



Bölüm 3: Bağlı Listeler

Veri Yapıları



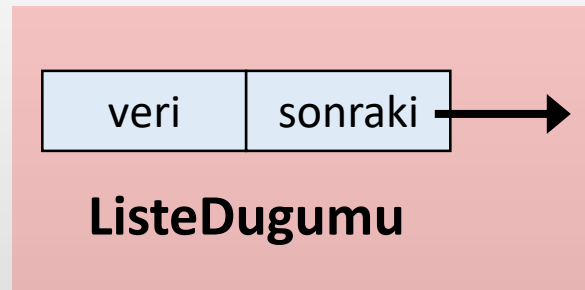
Tek Yönlü Bağlı Liste (Singly Linked List)

- Tek yönlü bağlı liste, elemanları birbirine tek yönlü bağlantılarla bağlanmış bir veri yapısıdır.
 - Her eleman, bir sonraki elemana doğru bir bağlantı içerir.
- Liste, ilk ve son eleman olmak üzere iki adet özel elemana sahiptir.
 - İlk eleman, listenin başını; son eleman ise listenin sonunu gösterir.
- Bağlı liste elemanları iki temel parçadan oluşur:
 - **Veri:** Elemanın taşıdığı bilgi veya veri.
 - **İşaretçi (Pointer):** Bir sonraki elemanın bellekteki adresini tutar.



Tek Yönlü Bağlı Liste (Singly Linked List)

- Bağlı liste, elemanların ardışık bellek konumlarında saklanmadığı bir lineer veri yapısıdır.
- Bağlı listedeki elemanlar, aşağıdaki resimde gösterildiği gibi işaretçiler kullanılarak birbirine bağlanır.





Tek Yönlü Bağlı Liste (Singly Linked List)

```
// Düğüm sınıfı
class Dugum {
    int veri;
    Dugum sonraki;

    Dugum(int veri) {
        this.veri = veri;
        this.sonraki = null;
    }
}
```



Tek Yönlü Bağlı Liste İşlemleri

- Tek yönlü bağlı listeler, aşağıdakiler gibi çeşitli işlemleri gerçekleştirmek için kullanılabilir:
- Eleman ekleme ve silme
- Eleman arama
- Listeyi yazdırma
- Listeyi ters çevirme
- İki listeyi birleştirme



Tek Yönlü Bağlı Liste Avantajları

- Kolay uygulanabilir
- Ekleme ve silme işlemleri hızlıdır
- Bellekte verimli bir şekilde kullanabilirler
- Dinamik boyut, yani eleman sayısı değiştirilebilir.



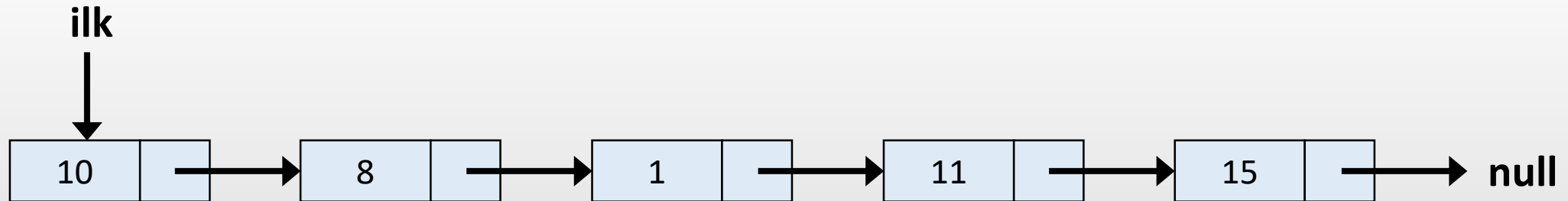
Tek Yönlü Bağlı Liste Dezavantajları

- Eleman arama işlemi yavaştır
- Listeyi ters çevirmek için ek işlem gereklidir
- İki listeyi birleştirmek için ek işlem gereklidir
- Rastgele erişim zordur, çünkü elemanlar peş peşe sıralıdır.
- Her elemanın işaretçiye ihtiyacı vardır, bu nedenle fazladan bellek kullanılabilir.



Temel Elemanlar

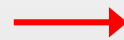
- Bağlı liste elemanları iki temel parçadan oluşur:
 - Veri: Elemanın taşıdığı bilgi veya veri.
 - İşaretçi (Pointer): Bir sonraki elemanın bellekteki adresini tutar.





Tek Yönlü Bağlı Liste Oluşturma

- Verilen kod, bağlı liste (linked list) tipinde bir veri yapısını oluşturur.
- İlk olarak, dört adet "Dugum" (Node) nesnesi oluşturur.
- Her düğüm, bir veri (10, 1, 8 ve 11) ve "sonraki" adında bir işaretçiyi içerir.
- Bu işaretçi, düğümün bir sonraki düğümü işaret etmesi için kullanılır.

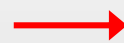


```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



Tek Yönlü Bağlı Liste Oluşturma

ilk \longrightarrow null



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
Dugum ucuncu = new Dugum(8);  
Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
ikinci.sonraki = ucuncu;  
ucuncu.sonraki = dorduncu;
```



Tek Yönlü Bağlı Liste Oluşturma

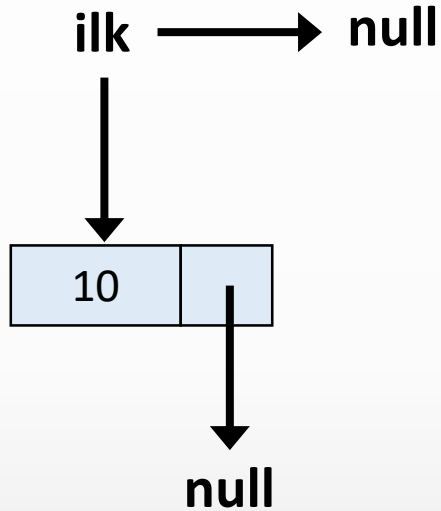
ilk \longrightarrow null

\longrightarrow

```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



Tek Yönlü Bağlı Liste Oluşturma

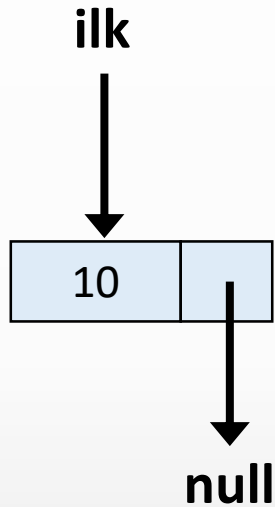


→

```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



Tek Yönlü Bağlı Liste Oluşturma

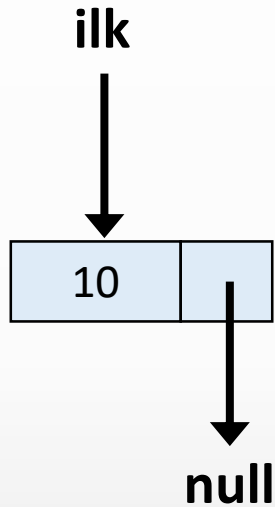


→

```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



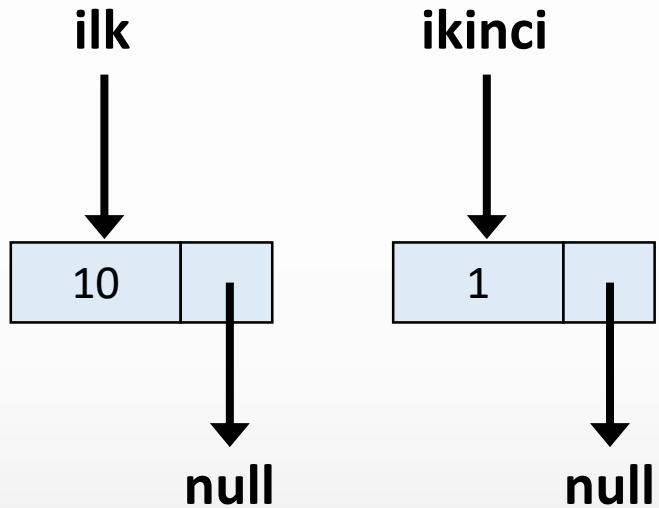
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
→ Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



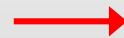
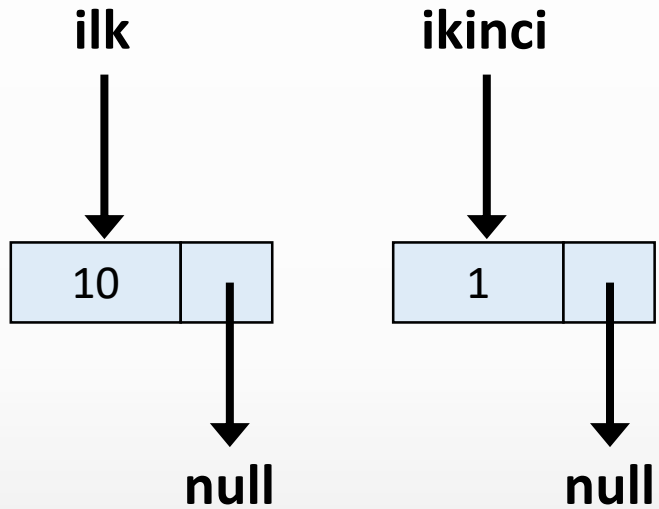
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
→ Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



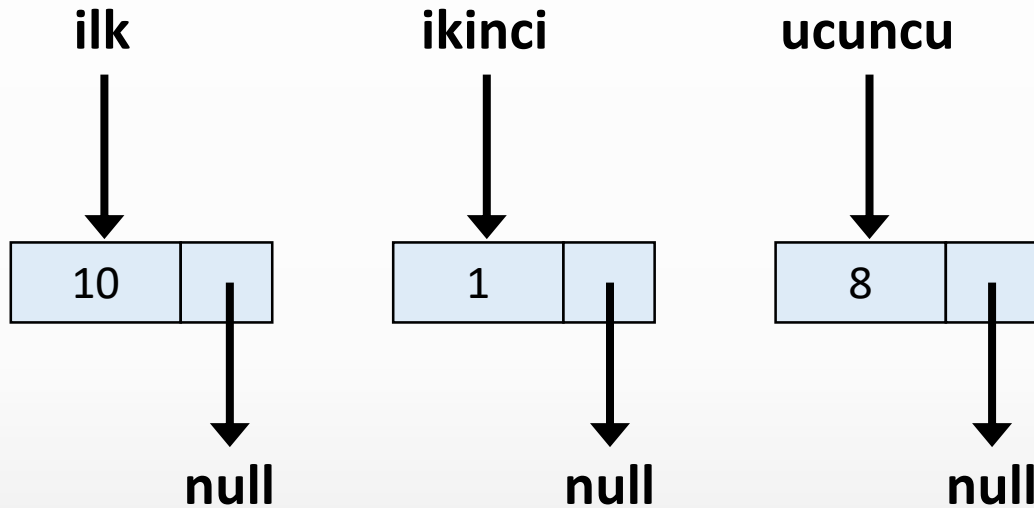
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



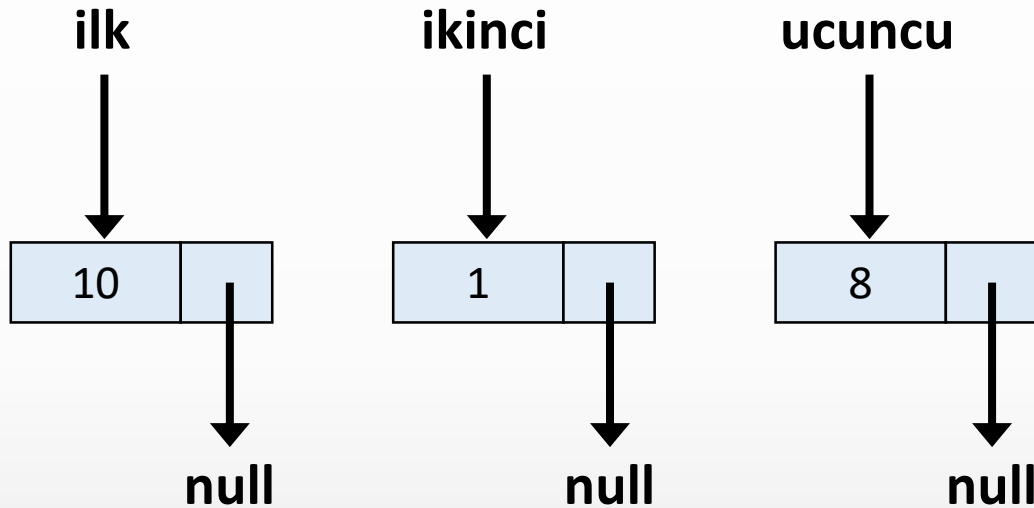

Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
→ Dugum ucuncu = new Dugum(8);  
Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
ikinci.sonraki = ucuncu;  
ucuncu.sonraki = dorduncu;
```



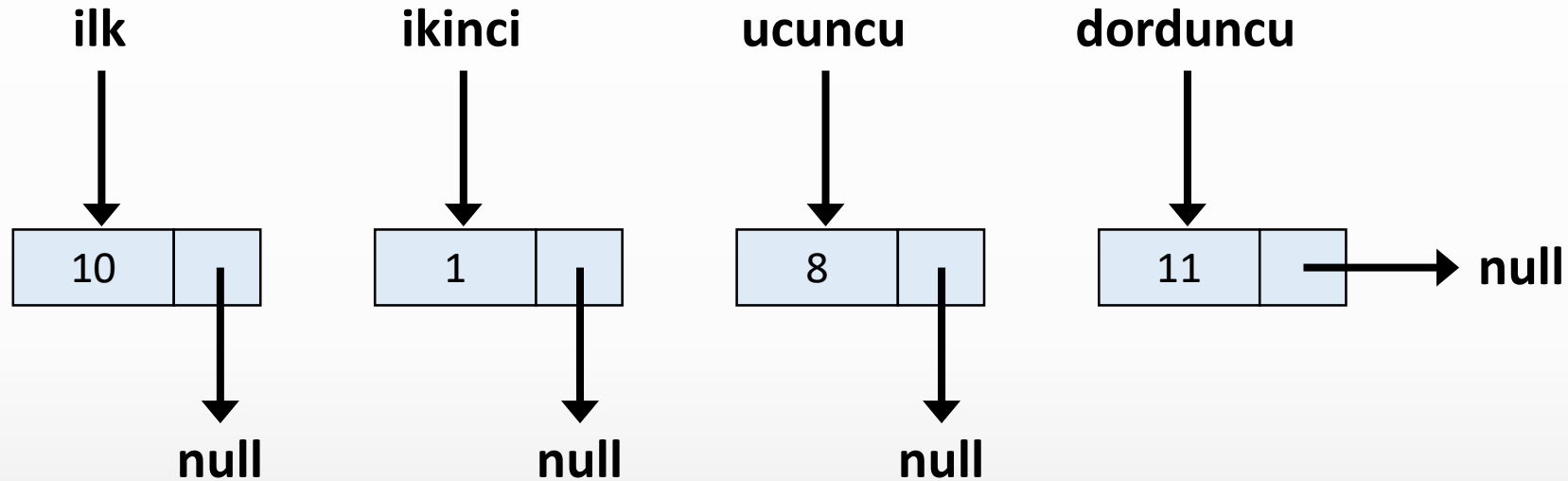
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
Dugum ucuncu = new Dugum(8);  
→ Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
ikinci.sonraki = ucuncu;  
ucuncu.sonraki = dorduncu;
```



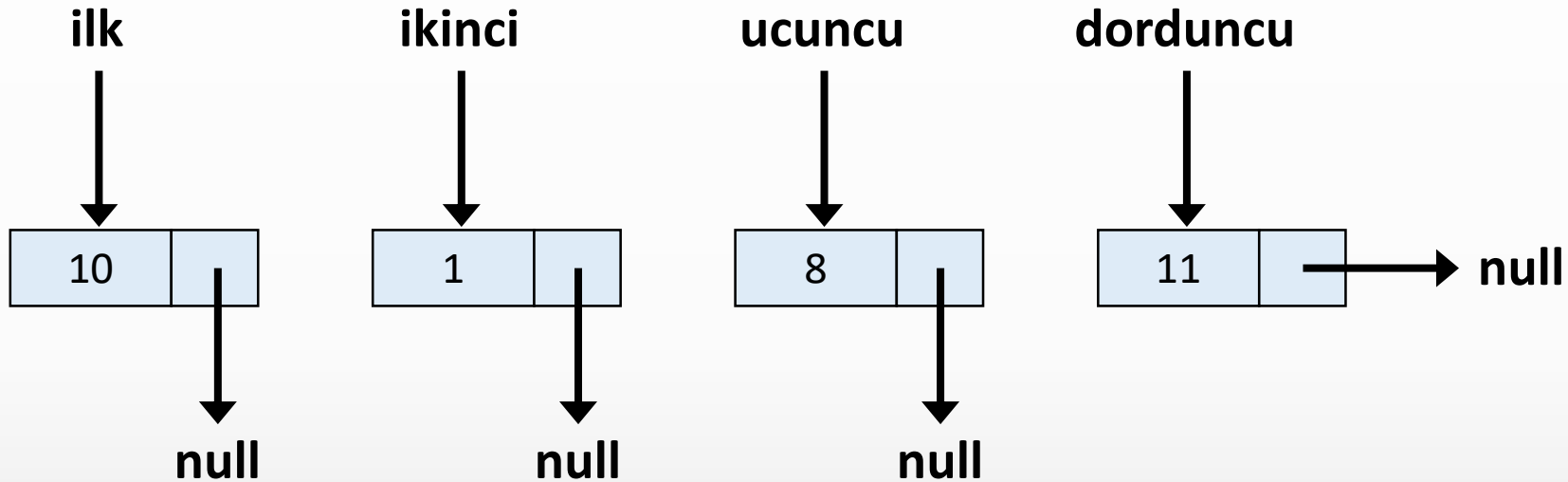
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
Dugum ucuncu = new Dugum(8);  
→ Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
ikinci.sonraki = ucuncu;  
ucuncu.sonraki = dorduncu;
```



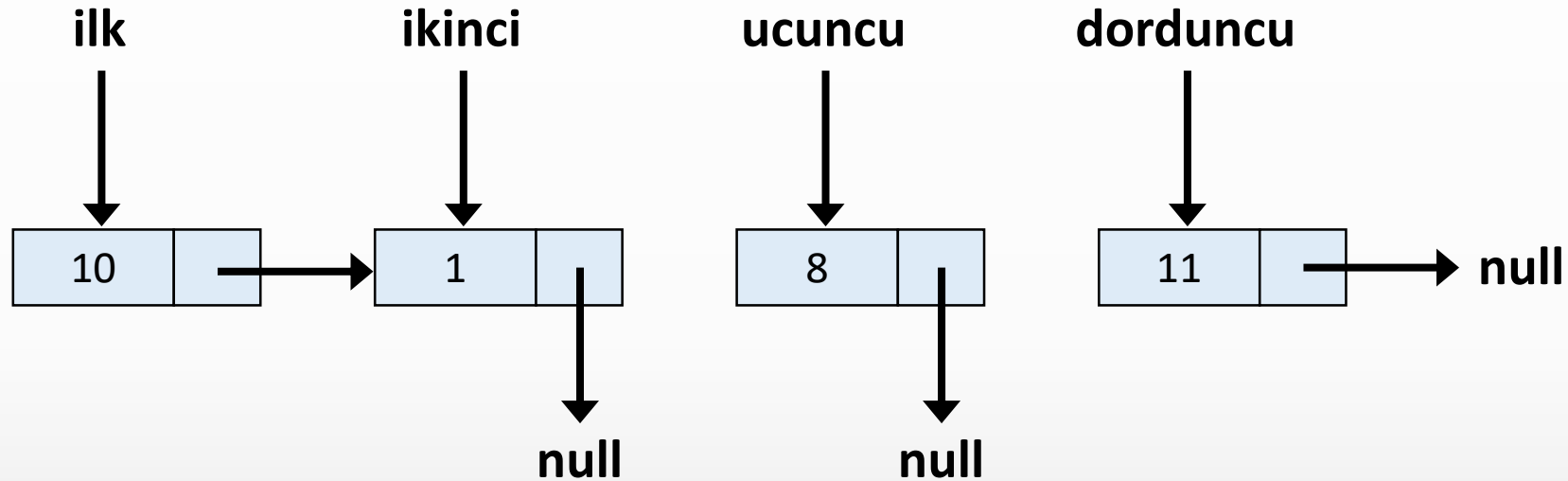
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
→ ilk.sonraki = ikinci;
   ikinci.sonraki = ucuncu;
   ucuncu.sonraki = dorduncu;
```



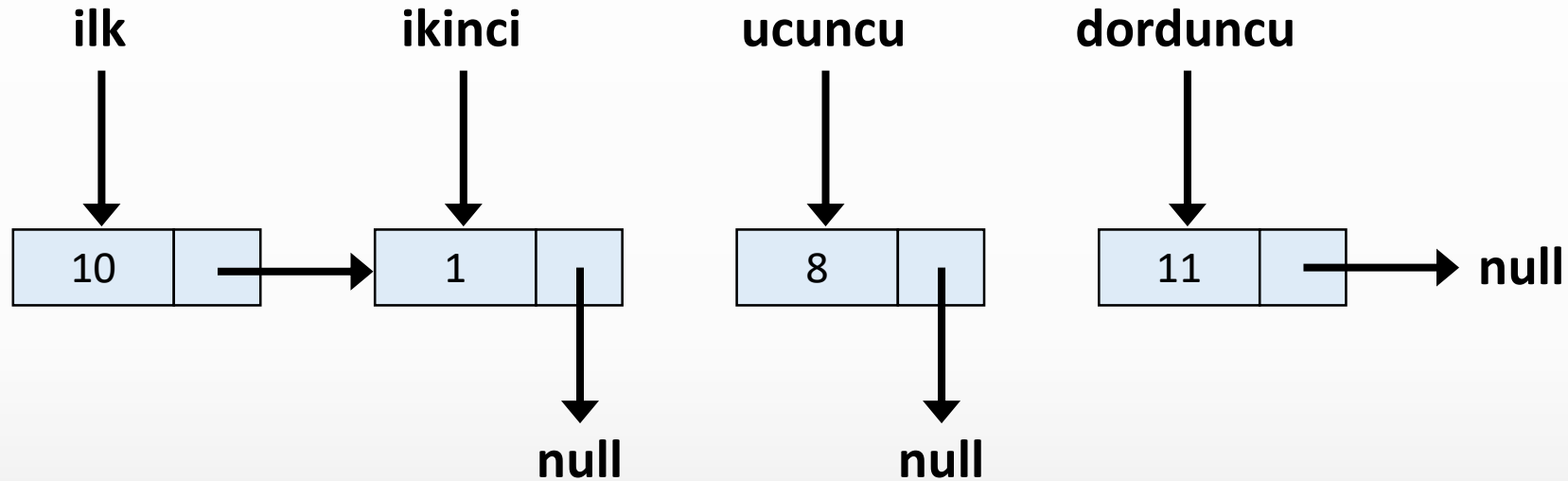
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
→ ilk.sonraki = ikinci;
   ikinci.sonraki = ucuncu;
   ucuncu.sonraki = dorduncu;
```



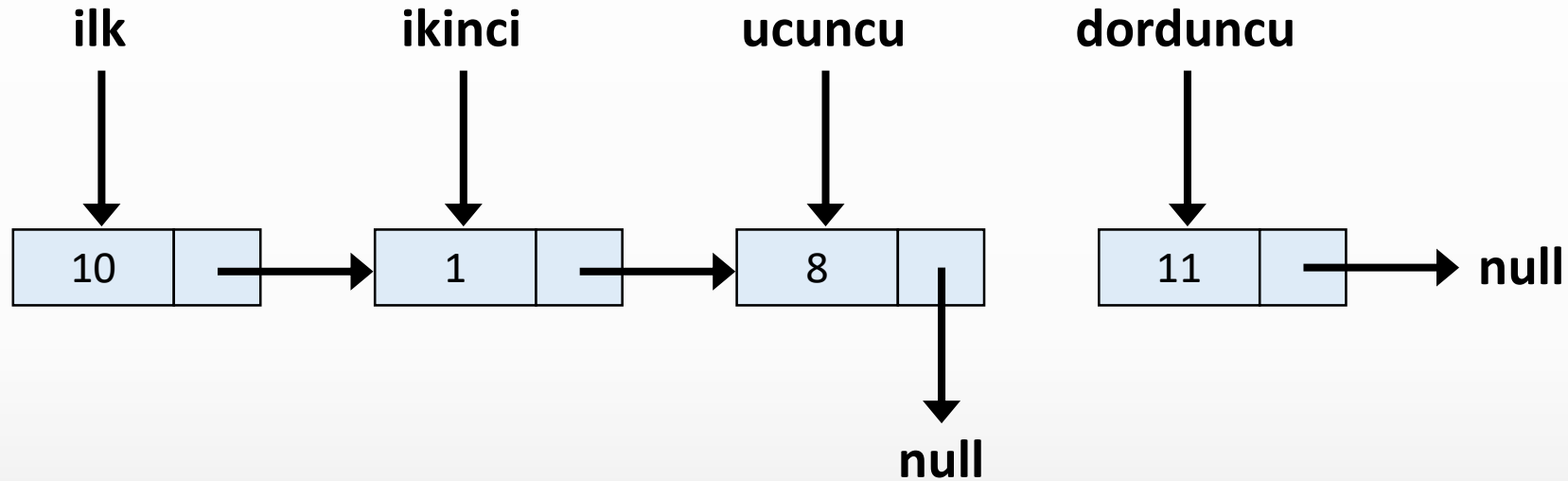
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
Dugum ucuncu = new Dugum(8);  
Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
→ ikinci.sonraki = ucuncu;  
ucuncu.sonraki = dorduncu;
```



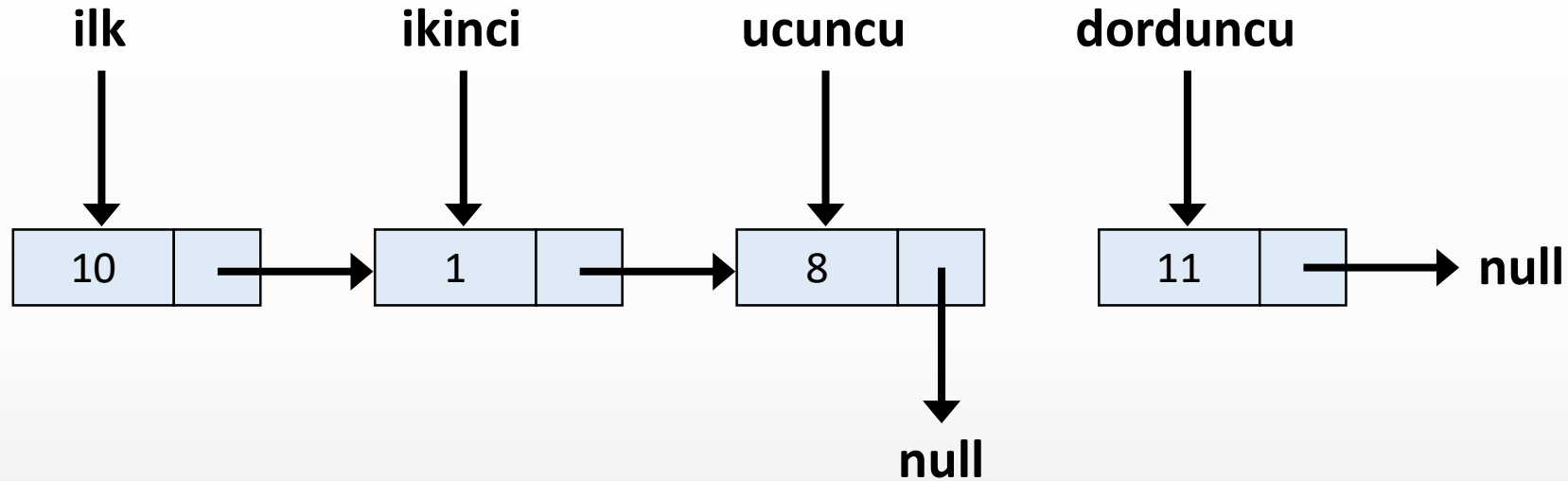
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
→ ikinci.sonraki = ucuncu;
ucuncu.sonraki = dorduncu;
```



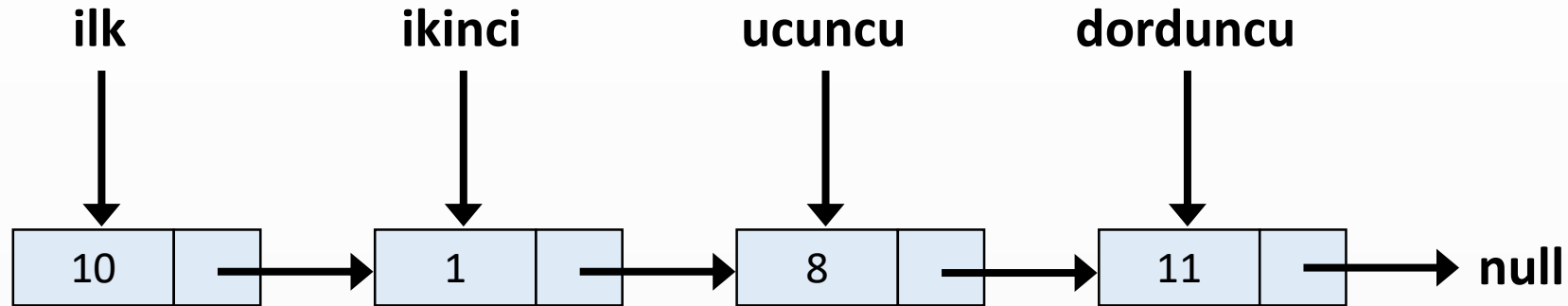
Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
→ ucuncu.sonraki = dorduncu;
```



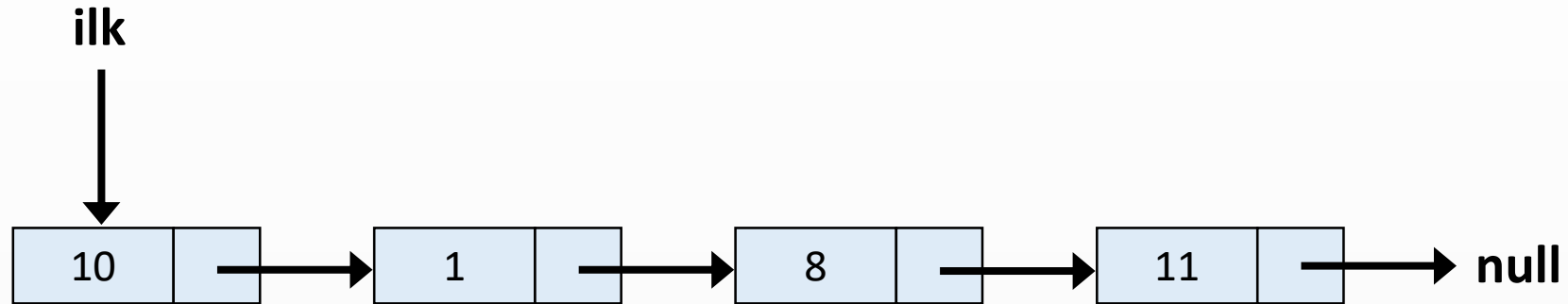

Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni
Dugum ilk = new Dugum(10);
Dugum ikinci = new Dugum(1);
Dugum ucuncu = new Dugum(8);
Dugum dorduncu = new Dugum(11);
ilk.sonraki = ikinci;
ikinci.sonraki = ucuncu;
→ ucuncu.sonraki = dorduncu;
```



Tek Yönlü Bağlı Liste Oluşturma



```
// ilk - örnek değişkeni  
Dugum ilk = new Dugum(10);  
Dugum ikinci = new Dugum(1);  
Dugum ucuncu = new Dugum(8);  
Dugum dorduncu = new Dugum(11);  
ilk.sonraki = ikinci;  
ikinci.sonraki = ucuncu;  
→ ucuncu.sonraki = dorduncu;
```

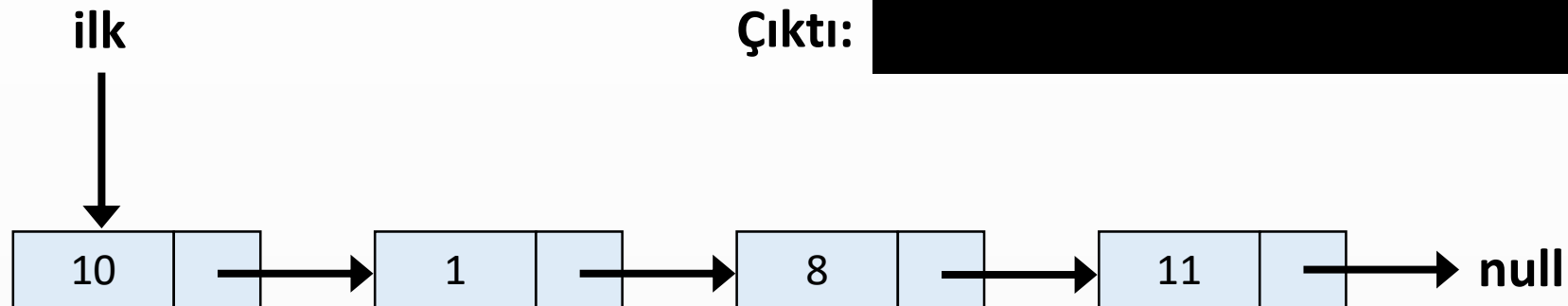


Liste Üzerinde Dolaşma ve Verileri Yazdırma

```
void listeyiYazdir() {  
    Dugum gecici = bas;  
    System.out.print("Bağlı Liste: ");  
    while (gecici != null) {  
        System.out.print(gecici.veri + " -> ");  
        gecici = gecici.sonraki;  
    }  
    System.out.println("null");  
}
```



Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



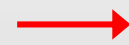
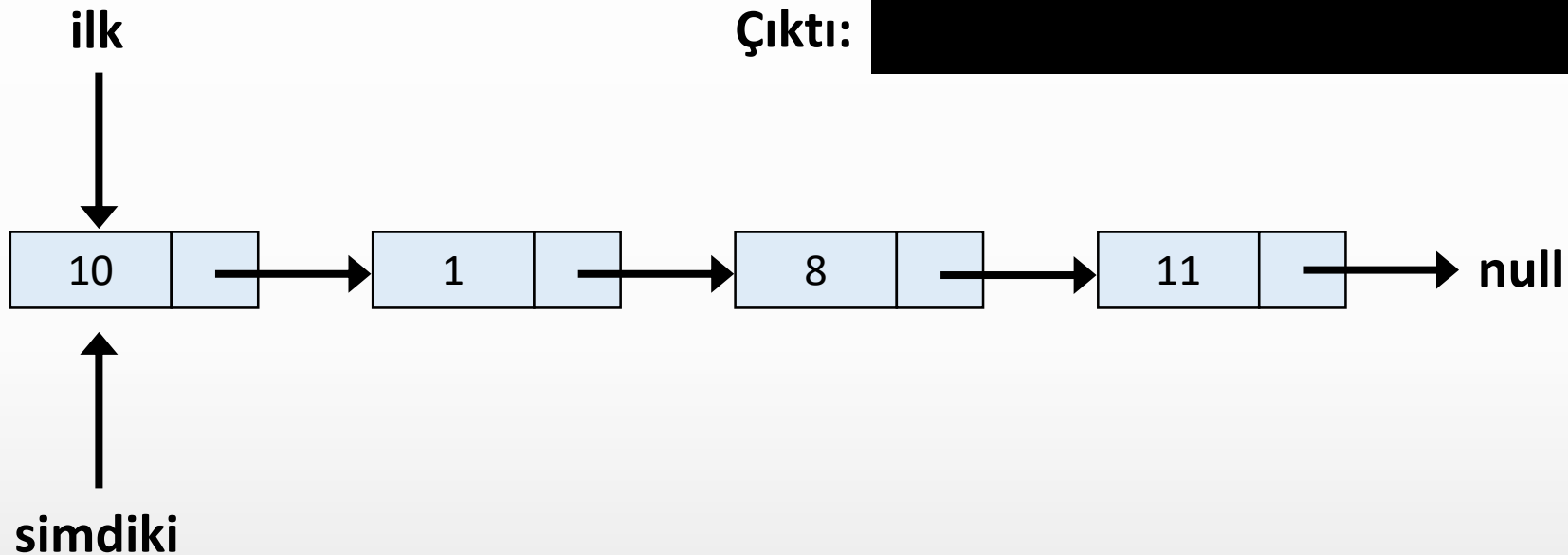
```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



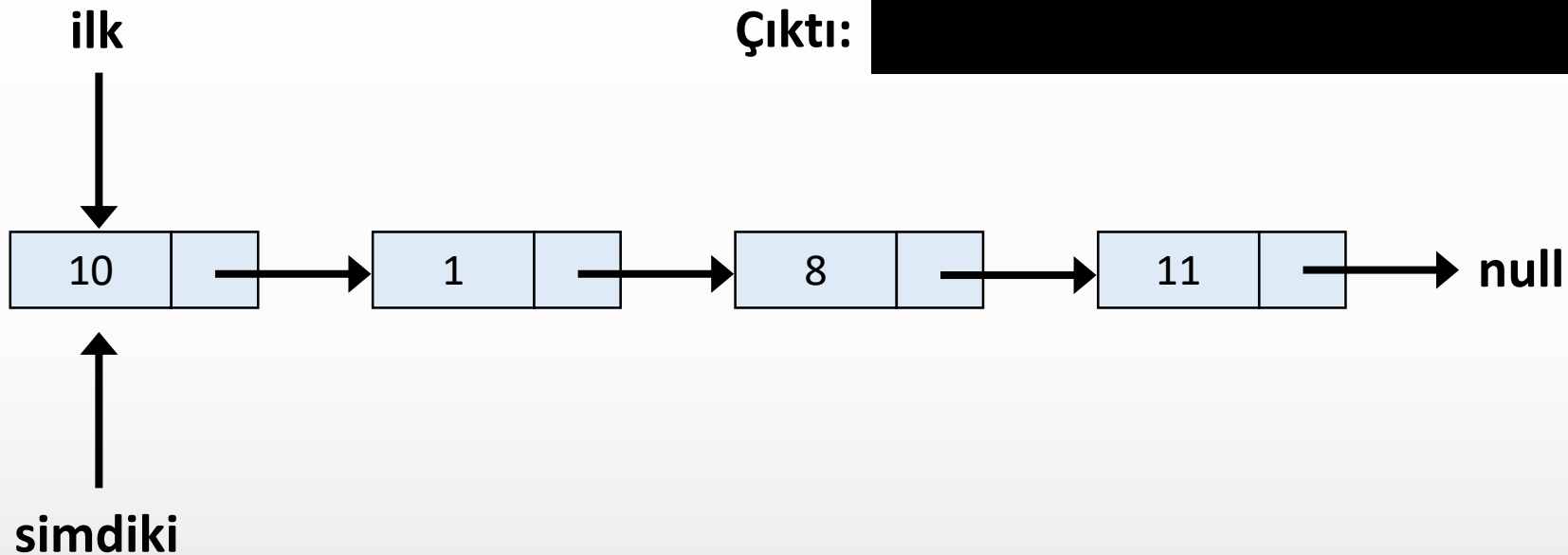
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



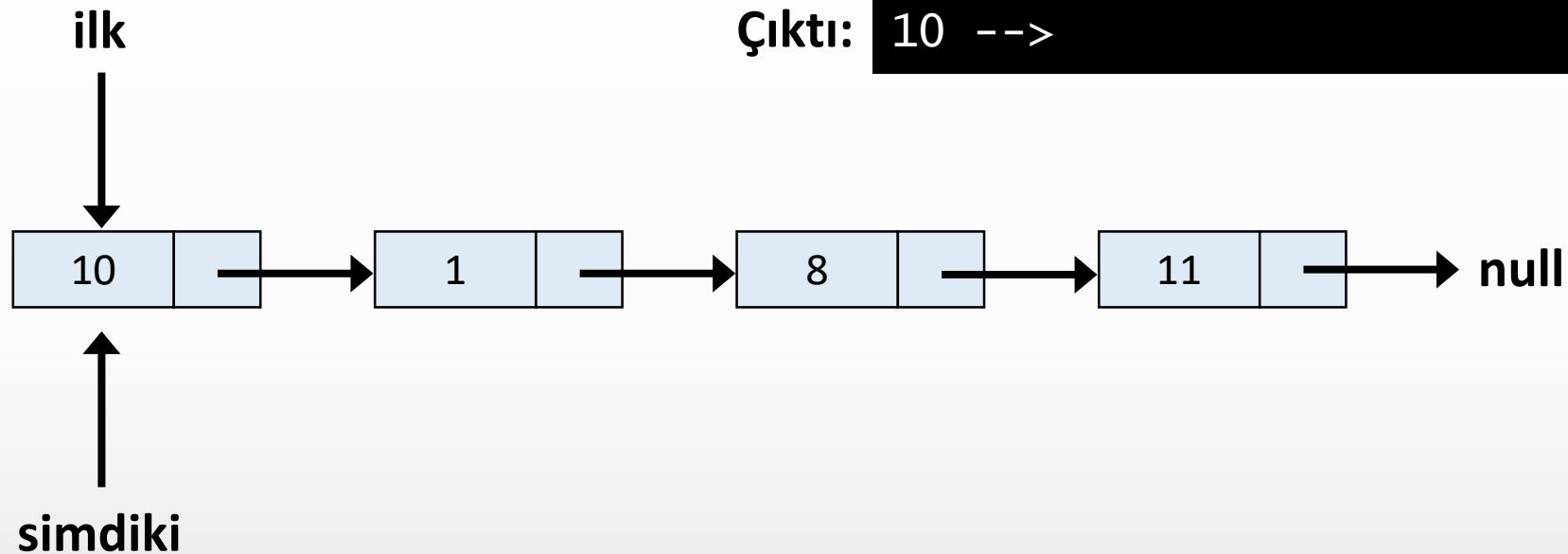
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



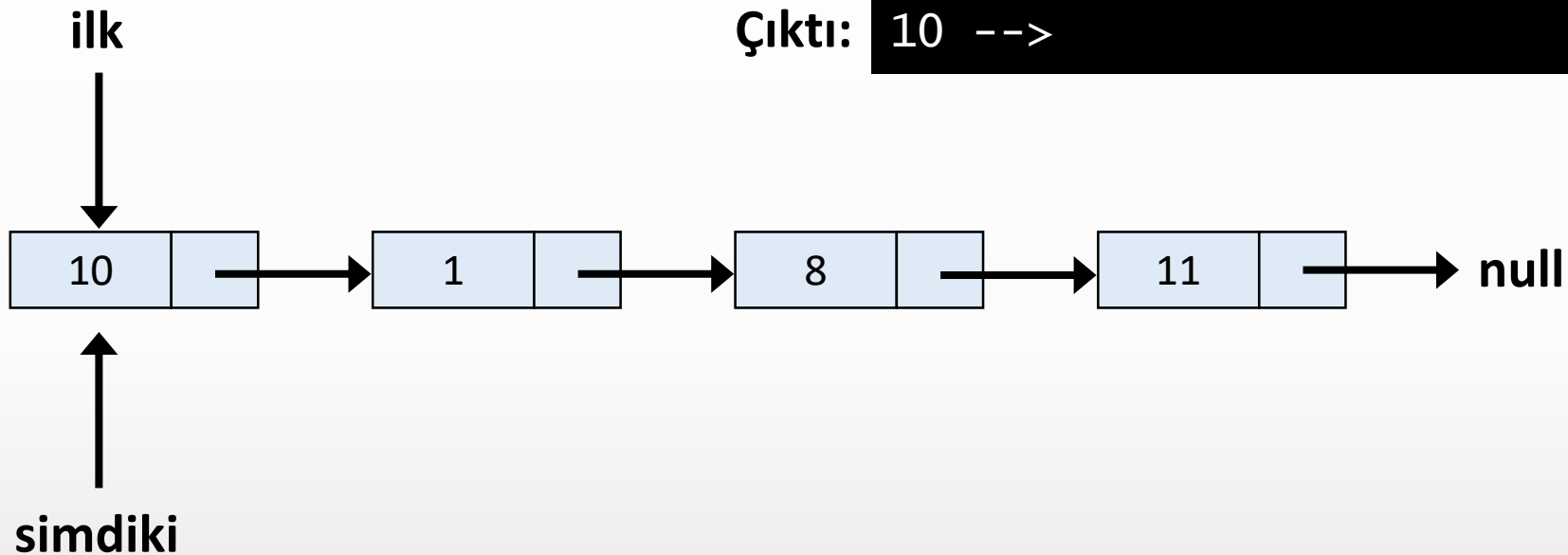

Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



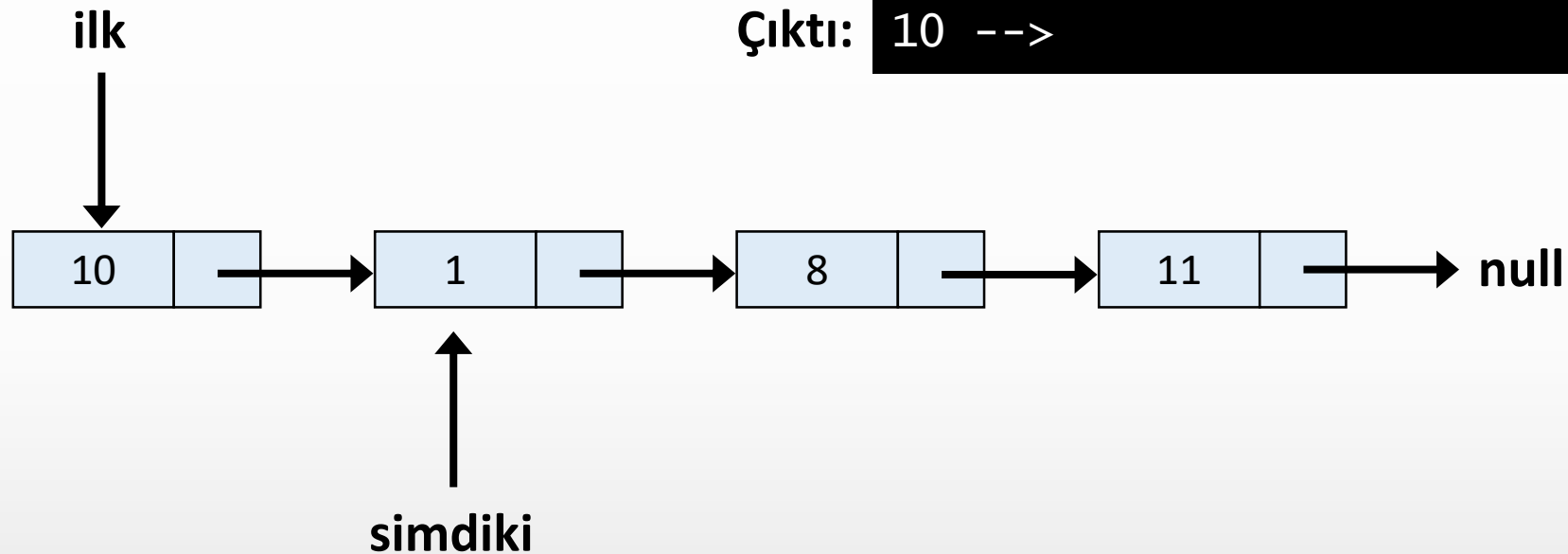
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



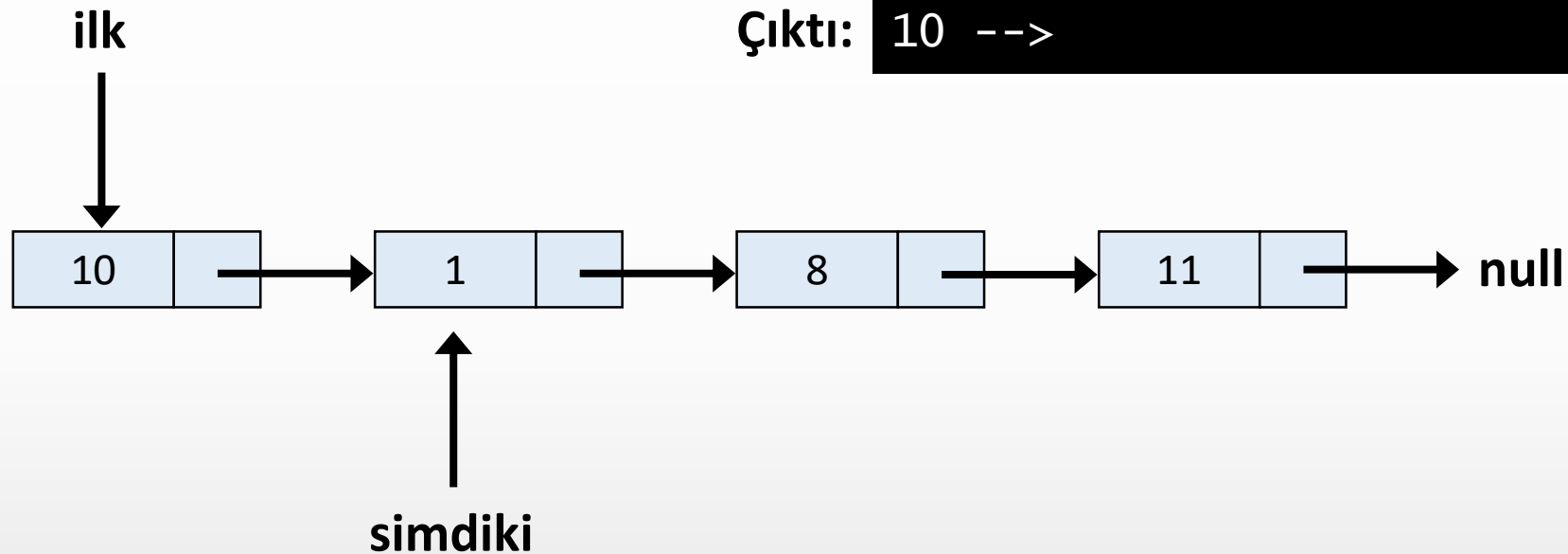
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;  
while(simdiki != null) {  
    System.out.print(simdiki.veri + " --> ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print("null");
```



