarına 14 yazar.
`Arrays.fill(dizi1, 0, 2, 10)` // 0,1 ve 2. indexe 10 sayısını yazar.

Elemanların yer değiştirilmesi `Collections.shuffle()` ile yapılacaktır. `Collections` bölümünde anlatılacak.

Bölüm Sonu Soruları

Enre Atunbilek Java Dersleri

Soru 1

100 elemanlı bir dizi oluşturun ve bu dizinin tüm elemanlarını rastgele olacak şekilde üretin. Bu işlemten sonra dizideki elemanların ortalamasını hesaplayarak bu ortalamanın altında kaç tane eleman olduğunu yazdırın.

Soru 2

Her bir elemanı rastgele oluşturulmuş 3*2'lik bir matriste bulunan sayıların toplamını yazdıran program yazınız. Ayrıca bu matriste yer alan en küçük ve en büyük sayıyı da mesaj olarak gösterebilirsiniz.

Soru 3

Iskambil destesini diziler ile olusturun. Sonrasında bu destedeki elemanları karıştırın ve de ilk kullanıcıya hangi 4 kartın geldiğini ekrana yazdırın.

Soru 4

Verilen iki sayının en küçük ortak katını bulan uygulamayı yazınız. $\text{Ekok}(s1,s2) * \text{ebob}(s1,s2) = (s1*s2)$; buna uygun olacak şekilde bulunuz

Soru 5

Kullanıcının girdiği öğrenci sayısı ve kullanıcının girdiği toplam soru sayısına bağlı olarak, öğrencilerin soruya verdiği cevapları tutan bir matris düzenleyin. Bu değerleri rastgele olarak olusturun. Değerler A,B,C,D,E olmalı.

Bir de doğru cevapları tutan tek boyutlu bir diziniz olsun. Bu da rastgele oluşturulmalı.

Sonrasında bu değerleri karşılaştırarak her bir öğrencinin kaç doğru ve yanlış olduğunu gösteren programı yazınız.

Çözüm 1

Emre Atunbilek Java Dersleri

Çözüm 1

```
public class Cozum1 {

    public static void main(String[] args) {
        int sayilar[] = new int[100];
        int tumSayilarinToplami = 0;

        for(int i=0; i<sayilar.length; i++){
            int rastgeleSayi = (int)(Math.random() * 100);
            sayilar[i] = rastgeleSayi;
            tumSayilarinToplami += rastgeleSayi;
            // tumSayilarinToplami = tumSayilarinToplami +
            rastgeleSayi;
        }
    }
}
```

```

        double ortalama = (double) tumSayilarinToplami /
(double)sayilar.length;

        int ortalamaAltindakiDegerlerinSayisi = 0;

        for (int i=0 ; i<sayilar.length; i++){
            if(sayilar[i] < ortalama){
                ortalamaAltindakiDegerlerinSayisi++;
            }
        }
        System.out.println("Ortalama : "+ortalama + " ve ortalamanın
altındaki değerlerin sayısı : "+ortalamaAltindakiDegerlerinSayisi);
    }
}

```

Çözüm 2

Emre Atunbilek Java Dersleri

```

public class Cozum2 {

    //Her bir elemanı rastgele oluşturulmuş 3*2lik bir matriste bulunan sayıların
    toplamını yazdıran
    // program yazınız.Ayrıca bu matriste yer alan en küçük ve en büyük sayıyı da
    mesaj olarak gösterebilirsiniz.
    public static void main(String[] args) {
        int[][] matris = new int[3][2];
        int enKucukSayi=900, enBuyukSayi=0;

        for(int satir=0; satir < matris.length; satir++){
            for(int sutun=0; sutun < matris[satir].length; sutun++){
                int rastgeleSayi = (int)(Math.random() * 100);
                matris[satir][sutun] = rastgeleSayi;
                System.out.println(satir+" . satırdaki " + sutun + " . sutunun
                elemanı : " + matris[satir][sutun]);

                if(enKucukSayi > rastgeleSayi){
                    enKucukSayi = rastgeleSayi;
                }
                if(enBuyukSayi < rastgeleSayi){
                    enBuyukSayi=rastgeleSayi;
                }
            }
        }

        System.out.println("Bu matristeki en küçük sayı : "+enKucukSayi+" en büyük
        sayı: "+enBuyukSayi);
    }
}

```

Çözüm 3

Emre Atunbilek Java Dersleri

İçerik

```
public class Cozum3 {

    public static void main(String[] args) {

        String kartTurleri[] = {"Kupa","Maça","Karo","Sinek"};
        String kartNumaralari[] = {"As" , "2", "3", "4", "5", "6", "7", "8","9" ,"10", "J","Q","K"};

        int deste[]=new int[52];

        desteyiOlustur(deste);

        desteyiKaristir(deste);

        desteyiGoster(deste, kartTurleri, kartNumaralari);

    }

    private static void desteyiKaristir(int[] deste) {

        for(int i=0; i<deste.length; i++){
            int rastgeleIndex = (int)(Math.random() * deste.length);
            int gecici = deste[i];

            deste[i] = deste[rastgeleIndex];
            deste[rastgeleIndex] = gecici;
        }

    }

    private static void desteyiGoster(int[] deste, String[] kartTurleri, String[] kartNumaralari) {

        for(int i=0; i<4 ; i++){

            String kartTuru = kartTurleri[deste[i] / 13];
            String kartNumarasi = kartNumaralari[deste[i] % 13];

            System.out.println(kartTuru + " " + kartNumarasi);

        }

    }

    private static void desteyiOlustur(int[] deste) {

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

            deste[i] = i ;

        }

    }

}
```

