

## 1.B

Varargs Java 5 ile gelen metoda istediğin kadar parametre yollayarak sonuç almanı sağlayan bir özelliktir. Yani parametre sayının değişken ancak metodun yapacağı iş sabitse Varargs kullanılabilir.

<https://ekremcandemir.wordpress.com/2015/06/23/13-varargs-degisken-uzunluklu-argumanlar/>

## 2.B

İlk elemanı F[0].

## 3.A

Primitive tipler int, double, boolean vb. şeklinde küçük harflerle tanımlanmış olan 8 tiptir. Integer ise wrapper'dır.

## 4.C

[ ]double lion; false

double[ ] tiger; true

double bear[ ]; true

## 5.C

```
public class Q5 {  
  
    public static void printStormName(String... names) {  
        // TODO Auto-generated method stub  
        System.out.println(Arrays.toString(names));  
    }  
    public static void printStormNames(String[] names) {  
        // TODO Auto-generated method stub  
        System.out.println(Arrays.toString(names));  
    }  
  
    public static void main(String[] args) {  
        printStormName("azim");  
        printStormName(new String[] {"azim"});  
        printStormNames("azim");  
        printStormNames(new String[] {"azim"});  
    }  
}
```

## 6.A

```
int dizi[] = new int[5];  
System.out.println(dizi.length);
```

## 7.C

```
int[][] multiples = new int[4][2]; // 2D integer array with 4 rows and 2 columns  
String[][] cities = new String[3][3]; // 2D String array with 3 rows and 3 columns
```

## 8.B

## 9.B

### 10.B

String olarak sıralama yapacaktır. Yani alfabetik sıralama.

### 11.B

### 12.C

Eğer initializer ({})) kullanılarak dizi elemanları belirtilecekse [] içerisinde dizinin boyutu belirtilmiyor. Bu yüzden 2 ve 4 yanlış.

### 13.B

Sadece float tiger = new float[1]; ifadesi doğrudur. Dizi tanımlarken ya {} ifadesi ile dizi elemanları belirtilecek ya da int[5] şeklinde dizi boyutları belirtilecektir.

### 14.C ??

In case of binary search, array elements must be in ascending order. If you have unsorted array, you can sort the array using *Arrays.sort(arr)* method.

### 15.A

Dizilerde olduğu zaman otomatik genişleme gibi bir özellik yoktur.

### 16.C

m3 satırı [ArrayIndexOutOfBoundsException](#) verecektir. Çünkü dizinin 1 satırdan oluşması gerekirken sanki 2. satır var gibi yazılmaya çalışılan ilk satır burasıdır. Yani m3 satırı olmasa da m4 satırında aynı hatayı alacaktık.

### 17.B

```
public class Q17 {  
    public static void main(String[] args) {  
        String[] os = new String[] { "mac", "linux", "windows" };  
        Arrays.sort(os);  
        System.out.println(Arrays.binarySearch(os, "mac"));  
    }  
}
```

### 18.A

New char [3,3] kısmı yanlış yazılmış.

### 19.???

### 20.B

### 21.D

Çoklu dizi elemanları süslü parantezler içerisinde tanımlanırken ;

Herbir süslü parantez bir satırı temsil eder, içerisindeki virgülle ayrılmış veriler de satır elemanlarıdır.

```
char [][] blocks = new char [][] {{'a','b','c'},{'d'},{'e','f'}};
```

kodunun çıktısı

a-b-c

d

e-f

şeklinde olacaktır. Buna uygun ifade D seçeneğidir.

## 22.D

```
public class Q22 {
    public static void addStationName(String[] names) {
        names[names.length-1] = "Times Square";
        // eger names.length şeklinde olursa oyle bir yer olmadığı için hata verecektir.
        for (int i = 0; i < names.length; i++) {
            System.out.println(names[i]);
        }

        public static void main(String[] args) {
            String[] send_ar = { "a", "b" };
            addStationName(send_ar);
        }
    }
}
```

Kod yukarıdaki şekilde olsa idi B şıkkı doğru olurdu.

## 23.C

days.size() ifadesi yanlıştır.

## 24.C

bools ve moreBools dizileri 3 boyutlu dizilerdir.

## 25.C

## 26.B

[1][3] diyerek olmayan bir elemanı çağırmış

## 27.D

While using the varargs, you must follow some rules otherwise program code won't compile. The rules are as follows:

- There can be only one variable argument in the method.
- Variable argument (varargs) must be the last argument.