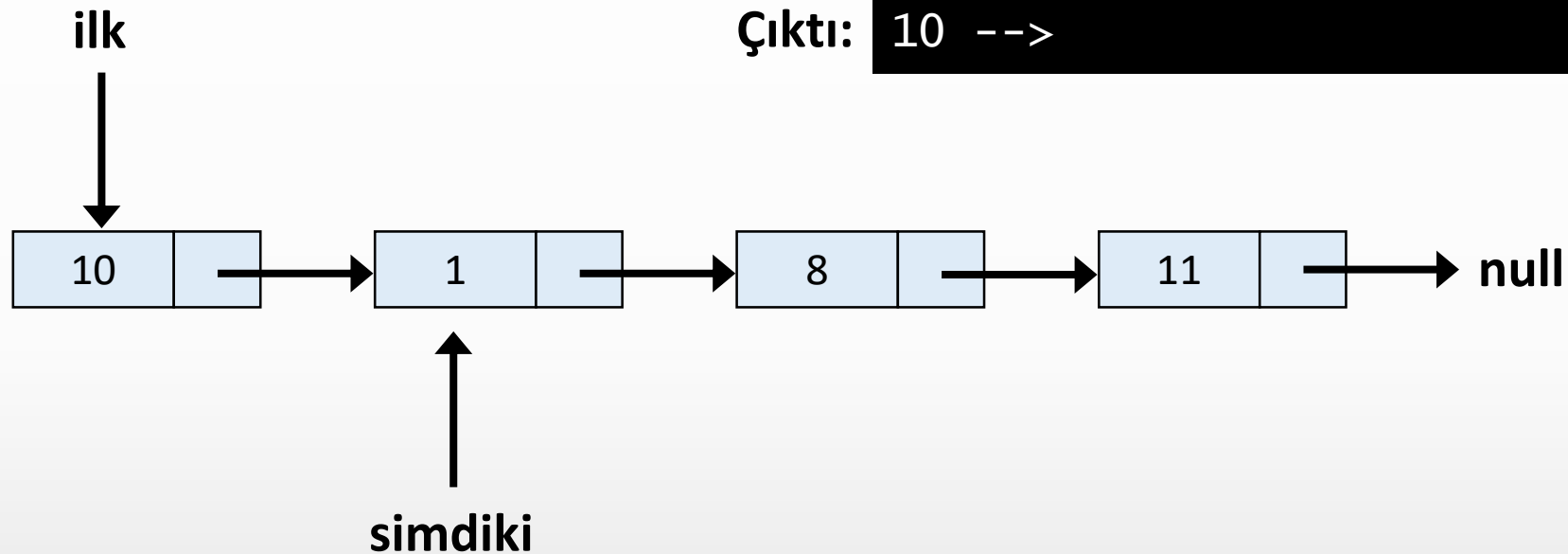
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;  
while(simdiki != null) {  
    System.out.print(simdiki.veri + " --> ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print("null");
```



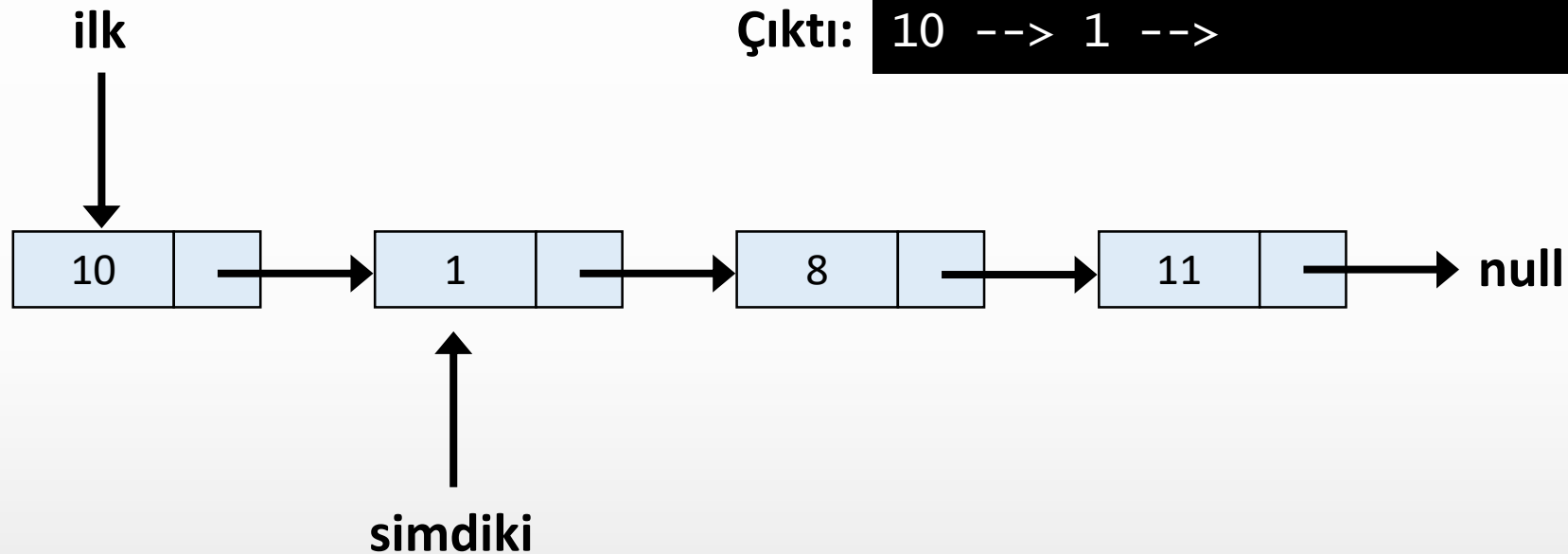
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



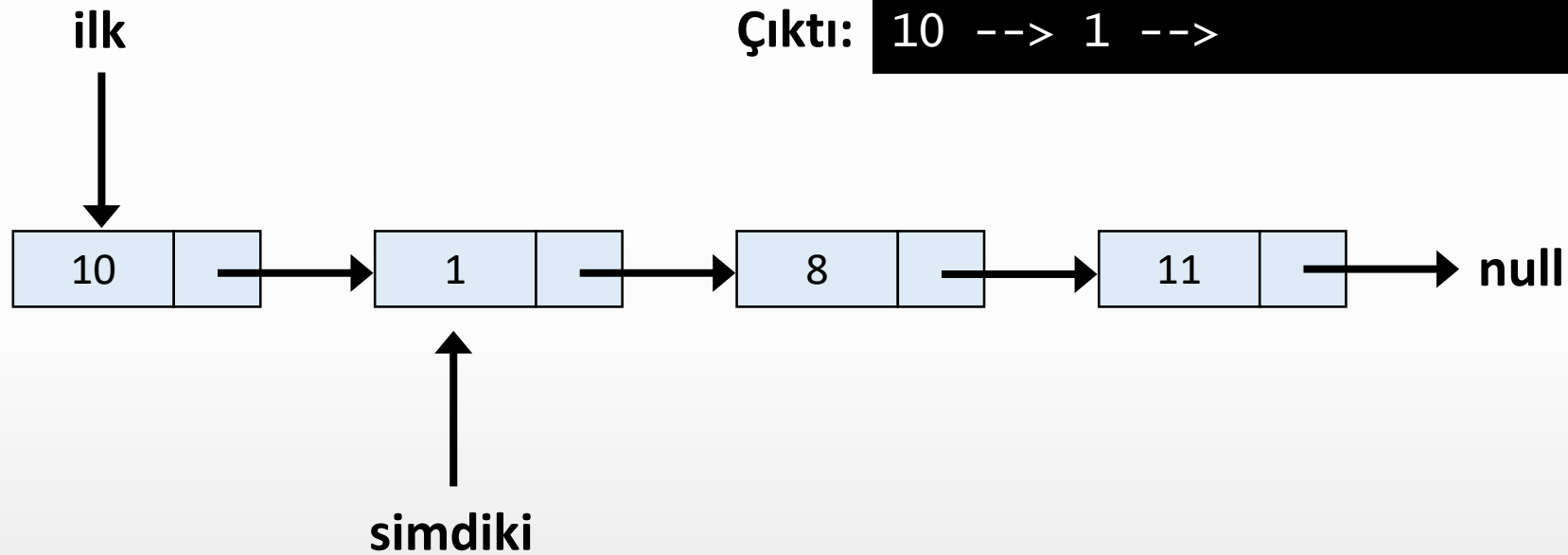
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;  
while(simdiki != null) {  
    System.out.print(simdiki.veri + " --> ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print("null");
```



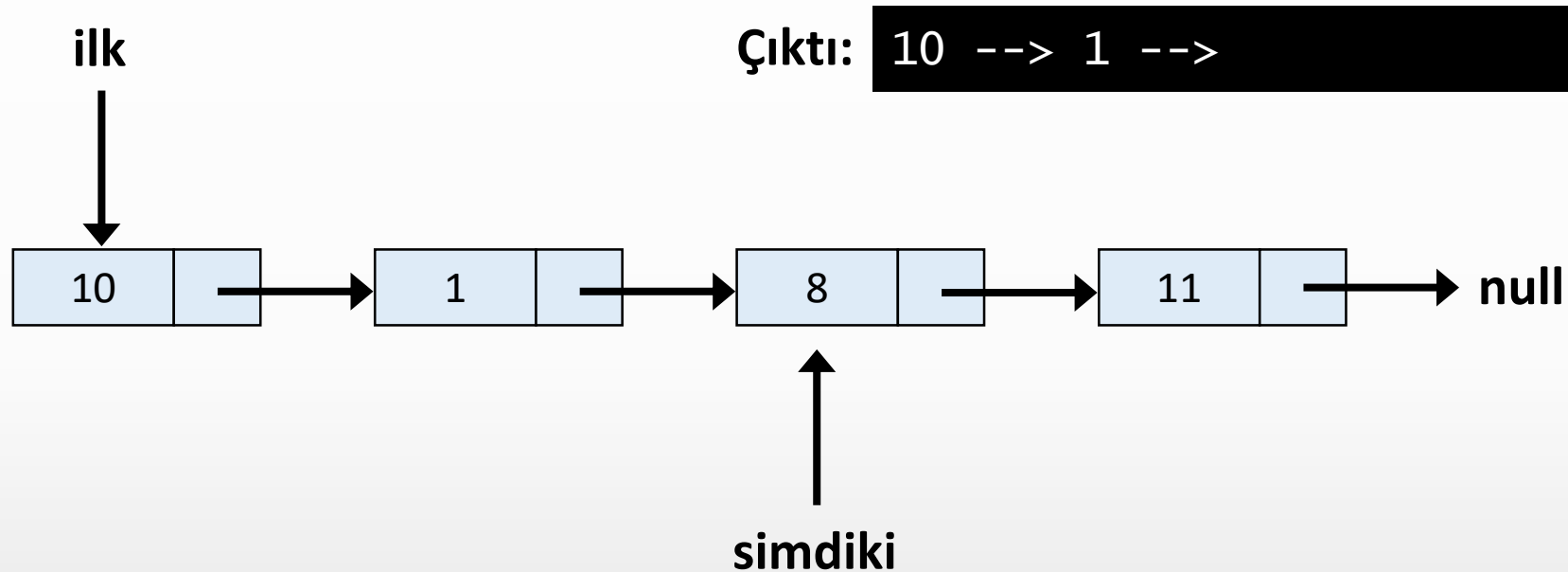
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;  
while(simdiki != null) {  
    System.out.print(simdiki.veri + " --> ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print("null");
```



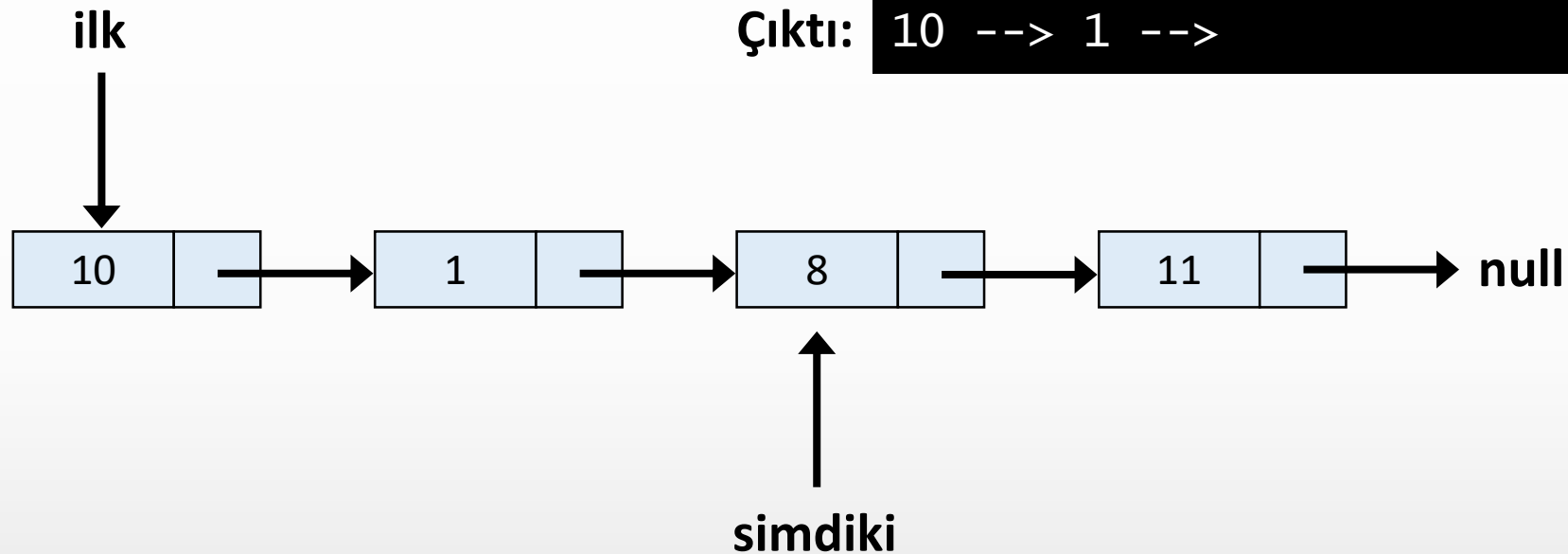
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



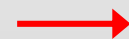
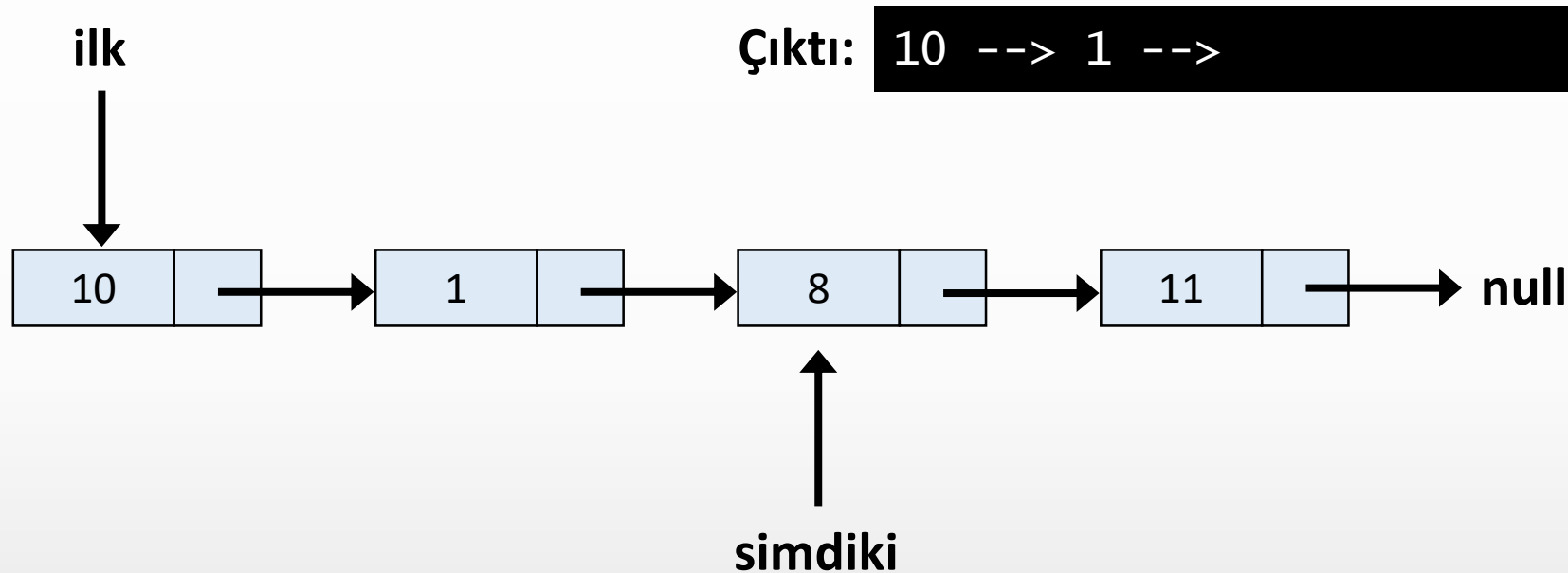

Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



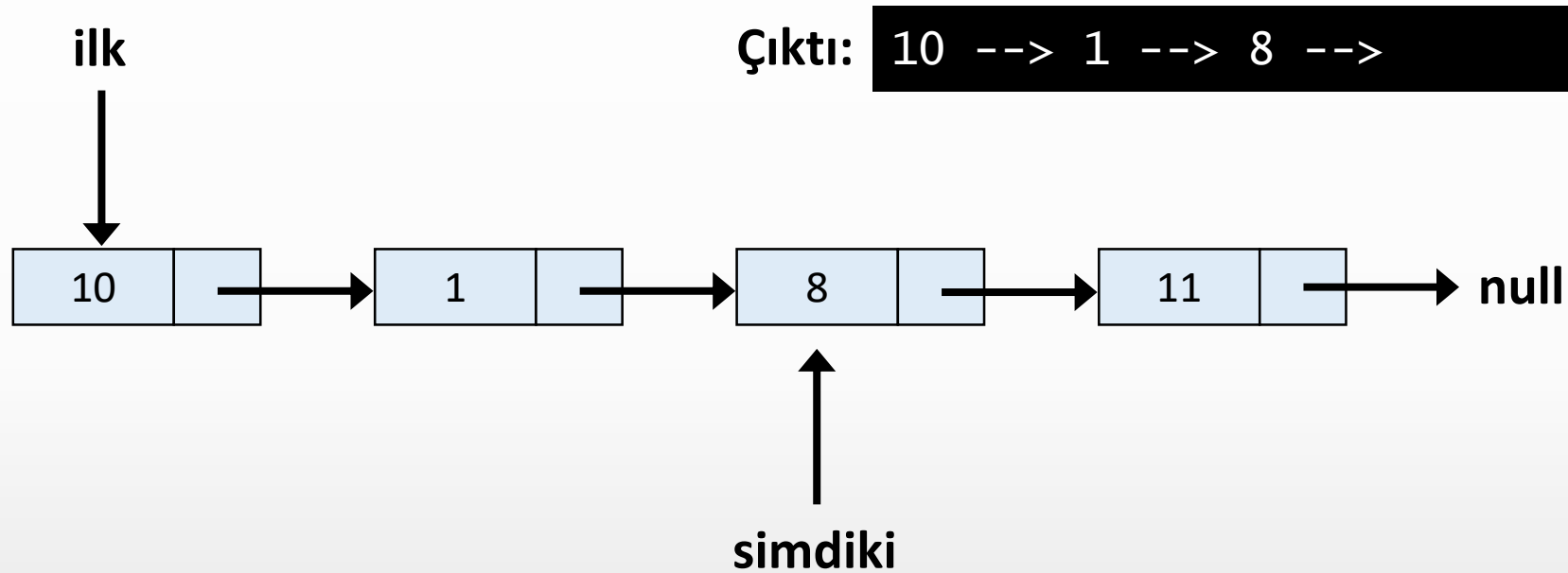
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



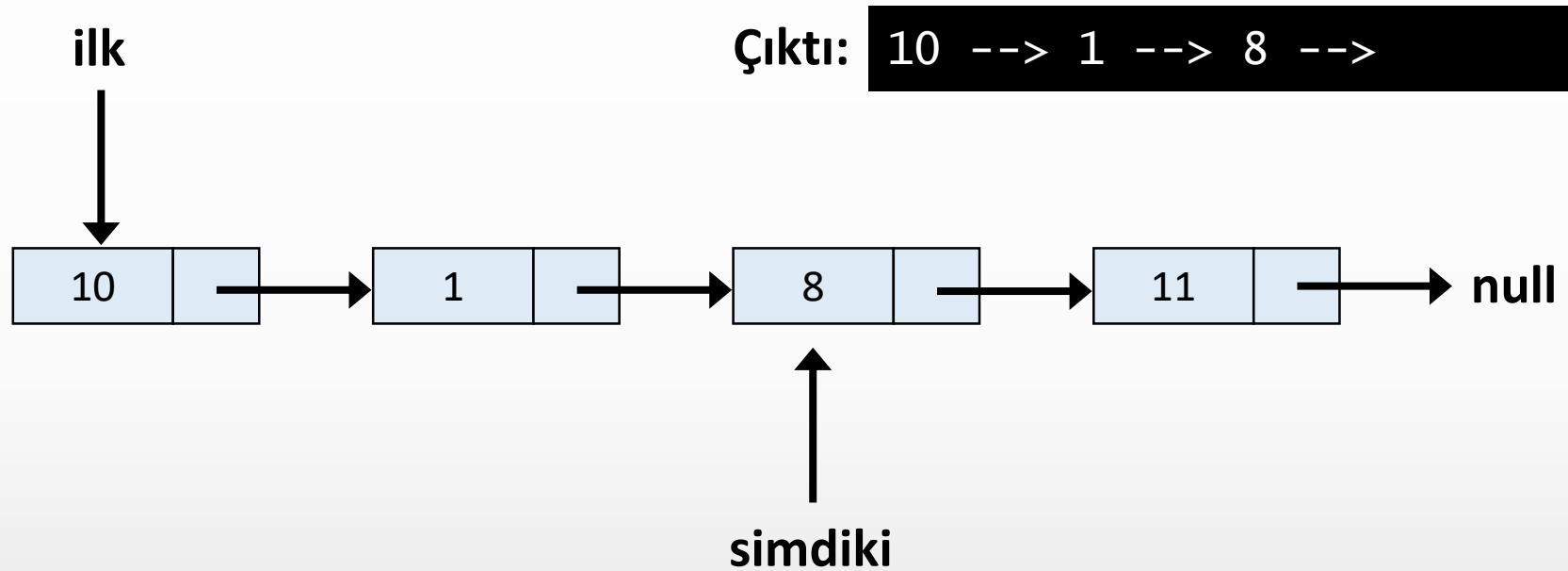
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



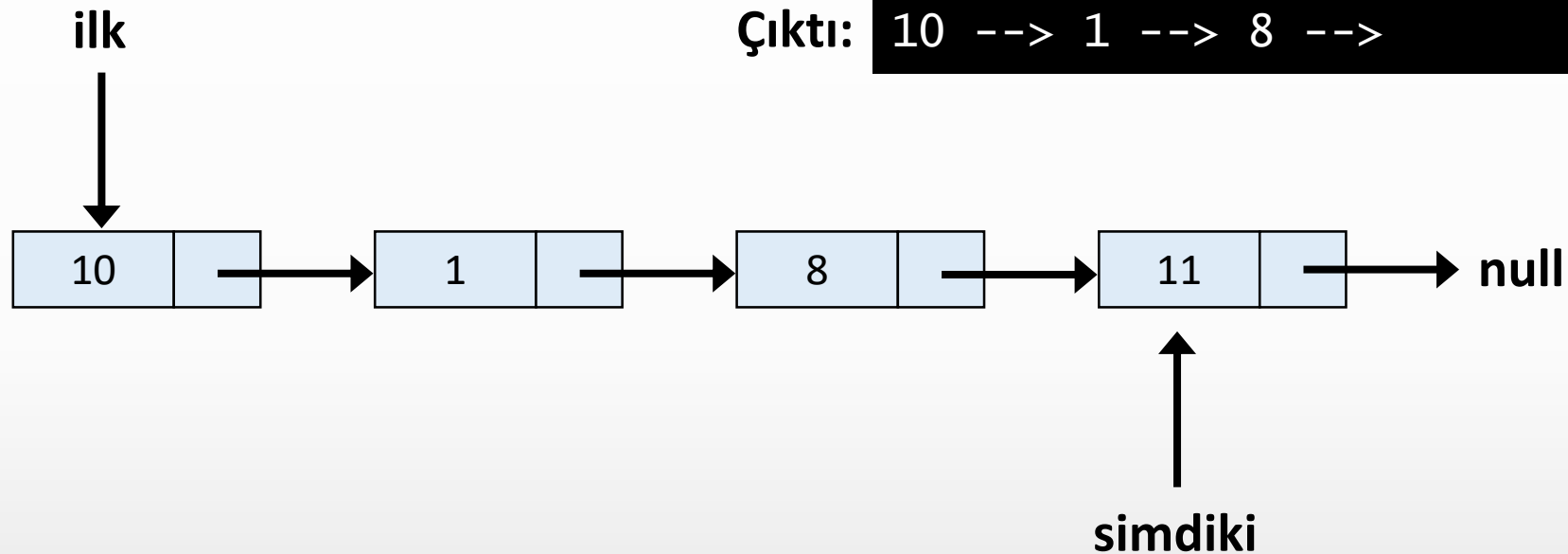
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



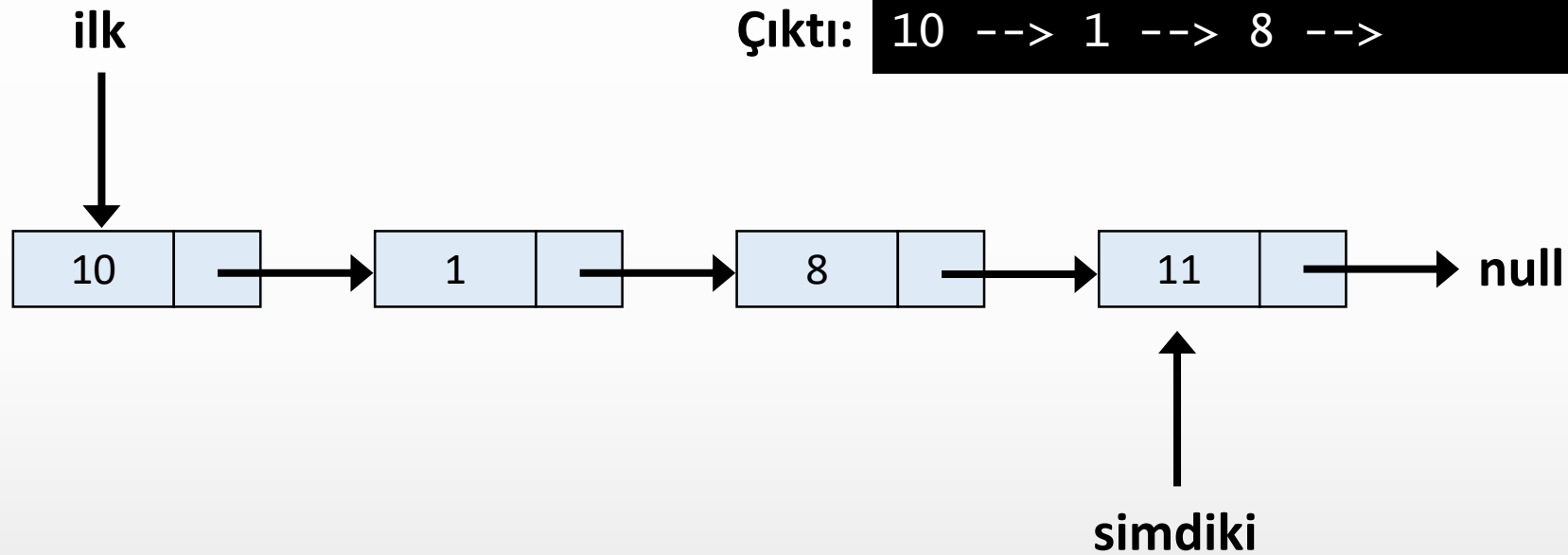
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



Tek Yönlü Bağlı Liste Elemanlarını Yazdırma

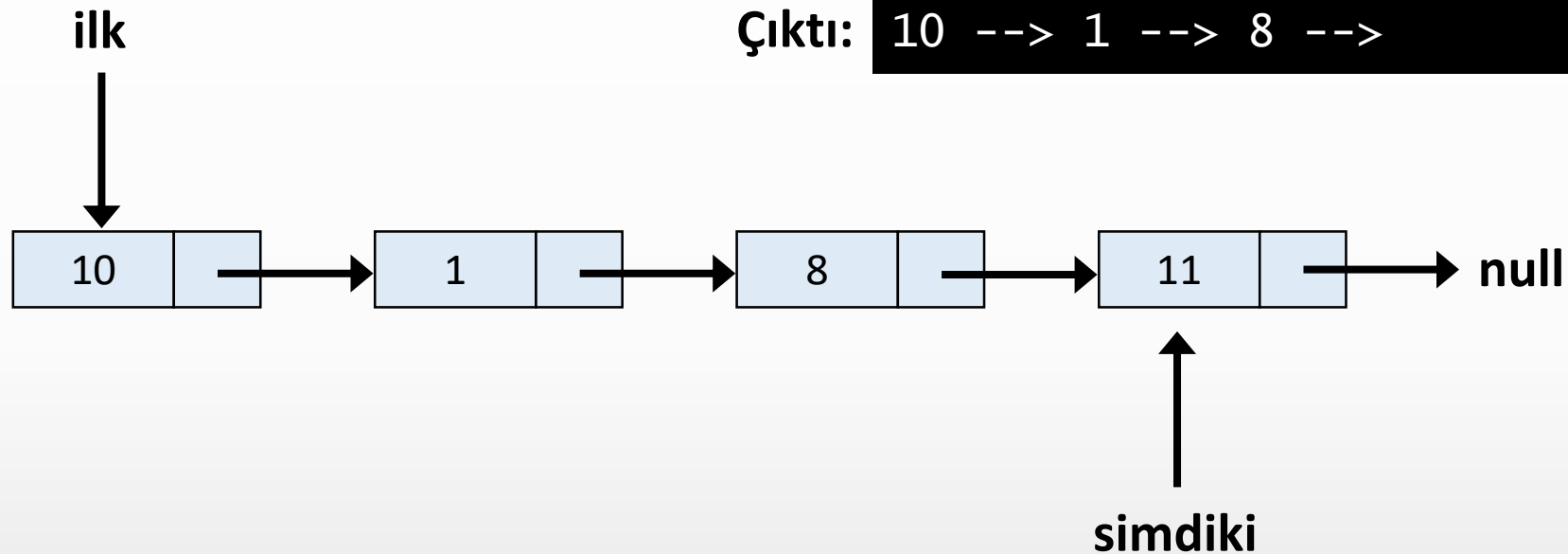


→

```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



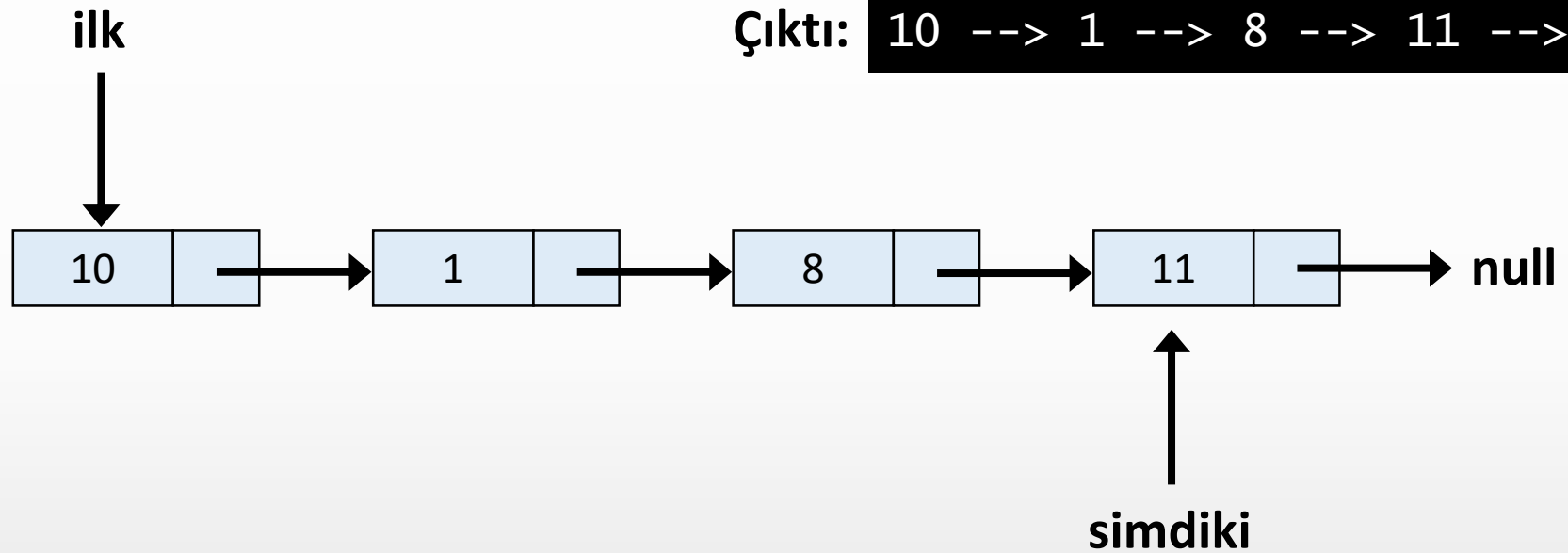
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



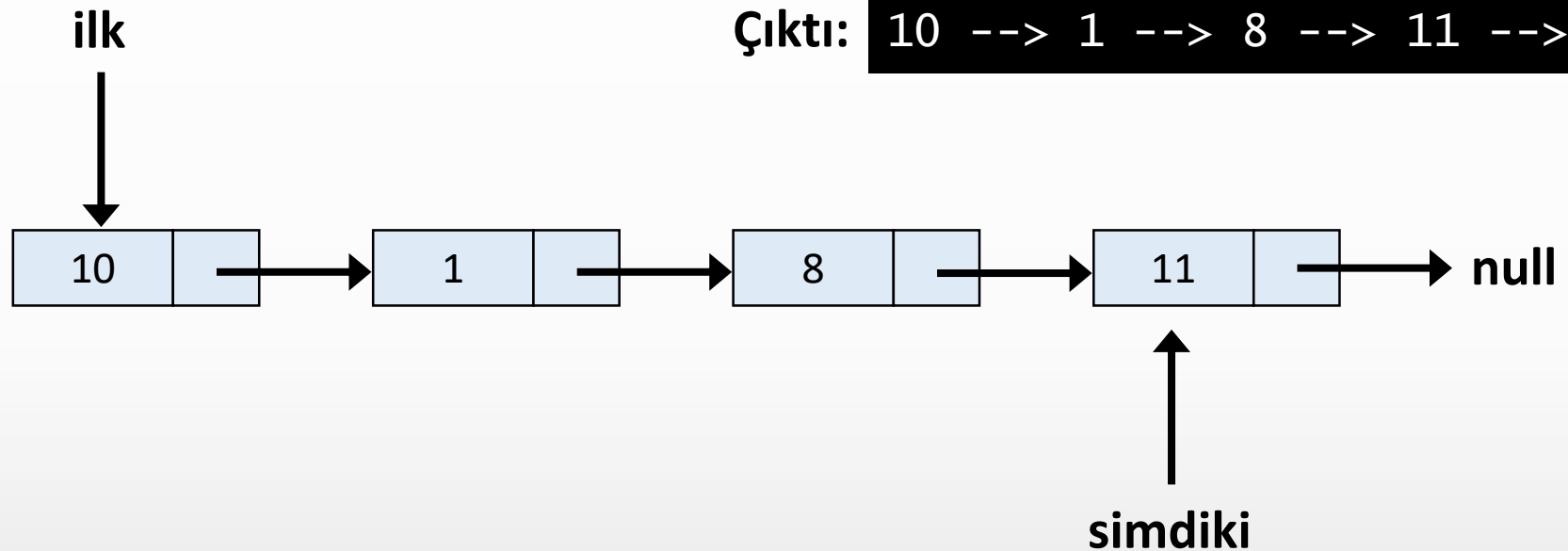
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