Çözüm 4

Emre Atunbilek Java Dersleri

public class Cozum4 {

```
    public static void main(String[] args) {
        int birinciSayi = 40 , ikinciSayi = 21;
```

```

        //int ekok = klasikEkokBul(birinciSayi, ikinciSayi);

        //System.out.println(birinciSayi+" ile "+ikinciSayi+" ekoku :"+ekok);

        //int ebob = ebobBul(15, 45);
        //System.out.println("ebob : "+ebob);

        int ekok = ekokBul(birinciSayi, ikinciSayi);
        System.out.println("ekok : "+ekok);
    }

    private static int ekokBul(int birinciSayi, int ikinciSayi) {

        //ebob * ekok = s1 * s2 ;

        int ekok = (birinciSayi * ikinciSayi) / ebobBul(birinciSayi, ikinciSayi);

        return ekok;
    }

    private static int ebobBul(int birinciSayi, int ikinciSayi){
        int ebob = 1;
        int kontrol = 2;

        while(kontrol <= birinciSayi && kontrol <= ikinciSayi){
            if(birinciSayi % kontrol == 0 && ikinciSayi % kontrol == 0){
                ebob = kontrol;
            }

            kontrol++;
        }

        if(ebob==1){
            System.out.println(birinciSayi + " ve " + ikinciSayi + " aralarında asaldır");
        }else{
            System.out.println(birinciSayi + " ve " + ikinciSayi + " sayılarının en büyük ortak böleni :
"+ebob);
        }

        return ebob;
    }

    private static int klasikEkokBul(int birinciSayi, int ikinciSayi) {

        int buyuk = Math.max(birinciSayi, ikinciSayi);
        int ekok = 0;

        for(int i = buyuk; i<= birinciSayi*ikinciSayi; i++){
            if(i % birinciSayi == 0 && i% ikinciSayi == 0){
                ekok = i;
                break;
            }
        }

        return ekok;
    }

}

```

Çözüm 5

Enre Atunbilek Java Dersleri
 public class Cozum5 {

```

public static void main(String[] args) {

    Scanner tara = new Scanner(System.in);
    System.out.println("Toplam öğrenci sayısını giriniz");
    int ogrenciSayisi = tara.nextInt();

    System.out.println("Toplam soru sayısını giriniz");
    int soruSayisi = tara.nextInt();

    //5,10 ==> 5 x 10 matris
    char[][] tumCevaplar = new char[ogrenciSayisi][soruSayisi];
    char[] cevapAnahtari = new char[soruSayisi];

    cevapAnahtariOlustur(cevapAnahtari);
    ogrenciCevaplariniYerlestir(tumCevaplar);

    ogrenciCevaplariniGoruntule(tumCevaplar);
    cevapAnahtariniGoruntule(cevapAnahtari);

    ogrencileriDegerlendir(cevapAnahtari, tumCevaplar);
}

private static void cevapAnahtariniGoruntule(char[] cevapAnahtari) {
    for (char secenek : cevapAnahtari){
        System.out.print(secenek+",");
    }
    System.out.println();
}

private static void ogrenciCevaplariniGoruntule(char[][] tumCevaplar) {
    for (int satir = 0; satir < tumCevaplar.length; satir++) {
        for (int sutun = 0; sutun < tumCevaplar[satir].length; sutun++) {
            System.out.print(tumCevaplar[satir][sutun]+ " ");
        }
        System.out.println();
    }
}

private static void ogrencileriDegerlendir(char[] cevapAnahtari, char[][] tumCevaplar) {

    for (int satir = 0; satir < tumCevaplar.length; satir++) {

        int dogruCevapSayisi = 0;
        for (int sutun = 0; sutun < tumCevaplar[satir].length; sutun++) {
            if (tumCevaplar[satir][sutun] == cevapAnahtari[sutun]) {
                dogruCevapSayisi++;
            }
        }

        System.out.println(satir+" indexindeki öğrencinin dogru cevap sayısı : "
+dogruCevapSayisi);
    }
}

private static void cevapAnahtariOlustur(char[] cevapAnahtari) {
    for (int i = 0; i < cevapAnahtari.length; i++) {
        cevapAnahtari[i] = cevapOlustur();
    }
}

private static void ogrenciCevaplariniYerlestir(char[][] tumCevaplar) {
    for (int satir = 0; satir < tumCevaplar.length; satir++) {
        for (int sutun = 0; sutun < tumCevaplar[satir].length; sutun++) {
            tumCevaplar[satir][sutun] = cevapOlustur();
        }
    }
}

public static char cevapOlustur() {
    int rastgeleSayi = 65 + (int) (Math.random() * 5);//0 1 2 3 4
    char uretilenSecenek = (char) rastgeleSayi;
}

```

```
        return uretilenSecenek;  
    }  
}
```