

## Metotlar II

```
public static int FonksiyonAdı(String a, double b) {  
    // kod  
}
```

**public**, erişim belirleyicisidir. Bununla ilgili daha fazla bilgiyi ilerleyen bölümlerde öğreneceksiniz.

**static**, metodu sınıfa ait olduğunu ve sınıfın bir örneği olmadığını belirtir.

**int**, dönüş tipidir. Olası dönüş tipleri **int**, **double**, **String**, **boolean**, **void**, vb. olabilir. **FonksiyonAdı**, fonksiyonun adıdır.

**String a**, **double b**, parametrelerdir. İstedığınız kadar parametreniz olabilir, ancak virgülle ayrılmalıdır. Parametreler, fonksiyona iletilen değişkenlerdir. Bu değişkenleri fonksiyonda kendi içinde tanımlanmış gibi kullanabilirsiniz.

- Parametre isimleri ve argüman isimleri farklı olabilir.

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 4;  
  
    // a, b sum fonksiyonuna argüman olarak iletiliyor  
    sum(a, b);  
}  
  
// parametre isimleri x, y  
public static int sum(int x, int y){  
    return x+y;  
}
```

- Java'da fonksiyonlar aşırı yüklenebilir (overloaded). Bu, aynı isme sahip ancak farklı parametrelere sahip birden çok fonksiyon olabileceği anlamına gelir.

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 4;  
    int c = 1;  
  
    sum(a, b); // A'yı çağırır  
    sum(a, b, c); // B'yi çağırır  
}  
  
// A  
public static int sum(int x, int y){  
    return x+y;  
}  
  
// B  
public static int sum(int x, int y, int z){  
    return x+y+z;  
}
```

**Eğlenceli ekstra bilgi:** `println()`, `System` sınıfında aşırı yüklenen bir metoddur. Aşağıdaki gibi yazabilirsiniz:

```
System.out.println("aynı"); //String  
System.out.println(24); //Integer
```

```
System.out.println(24.6); //Double  
System.out.println(a==b); //Boolean
```

- Java'da, ilkel tipler (int, char, double, float vb.) değer ile geçirilir. Nesneler ise referans ile geçirilir. (referans ile geçirme hakkında daha sonra öğreneceksiniz)

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 4;  
  
    // a, b değişkenleri yok edilemez  
    destroyer(a, b);  
  
    System.out.println(a); // 5  
    System.out.println(b); // 4  
}  
  
public static void destroyer(int a, int b){  
    // a, b'nin yerel kopyalarını değiştiriyoruz  
    a = 0;  
    b = 0;
```

## Egzersizler

1. Aşağıdaki döngüde hangi ifadeyi kullanmalıyız ki girdisinin tersini döndürecek şekilde fonksiyon tamamlansın?

```
public class punchcard {
    public static String tersCevir(String s) {
        String res = "";
        for(int i=0; i<s.length(); i++) {
            -----
        }
        return res;
    }

    public static void main(String[] args) {
        String s = "Merhaba Dünya!";
        System.out.println(tersCevir(s));
    }
}
```

- a) `res += s.charAt(i);`
- b) `res = s.charAt(i) + res;`
- c) `s.charAt(i) = s.charAt(s.length()-i);`
- d) `s.charAt(i) = s.charAt(s.length()-i-1);`

cevap: b - `res = s.charAt(i) + res;`

---

2. Bu programı çalıştırdığımızda ekrana ne yazdırılır?

```
public class punchcard {
    public static void swap(int a, int b) {
        int temp = a;
        a = b;
        b = temp;
    }

    public static void main(String[] args) {
        int a = 5;
        int b = 2;
        swap(a, b);
        System.out.println(a);
    }
}
```

- a) 0
- b) 2
- c) 5
- d) Null
- e) hata

cevap: c - 5

---

3. Aşağıdaki fonksiyonda ne yanlış?

```
public class punchcard {
    public static String birlestir(String s1, String s2) {
        // iki string'i birleştirir
        System.out.println(s1 + s2);
    }

    public static void main(String[] args) {
        // ana metod
        birlestir("Merhaba", "Dünya");
    }
}
```

```
}  
}
```

- a) Yorumlar fonksiyonlar içinde olamaz.
- b) static fonksiyonlar parametre alamaz.
- c) Dönüş tipi String, ancak fonksiyon hiçbir şey döndürmüyor.
- d) + operatörü stringleri birleştiremez.

cevap: c - Dönüş tipi String, ancak fonksiyon hiçbir şey döndürmüyor.

---

4. Aşağıdaki programdaki ne yanlış?

```
public class punchcard {  
    public static int min(int a, int b) {  
        if(a < b)  
            return a;  
        else  
            return b;  
    }  
    public static void main(String[] args) {  
        int a = 3;  
        int b = 2;  
        int c = 5;  
        a = min(a,b,c);  
    }  
}
```

- a) If-else yapısı süslü parantez içermiyor.
- b) min() fonksiyonunun parametre leri main içindeki a, b değişkenleriyle çakışıyor.
- c) a değişkeni atama operatörünün her iki tarafında da bulunuyor, bu da tanımsız davranışa neden oluyor.
- d) min fonksiyonuna iki değer yerine üç değer verilmeli.

5. Aşağıdaki programdaki ne yanlış?

