

ALGORİTMA VE PROGRAMLAMA I

HAFTA 8 DÖNGÜLER

Ders İçeriği: Hafta 8

- Giriş
- for Döngüsü
- while Döngüsü
- do-while Döngüsü
- break ve continue Anahtar Kelimeleri
- Örnekler

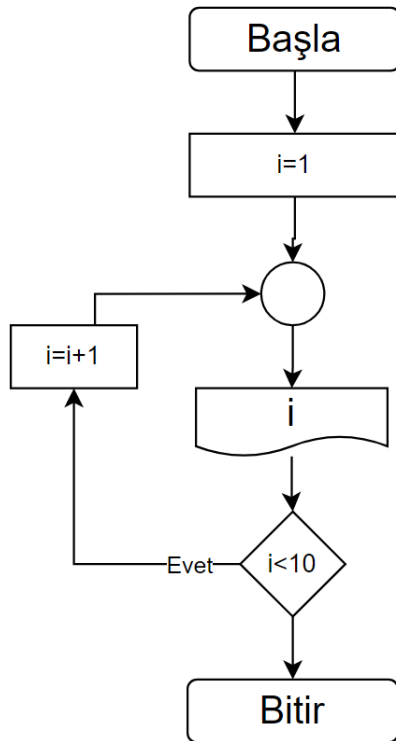
Giriş

- Bir grup işlemin belirli koşullar altında tekrarlanması için kullanılan programlama yapılarına döngü denir.
- Üçe ayrılır:
 - Sayaçlı döngüler (for): Döngü işlemleri bir sayaca bağlı olarak gerçekleştirilir.
 - Ön koşullu döngüler (while): Döngü işlemleri, döngü öncesinde kontrol edilen koşula bağlı olarak gerçekleştirilir.
 - Son koşullu döngüler (do-while): Döngü işlemleri, döngü sonunda kontrol edilen koşula bağlı olarak gerçekleştirilir. Bu durumda döngü en az bir kez çalıştırılır.
- Döngü deyimleri istenildiği kadar ve iç içe kullanılabilir.

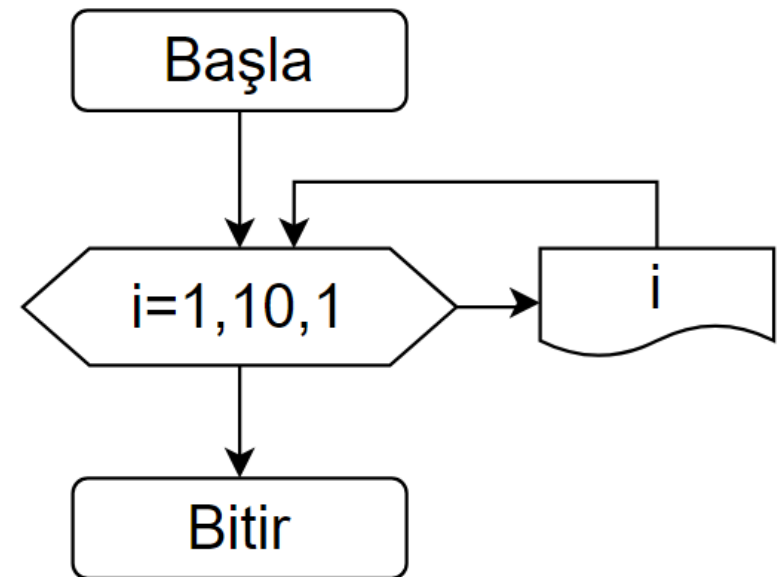
Giriş

- Örneğin 1'den 10'a kadar sayıları ekranda görüntüleyen algoritmayı inceleyelim.