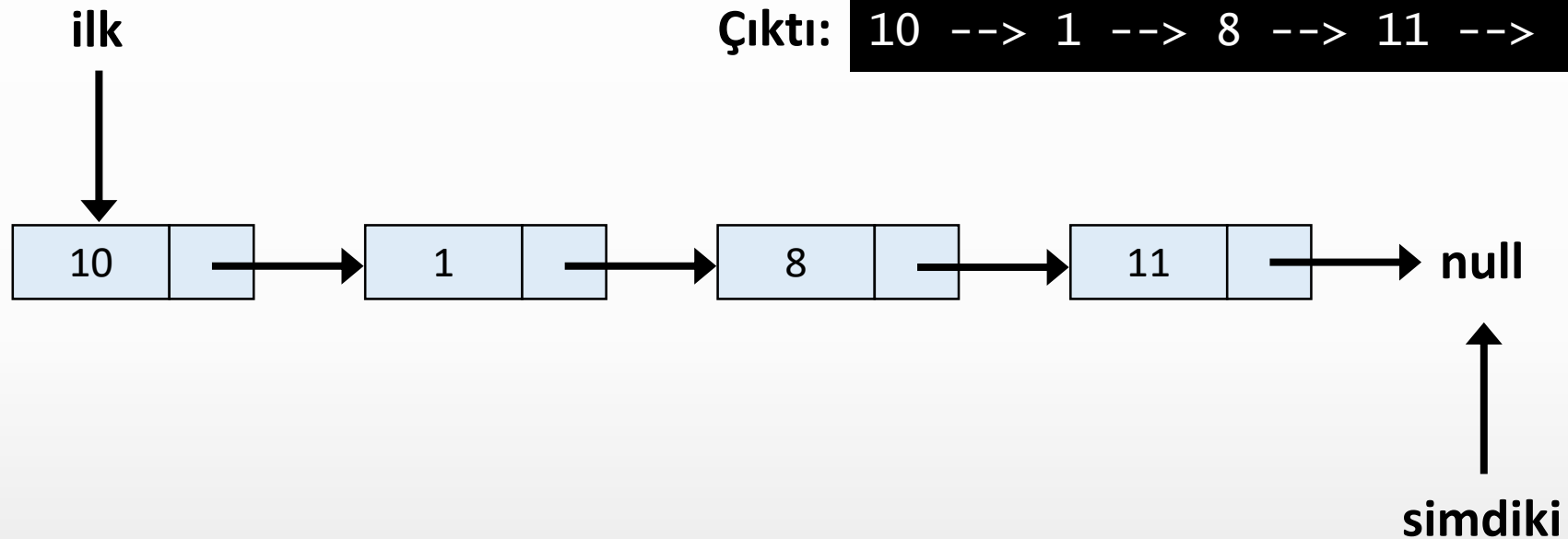

Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



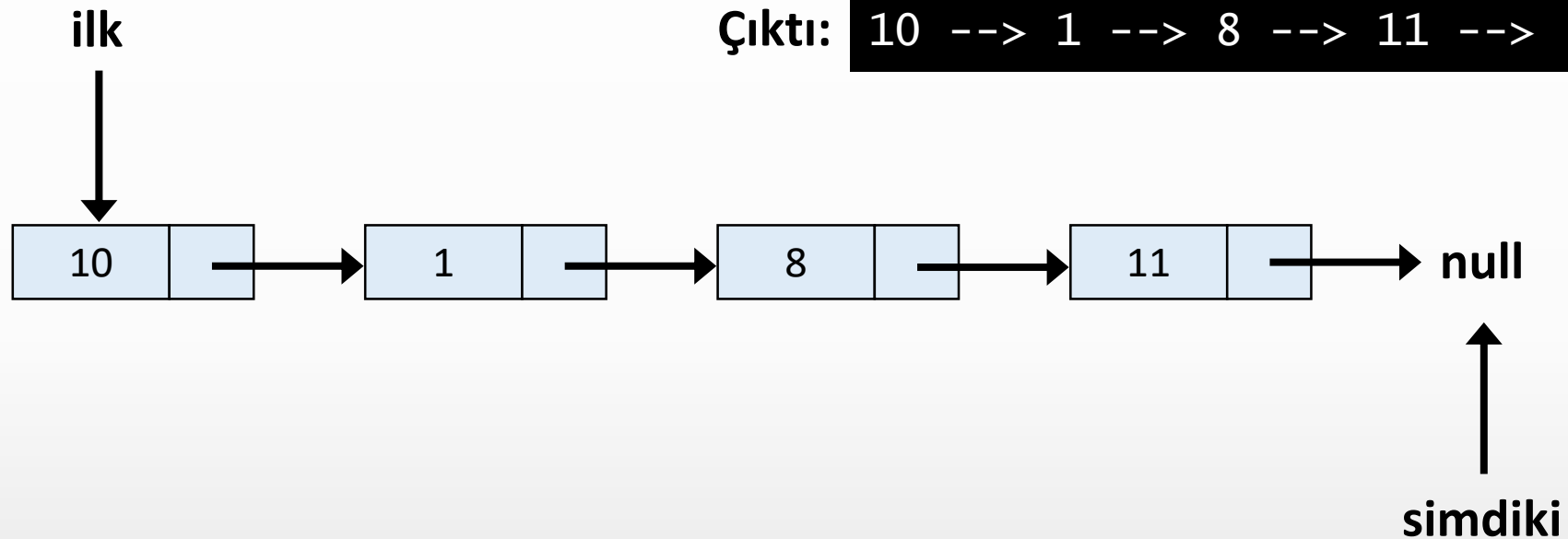
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



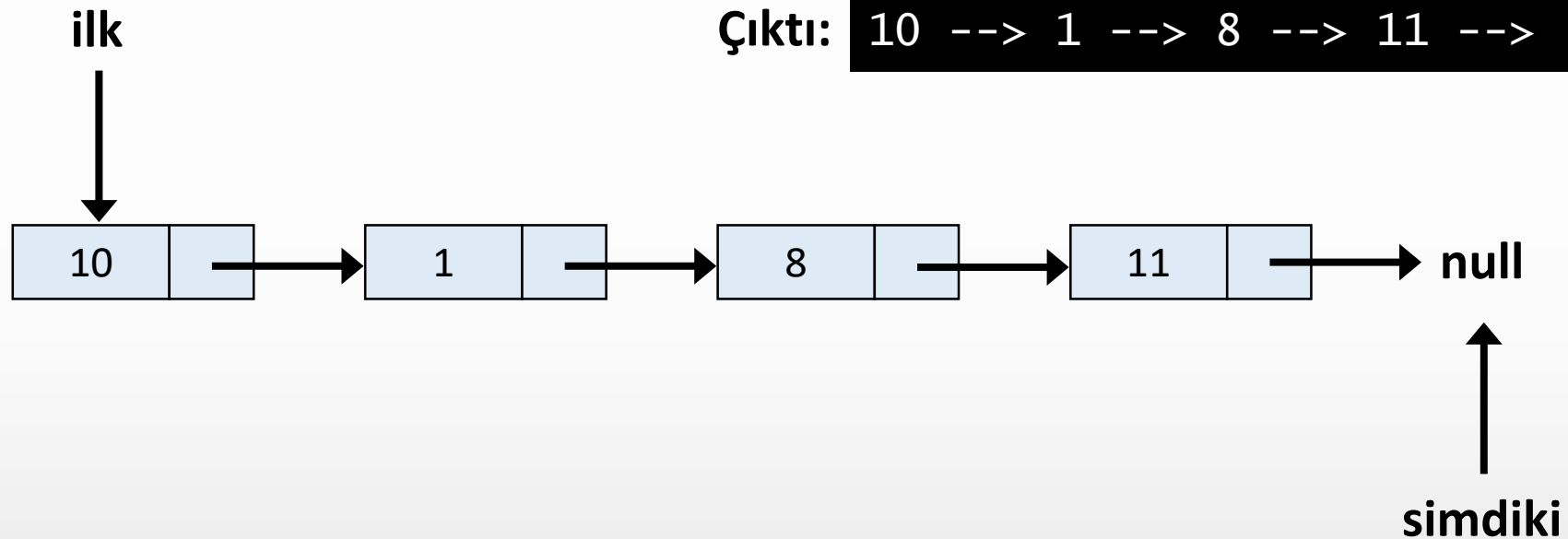
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



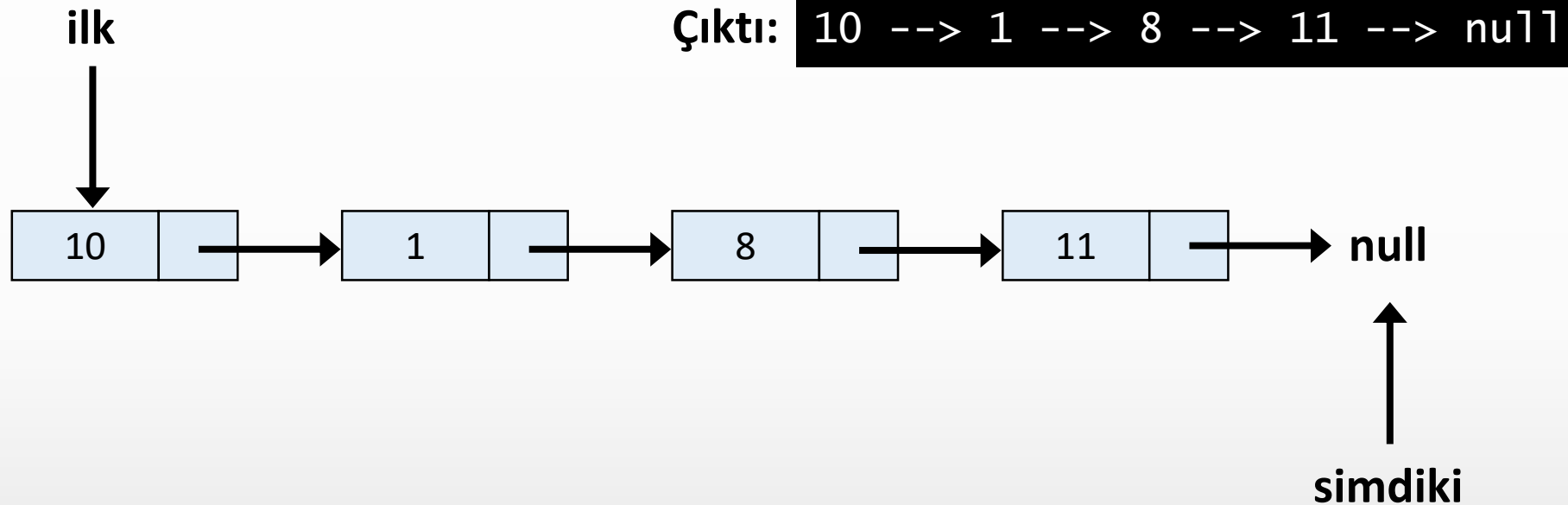
Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```



Tek Yönlü Bağlı Liste Elemanlarını Yazdırma



```
Dugum simdiki = ilk;
while(simdiki != null) {
    System.out.print(simdiki.veri + " --> ");
    simdiki = simdiki.sonraki;
}
System.out.print("null");
```

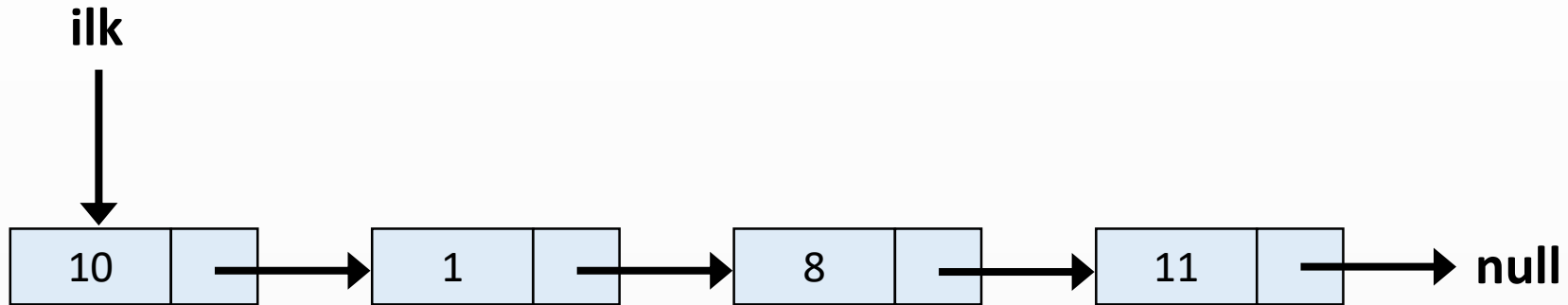


Liste Uzunluğunu Hesaplama

```
int listeUzunlugu() {  
    int uzunluk = 0;  
    Dugum gecici = bas;  
    while (gecici != null) {  
        uzunluk++;  
        gecici = gecici.sonraki;  
    }  
    System.out.println("Bağlı Liste Uzunluğu: " + uzunluk);  
    return uzunluk;  
}
```



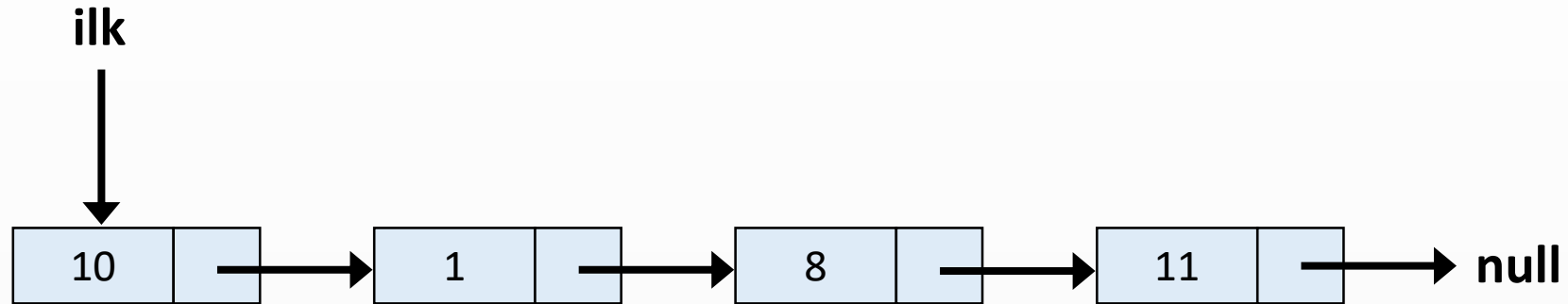
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



Tek Yönlü Bağlı Listenin Uzunluğunu Bulma

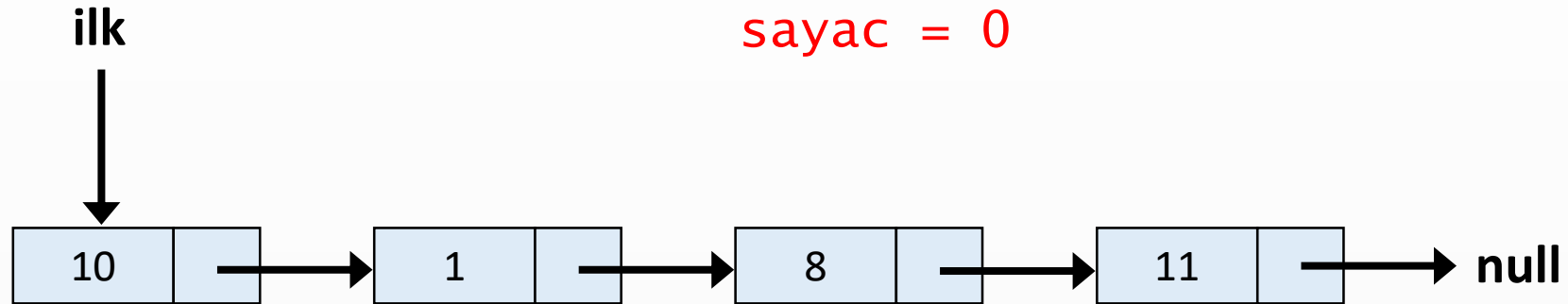


→

```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```




Tek Yönlü Bağlı Listenin Uzunluğunu Bulma

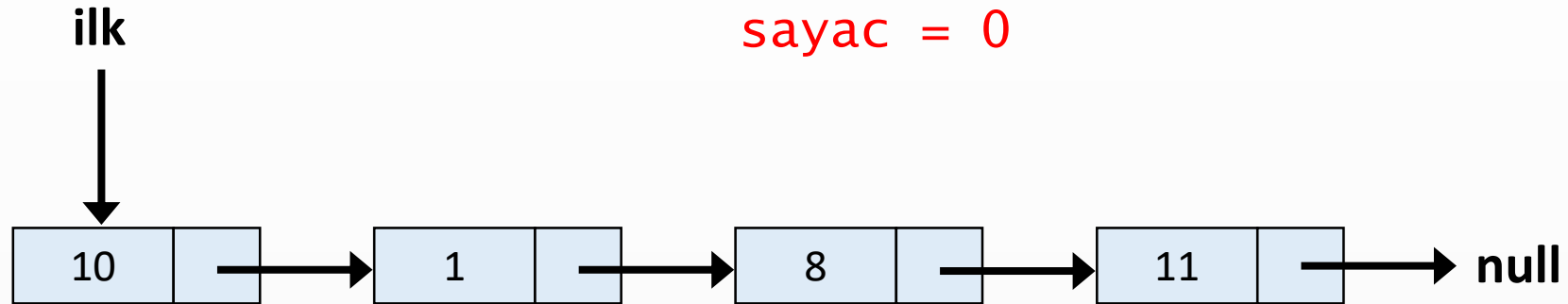


→

```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



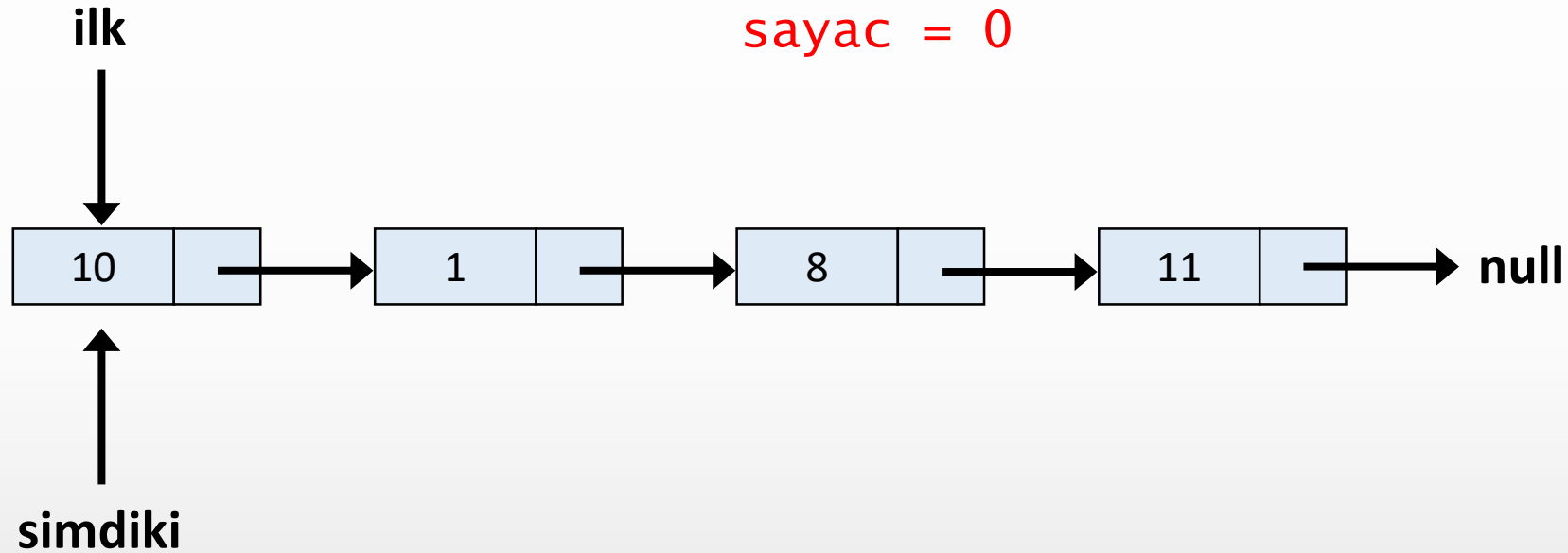
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
→ int sayac = 0;  
   Dugum simdiki = ilk;  
   while(simdiki != null) {  
       sayac++;  
       simdiki = simdiki.sonraki;  
   }  
   return sayac;
```



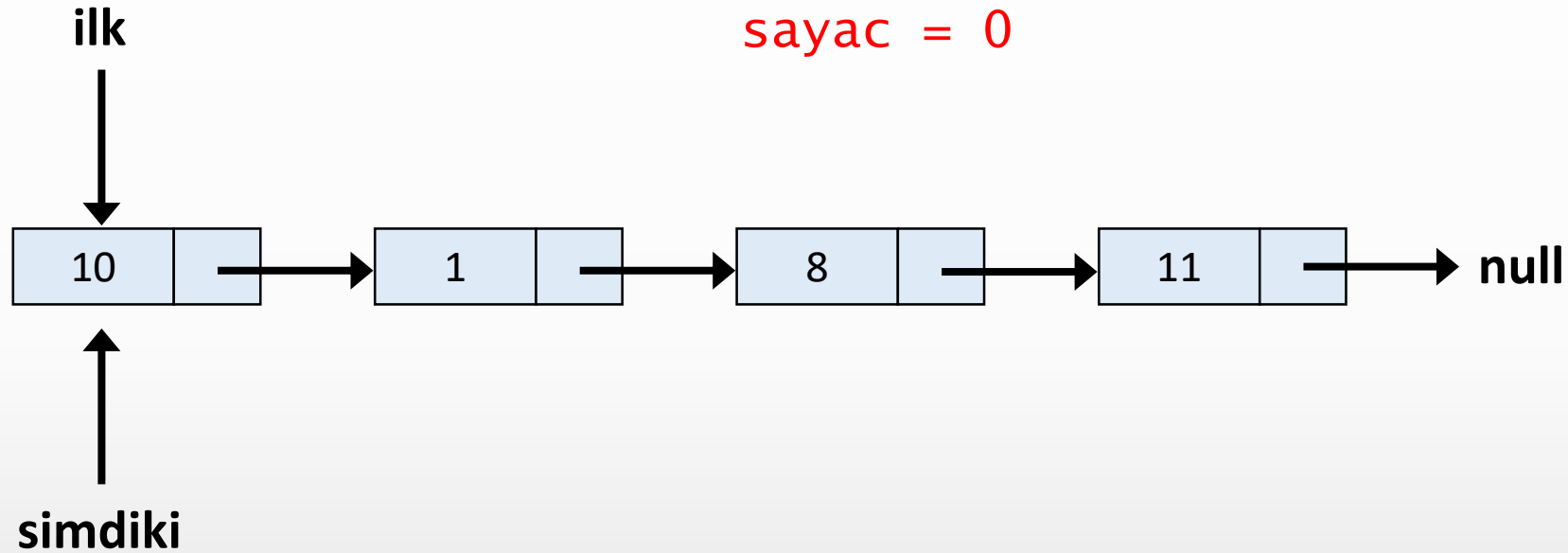
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
→ int sayac = 0;
   Dugum simdiki = ilk;
   while(simdiki != null) {
       sayac++;
       simdiki = simdiki.sonraki;
   }
   return sayac;
```



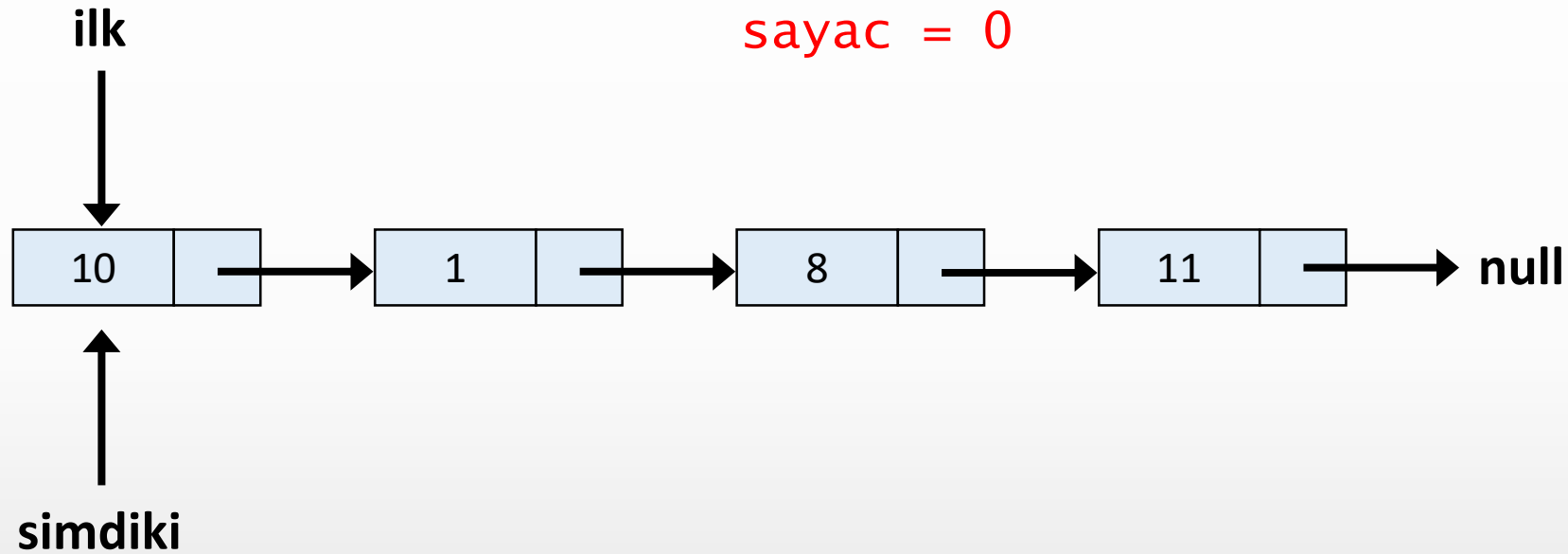
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



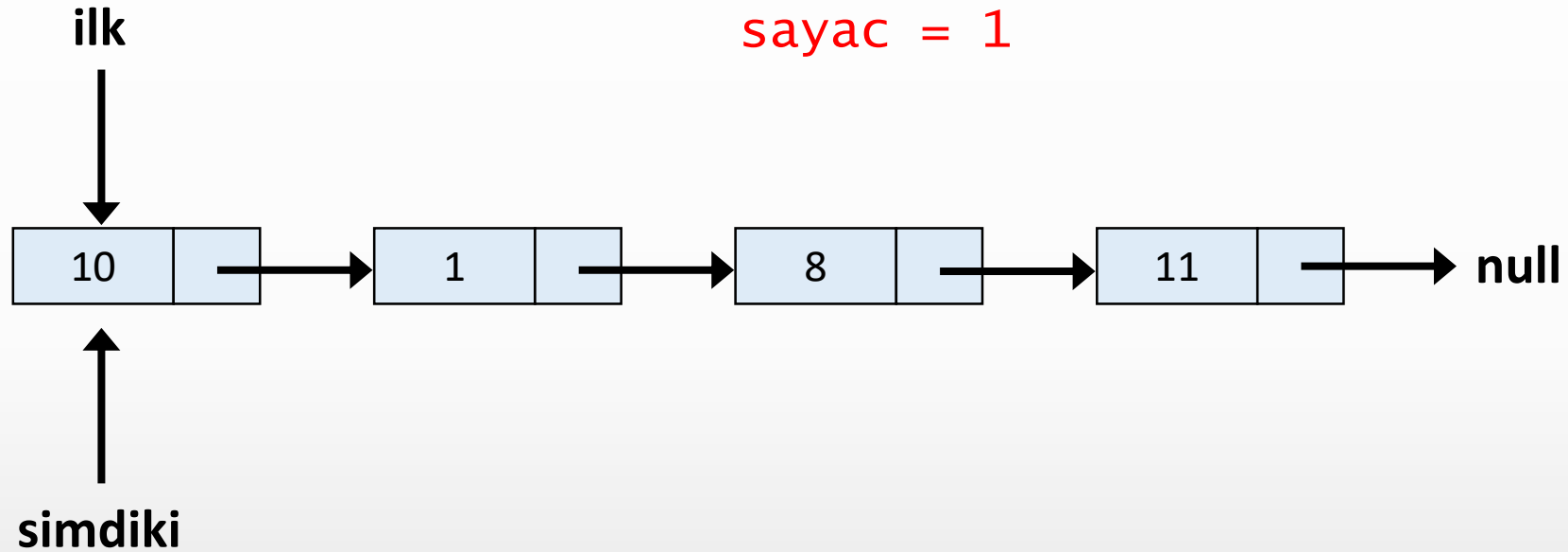
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



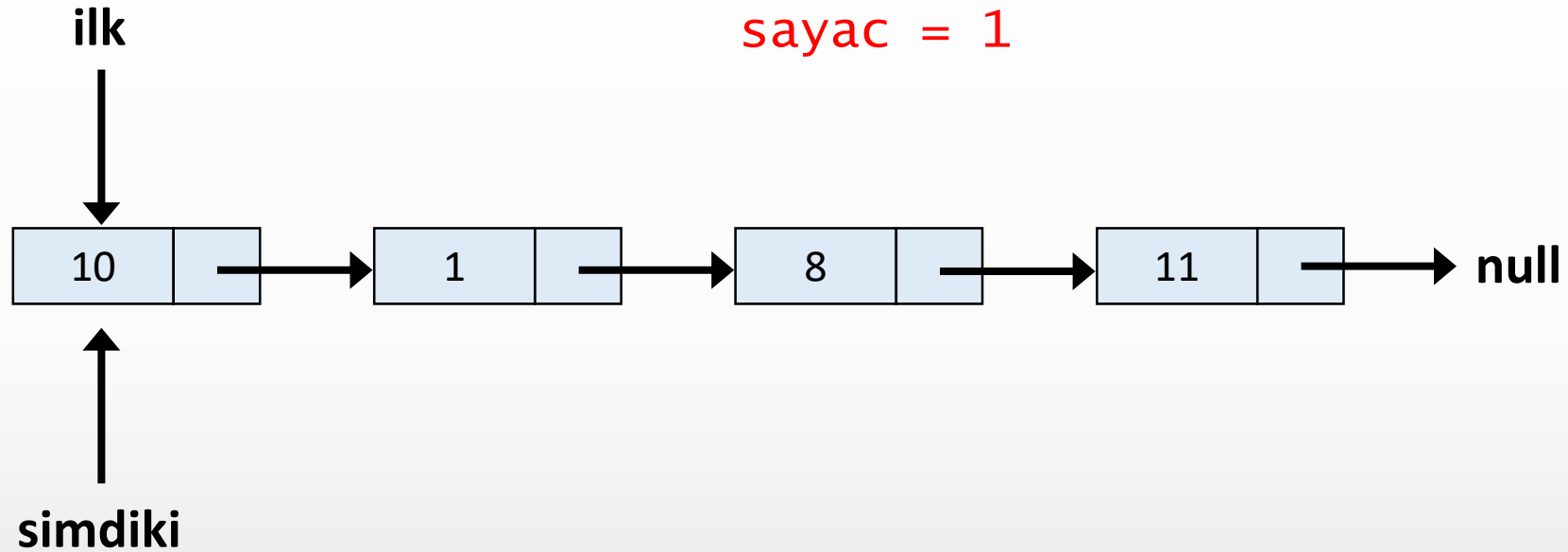
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



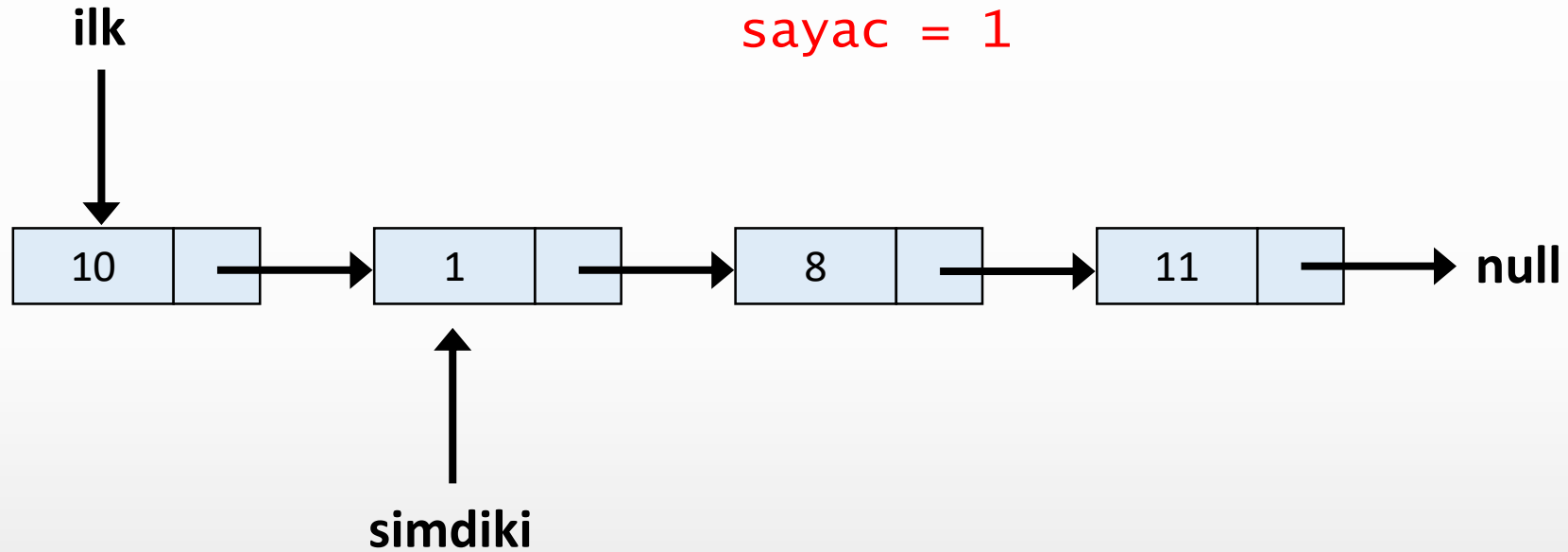
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



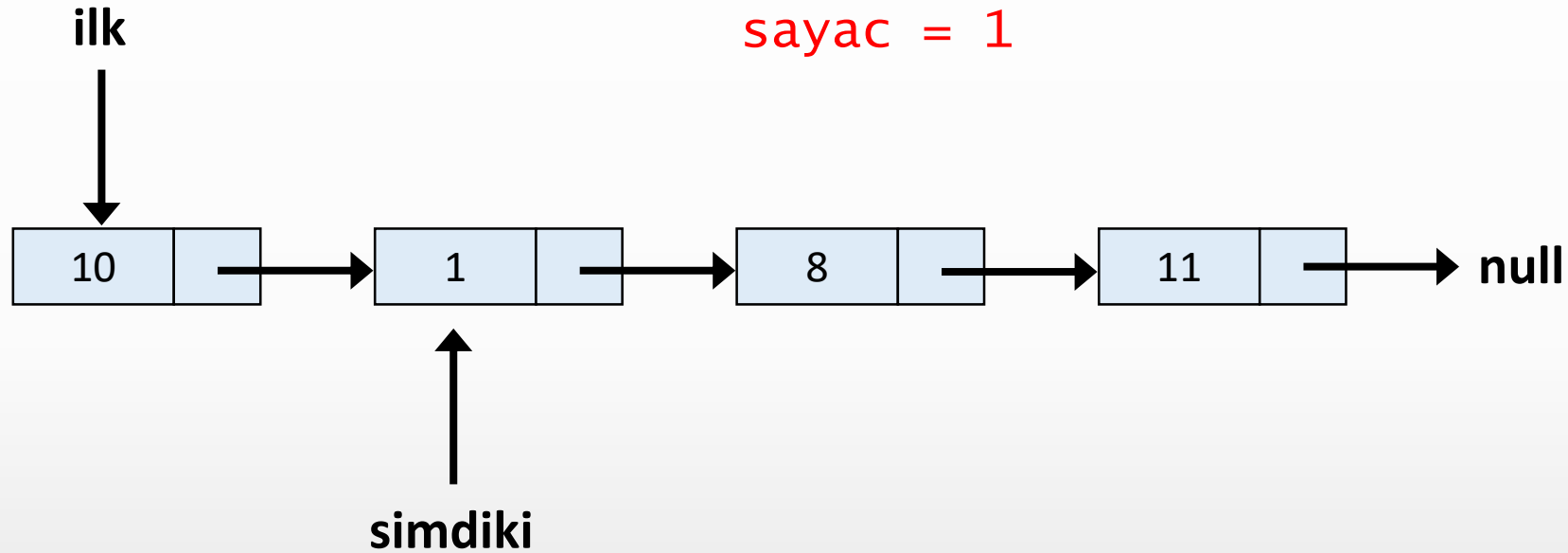
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



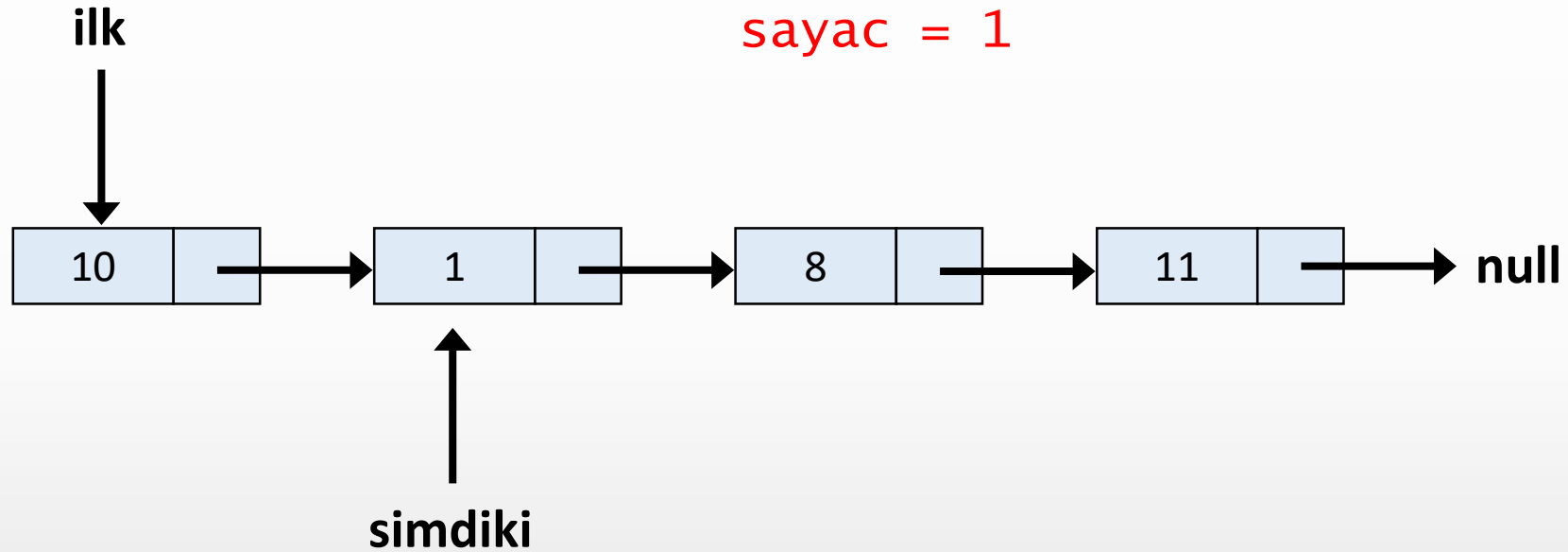

Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
→ while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



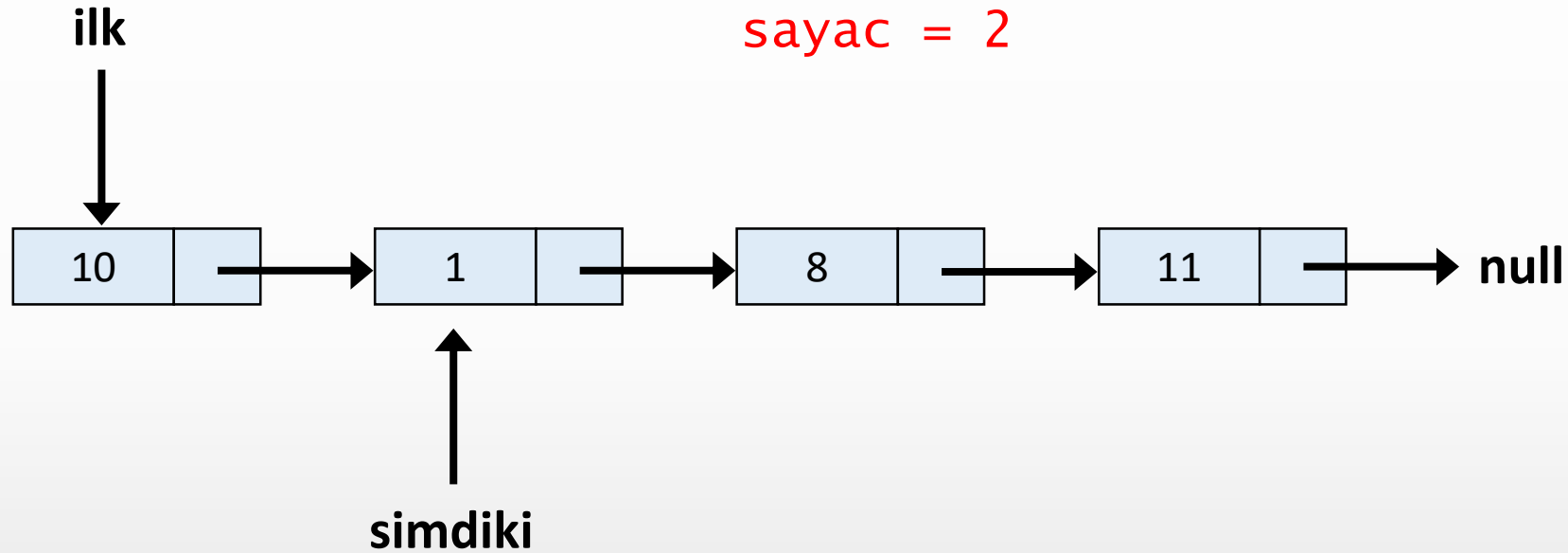
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