```
public class punchcard {  
    public static innocentFunction() {  
        System.out.println("Ben masumum.");  
    }  
    public static void main(String[] args) {  
        innocentFunction();  
    }  
}
```

- a) Aynı kapsam içinde iki static fonksiyon olamaz.
- b) Bir fonksiyonun dönüş tipi olmalıdır.
- c) Bir fonksiyonun içinde birden fazla ifade olmalıdır.
- d) Bir fonksiyon bir şey döndürmelidir.

cevap: b - Bir fonksiyonun dönüş tipi olmalıdır.

---

6. Java'da bir metod ismi aşağıdakilerden hangisiyle başlayamaz?

- a) Sayı ile
- b) #
- c) - (tire)
- d) Yukarıdakilerin hepsi

cevap: d - Yukarıdakilerin hepsi

---

7. Aşağıdaki kod ne yapar?

```
public class punchcard {
    public static int gizem(int a, int b) {
        if(b==0)
            return a;
        else
            return gizem(a, b-1) + 1;
    }
    public static void main(String[] args) {
        System.out.println(gizem(2, 25));
    }
}
```

- a) Yinelemeli çıkarma ile iki sayıyı çıkarır.
- b) İki sayıyı bir dizi işlemiyle çarpar.
- c) İki sayıyı çok verimsiz bir şekilde toplar.
- d) İki sayıyı iteratif bölme ile böler.

*cevap:* Çok verimsiz bir şekilde iki sayıyı toplar.

---

8. Fonksiyonumun pozitif tamsayıların çarpımını hesaplamasını istiyorum. Boşluğa ne koymalıyım?

```
public class punchcard {
    public static int carp(int a, int b) {
        if(a == 1)
            -----
        else
            return carp(a-1, b) + b;
    }

    public static void main(String[] args) {
        System.out.println(carp(3, 5));
    }
}
```

- a) return 1;
- b) return 0;
- c) return a;
- d) return b;

*cevap:* d - return b;

---

9. Aşağıdaki programı çalıştırdığımızda ne olur?

```
public class punchcard {
    public static void a() {
        b();
    }
    public static void b() {
        a();
    }

    public static void main(String[] args) {
        a();
    }
}
```

- a) Program, herhangi bir çıktı vermeden sonsuz bir döngüye girer. (StackOverflowError)
- b) Program başarılı bir şekilde çalışır ve bir çıktı verdikten sonra sonlanır.

- c) Özyinelemeli metod çağrıları nedeniyle program derleme hatası ile karşılaşır.
- d) Program herhangi bir hata ile karşılaşmaz, ancak çalışmaz.

*cevap:* a - Program, herhangi bir çıktı vermeden sonsuz bir döngüye girer. (StackOverflowError)

---

10. Bir(n) pozitif olmayan n değeriyle çağrıldığında ne olur?

```
public class punchcard {
    public static void a(int n) {
        if(n==0)
            System.out.println("a() içinde biter");
        else
            b(n-1);
    }
    public static void b(int n) {
        if(n==0)
            System.out.println("b() içinde biter");
        else
            a(n-1);
    }

    public static void main(String[] Args) {
        a(5);
    }
}
```

- a) a() içinde biter
- b) b() içinde biter
- c) derleme hatası
- d) çalışma zamanı hatası

*cevap:* b - b() içinde biter

Eğer n çift ise a() içinde biter, tek ise b() içinde biter. Nedenini görebiliyor musunuz?

Şöyle varsayalım n=5

a(5) --> b(4) --> a(3) --> b(2) --> a(1) --> b(0) "b() içinde biter" yazdırır

---

11. Aşağıdaki kodun çıktısı nedir?

```
public class punchcard {
    public static void Fonksiyon(int a) {
        System.out.print("Girdi integer " + a);
    }
    public static void Fonksiyon(double a) {
        System.out.print("Girdi double " + a);
    }

    public static void main(String[] Args) {
        Fonksiyon(2);
    }
}
```

- a) Girdi integer 2
- b) Girdi double 2.0
- c) Girdi integer 2 Girdi double 2.0
- d) hata

*cevap:* a - Girdi integer 2

---

12. Son kodda ilk fonksiyonu kaldırırsak ne olur? Çıktı ne olur?

```
public class punchcard {  
    public static void Fonksiyon(double a) {  
        System.out.println("Girdi double " + a);  
    }  
  
    public static void main(String[] Args) {  
        Fonksiyon(2);  
    }  
}
```

- a) Girdi integer 2
- b) Girdi double 2.0
- c) Girdi integer 2 Girdi double 2.0
- d) hata

*cevap:* b - Girdi double 2.0