Koşul İfadesiyle

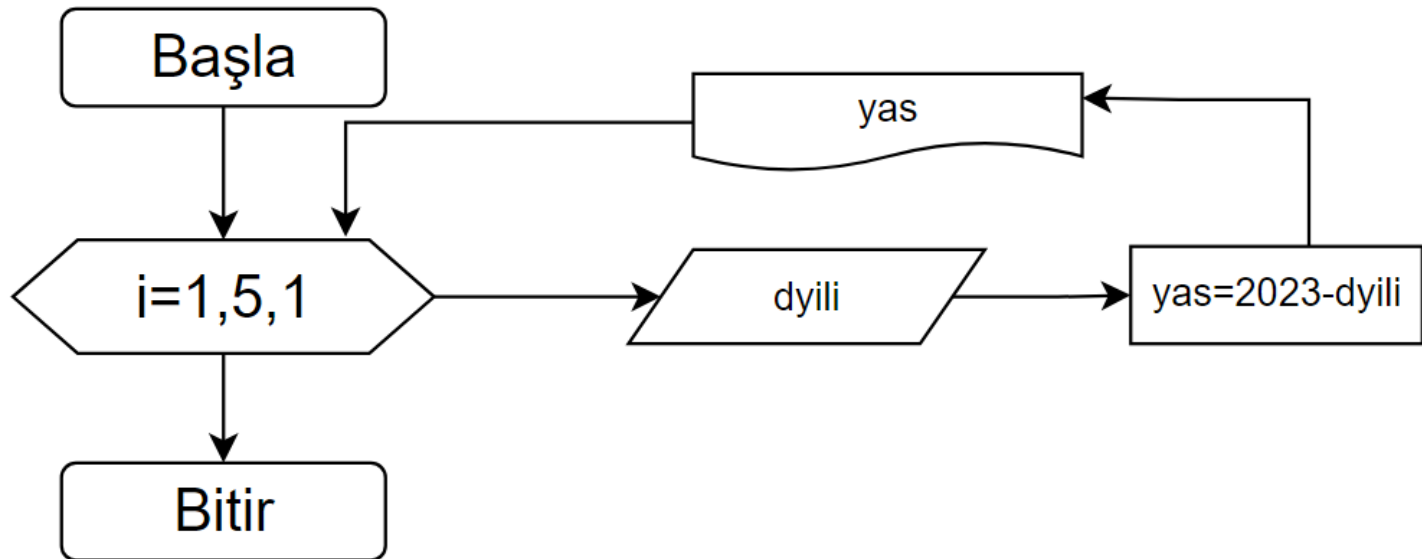


Döngü İfadesiyle



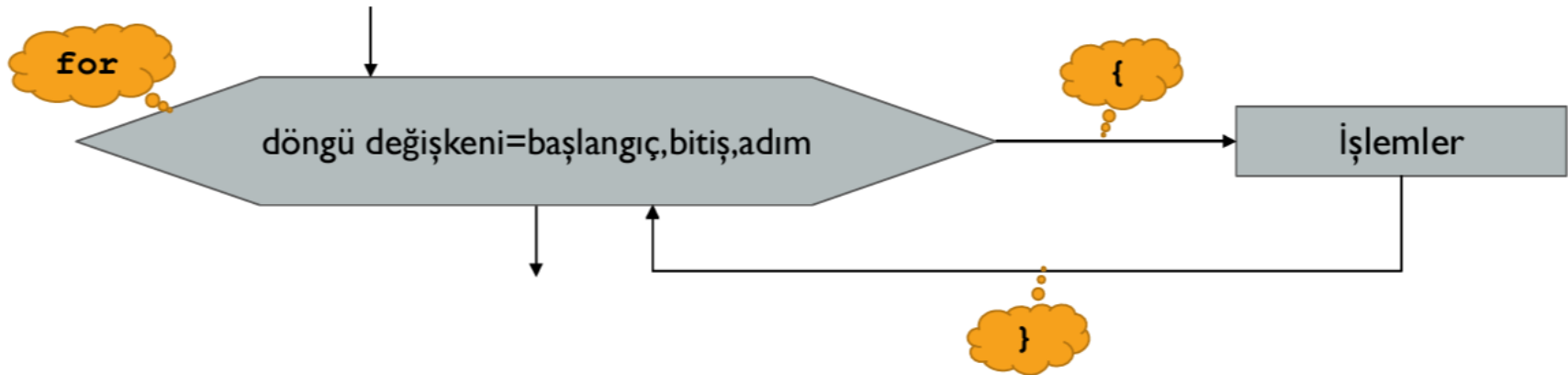
Giriş

- Örneğin klavyeden 5 kişinin doğum yılını girip 2023 yılındaki yaşını hesaplayan algoritmayı inceleyelim.



for Döngüsü

- Döngünün kaç kez çalışacağını bilindiği durumlarda kullanılır.
- Sayaç kontrollüdür. Sayaç, tekrar edilme sayısını tutan bir değişkendir.



```
for (tip başlangıç değeri; koşul; adım) {  
    .....  
    .....  
}  İşlemler
```

for Döngüsü

➤ Çalışma Mantığı:

- Döngü değişkeni sadece bir kez çalışır.
- Döngü koşulu döngüye girmeden ve her döngü tamamlandığında kontrol edilir.
- Döngü sonunda adım miktarı çalıştırılır.