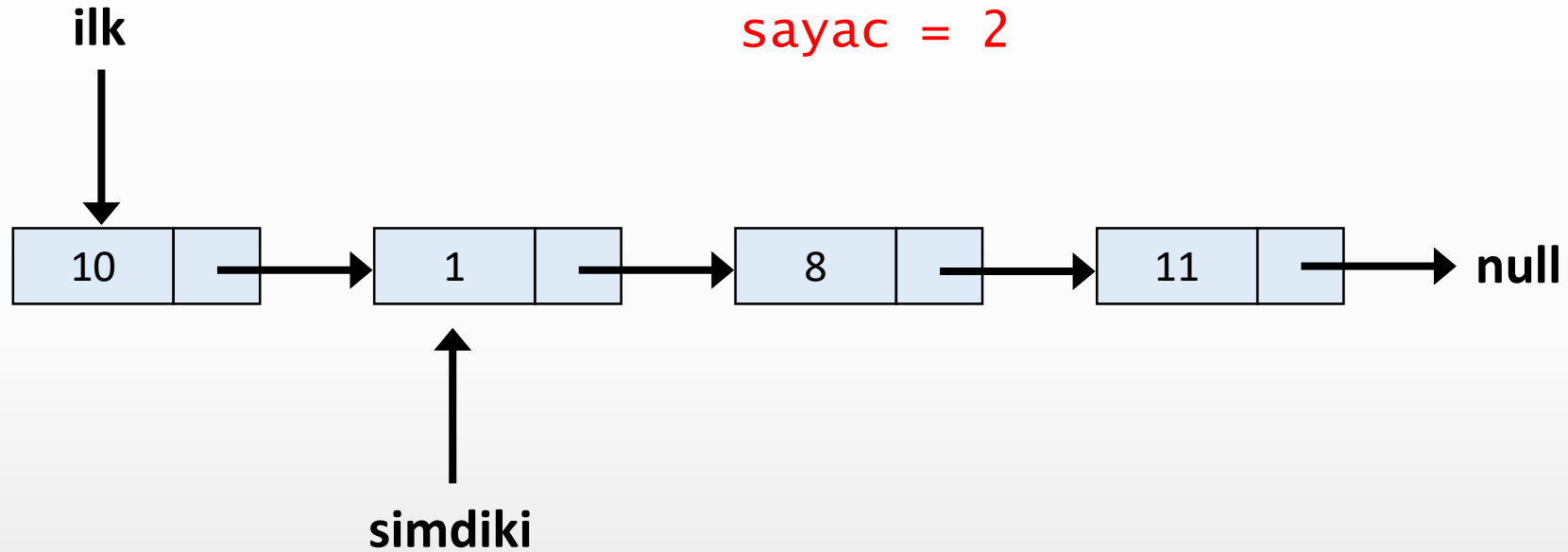
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



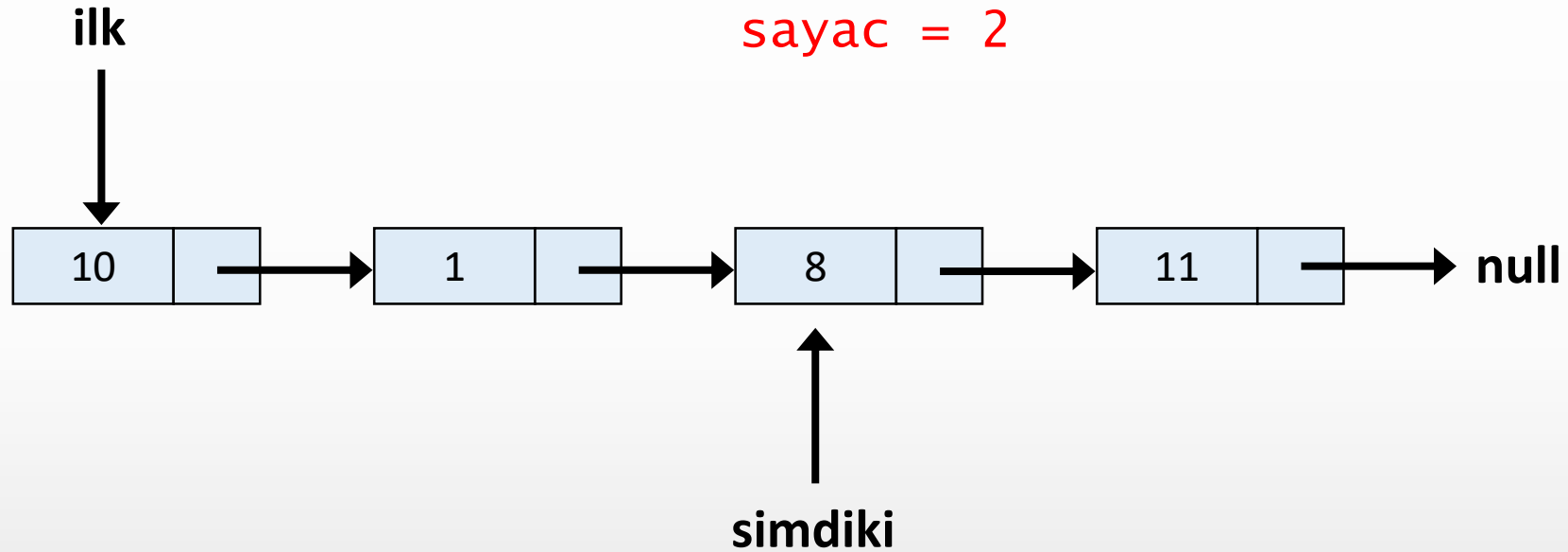
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



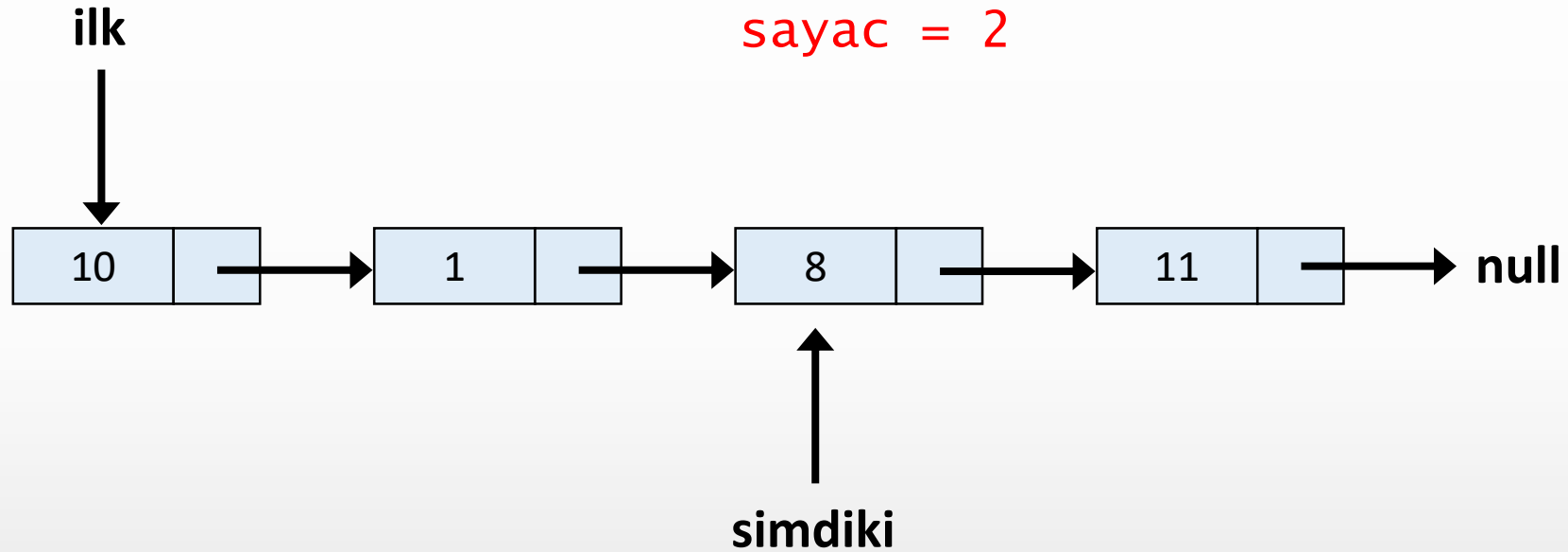
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



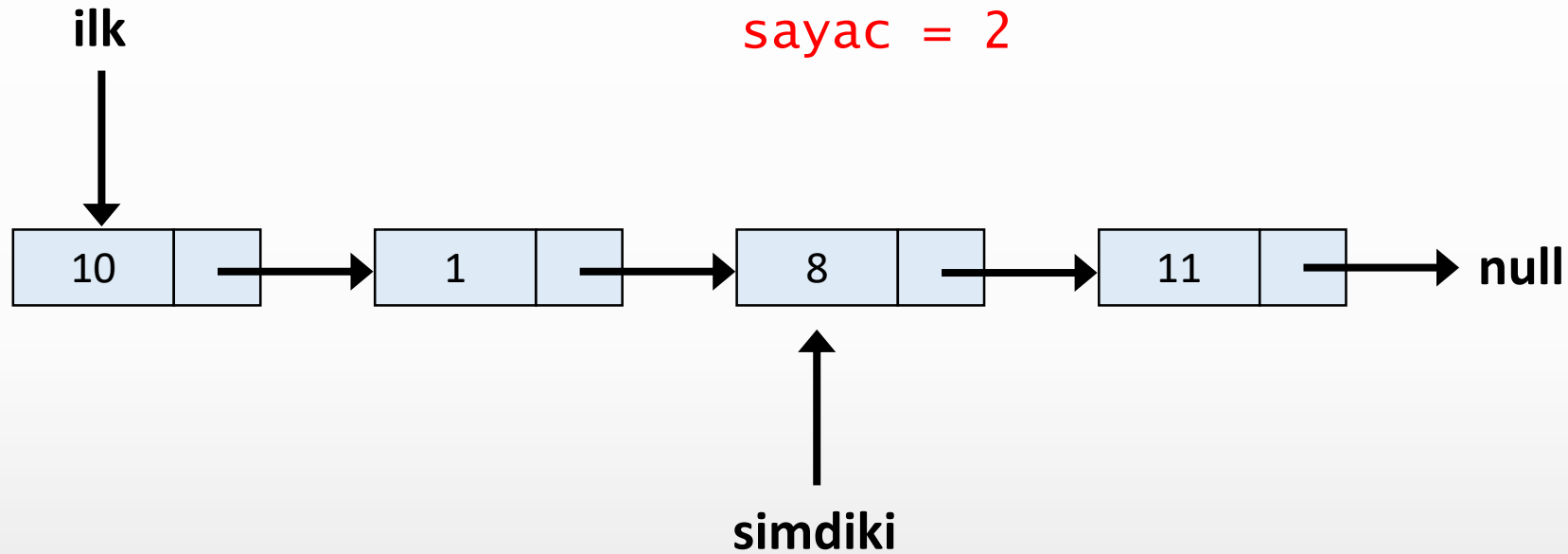
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
→ while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



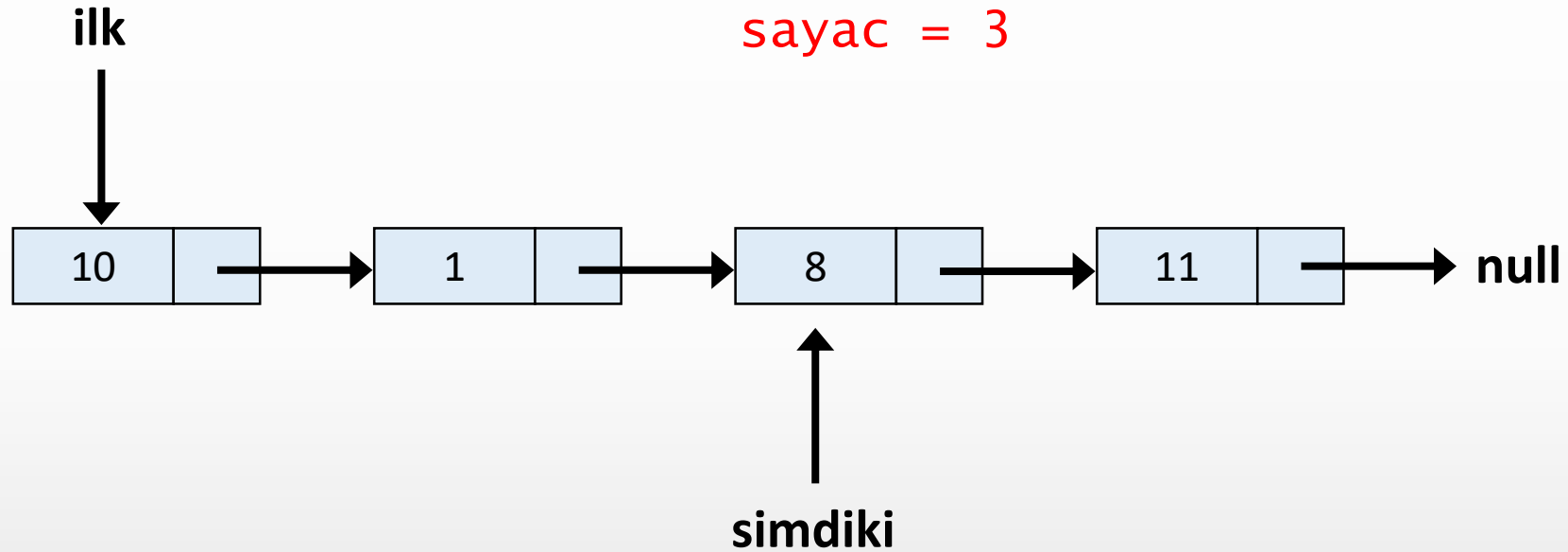
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



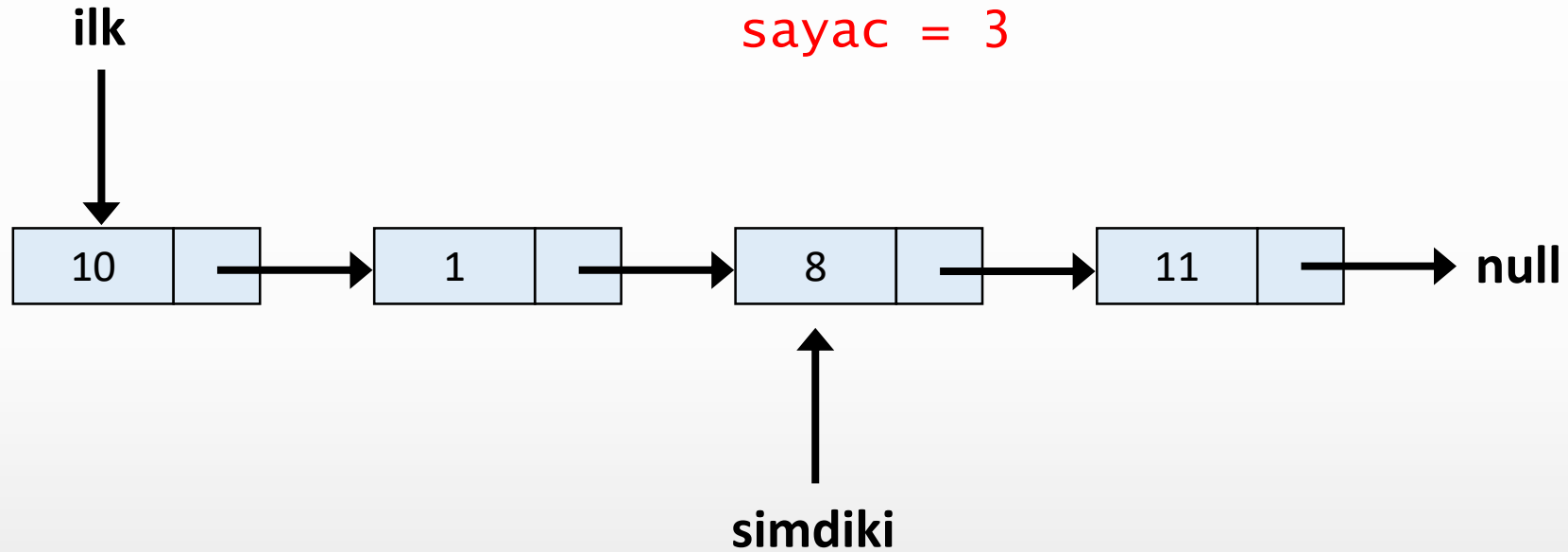
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



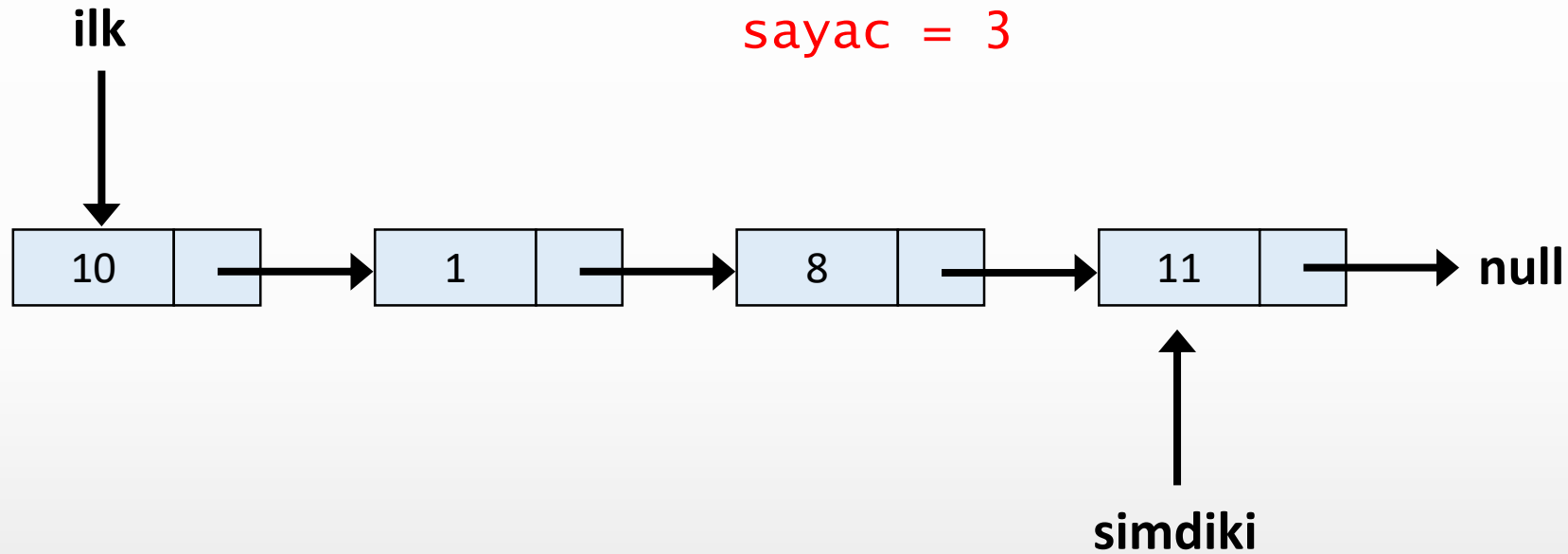

Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



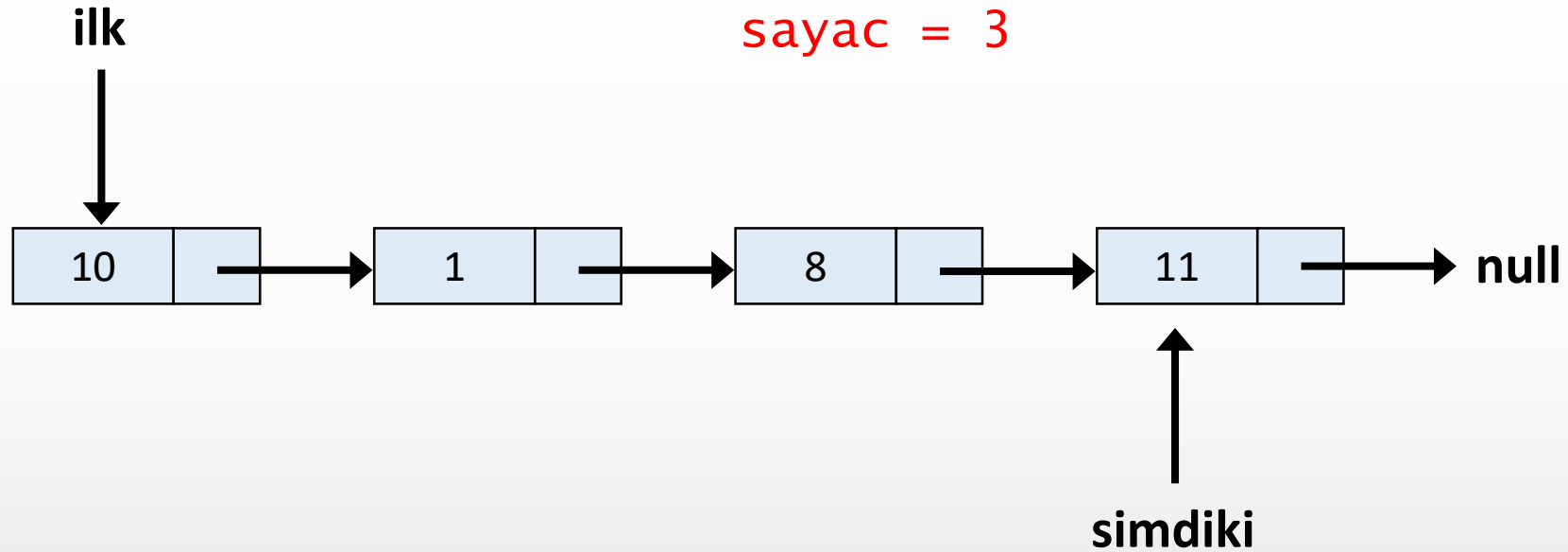
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



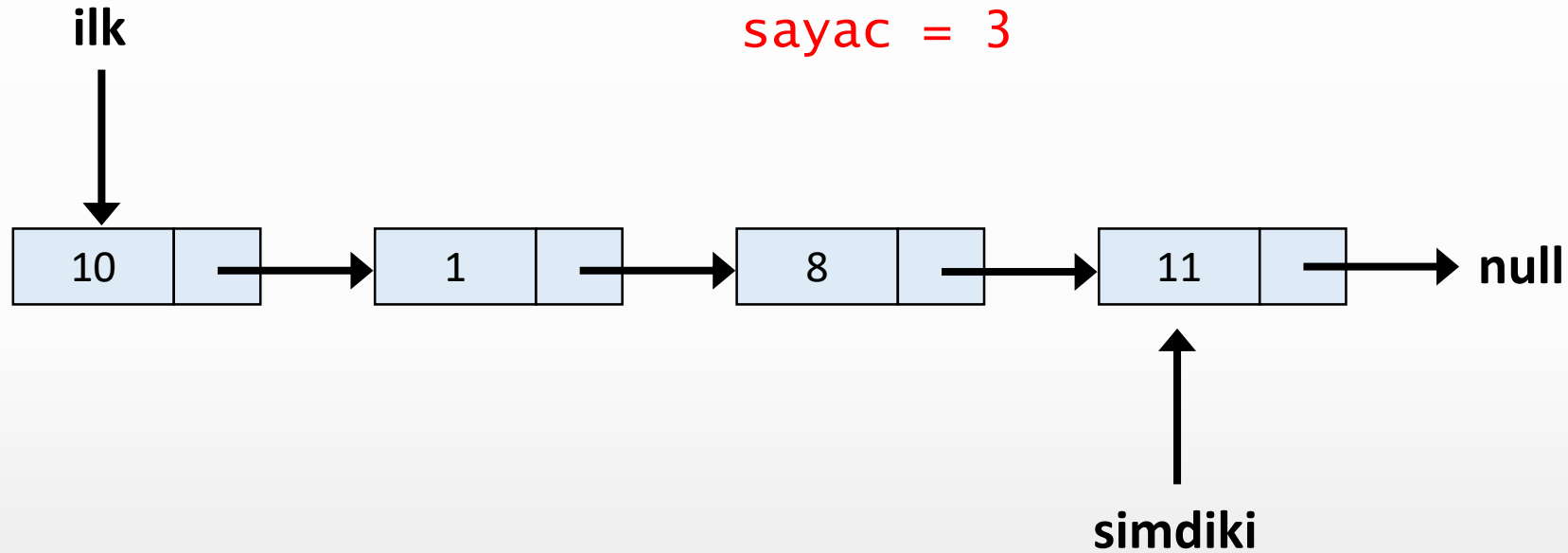
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
→ while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



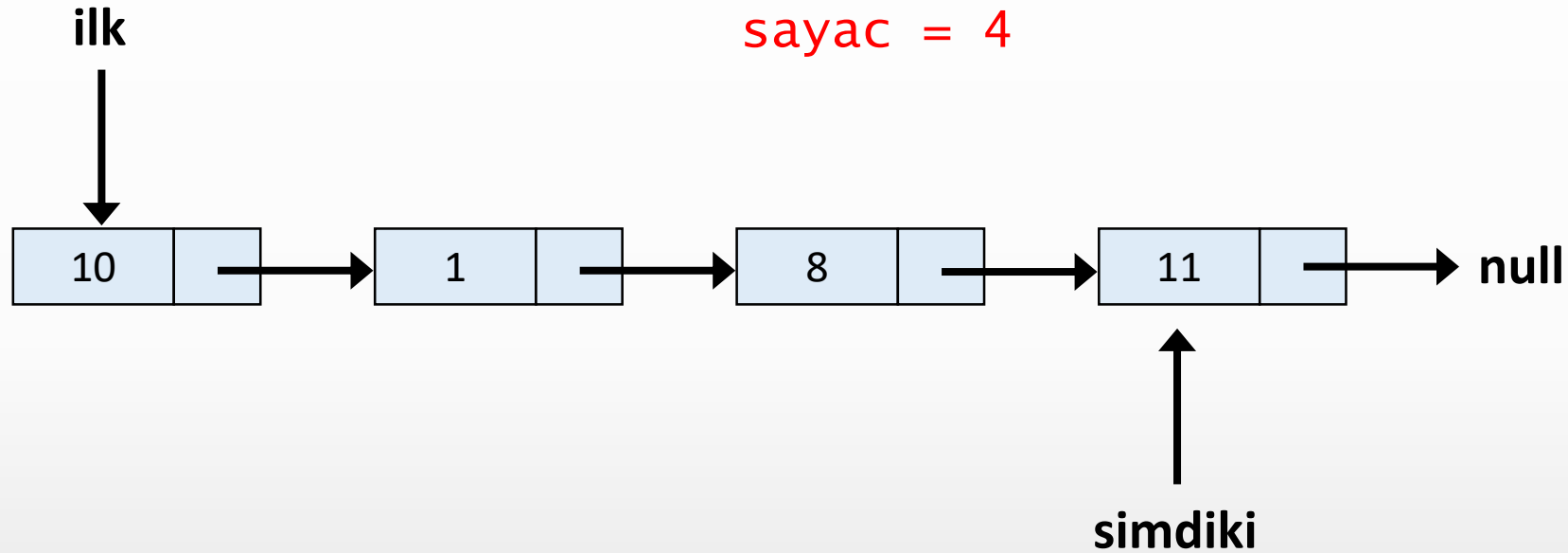
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



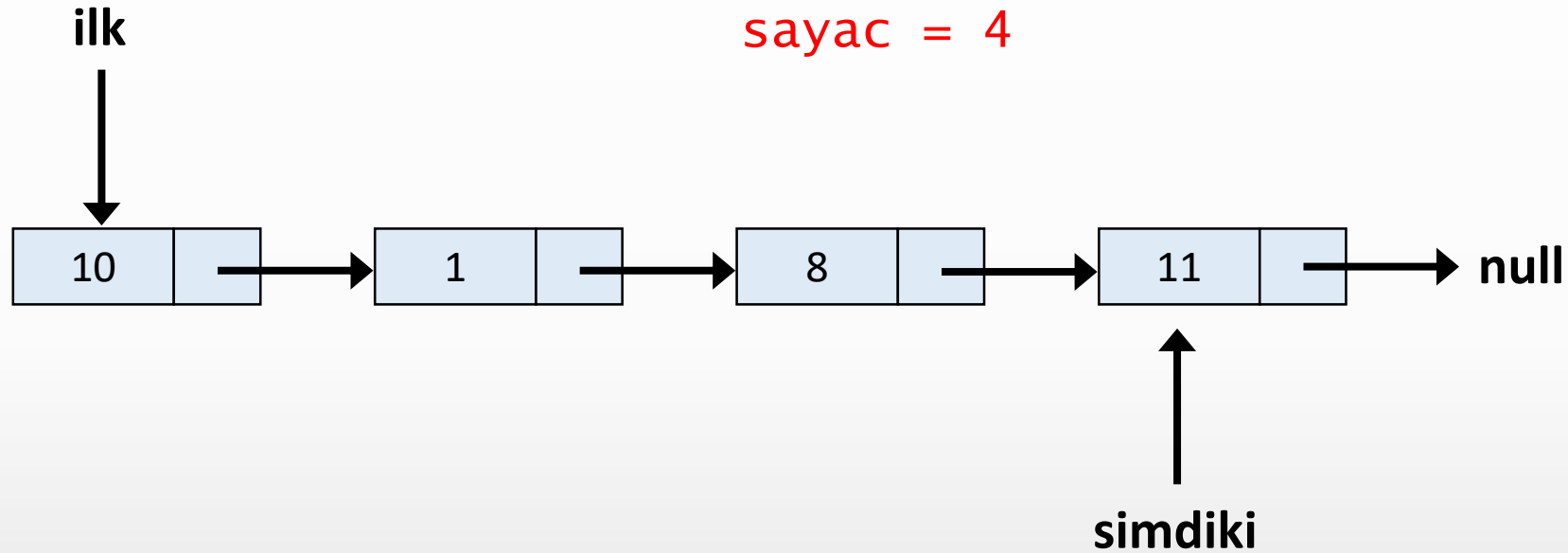
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



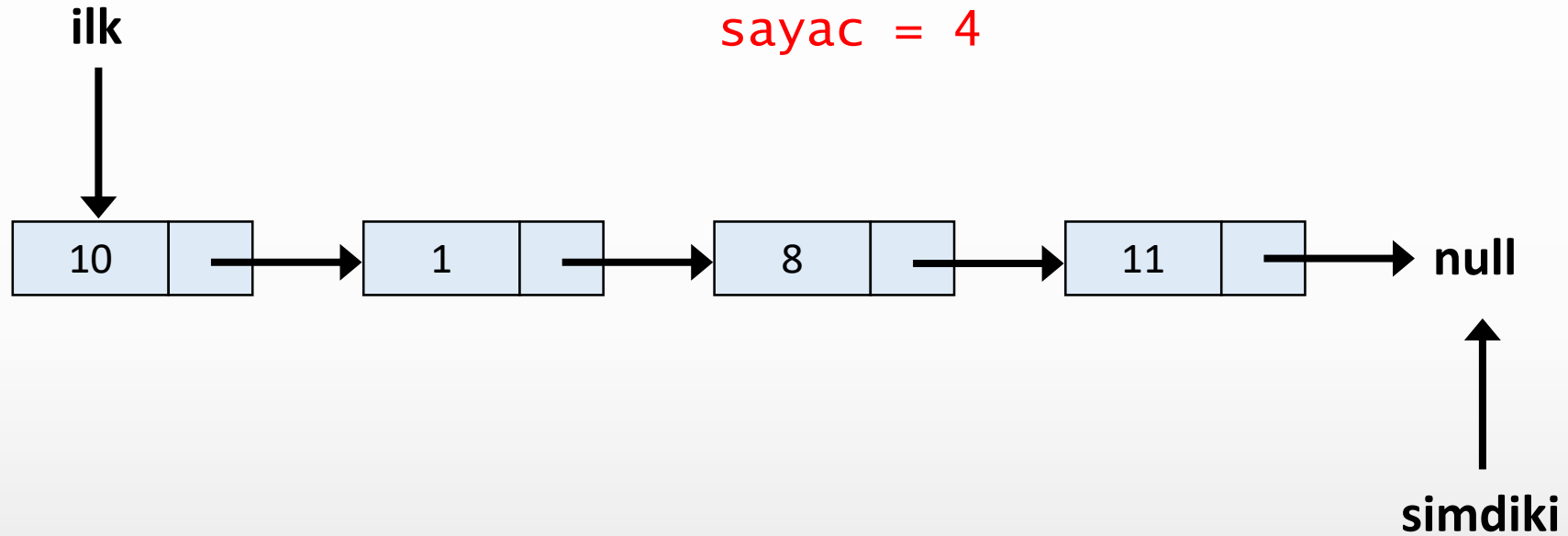
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



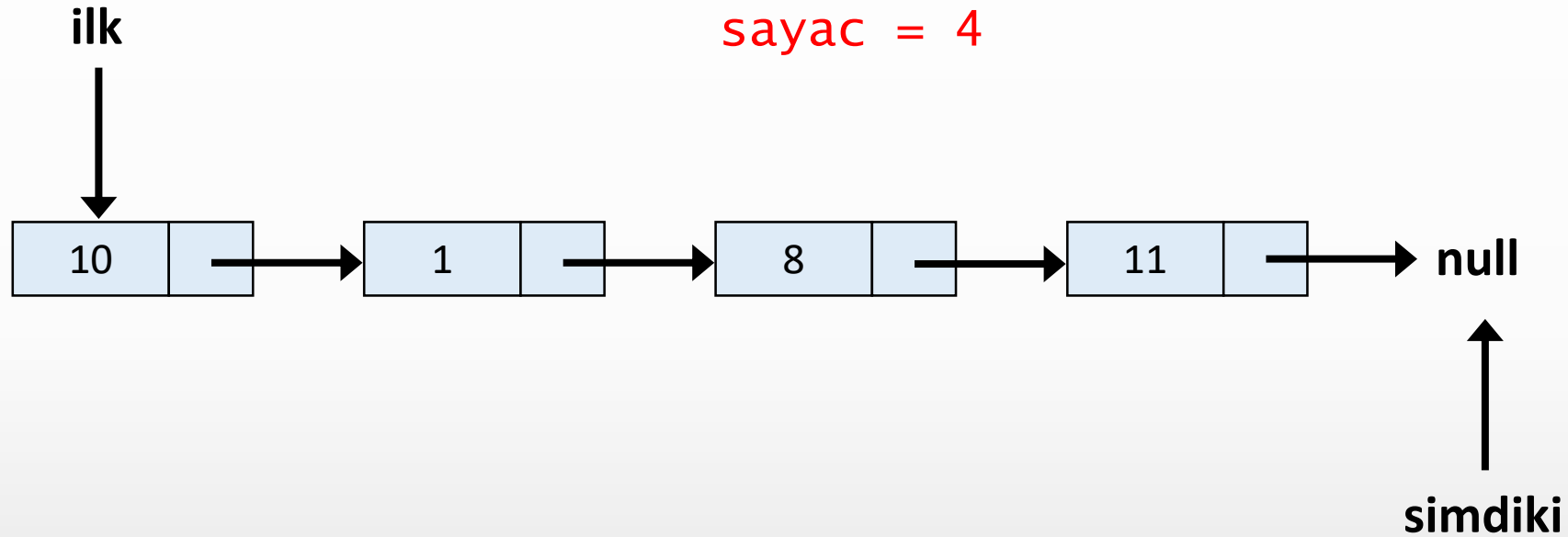
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```



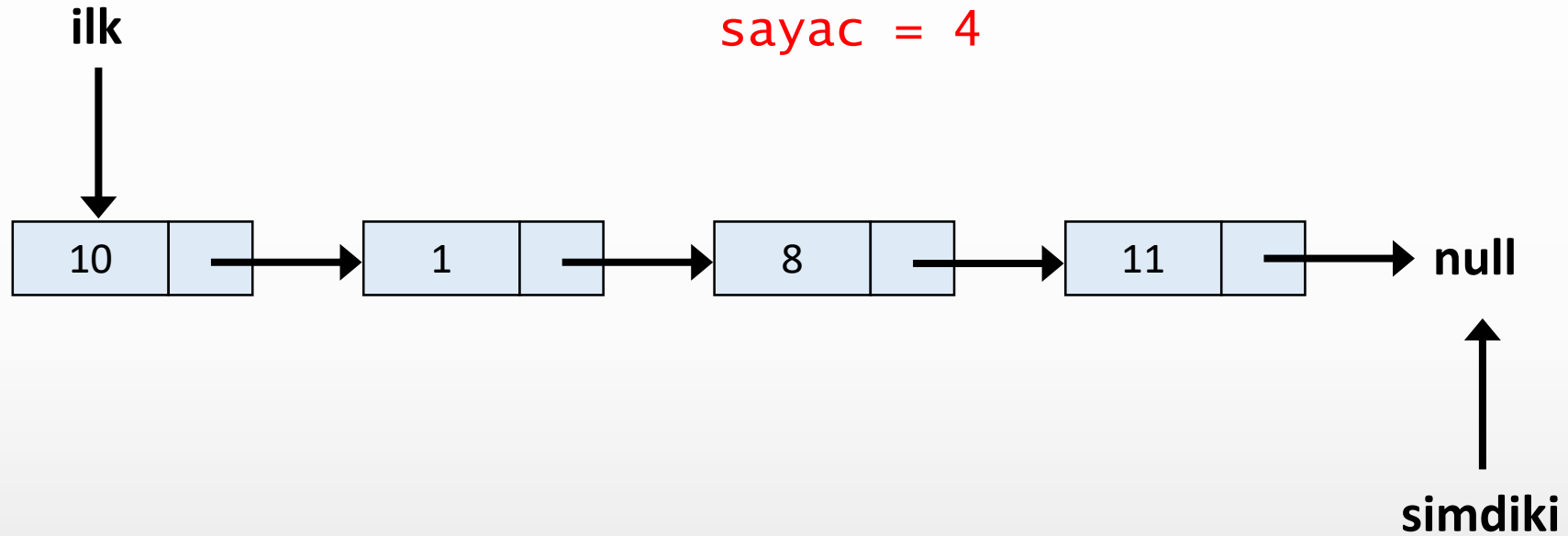
Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
return sayac;
```




Tek Yönlü Bağlı Listenin Uzunluğunu Bulma



```
int sayac = 0;
Dugum simdiki = ilk;
while(simdiki != null) {
    sayac++;
    simdiki = simdiki.sonraki;
}
→ return sayac;
```



Liste Başına Düğüm Ekleme