## 28.D

```
public class Q28 {
    static int[][] game = new int[6][6];

    public static void main(String[] args) {
        game[3][3] = 6;
        Object[] obj = game;
    }
}
```

```

        obj[3] = "X";
        // int diziye string atanmaya çalışılmıştır.
        System.out.println(game[3][3]);
    }
}

```

## 29.C ...neden -3 veriyor

```

public class Q29 {
    public static void main(String[] args) {
        String[] os = new String[] { "Mac", "Linux", "Windows" };
        Arrays.sort(os);
        System.out.println(Arrays.binarySearch(os, "RedHat"));
        // RedHat dizi elamni olmadigi için bulamayacak ve default -3 degeriini
        // döndürecek
    }
}

```

## 30.B

```

public class Q30 {
    public static void main(String... names) {
        System.out.println(names[0]);
    }
}
// java Q30 Wolfie seklinde çalıştırdığımızda
// ilk eleman olarak Wolfie yi atamıs olacağız ve çalıştırdığımız da onu
// göreceğiz. İlk deger vermeden çalıştırırsak C secenegi dogru olur.

```

## 31.A

```

public class Q31 {
    public static void main(String target[]) {
        System.out.println(target.length);
        // herhangi bir eleman atamadığımız için 0 dönecektir.
    }
}

```

## 32.B

```

public class Q32 {
    public static void main(String[] args) {
        String one = args [0];
        Arrays.sort(args);
        int result = Arrays.binarySearch(args, one);
        System.out.println(result);
        //ilk eleman search edilecek olduğu için de 1 donecek
    }
}

```

## 33.D

A, B ve C seçenekleri 2\*1 dizi iken D seçeneği 1\*2 dizidir.

## 34.C

3.satirin 3.elemanina erisilmsi gerekiyor.

## 35.A

6 tanesini yazdırdıktan sonra exception verecektir

### 36.C

Tek eleman atanan dizinin 2. Elemanı çağrıldığı için outOfBounds.

### 37.D

Kod normal olarak çalışacaktır.

### 38.C

Diziye 2 eleman atandığı için sonuç 2 olacaktır.

### 39.B ??

### 40.B

```
public class Q40 {
    public static void main(String... args) {
        Arrays.sort(args);
        System.out.println(Arrays.toString(args));
        //bos bir dizi olduğu için [] yazacaktır.
        //eger eleman atansaydı elemanları da yazacaktır
    }
}
```

### 41.B

```
public class Q41 {
    public static void main(String[] args) {
        String[] os = new String[] { "mac", "linux", "windows" };
        System.out.println(Arrays.binarySearch(os, "linux"));
    }
}
```

### 42.B

```
game[3][3] = "X";
```

satırında int diziye string atanmaya çalışıldığı için derlenmeyecektir

### 43.A

```
public class Q43 {
    public static void main(String[] args) {
        String[][] listing = new String[][] { { "Book" }, { "Game", "29.99" } };
        System.out.println(listing.length + " " + listing[0].length);
        //2 boyutlu dizinin length istersek row sayısı olacaktır
    }
}
```

### 44.C

Dizinin olmayan elemanı çağrılıyor

### 45.A

```
public class Q45 {
    public static void main(String[] args) {
        String[] days = new String[] { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
        "Friday", "Saturday" };
        for (int i = 1; i < days.length; i++) {
```

```

        System.out.println(days[i]);
        // i=1 den basladigi icin ilk eleman yazilmayacak
    }
}

```

## 46.B

```

public class Q46 {
    public static void main(String target[]) {
        System.out.println(target.length);
    }
}

```

// terminalden calistirilirken string bir eleman atandigi icin 1 donecek

## 47.A ?? 41 'le karsilatur.

```

public class Q47 {
    public static void main(String[] args) {
        String[] os = new String[] { "Linux", "Mac", "Windows" };
        System.out.println(Arrays.binarySearch(os, "Linux"));
    }
}

```

## 48. ??

## 49.?? Bi tane daha benzeri vardi

## 50.C

```

public class Q50 {
    public static void main(String[] args) {
        Arrays.sort(args);
        String result = Arrays.binarySearch(args, args[0]);
        //binarySearch metodu int deger dondugu icin derlenemez
        System.out.println(result);
    }
}

```