➤ Dikkat!

- for döngüsündeki ifadeler « ; » ile ayrılır.
- Döngü deyiminin sonunda « ; » bulunmaz.
- Döngü içindeki deyimler { } arasına yazılır.
- Tek satır deyim var ise { } kullanılmayabilir.

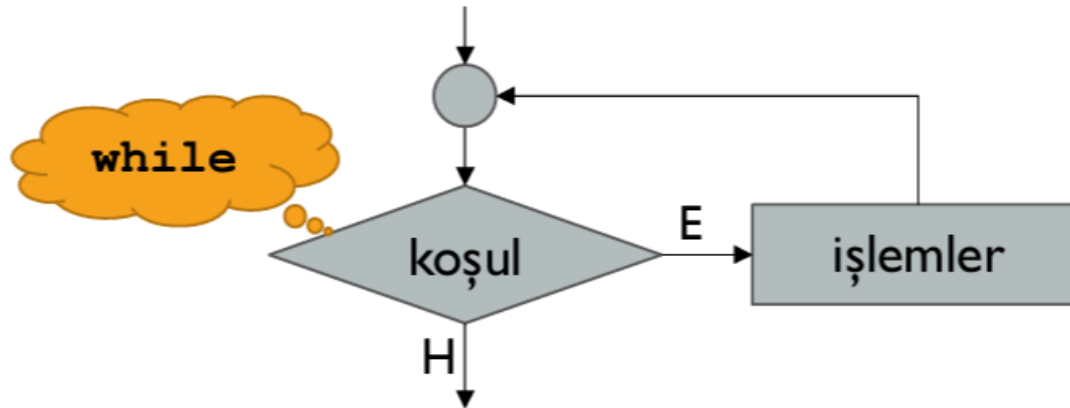
for Döngüsü

- Örnek: for döngüsü ile ekrana 10 kez Java Programlama yazan program.

```
public class Main {  
    public static void main(String[] args) {  
  
        for (int sayac = 1; sayac <= 10; sayac++) {  
            System.out.println("Java Programlama");  
        }  
    }  
}
```

while Döngüsü

- Döngünün belli bir koşul sağlandığı sürece tekrarlanması için kullanılan bir denetim yapısıdır.
- Koşul kontrollüdür. Koşul mantıksal sonucu true ya da false değer alır ve sonuç true oldukça döngü çalışır.



```
while ( koşul ) {
```

```
..... } işlemler
```

```
}
```

while Döngüsü

➤ Çalışma Mantığı:

- Koşul her döngü başında kontrol edilir.
- Koşuldaki mantıksal ifade true olduğu sürece döngü çalışır.
- Mantıksal ifade false değerini alana kadar yinelenir dolayısıyla mantıksal ifadenin sonlu sayıda yineleme sonrası değişmesi gerekir.
- Aksi halde sonsuz döngü denilen durum oluşur.

➤ Dikkat!

- while döngüsünün sonuna « ; » yazılmaz.
- Döngü içindeki deyimler { } arasına yazılır.
- Tek satır deyim var ise { } kullanılmayabilir.
- Koşul her zaman true ise sonsuz döngü oluşur.