```
void basaEkle(int veri) {  
    Dugum yeniDugum = new Dugum(veri);  
    yeniDugum.sonraki = bas;  
    bas = yeniDugum;  
}
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

ilk
↓
null

```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

ilk
↓
null

deger = 11

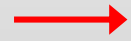
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

ilk
↓
null

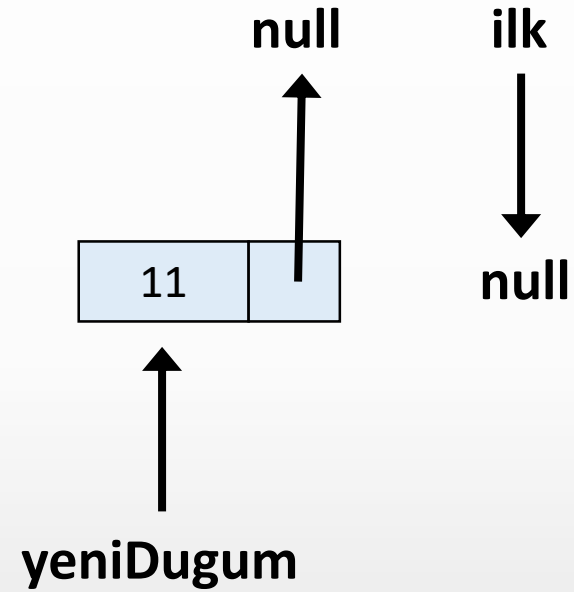
deger = 11



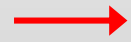
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



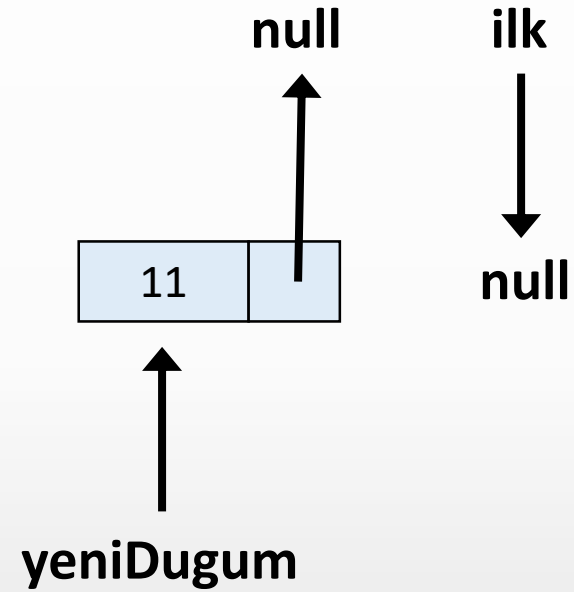
deger = 11



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



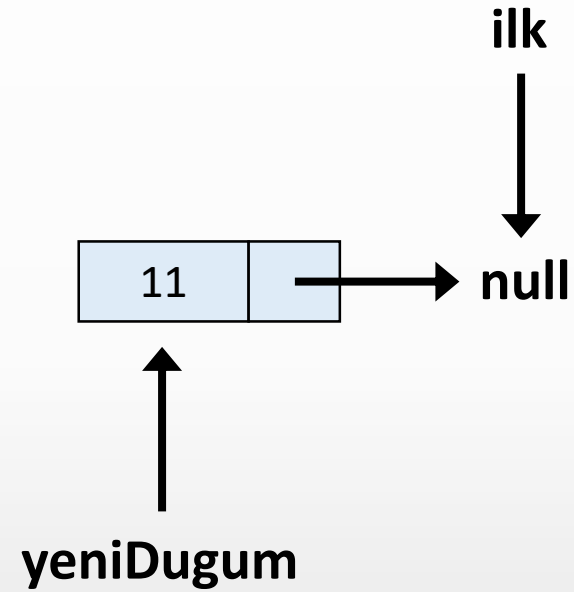
deger = 11



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```




Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



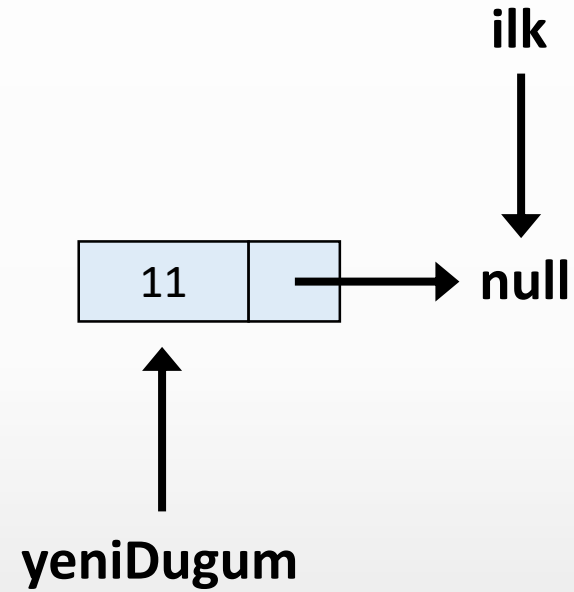
deger = 11



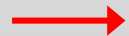
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



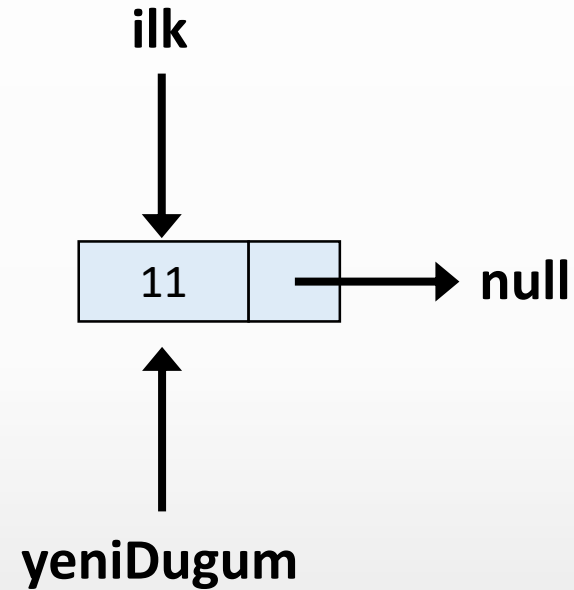
deger = 11



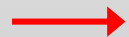
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



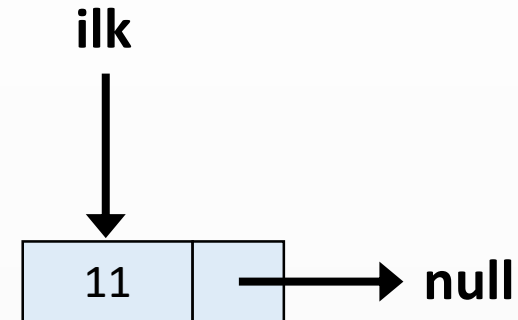
deger = 11



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



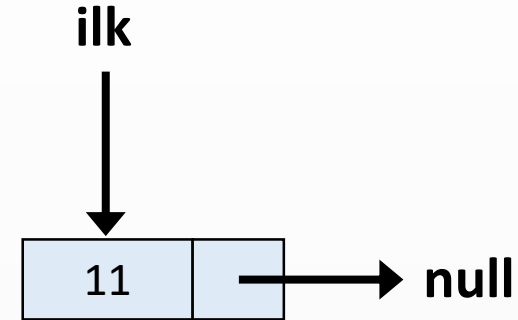
Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

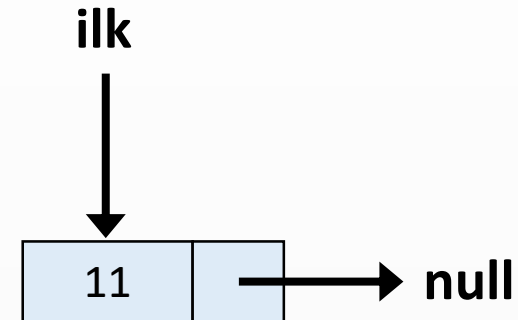


deger = 8

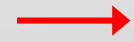
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



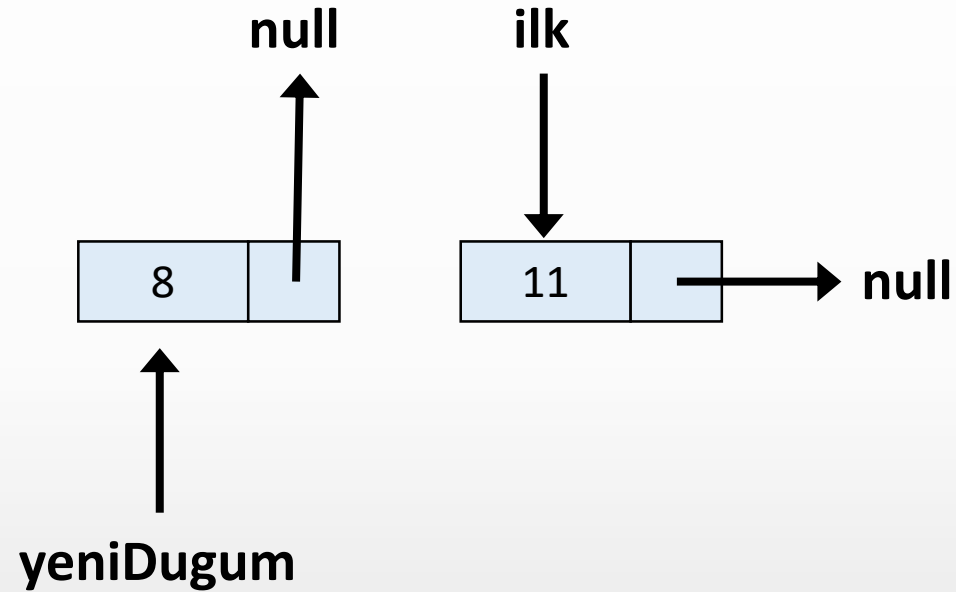
deger = 8



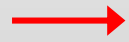
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



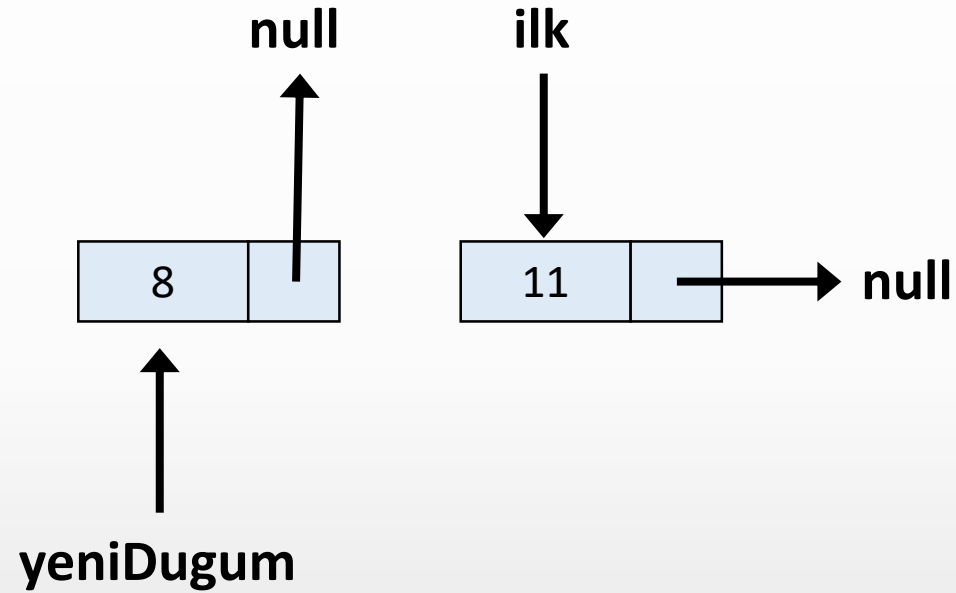
deger = 8



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



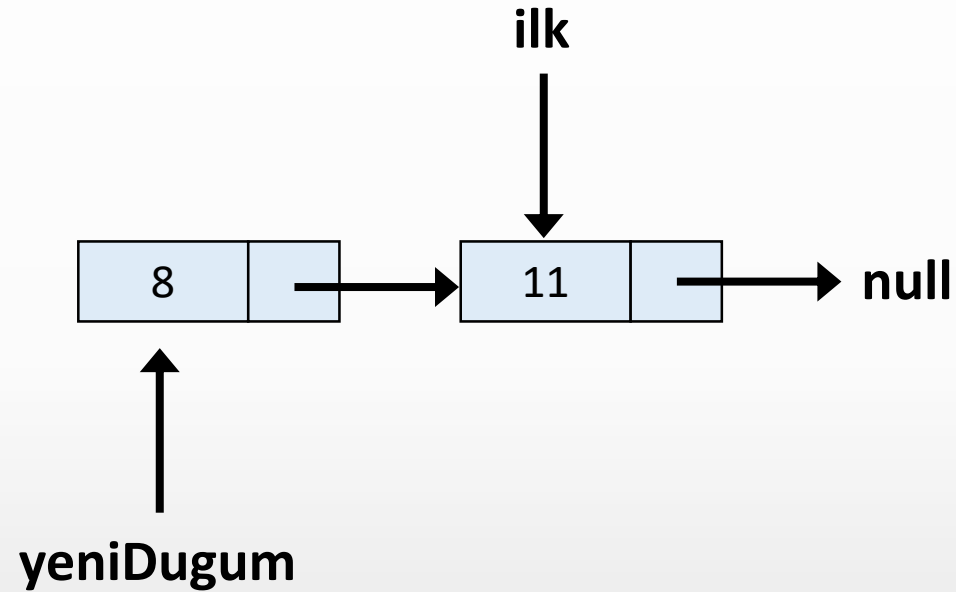
deger = 8



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```




Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



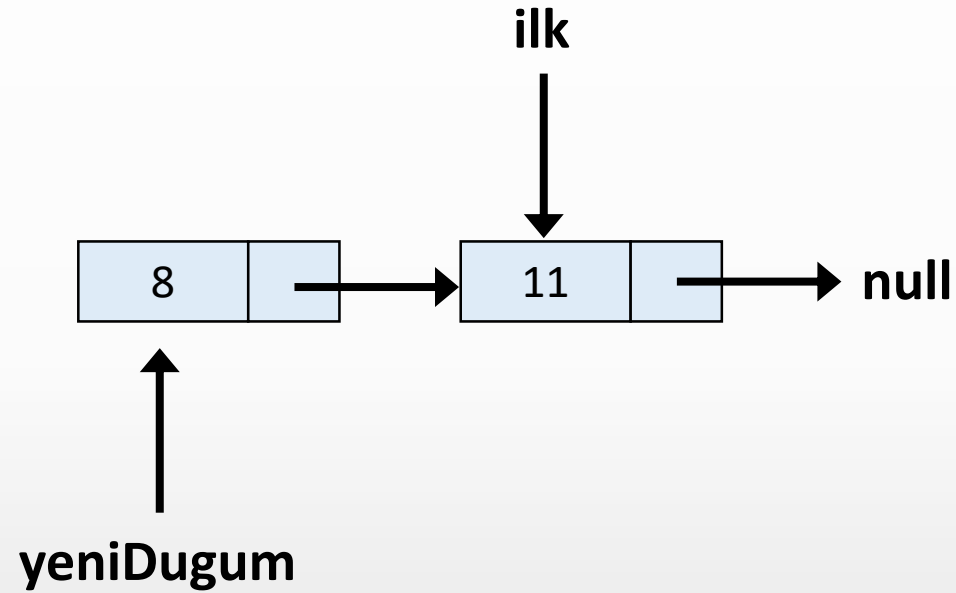
deger = 8



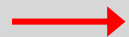
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



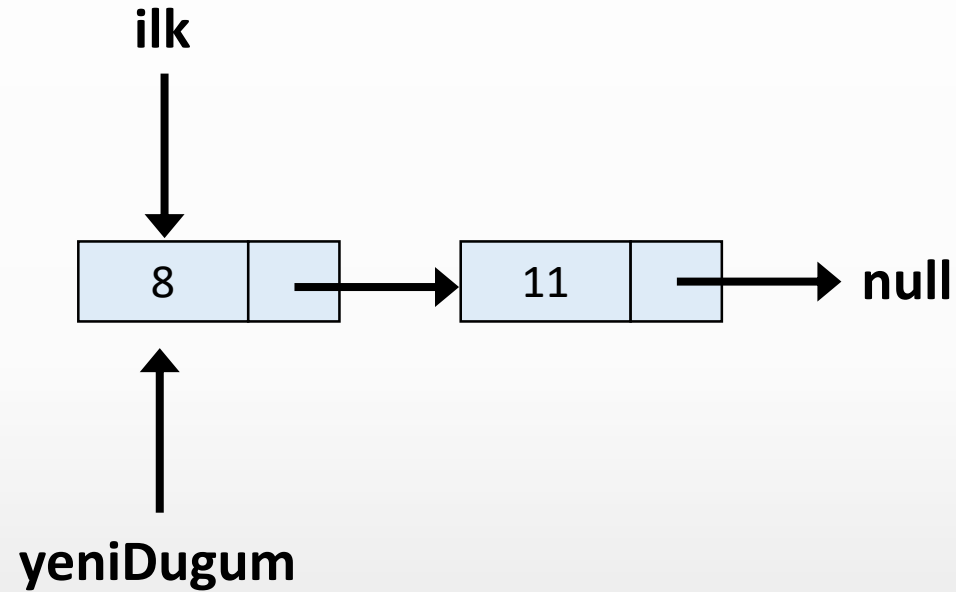
deger = 8



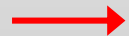
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



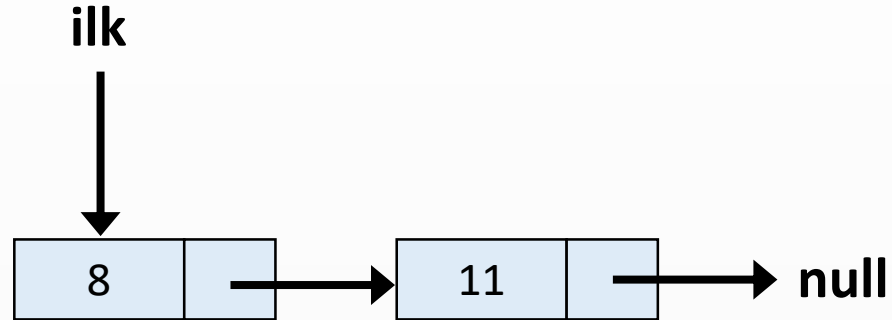
deger = 8



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



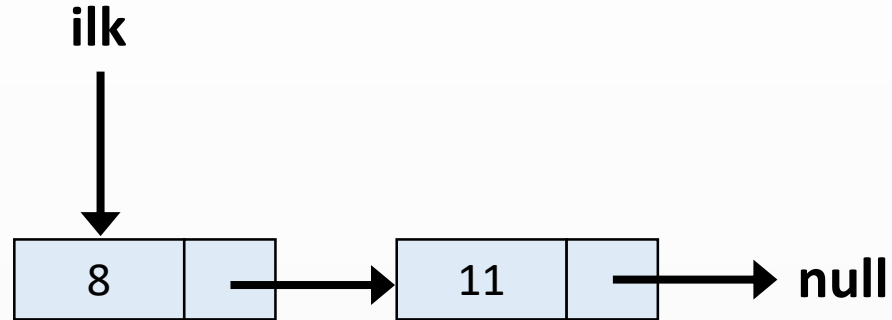
Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme

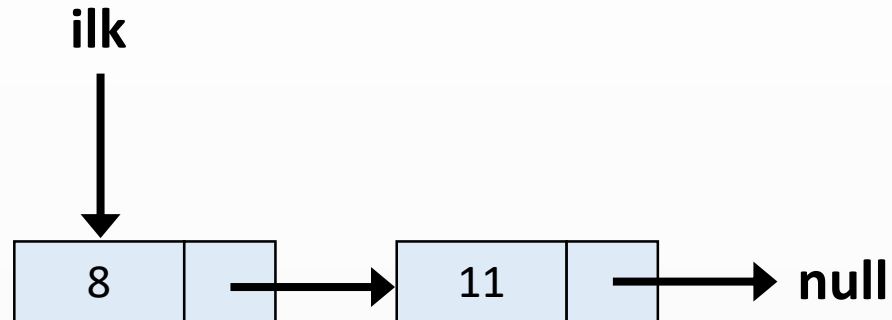


deger = 1

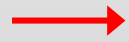
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



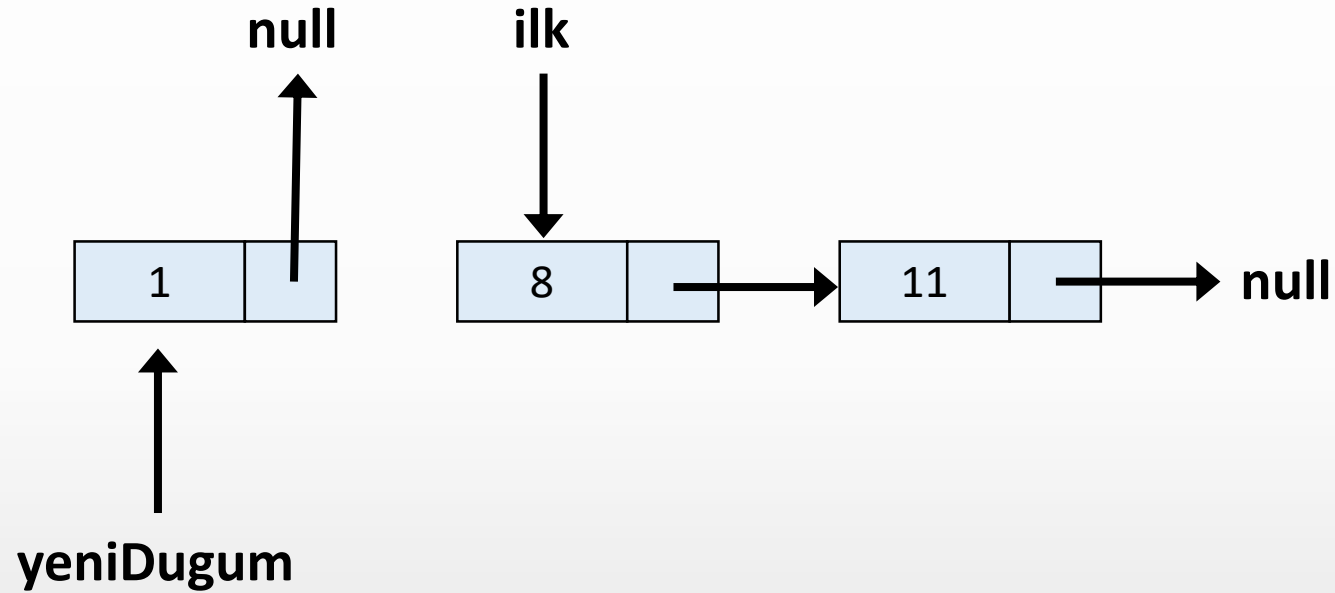
deger = 1



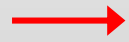
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



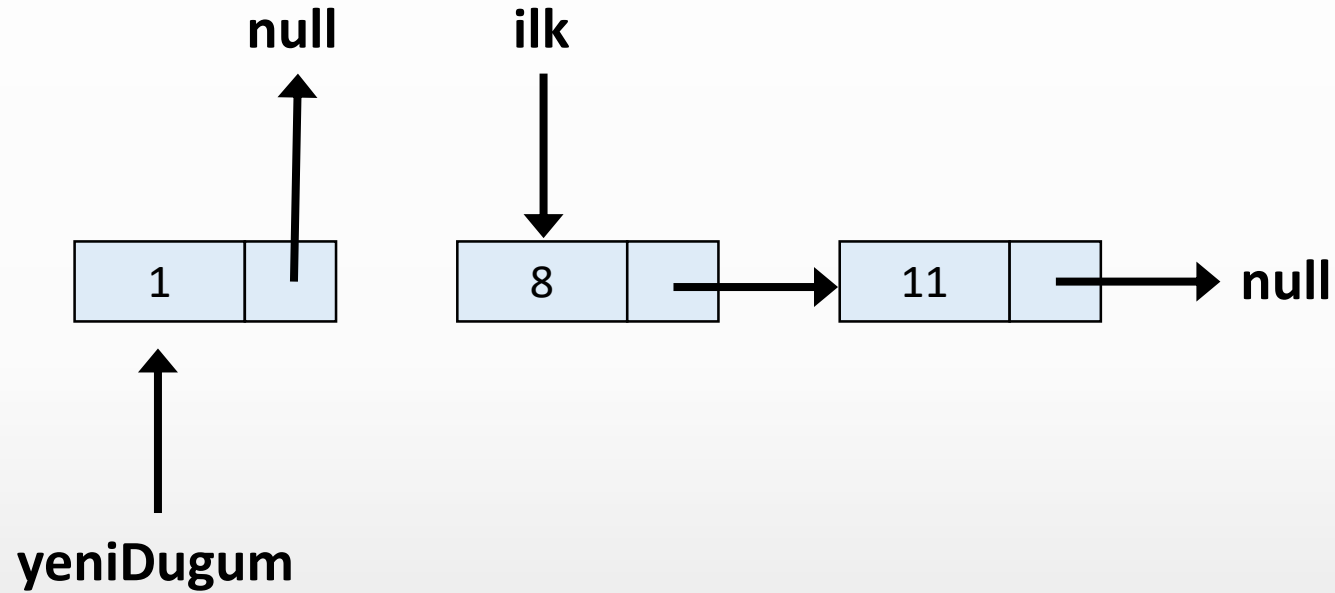
deger = 1



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



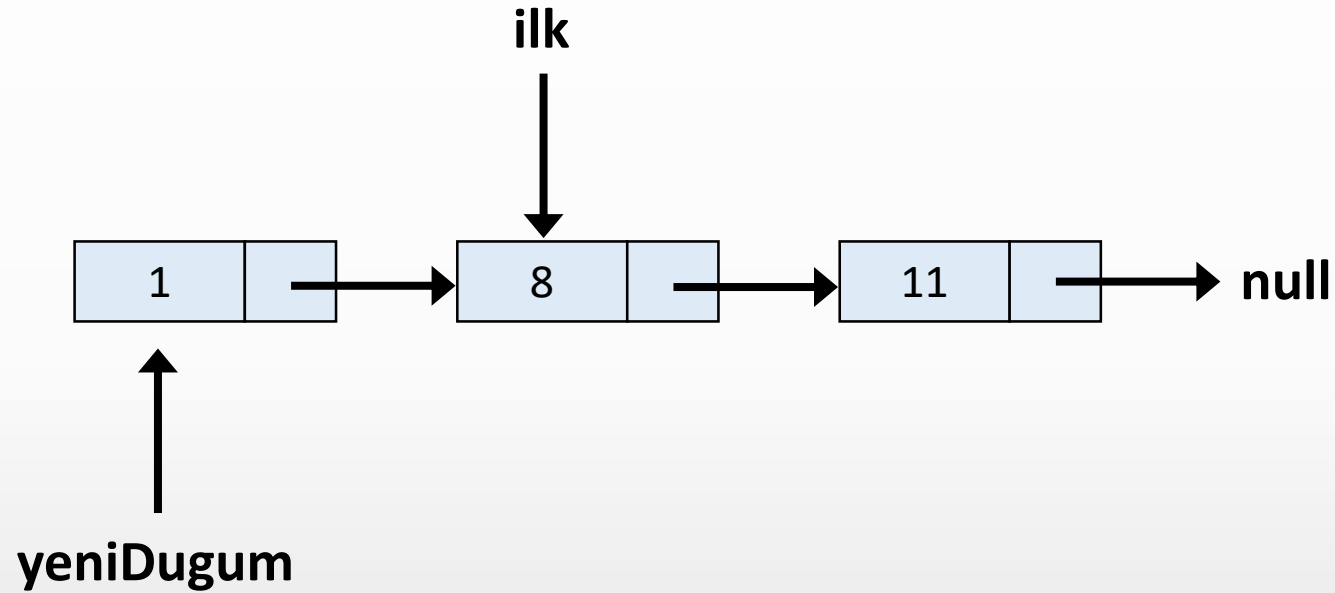
deger = 1



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```




Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



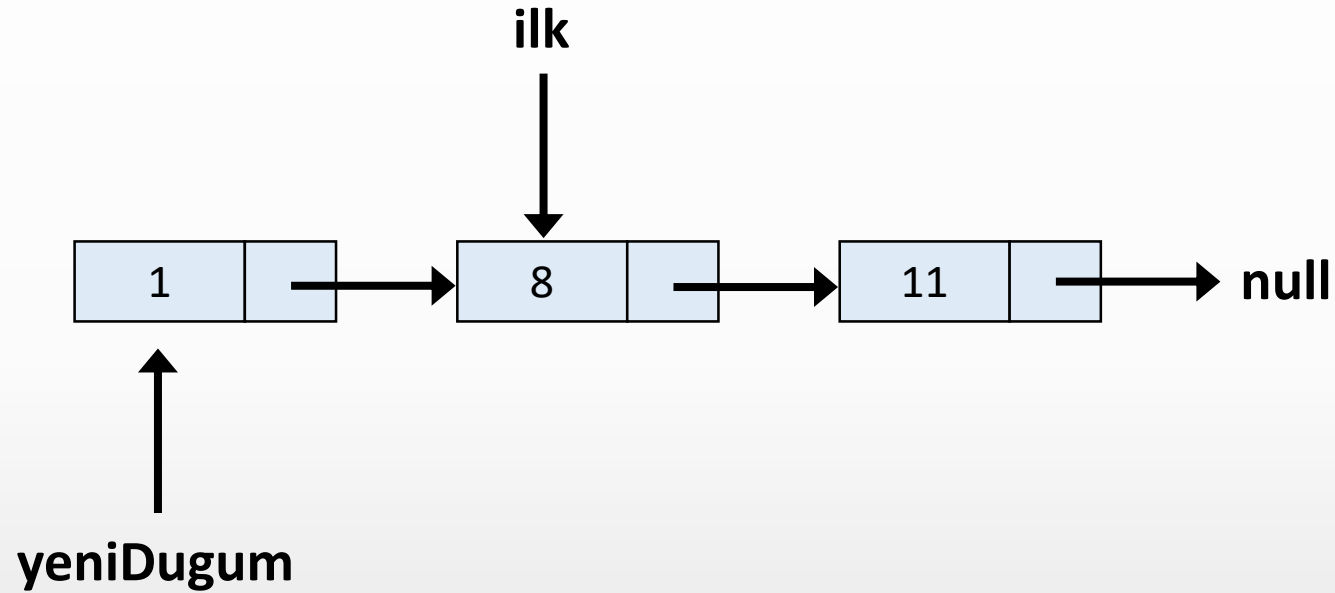
deger = 1



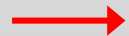
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



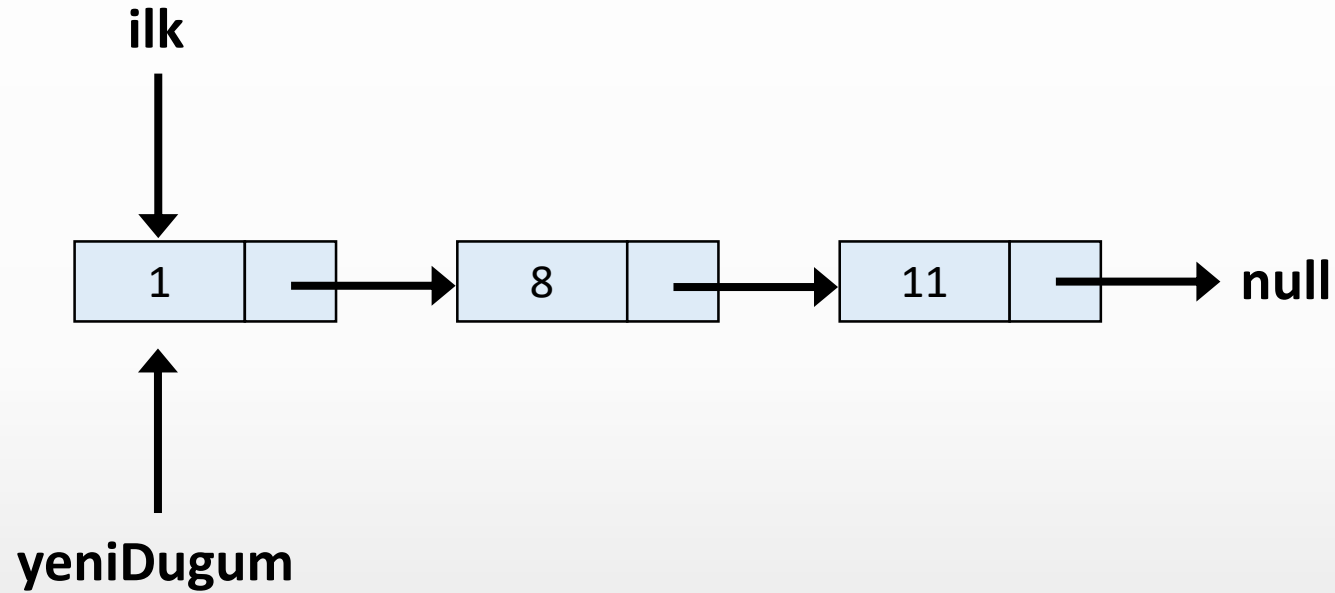
deger = 1



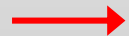
```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



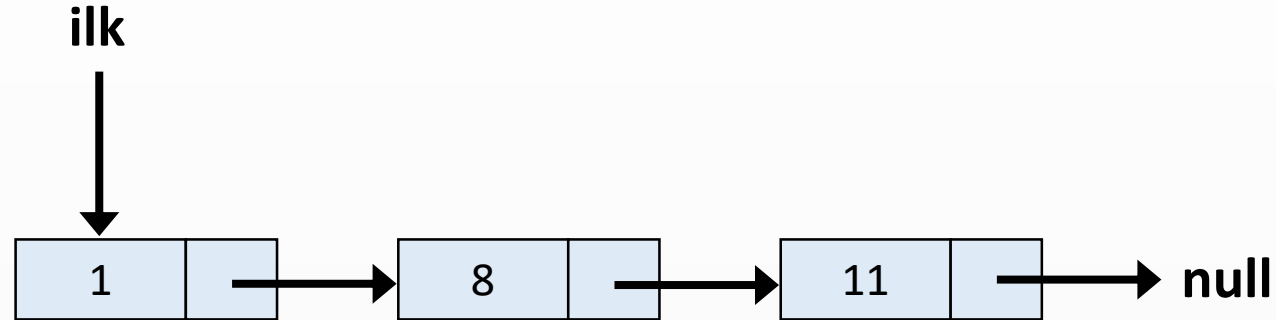
deger = 1



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Tek Yönlü Bağlı Listenin Başına Düğüm Ekleme



```
Dugum yeniDugum = new Dugum(deger);  
yeniDugum.sonraki = ilk;  
ilk = yeniDugum;
```



Liste Sonuna Düğüm Ekleme

```
void sonaEkle(int veri) {  
    Dugum yeniDugum = new Dugum(veri);  
    if (bas == null) {  
        bas = yeniDugum;  
        return;  
    }  
    Dugum gecici = bas;  
    while (gecici.sonraki != null) {  
        gecici = gecici.sonraki;  
    }  
    gecici.sonraki = yeniDugum;  
}
```



Tek Yönlü Bağlı Listenin Sonuna Düğüm Ekleme

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



ilk
↓
null

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



ilk
↓
null

deger = 11

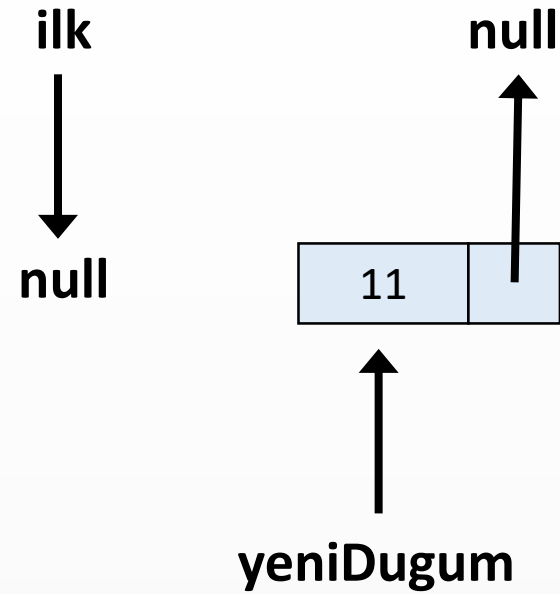
```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```




ilk
↓
null

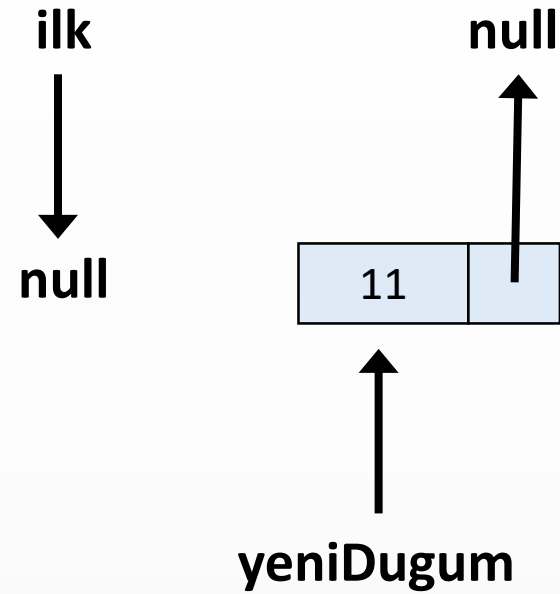
deger = 11

```
→ Dugum yeniDugum = new Dugum(deger);  
   if(ilk == null) {  
       ilk = yeniDugum;  
       return;  
   }  
   Dugum simdiki = ilk;  
   while(simdiki.sonraki != null) {  
       simdiki = simdiki.sonraki;  
   }  
   simdiki.sonraki = yeniDugum;
```



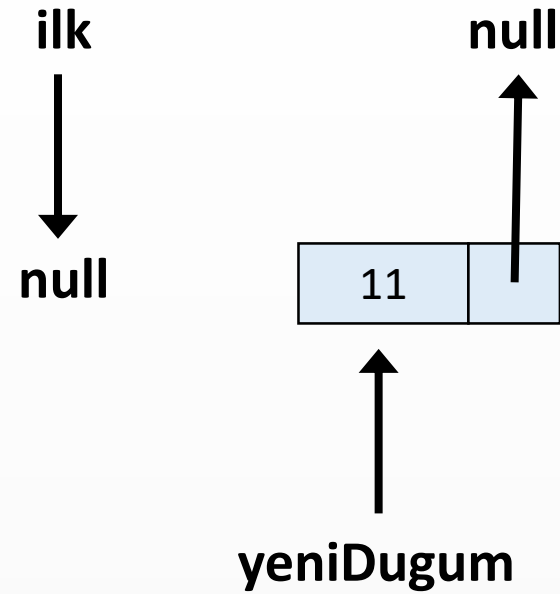
`deger = 11`

```
→ Dugum yeniDugum = new Dugum(deger);  
  if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
  }  
  Dugum simdiki = ilk;  
  while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
  }  
  simdiki.sonraki = yeniDugum;
```



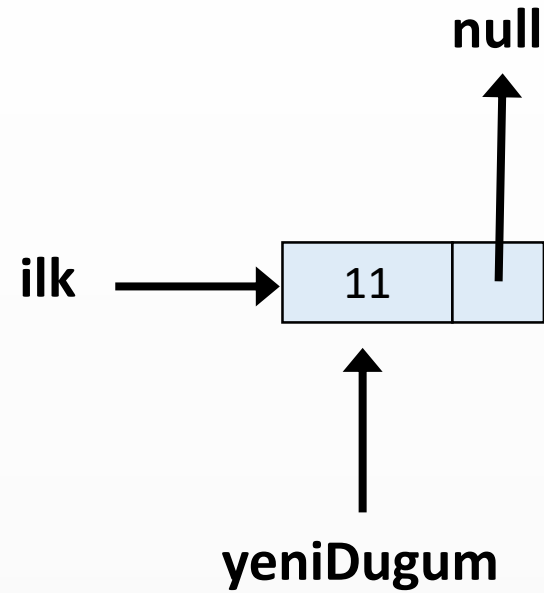
`deger = 11`

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



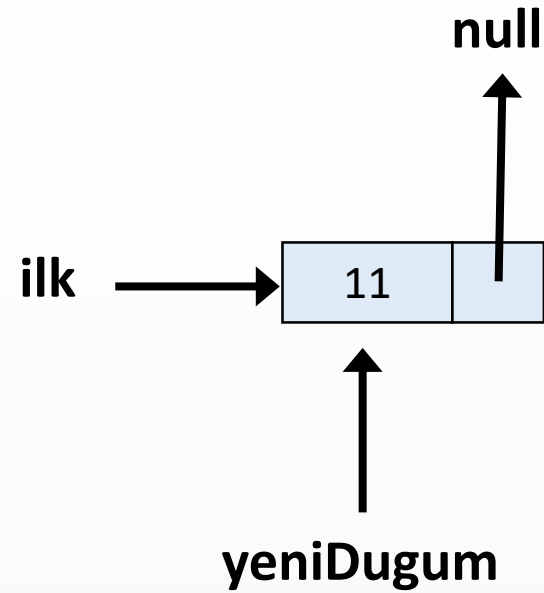
`deger = 11`

```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```



deger = 11

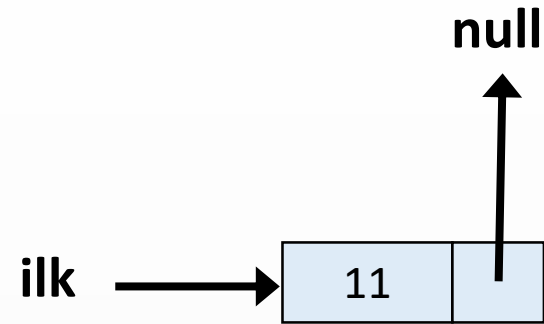
```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



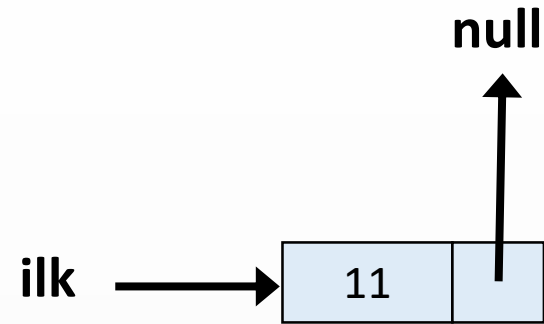
deger = 11

→

```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```

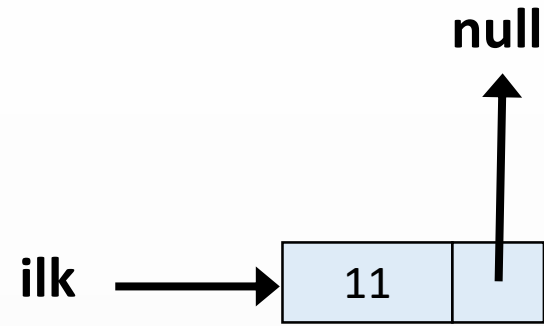


```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



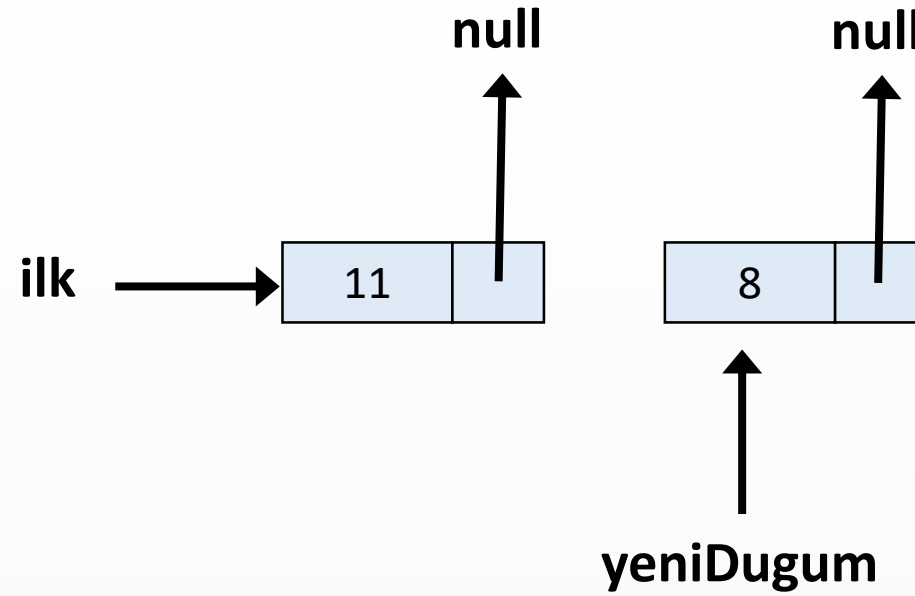
deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```

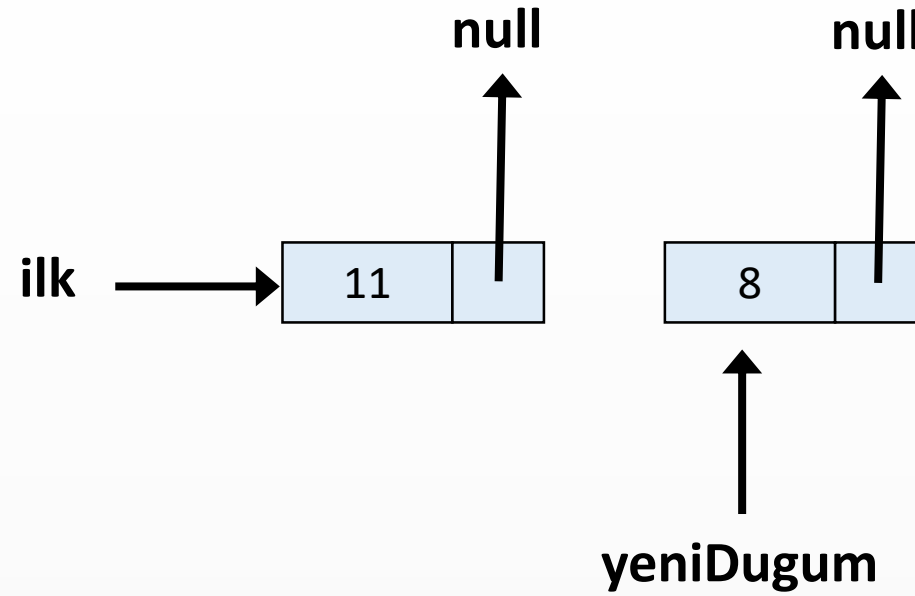
deger = 8

```
→ Dugum yeniDugum = new Dugum(deger);  
   if(ilk == null) {  
       ilk = yeniDugum;  
       return;  
   }  
   Dugum simdiki = ilk;  
   while(simdiki.sonraki != null) {  
       simdiki = simdiki.sonraki;  
   }  
   simdiki.sonraki = yeniDugum;
```



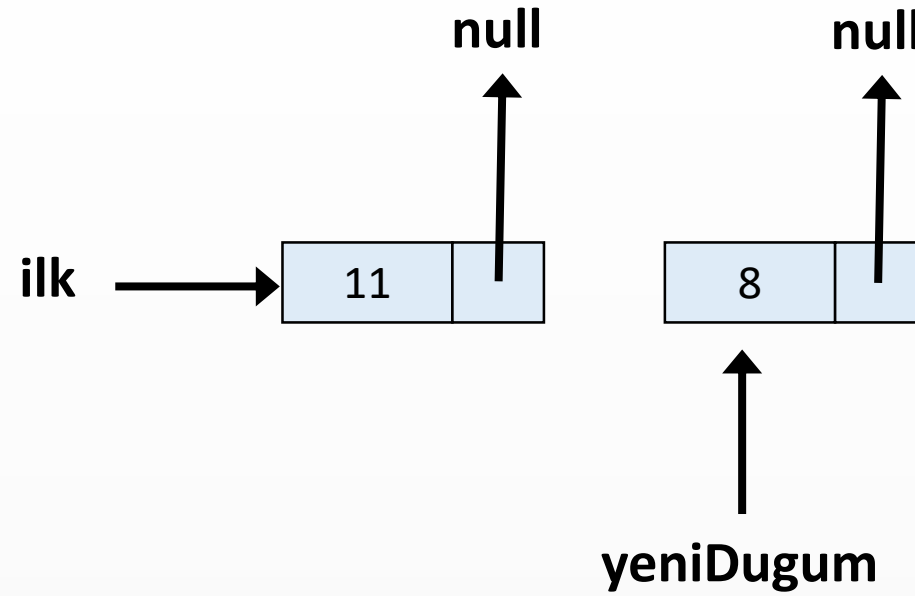
deger = 8

```
→ Dugum yeniDugum = new Dugum(deger);  
   if(ilk == null) {  
       ilk = yeniDugum;  
       return;  
   }  
   Dugum simdiki = ilk;  
   while(simdiki.sonraki != null) {  
       simdiki = simdiki.sonraki;  
   }  
   simdiki.sonraki = yeniDugum;
```



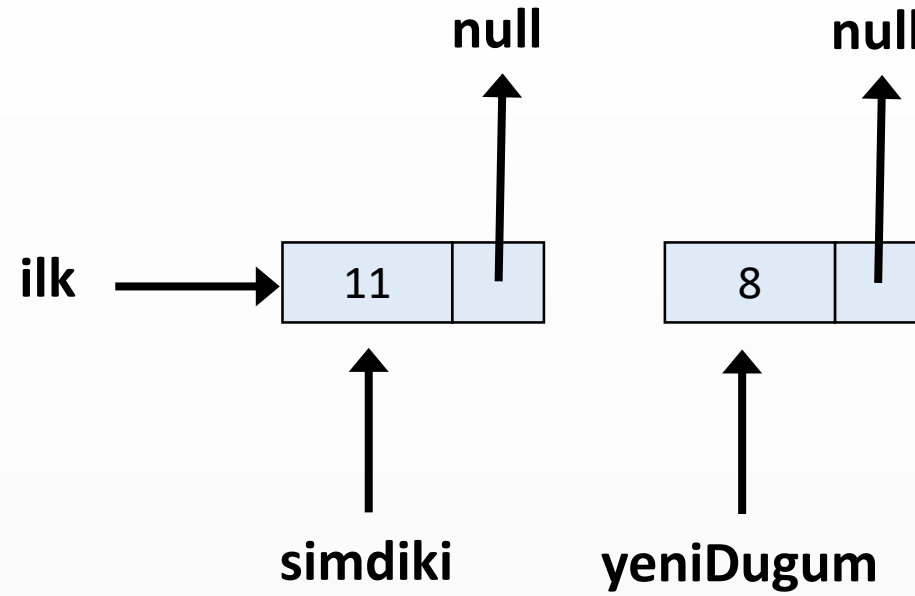
deger = 8

```
→ Dugum yeniDugum = new Dugum(deger);  
   if(ilk == null) {  
       ilk = yeniDugum;  
       return;  
   }  
   Dugum simdiki = ilk;  
   while(simdiki.sonraki != null) {  
       simdiki = simdiki.sonraki;  
   }  
   simdiki.sonraki = yeniDugum;
```



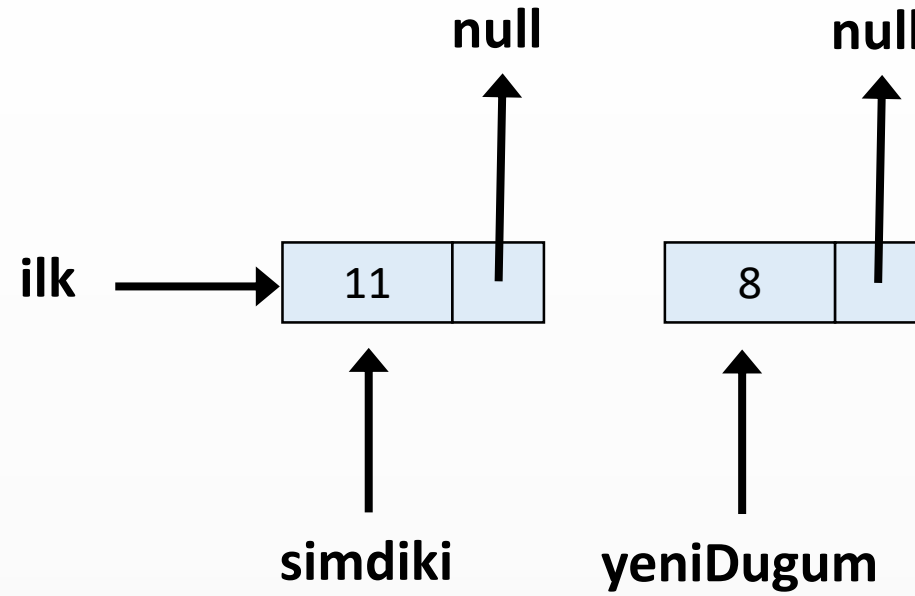
deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



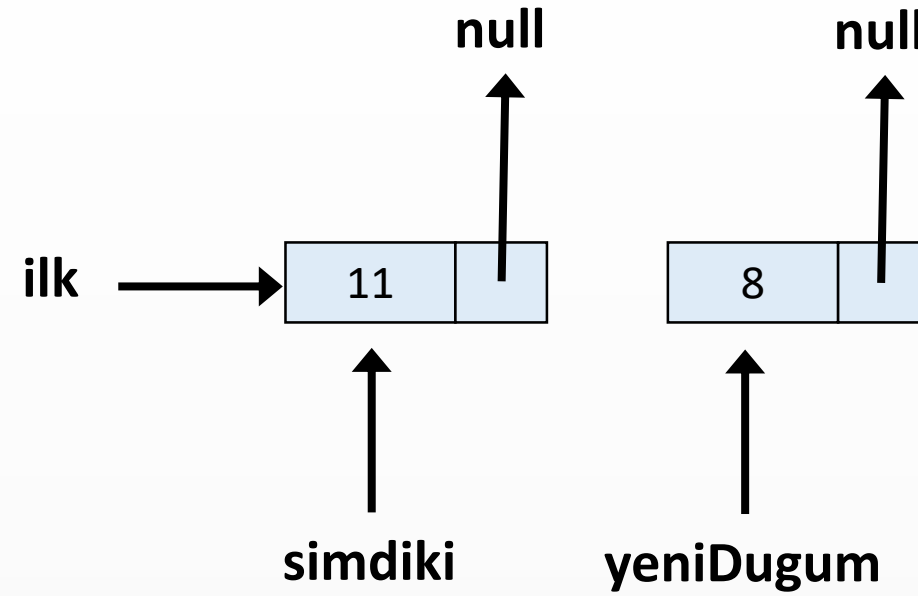
deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



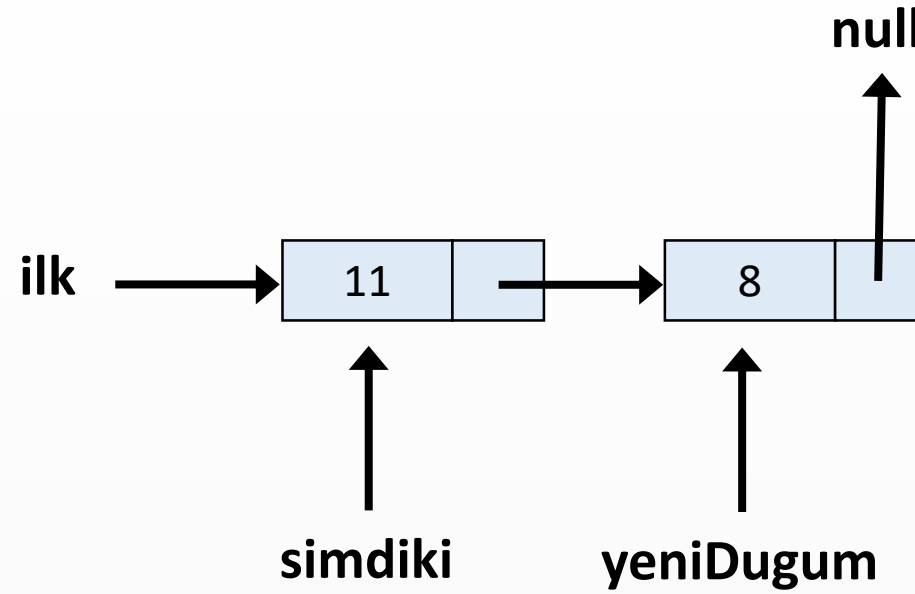
deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



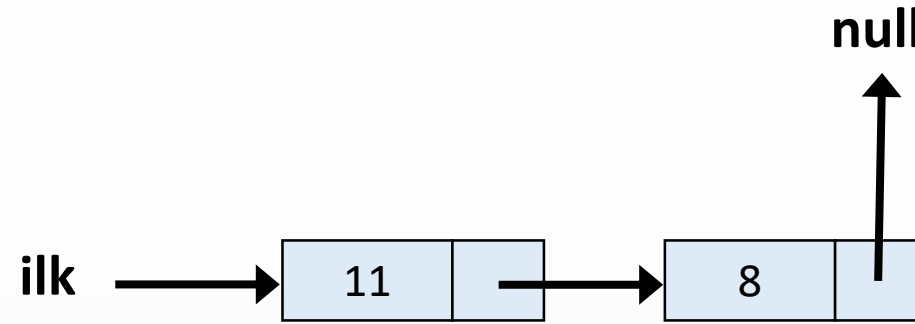
deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
→ simdiki.sonraki = yeniDugum;
```

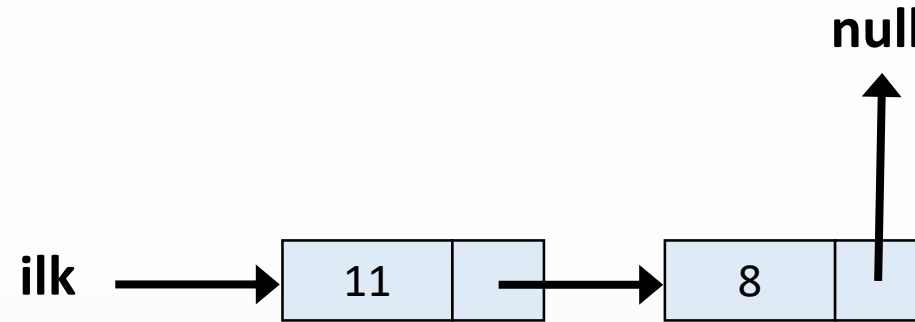


deger = 8

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
→ simdiki.sonraki = yeniDugum;
```

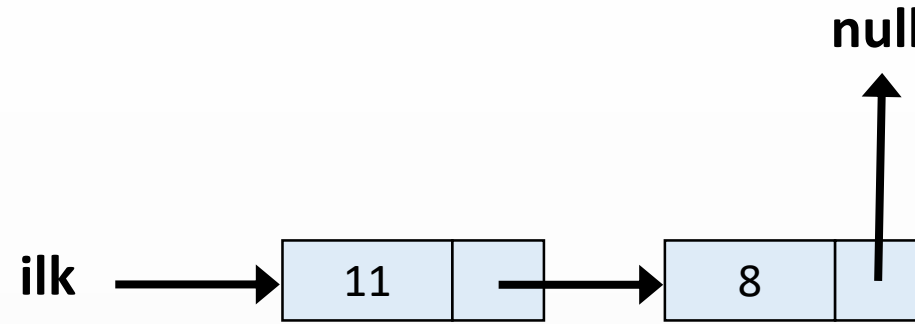



```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```



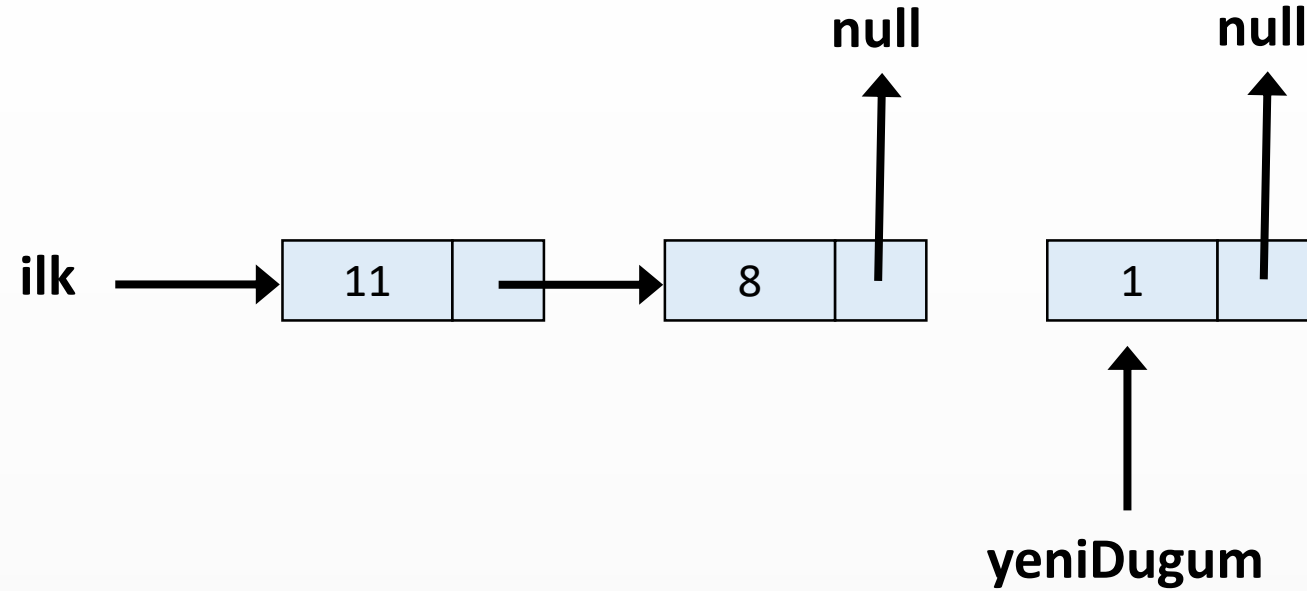
deger = 1

```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```



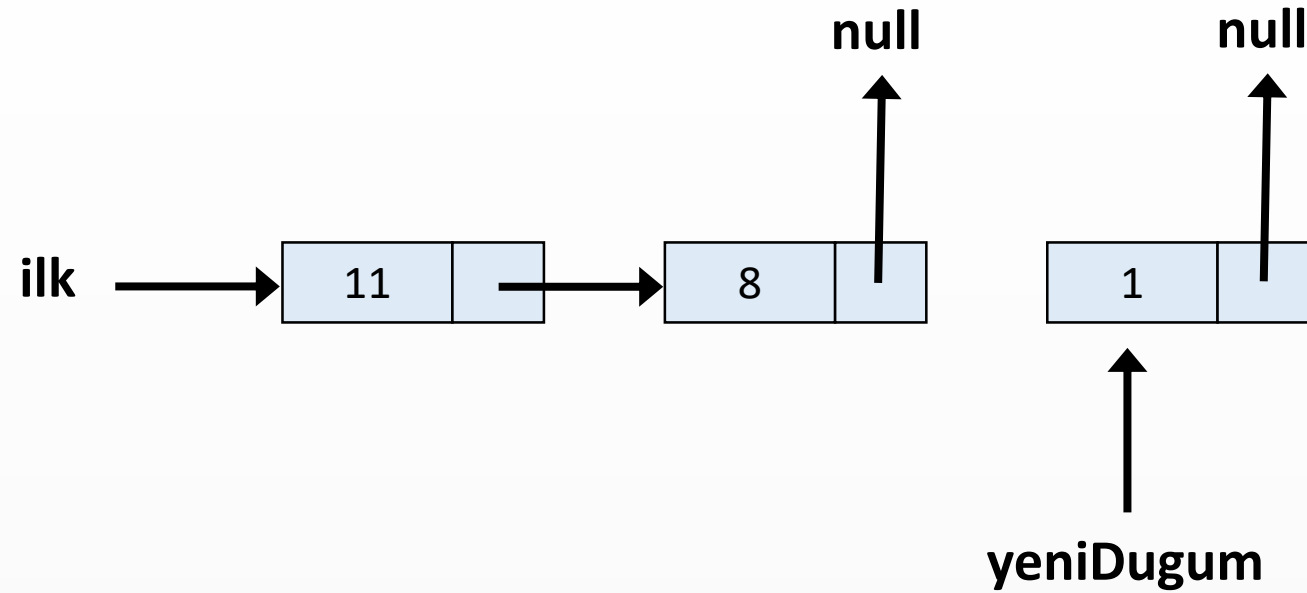
deger = 1

```
→ Dugum yeniDugum = new Dugum(deger);  
  if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
  }  
  Dugum simdiki = ilk;  
  while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
  }  
  simdiki.sonraki = yeniDugum;
```

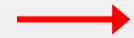


`deger = 1`

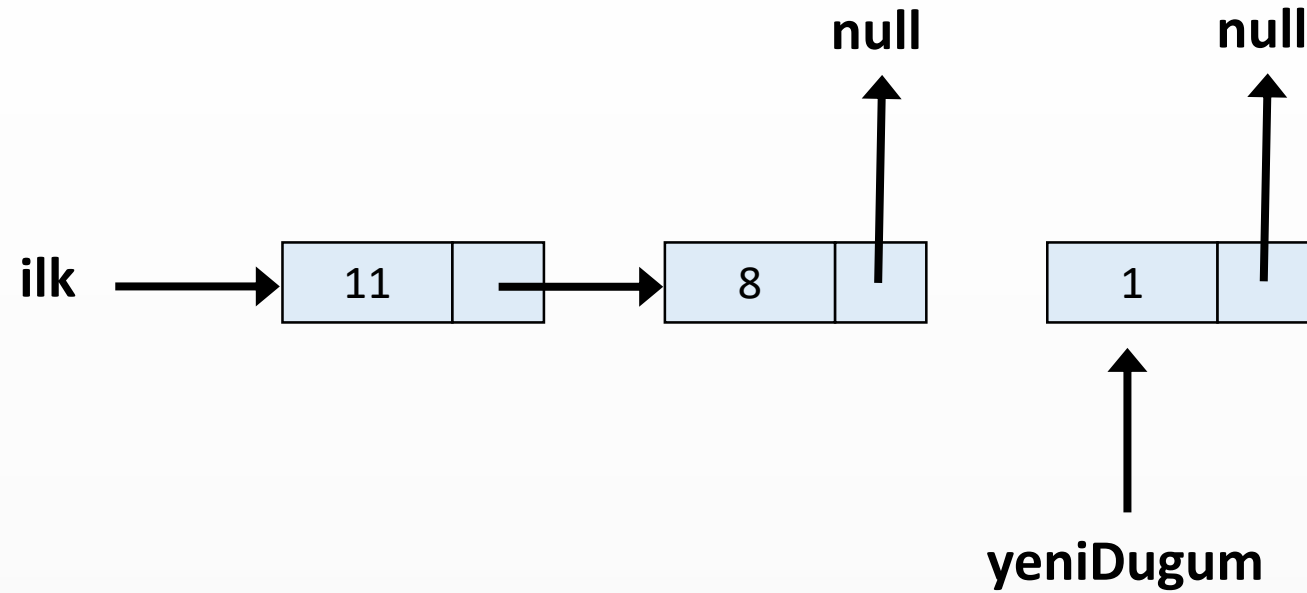
```
→ Dugum yeniDugum = new Dugum(deger);  
  if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
  }  
  Dugum simdiki = ilk;  
  while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
  }  
  simdiki.sonraki = yeniDugum;
```



`deger = 1`

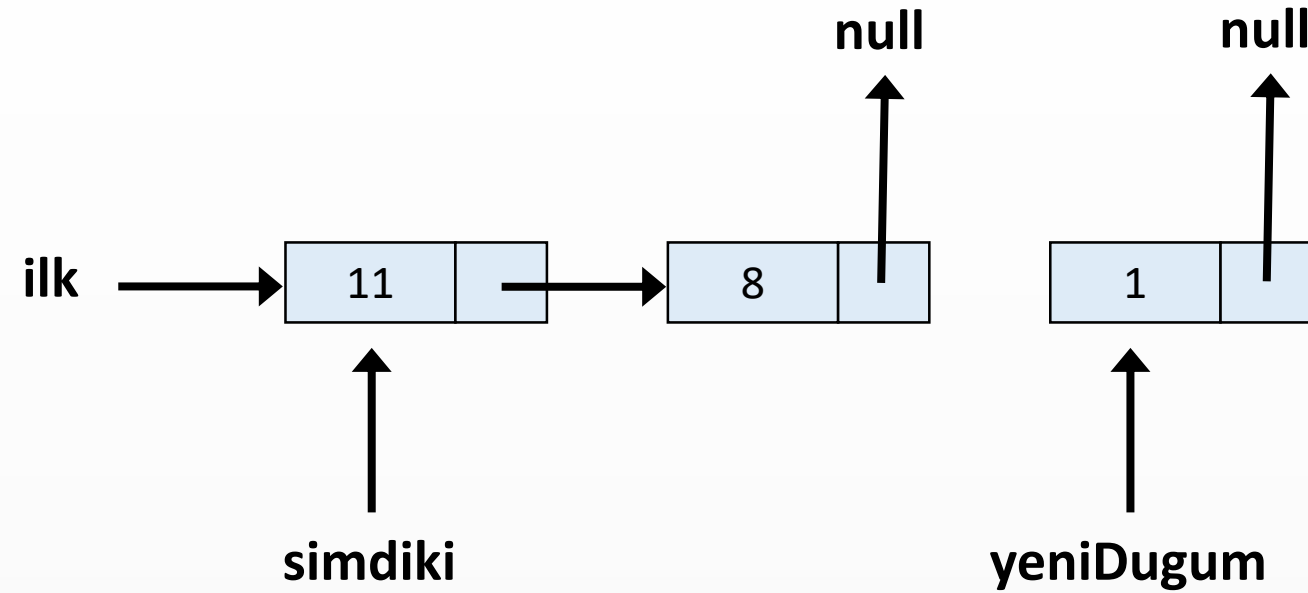


```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



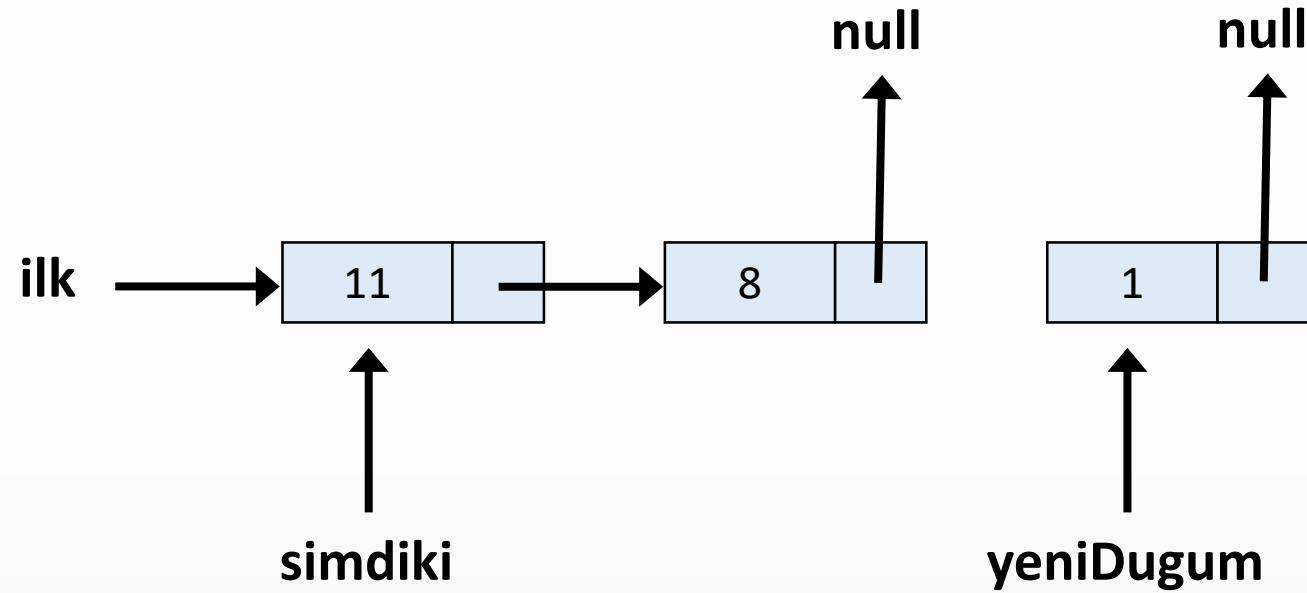
deger = 1

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



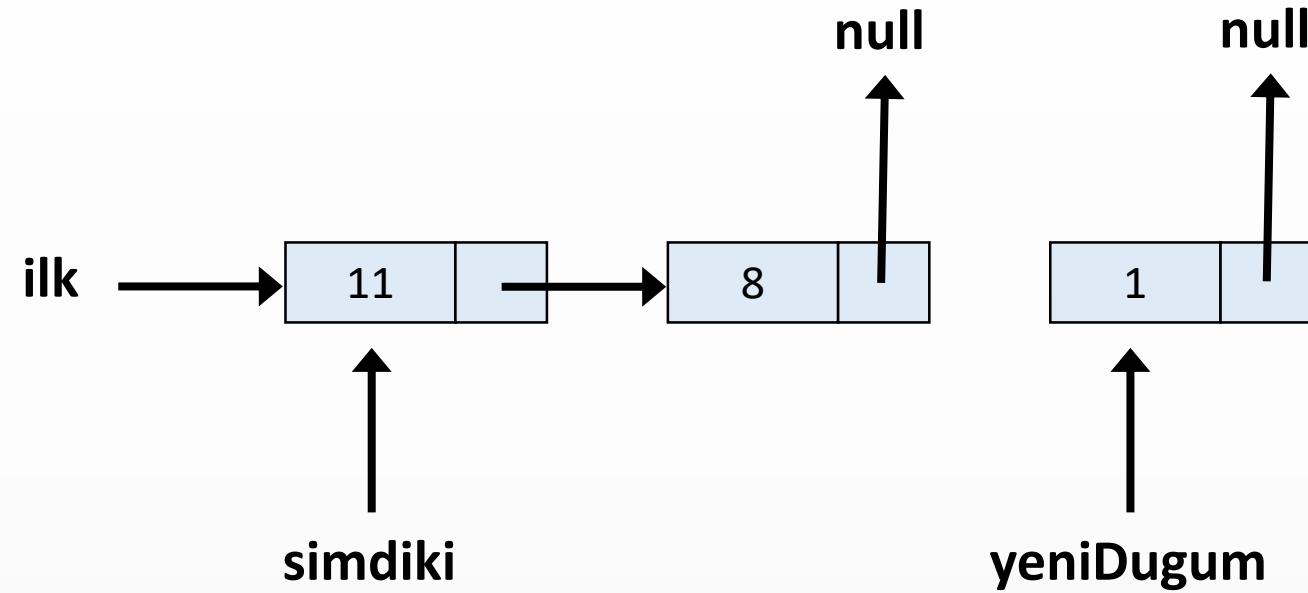
deger = 1

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```

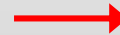


deger = 1

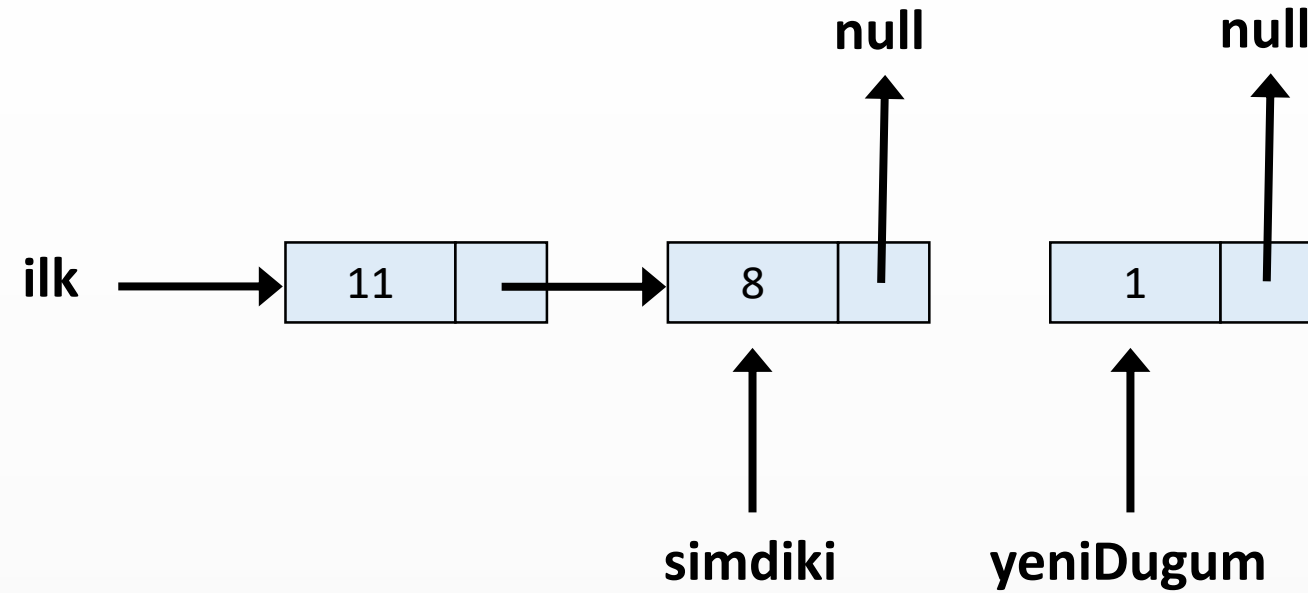
```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```

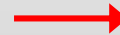
deger = 1



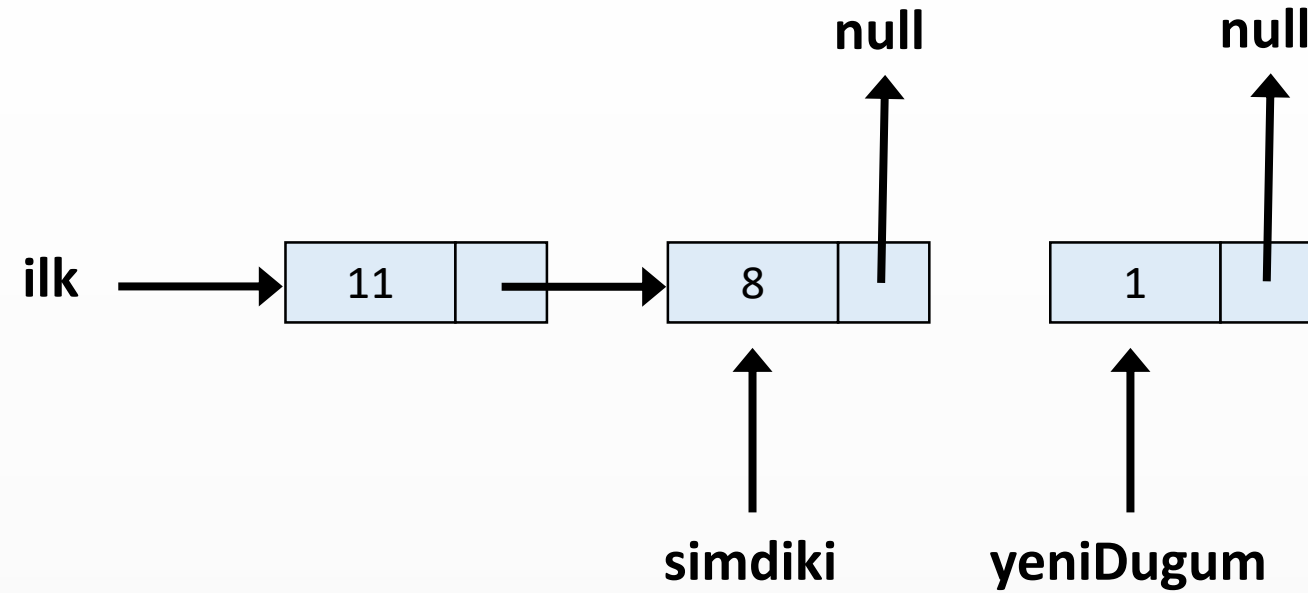
```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



deger = 1

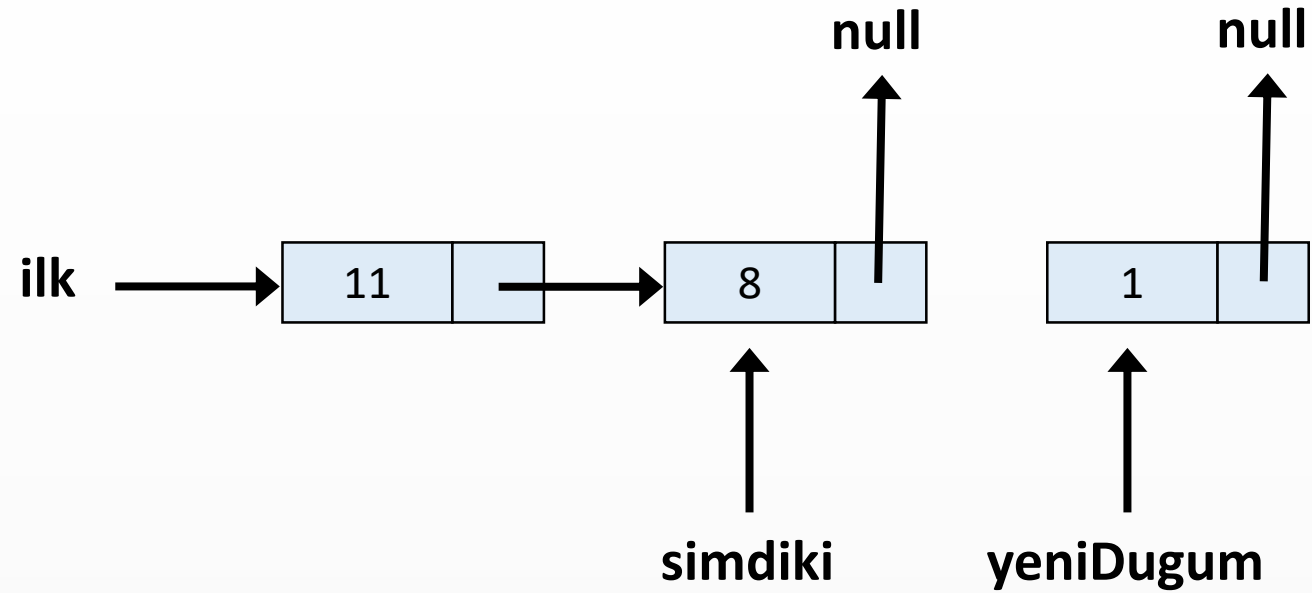


```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



deger = 1

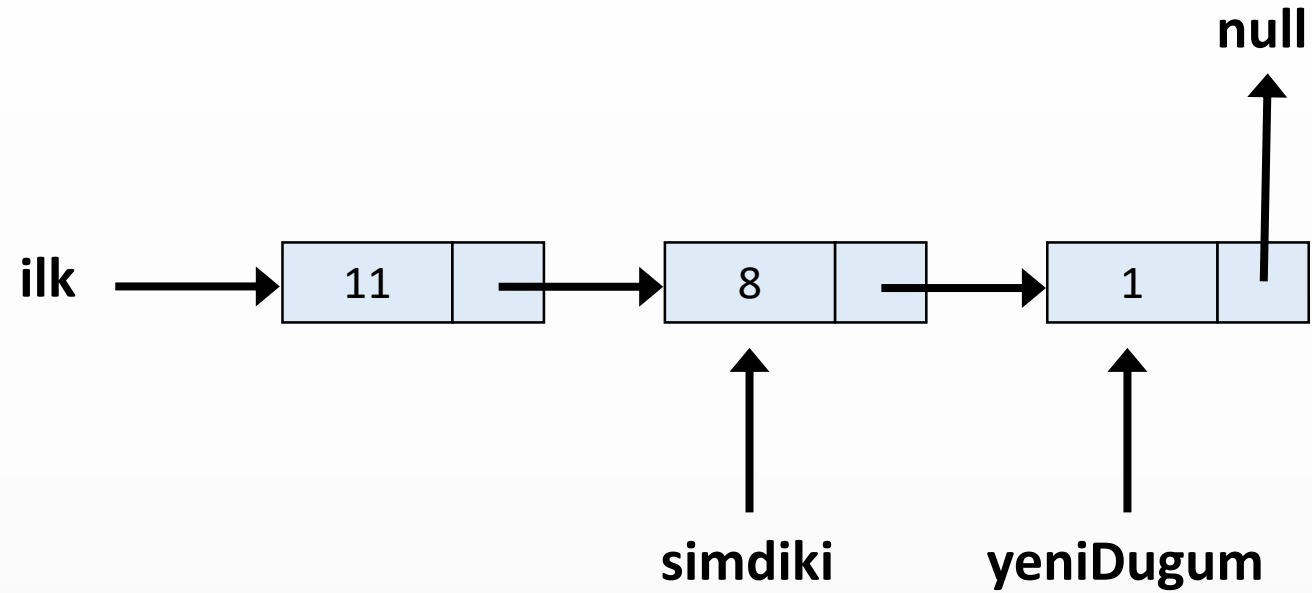
```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



deger = 1

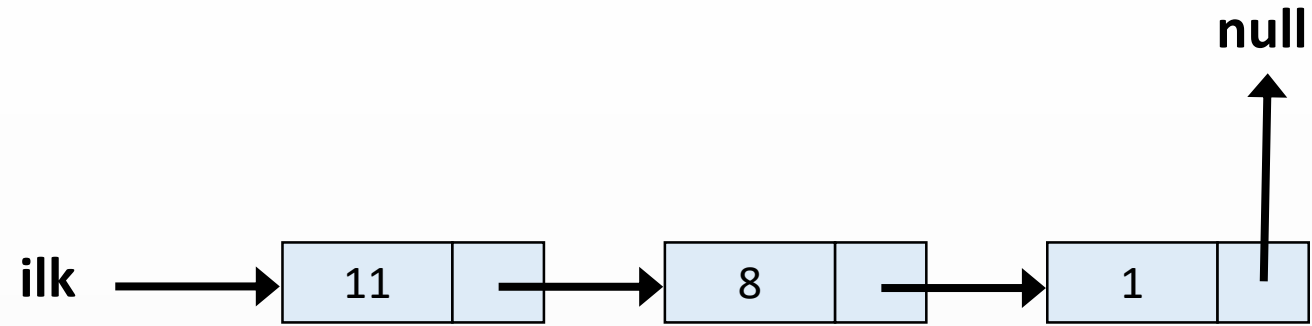


```
Dugum yeniDugum = new Dugum(deger);
if(ilk == null) {
    ilk = yeniDugum;
    return;
}
Dugum simdiki = ilk;
while(simdiki.sonraki != null) {
    simdiki = simdiki.sonraki;
}
simdiki.sonraki = yeniDugum;
```



deger = 1

```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
→ simdiki.sonraki = yeniDugum;
```



```
Dugum yeniDugum = new Dugum(deger);  
if(ilk == null) {  
    ilk = yeniDugum;  
    return;  
}  
Dugum simdiki = ilk;  
while(simdiki.sonraki != null) {  
    simdiki = simdiki.sonraki;  
}  
simdiki.sonraki = yeniDugum;
```



Belirli Bir Konuma Düğüm Ekleme

```
void konumaEkle(int veri, int konum) {  
    Dugum yeniDugum = new Dugum(veri);  
    Dugum gecici = bas;  
    for (int i = 0; i < konum - 1; i++) {  
        if (gecici == null) {  
            System.out.println("Belirtilen konum bulunamadı.");  
            return;  
        }  
        gecici = gecici.sonraki;  
    }  
    yeniDugum.sonraki = gecici.sonraki;  
    gecici.sonraki = yeniDugum;  
}
```

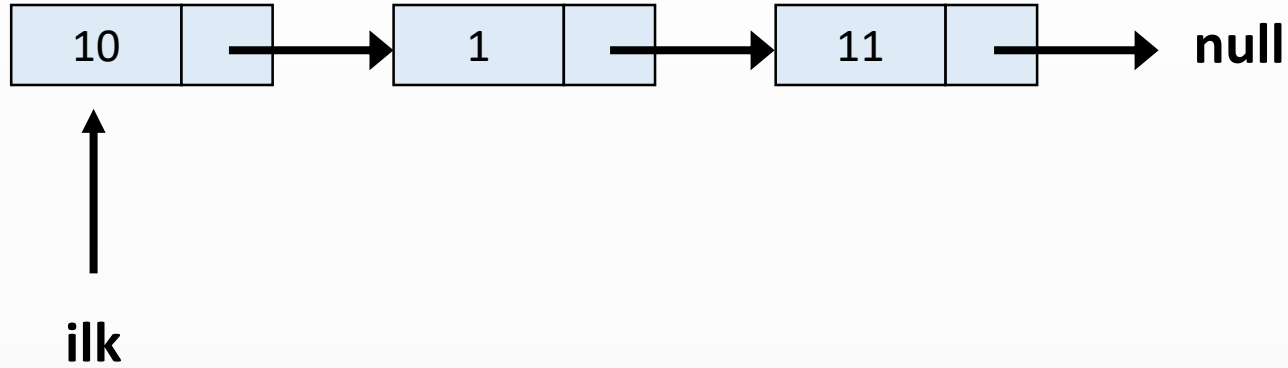


Listeden İlk Elemanı Silme

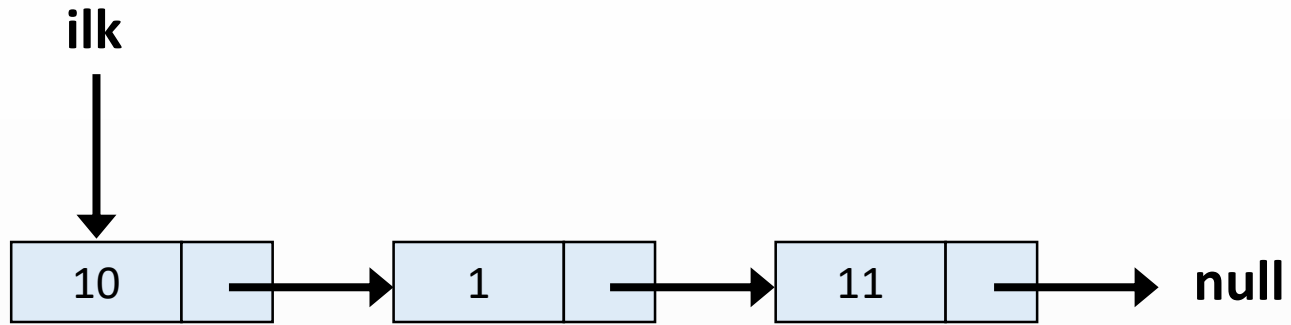
```
void ilkElemaniSil() {  
    if (bas == null) {  
        System.out.println("Liste zaten boş.");  
        return;  
    }  
    Dugum silinecekDugum = bas;  
    bas = bas.sonraki;  
    silinecekDugum = null; // Silinen düğümü bellekten temizle  
}
```