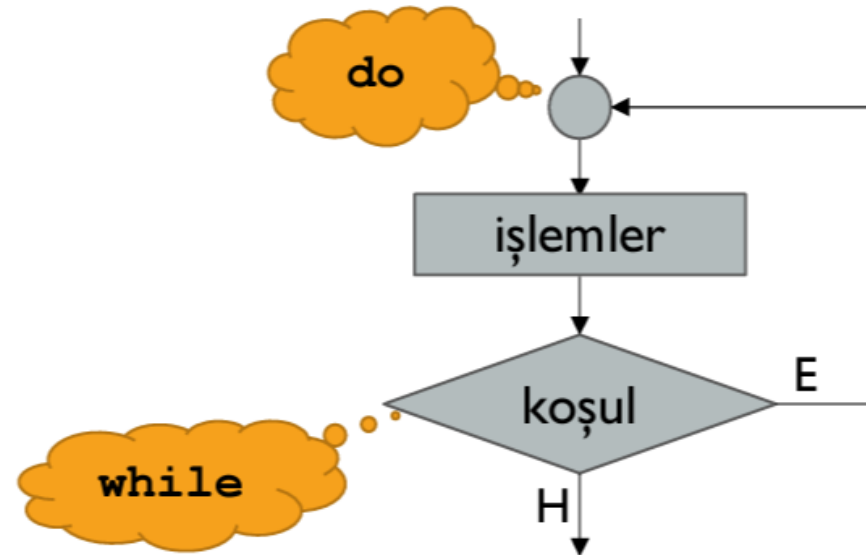
while Döngüsü

- Örnek: while döngüsü ile ekrana 10 kez Java Programlama yazan program.

```
public class Main {  
    public static void main(String[] args) {  
  
        int sayac=1;  
        while (sayac <= 10) {  
            System.out.println("Java Programlama");  
            sayac++;  
        }  
    }  
}
```

do - while Döngüsü

- while döngüsü ile benzer bir mantıkla çalışır.
- Farklı yanı ise koşul, döngü işlemlerinin sonunda kontrol edilir ve döngü deyimleri en az bir kez çalışmış olur.



do {

..... } işlemler
.....

} while (koşul);

do - while Döngüsü

➤ Çalışma Mantığı:

- Koşul her döngü sonunda kontrol edilir.
- Bu sebeple koşul ne olursa olsun döngü deyimleri en az bir kez çalıştırılmış olur.
- Koşuldaki mantıksal ifade true olduğu sürece döngü çalışır.

➤ Dikkat!

- do - while döngüsünde while ifadesinin sonuna « ; » yazılır.
- Döngü içindeki deyimler { } arasına yazılır.
- Tek satır deyim var ise { } kullanılmayabilir.
- Koşul her zaman true ise sonsuz döngü oluşur.

do - while Döngüsü

- Örnek: do - while döngüsü ile ekrana 10 kez Java Programlama yazan program.

```
public class Main {  
    public static void main(String[] args) {  
  
        int sayac=1;  
        do {  
            System.out.println("Java Programlama");  
            sayac++;  
        } while (sayac <= 10);  
    }  
}
```

break ve continue Anahtar Kelimeleri

➤ break Anahtar Kelimesi

- break anahtar kelimesine ulaşıldığında kod bloğundan çıkılır.
- Döngünün tamamen sona ermesi sağlanır.

➤ continue Anahtar Kelimesi

- continue deyimine ulaşıldığında, devamındaki komutlar atlanır.
- Döngünün başına tekrar gelinir ve devam koşulu test edilir.

break ve continue Anahtar Kelimeleri

➤ Örnek: break ve continue

```
public static void main(String[] args) {  
    int i;  
    for(i=1;i<=10;i++){  
        if(i==5)  
            break;  
        System.out.println(i);  
    }  
}
```

1
2
3
4

break ve continue Anahtar Kelimeleri

➤ Örnek: break ve continue

```
public static void main(String[] args) {  
    int i;  
    for(i=1;i<=10;i++){  
        if(i==5)  
            continue;  
        System.out.println(i);  
    }  
}
```

1
2
3
4
6
7
8
9
10

Örnekler

- Örnek 1: Aşağıdaki kodun ekran çıktısını inceleyelim.

```
public class Main {  
    public static void main(String[] args) {  
        int i, j;  
        for (i = 1; i <= 3; i++)  
            System.out.print("A, ");  
        for (j = 1; j <= 10; j += 3)  
            System.out.print("B, ");  
        System.out.println("C, ");  
    }  
}
```

A, A, A, B, B, B, B, C,

Örnekler

- Örnek 2: Aşağıdaki kodun ekran çıktısını inceleyelim.

```
public class Main {  
    public static void main(String[] args) {  
        int i, j;  
        for (i = 0; i < 4; i++)  
            for (j = i; j > 0; j --)  
                System.out.print("A");  
        System.out.print("B");  
    }  
}
```

AAAAAAB

Örnekler

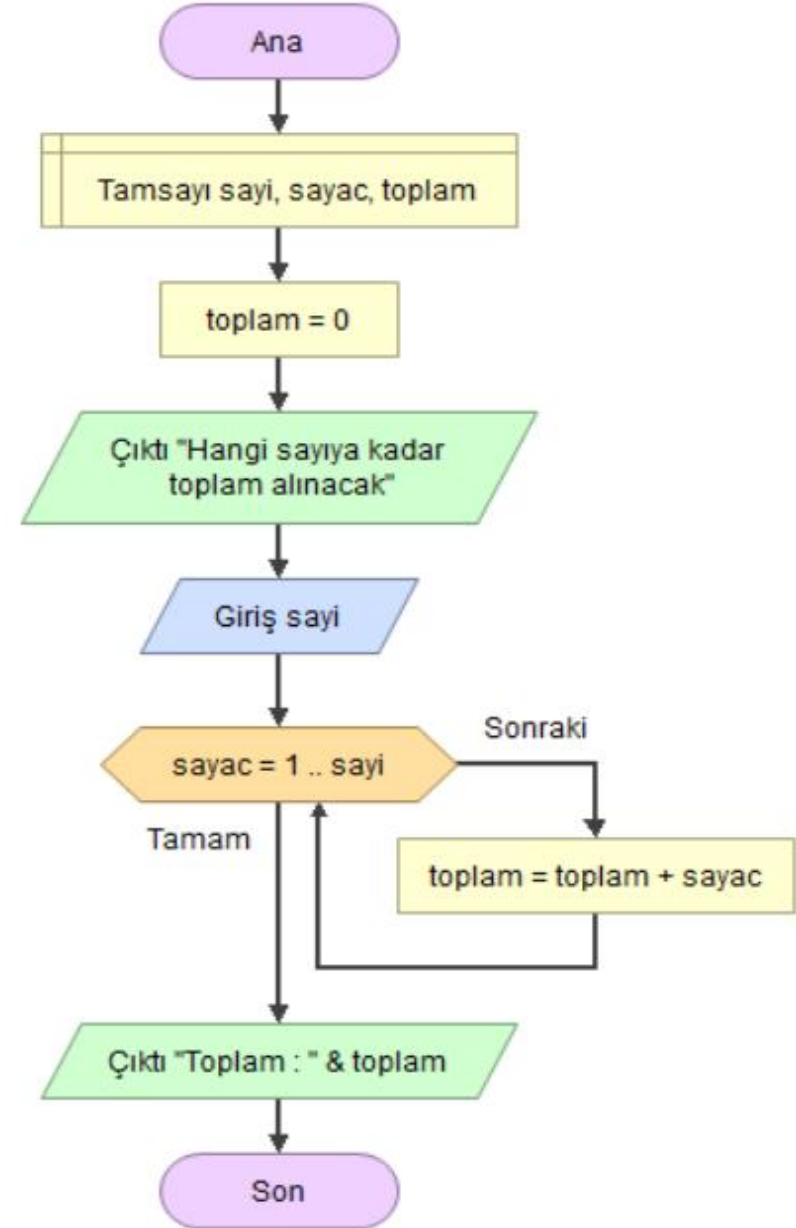
➤ Örnek 3: Aşağıdaki kodun ekran çıktısını inceleyelim.

```
public class Main {  
    public static void main(String[] args) {  
        int i, j;  
        for (i = 0; i < 4; i++) {  
            for (j = i; j > 0; j--)  
                System.out.print("A");  
            System.out.printf("B\n");  
        }  
    }  
}
```

B
AB
AAB
AAAB

Örnekler

- Örnek 4: Klavyeden girilen sayıya kadar olan sayıların toplamını bulan akış şemasını oluşturalım ve programını yazalım.