Tek Yönlü Bağlı Listede İlk Düğümü Silme

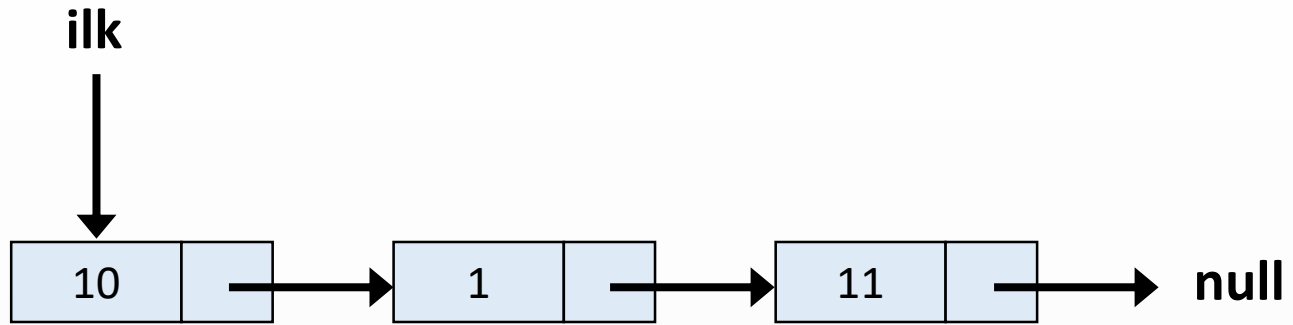


```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```

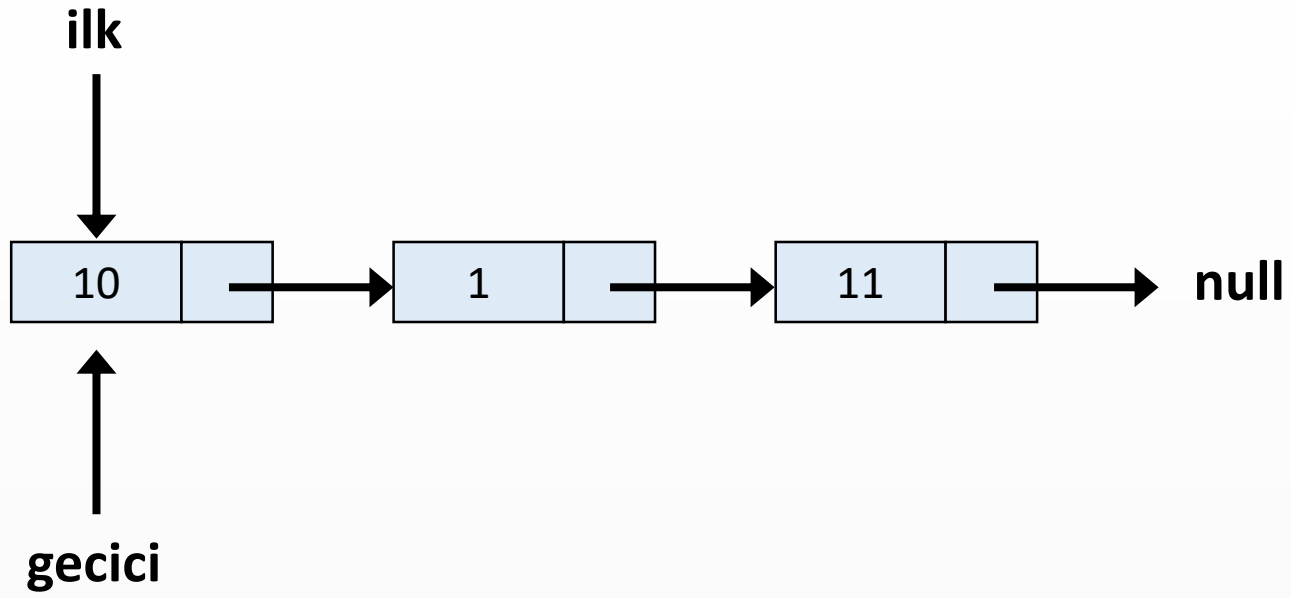


→

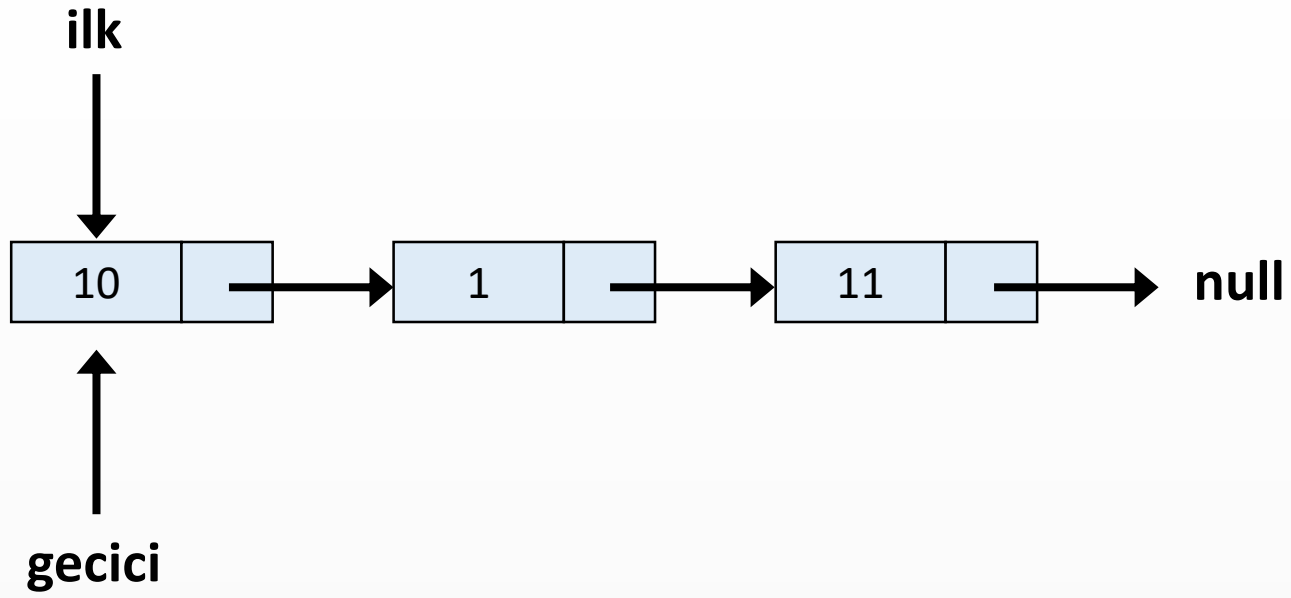
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



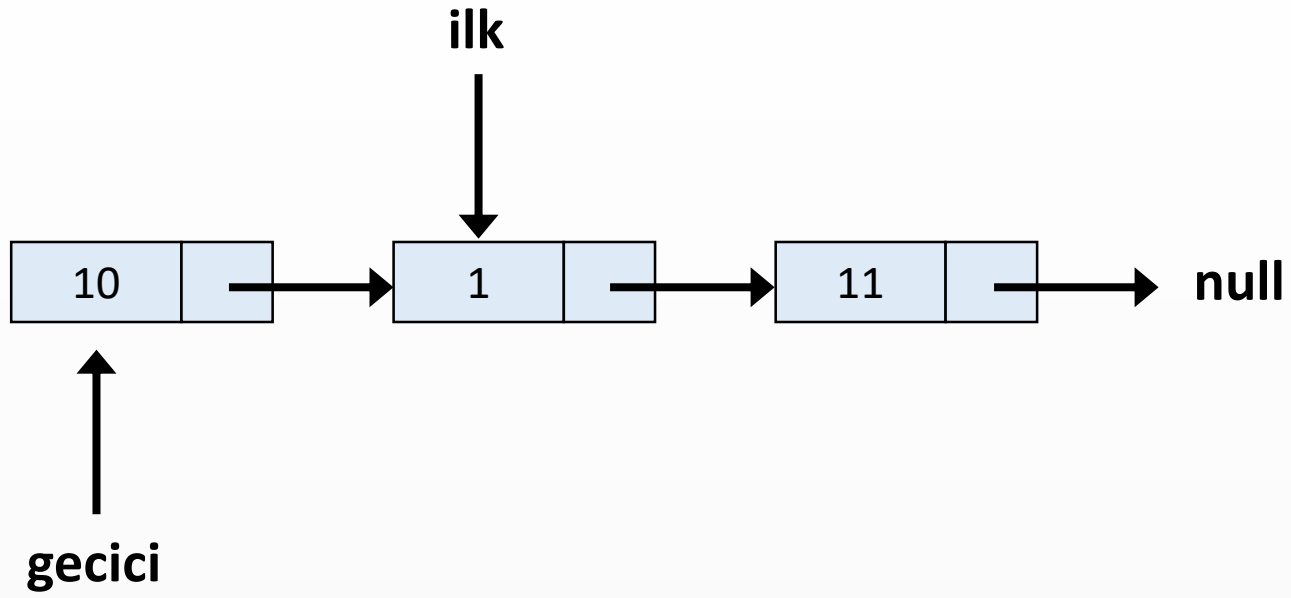
```
if(ilk == null) {  
    return null;  
}  
→ Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



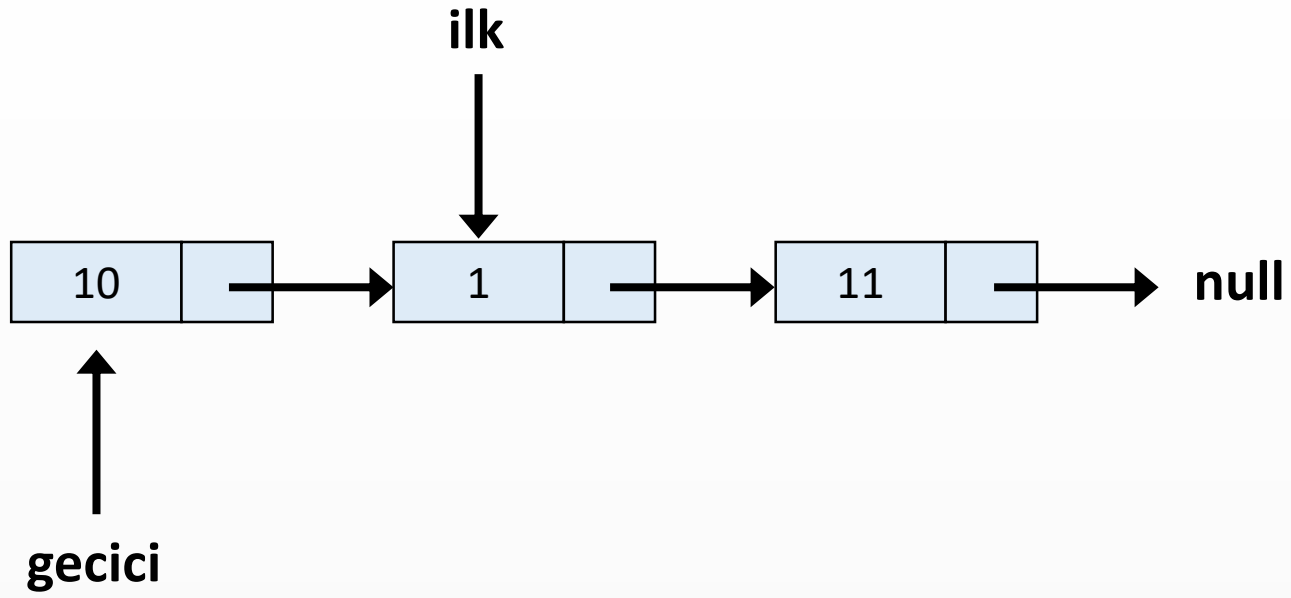
```
if(ilk == null) {  
    return null;  
}  
→ Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



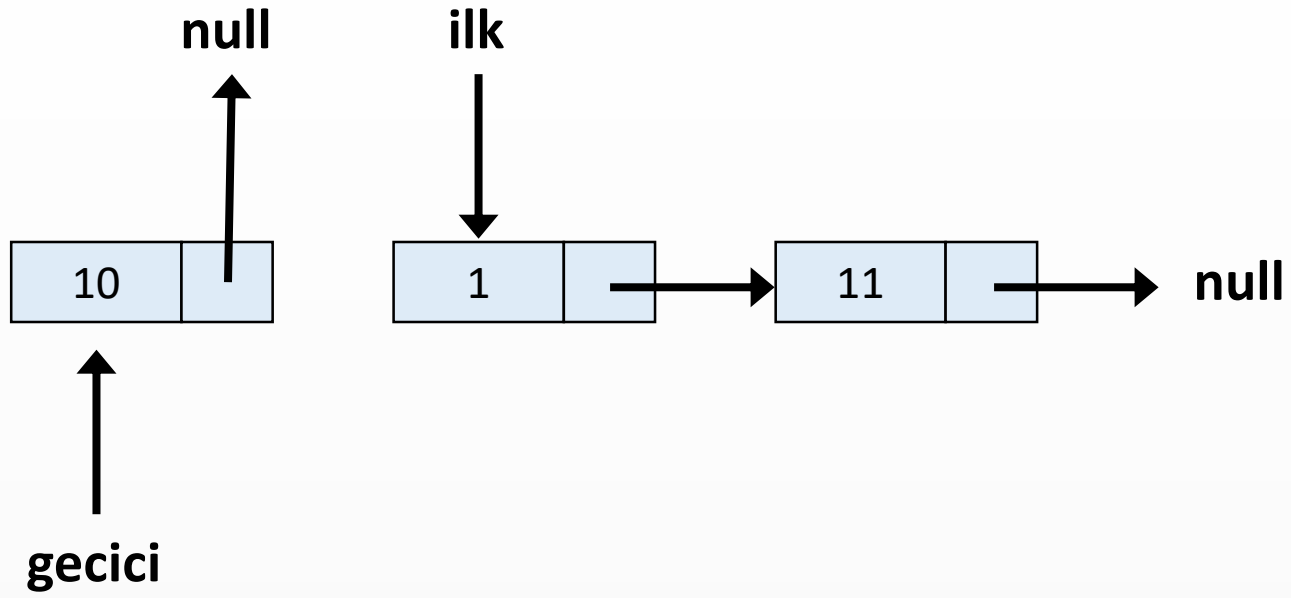
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



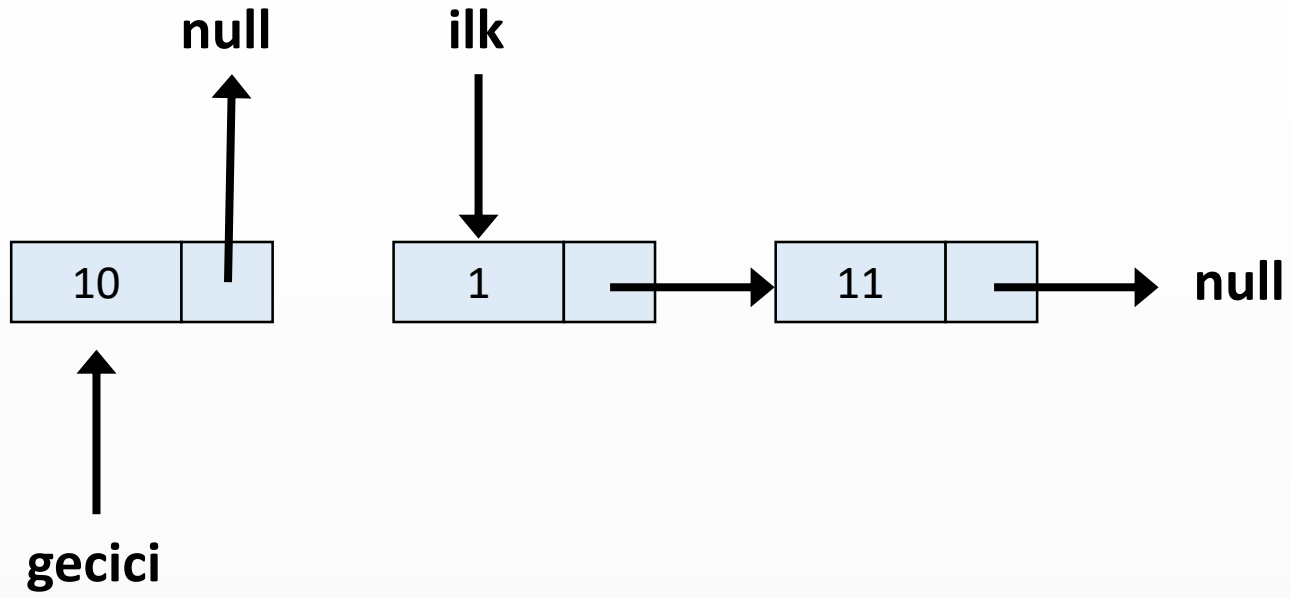
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



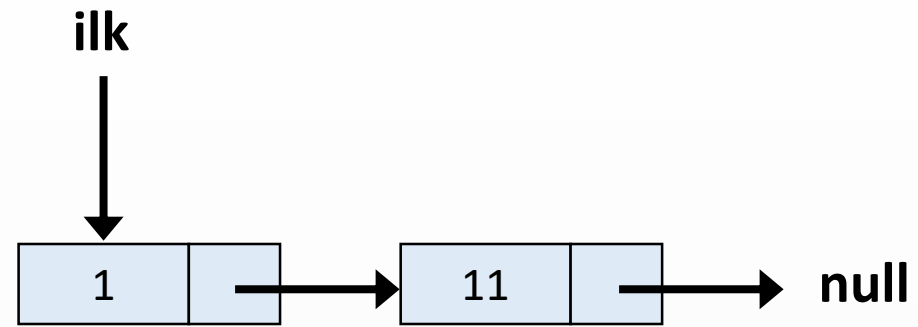
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```



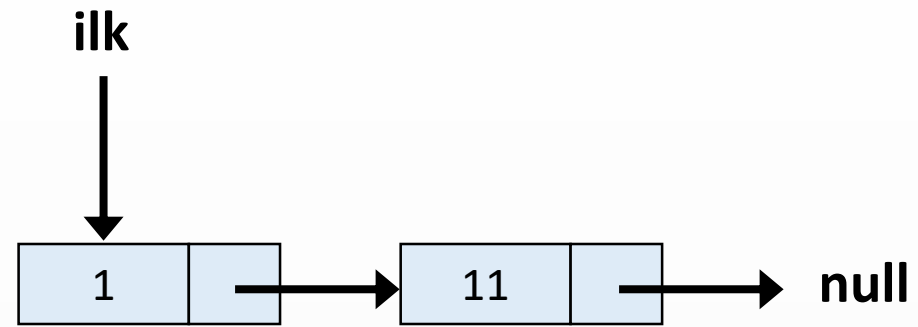
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```

```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```

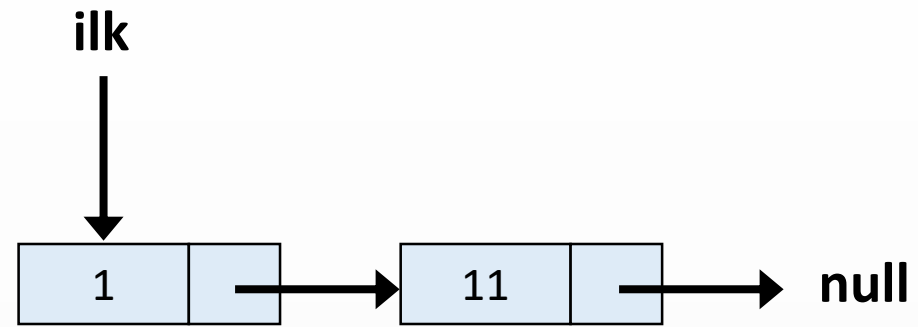


```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```

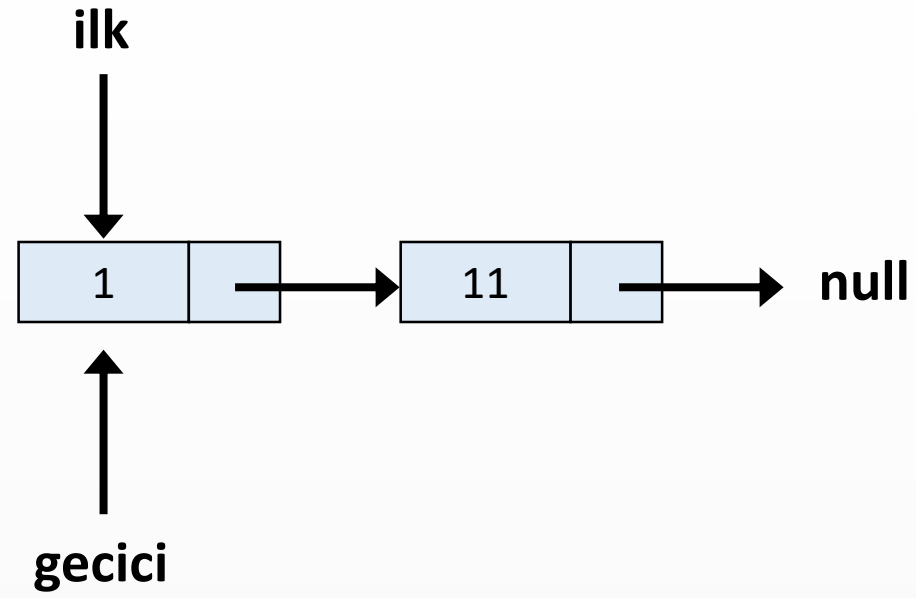


→

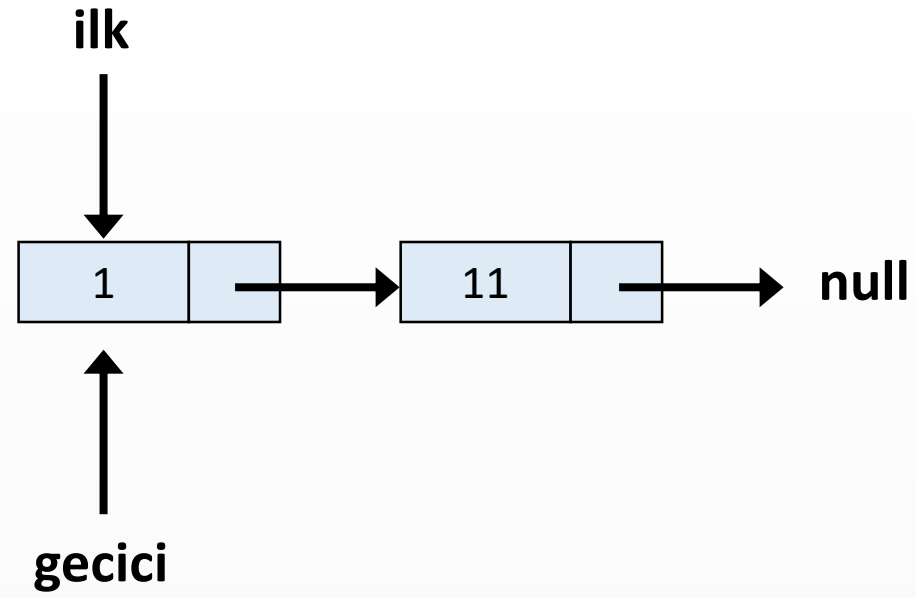
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



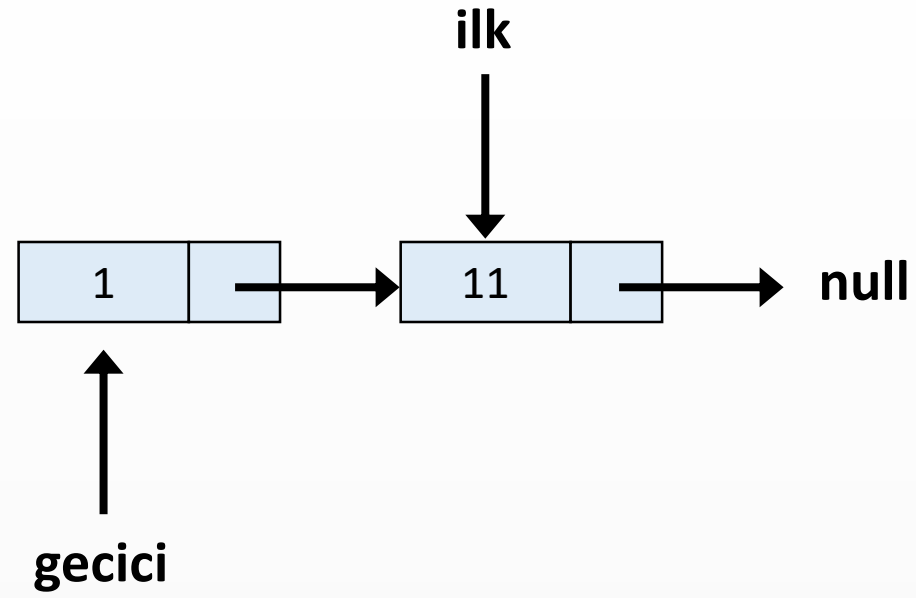
```
if(ilk == null) {  
    return null;  
}  
→ Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



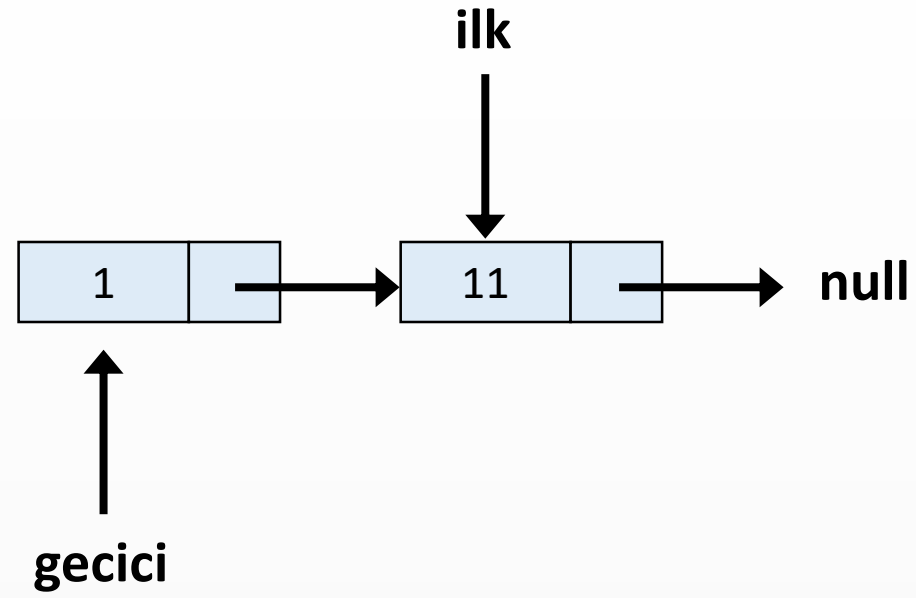
```
if(ilk == null) {  
    return null;  
}  
→ Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



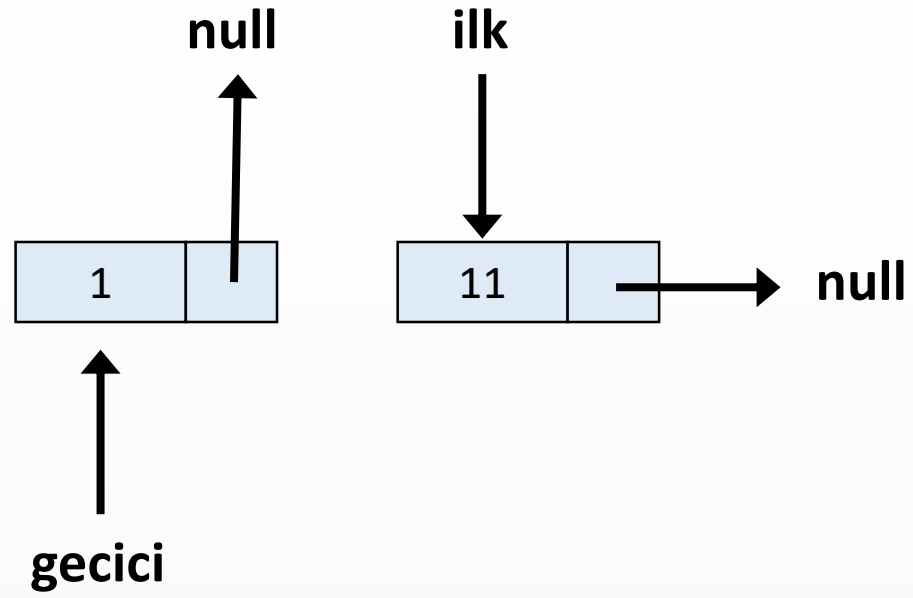
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



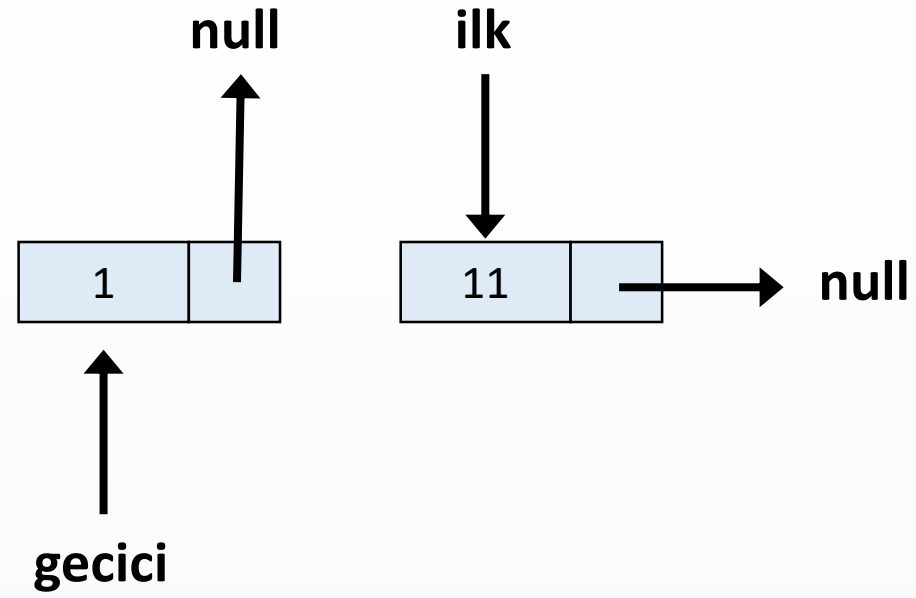
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```

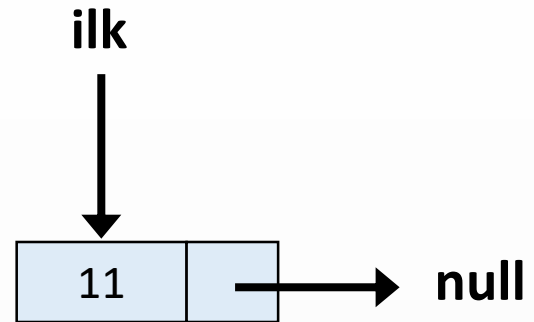



```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```



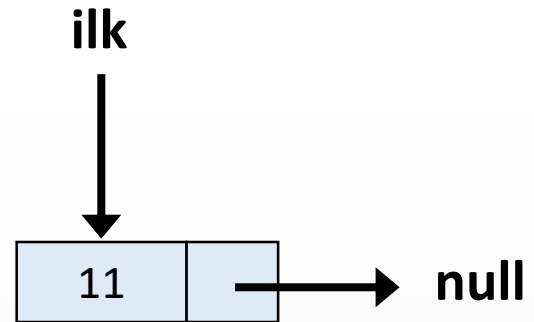
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



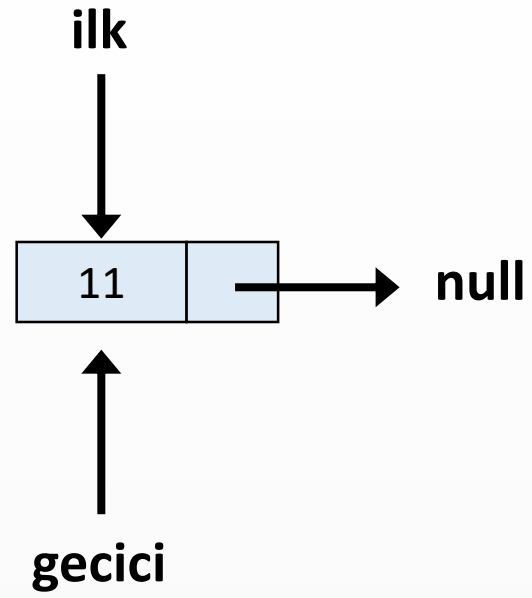


→

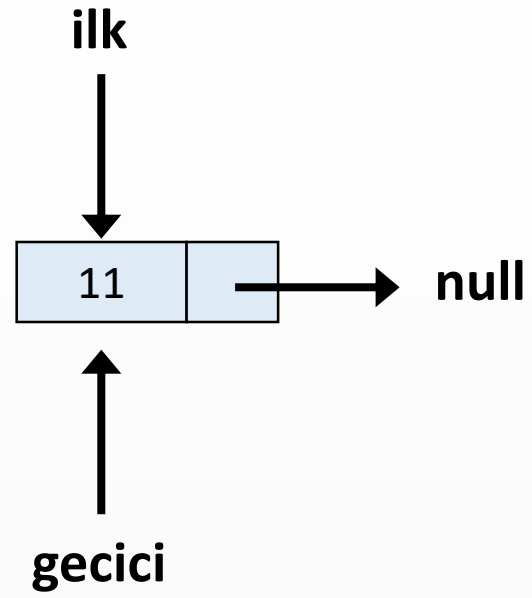
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



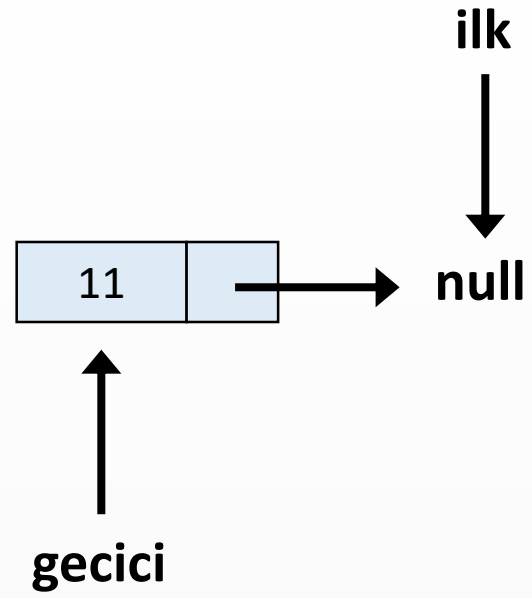
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



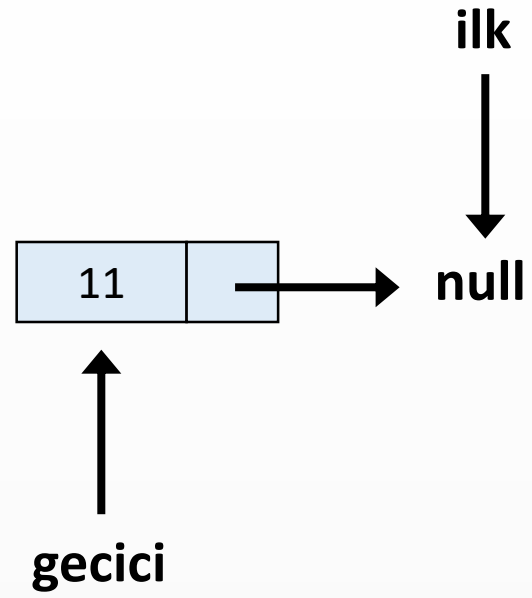
```
if(ilk == null) {  
    return null;  
}  
→ Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



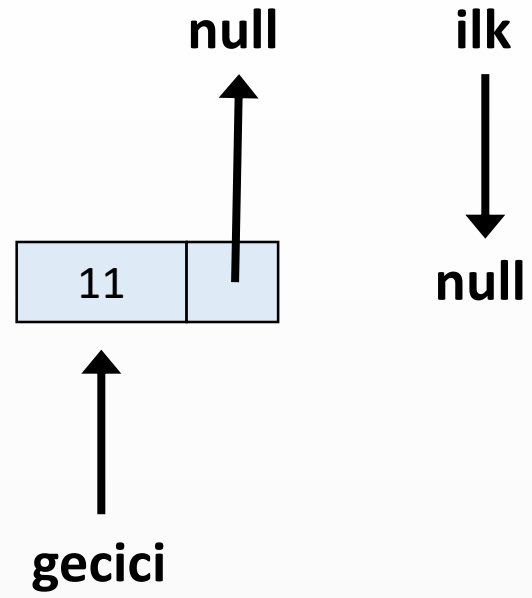
```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



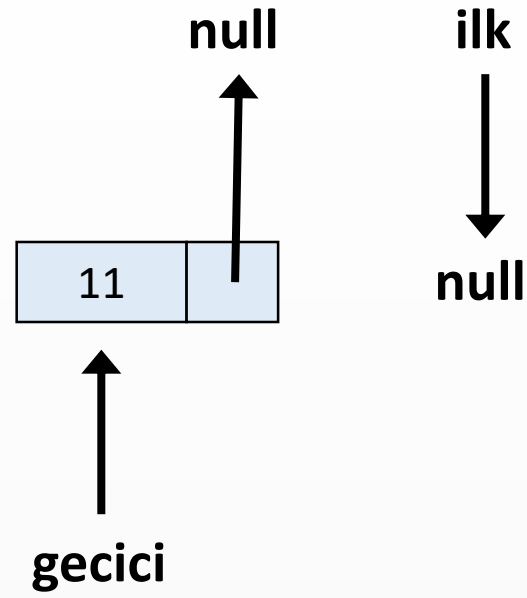
```
if(ilk == null) {
    return null;
}
Dugum gecici = ilk;
ilk = ilk.sonraki;
gecici.sonraki = null;
return gecici;
```



```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```

```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
→ gecici.sonraki = null;  
return gecici;
```



```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```





ilk
↓
null

```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



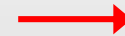
ilk
↓
null

→

```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



ilk
↓
null



```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



ilk
↓
null

```
if(ilk == null) {  
    return null;  
}  
Dugum gecici = ilk;  
ilk = ilk.sonraki;  
gecici.sonraki = null;  
return gecici;
```



Listeden Son Elemanı Silme

```
void sonElemaniSil() {  
    if (bas.sonraki == null) {  
        // Liste sadece bir eleman içeriyorsa, bu elemanı siler  
        bas = null;  
        return;  
    }  
    Dugum gecici = bas;  
    while (gecici.sonraki.sonraki != null) {  
        gecici = gecici.sonraki;  
    }  
    gecici.sonraki = null;  
}
```



Listeden Belirli Bir Elemanı Silme

```
void elemaniSil(int silinecekVeri) {  
    if (bas.veri == silinecekVeri) {  
        bas = bas.sonraki; return;  
    }  
    Dugum onceki = bas;  
    Dugum gecici = bas.sonraki;  
    while (gecici != null) {  
        if (gecici.veri == silinecekVeri) {  
            onceki.sonraki = gecici.sonraki; return;  
        }  
        onceki = gecici;  
        gecici = gecici.sonraki;  
    }  
    System.out.println("Belirtilen veri bulunamadı.");  
}
```




Listede Belirli Bir Elemanı Arama