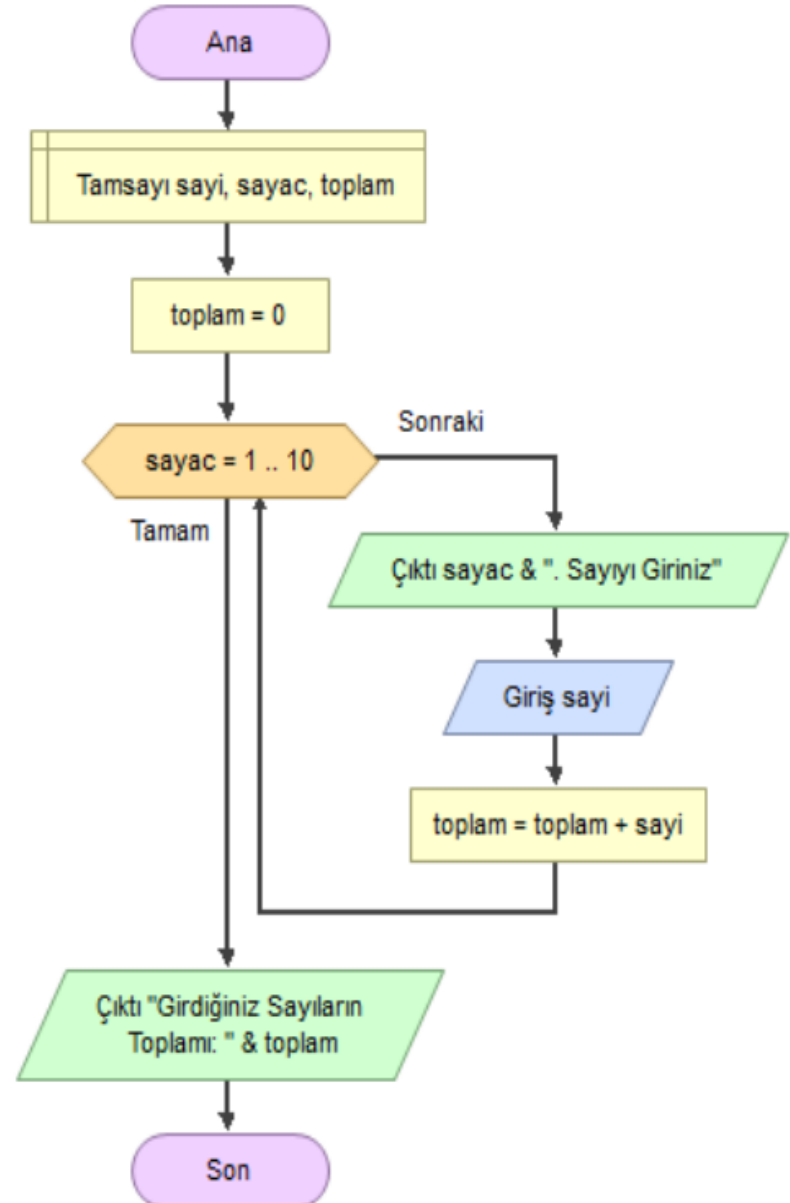
Örnekler

➤ Örnek 4:

```
public static void main(String[] args) {  
    int sayi, toplam=0;  
    Scanner klavye=new Scanner(System.in);  
    System.out.println("Bir sayı giriniz");  
    sayi= klavye.nextInt();  
    for(int i=1;i<=sayi;i++)  
        toplam = toplam + i;  
    System.out.printf("Sayıların toplamı: %d", toplam);  
}
```

Örnekler

- Örnek 5: Klavyeden girilen 10 tane sayıyı toplayan akış şemasını oluşturalım ve programını yazalım.



Örnekler

➤ Örnek 5:

```
public static void main(String[] args) {  
    int sayi, toplam=0;  
    Scanner klavye=new Scanner(System.in);  
    for(int i=1;i<=10;i++){  
        System.out.println(i+" sayıyı giriniz");  
        sayi= klavye.nextInt();  
        toplam = toplam + sayi;  
    }  
    System.out.printf("Sayıların toplamı: %d", toplam);  
}
```

Örnekler

- Örnek 6: Klavyeden girilen 10 tane sayıdan çift olanların sayısını ve toplamını, tek olanların sayısını ve toplamını hesaplayan programı yazalım.

```
int sayi, toplamTek=0, toplamCift=0;
int sayacTek=0, sayacCift=0;
Scanner klavye=new Scanner(System.in);
for(int i=1;i<=10;i++){
    System.out.println(i+" sayıyı giriniz");
    sayi= klavye.nextInt();
    if (sayi%2 == 0){
        sayacCift++;
        toplamCift += sayi;
    }
    else{
        sayacTek++;
        toplamTek += sayi;
    }
}
System.out.printf("Çift sayıların sayısı ve toplamı: %d", sayacCift, toplamCift);
System.out.printf("Tek sayıların sayısı ve toplamı: %d", sayacTek, toplamTek);
```

Örnekler

- Örnek 7: Klavyeden girilen pozitif tamsayının faktöriyelini hesaplayan programı yazalım.

```
public static void main(String[] args) {  
    int sayi, sayac;  
    long sonuc=1;  
    Scanner klavye=new Scanner(System.in);  
    System.out.println("Pozitif sayı giriniz");  
    sayi= klavye.nextInt();  
    while(sayi<=0) {  
        System.out.println("Yanlış!! Pozitif sayı giriniz");  
        sayi = klavye.nextInt();  
    }  
    for(int i=1;i<=sayi;i++)  
        sonuc=sonuc*i;  
    System.out.printf("%d! = %d", sayi, sonuc);  
}
```