```
boolean elemaniAra(int arananVeri) {  
    Dugum gecici = bas;  
    int konum = 0;  
    while (gecici != null) {  
        konum++;  
        if (gecici.veri == arananVeri) {  
            System.out.println("Belirtilen eleman " + konum + " konumda bulundu.");  
            return true;  
        }  
        gecici = gecici.sonraki;  
    }  
    System.out.println("Belirtilen eleman bulunamadı.");  
    return false;  
}
```

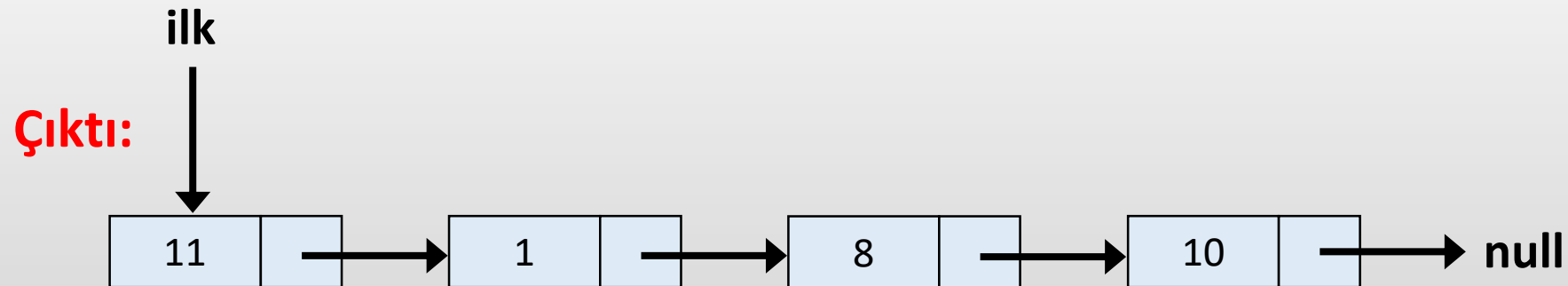
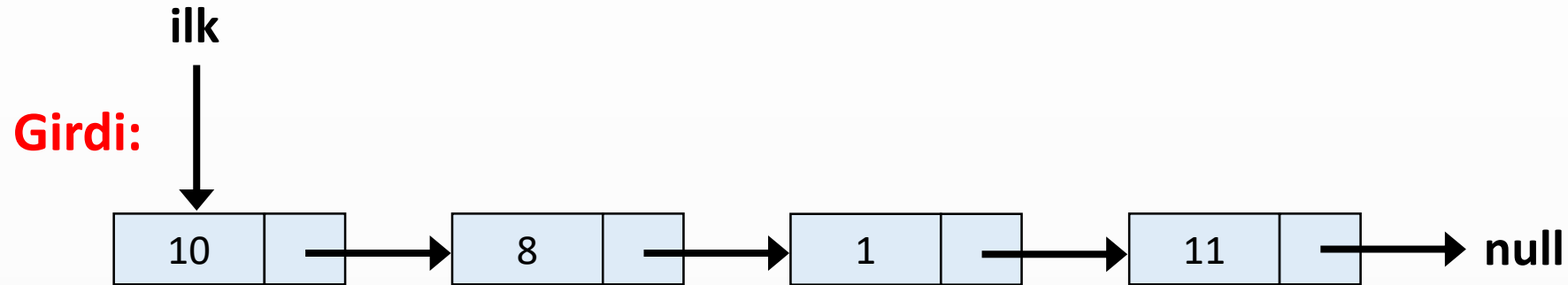


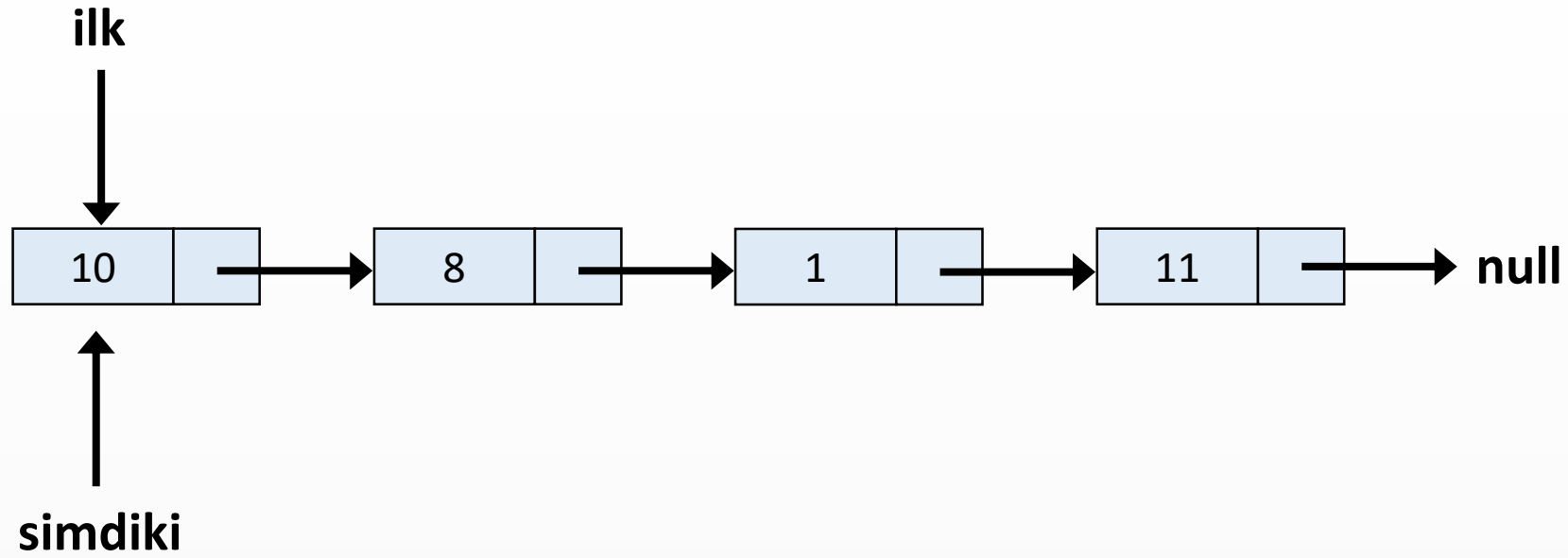
Listeyi Ters Çevirme

```
void listeyiTersCevir() {  
    Dugum onceki = null;  
    Dugum suanki = bas;  
    Dugum sonraki = null;  
  
    while (suanki != null) {  
        sonraki = suanki.sonraki;  
        suanki.sonraki = onceki;  
        onceki = suanki;  
        suanki = sonraki;  
    }  
    bas = onceki;  
}
```

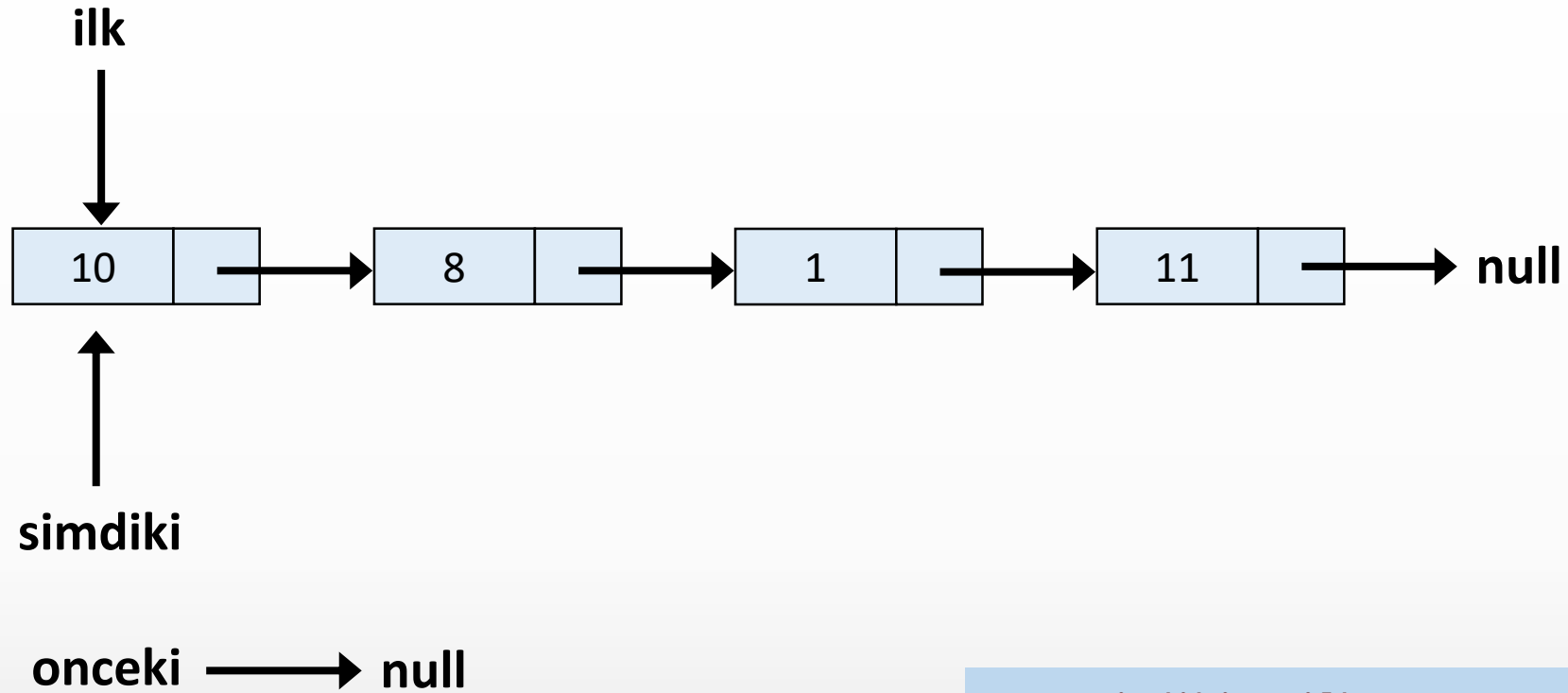


Tek Yönlü Bağlı Listeyi Tersine Çevirme

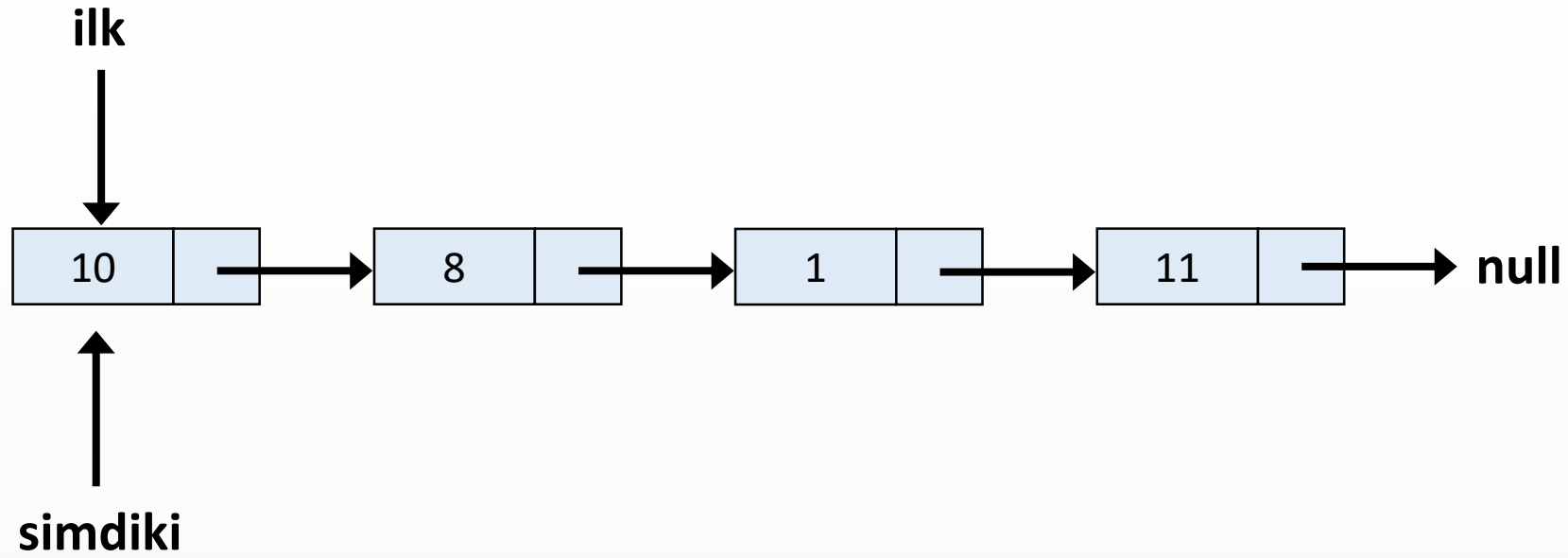




```
→ Dugum simdiki = ilk;
   Dugum onceki = null;
   Dugum sonraki = null;
   while(simdiki != null) {
       sonraki = simdiki.sonraki;
       simdiki.sonraki = onceki;
       onceki = simdiki;
       simdiki = sonraki;
   }
   return onceki;
```



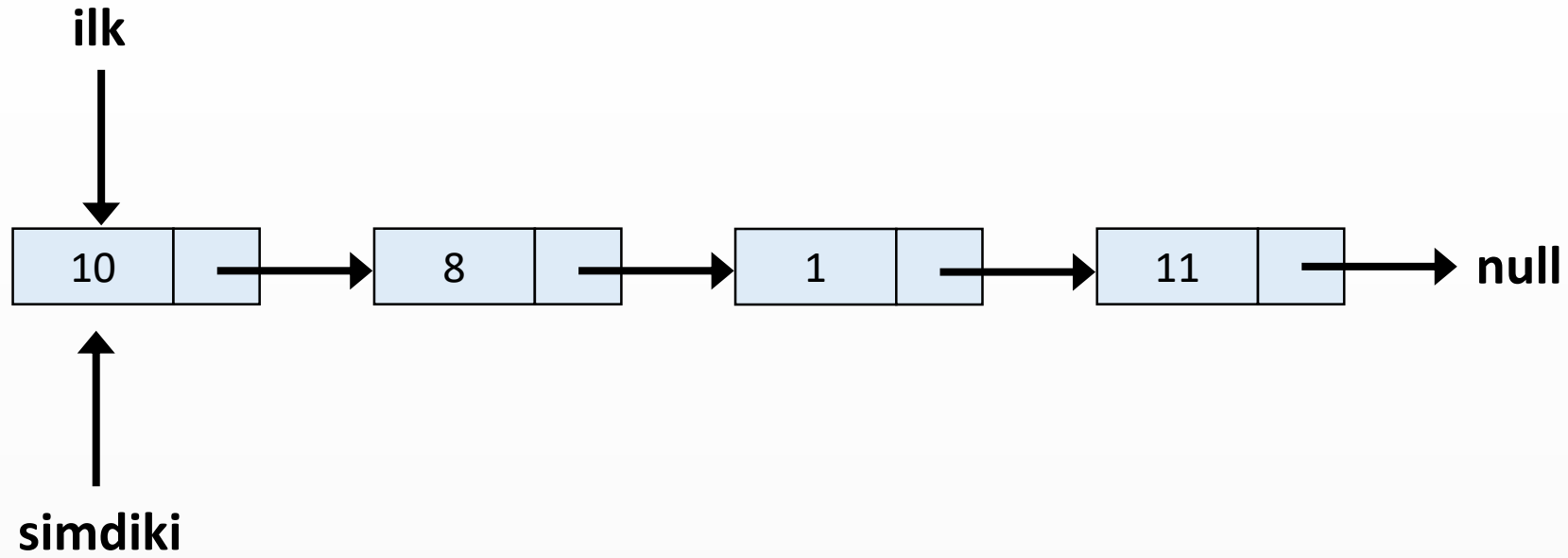
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



onceki → null

sonraki → null

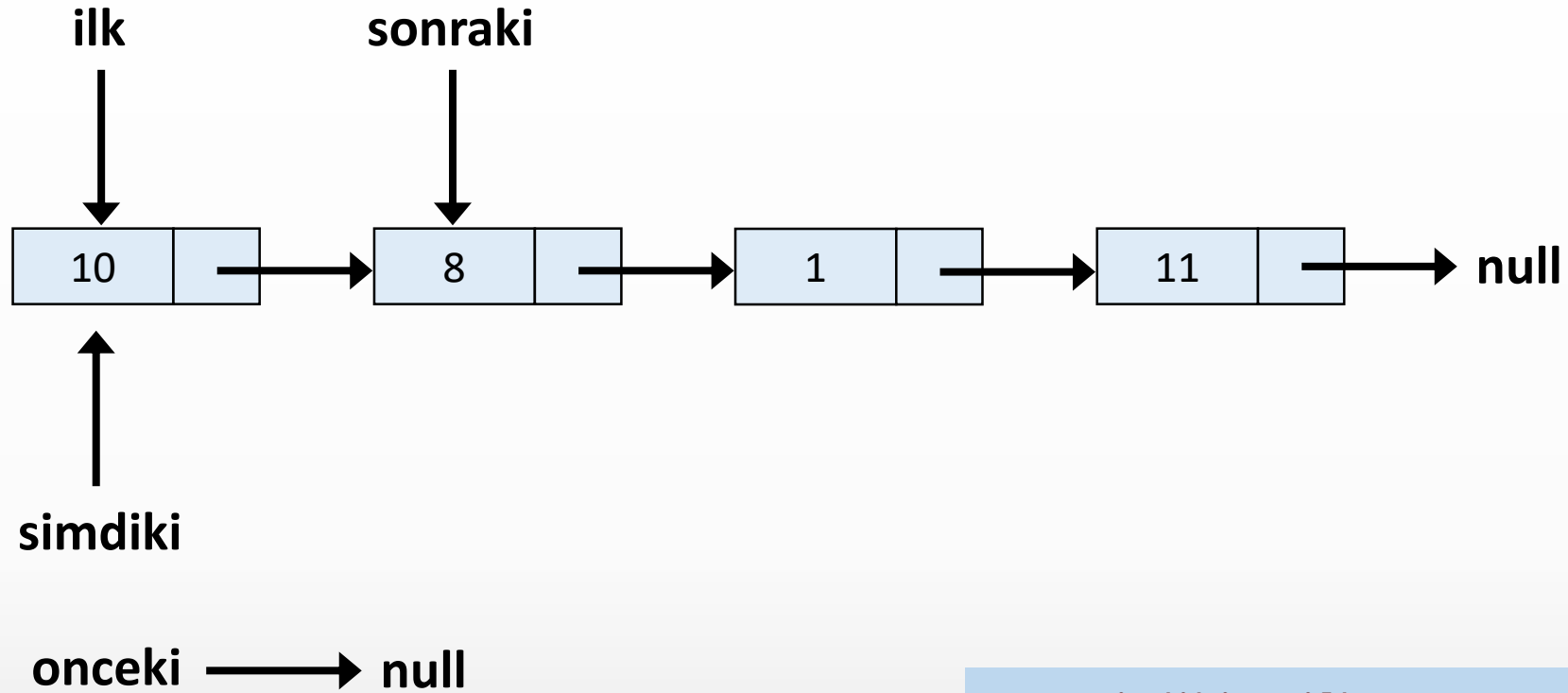
```
Dugum simdiki = ilk;
Dugum onceki = null;
Dugum sonraki = null;
while(simdiki != null) {
    sonraki = simdiki.sonraki;
    simdiki.sonraki = onceki;
    onceki = simdiki;
    simdiki = sonraki;
}
return onceki;
```



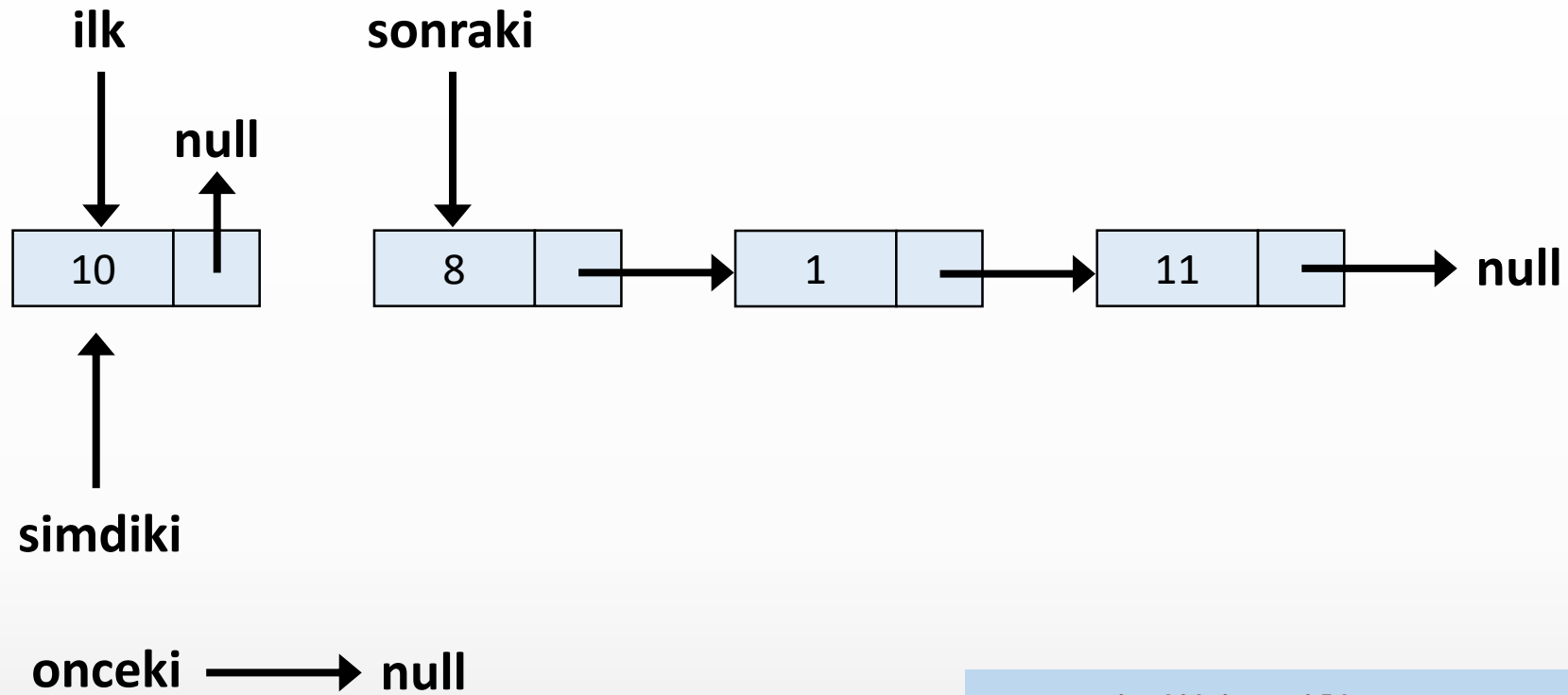
onceki → null

sonraki → null

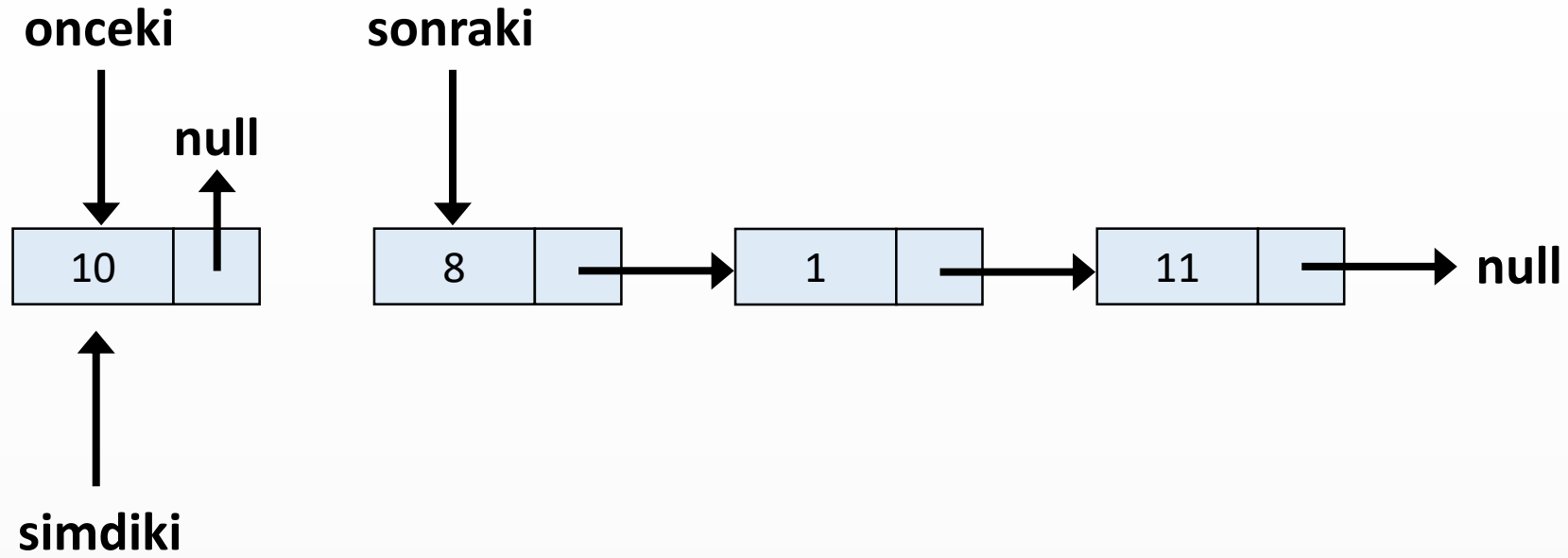
```
Dugum simdiki = ilk;
Dugum onceki = null;
Dugum sonraki = null;
→ while(simdiki != null) {
    sonraki = simdiki.sonraki;
    simdiki.sonraki = onceki;
    onceki = simdiki;
    simdiki = sonraki;
}
return onceki;
```



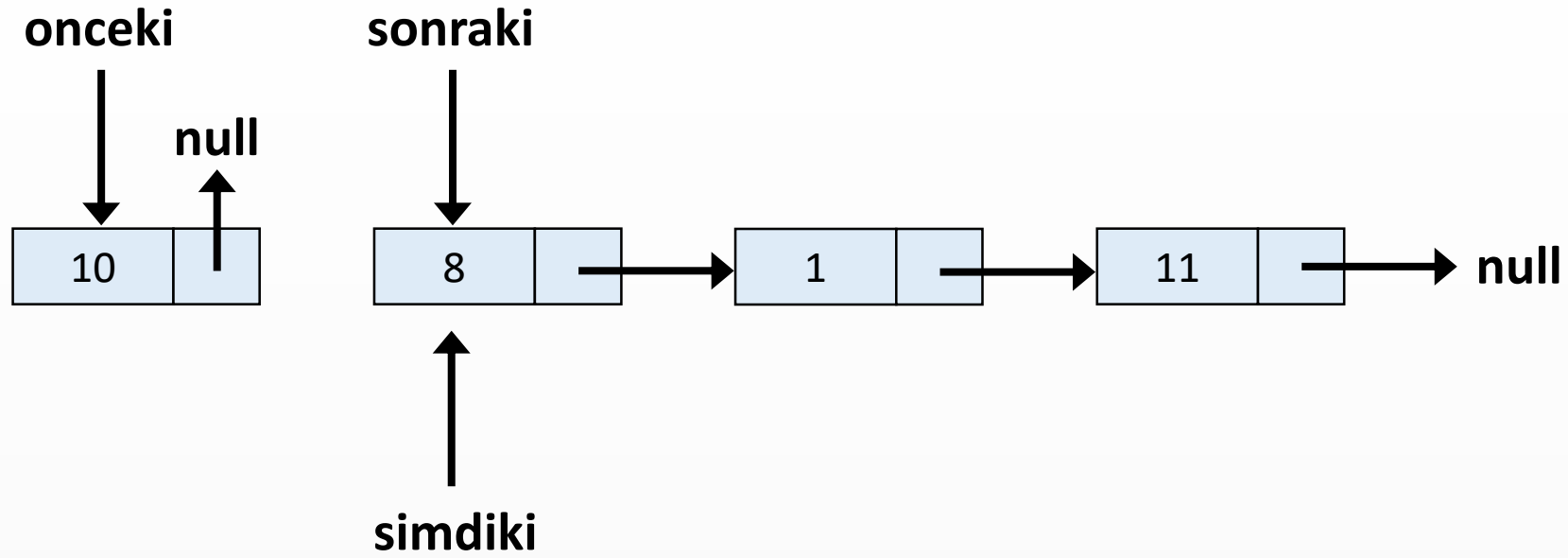
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```

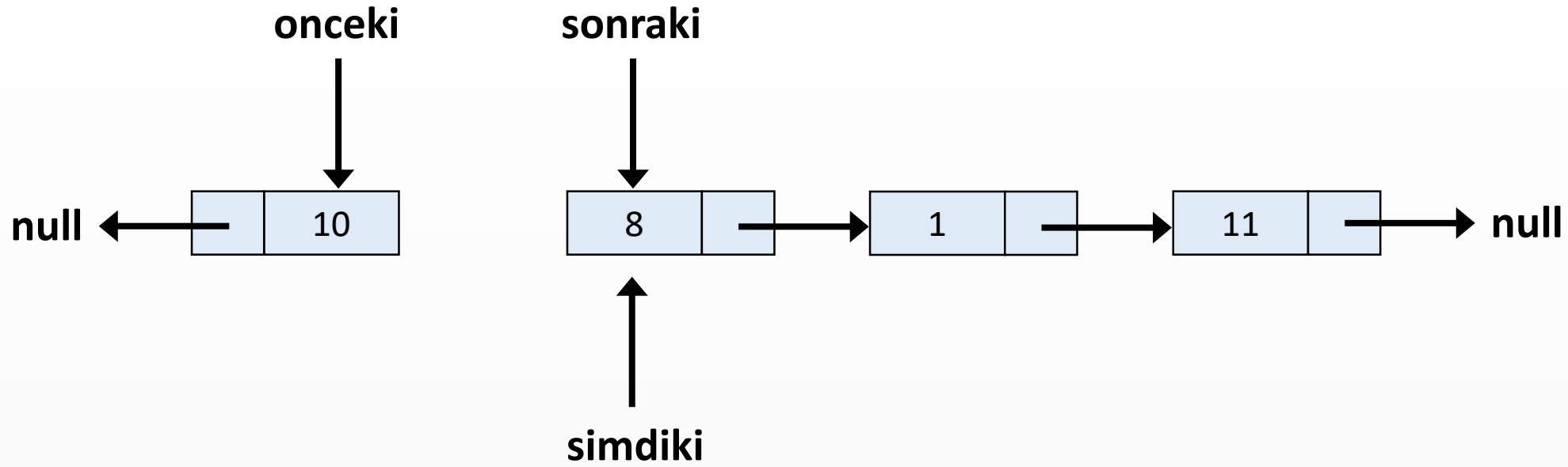
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



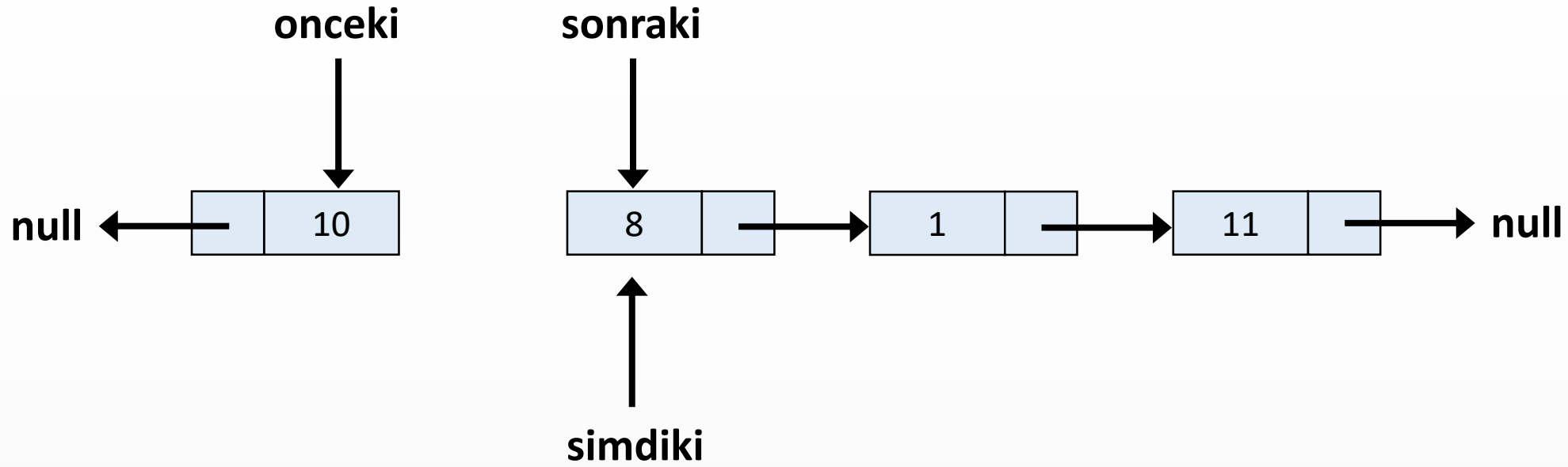
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



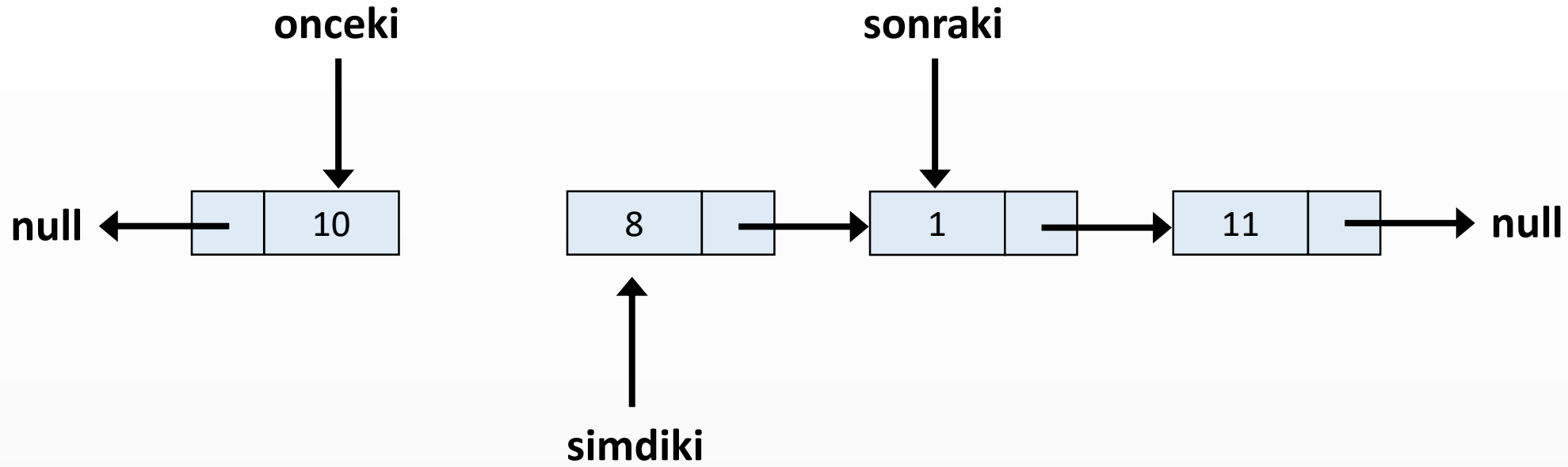
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



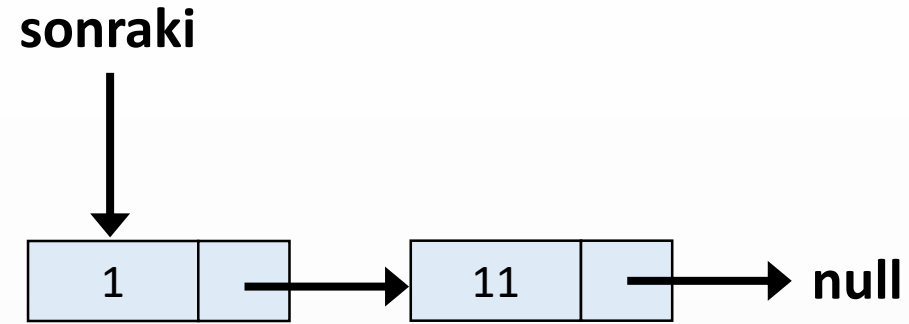
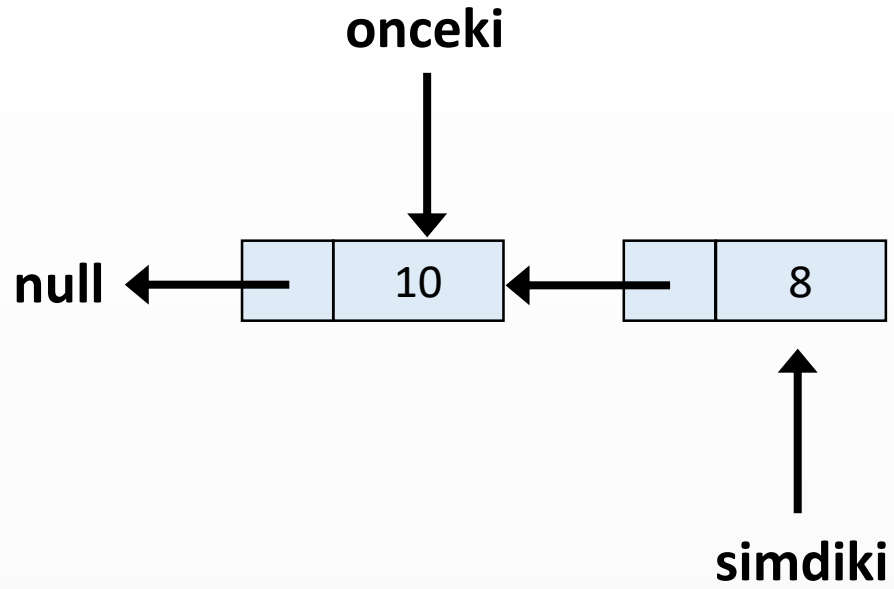
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



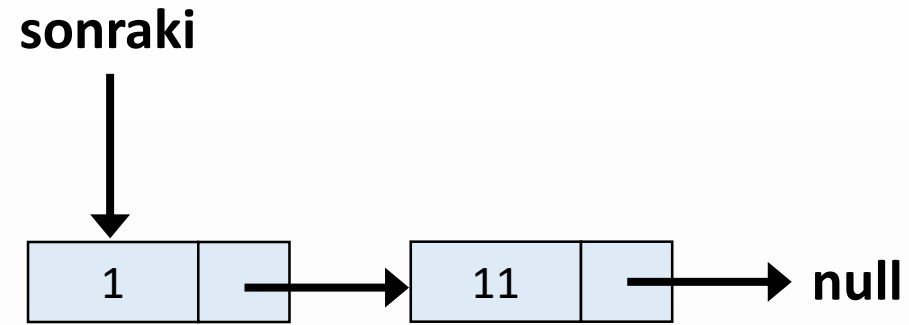
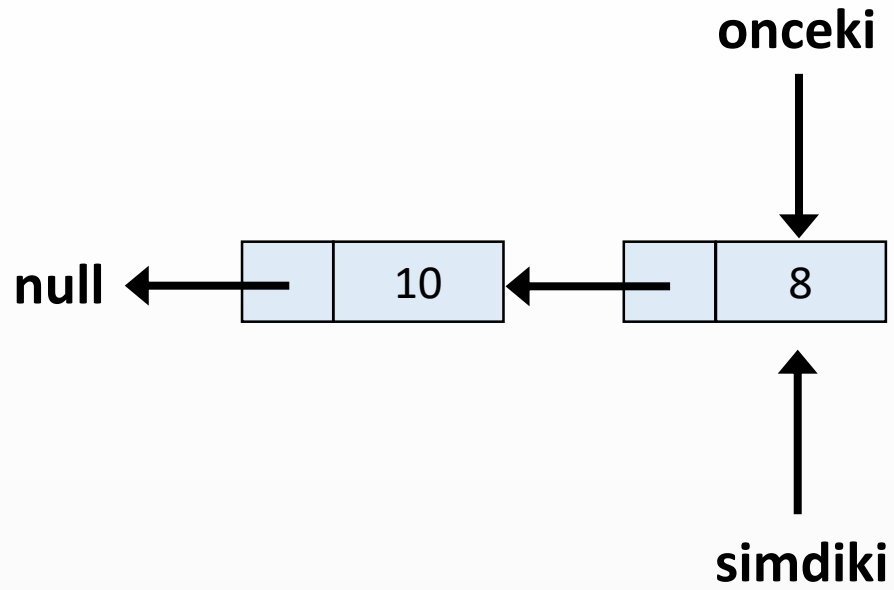
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
→ while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



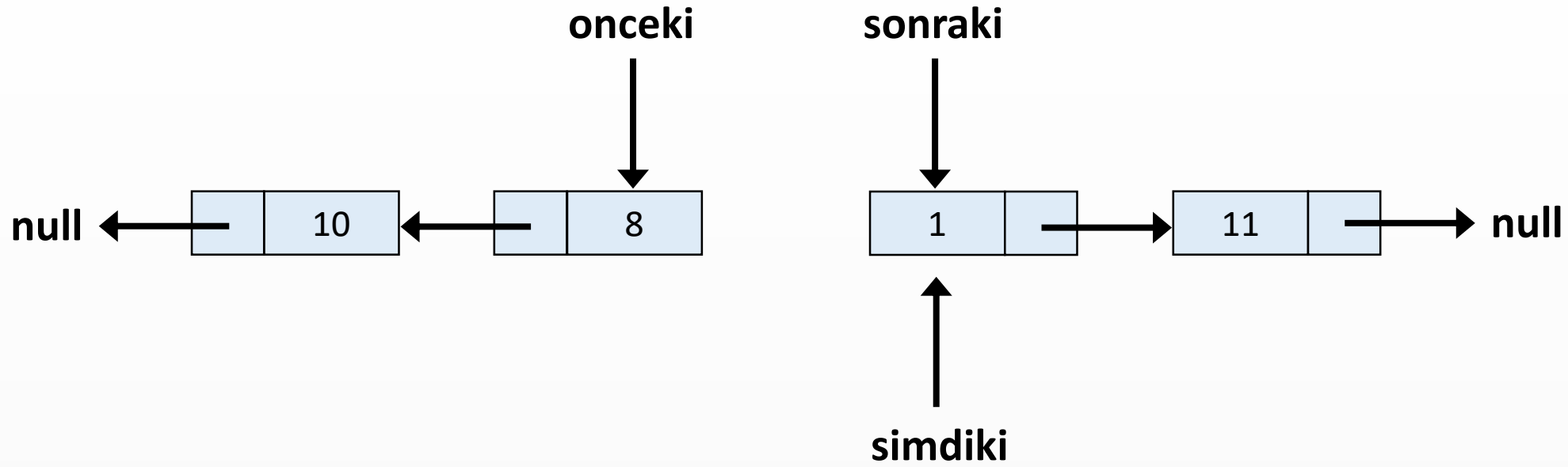
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



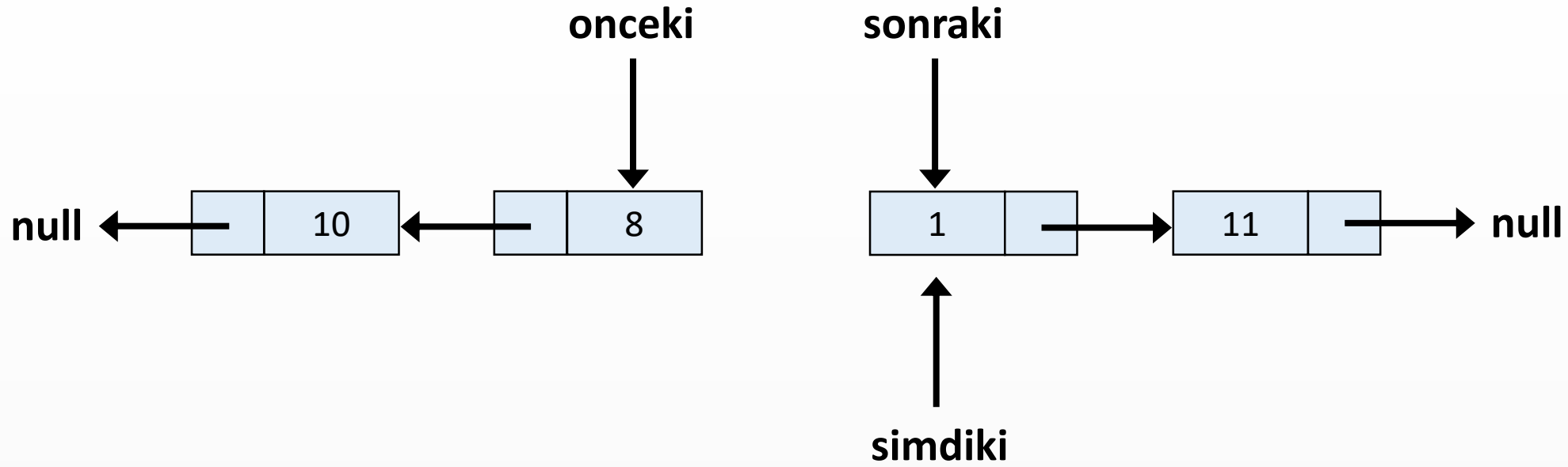
```
Dugum simdiki = ilk;
Dugum onceki = null;
Dugum sonraki = null;
while(simdiki != null) {
    sonraki = simdiki.sonraki;
    simdiki.sonraki = onceki;
    onceki = simdiki;
    simdiki = sonraki;
}
return onceki;
```



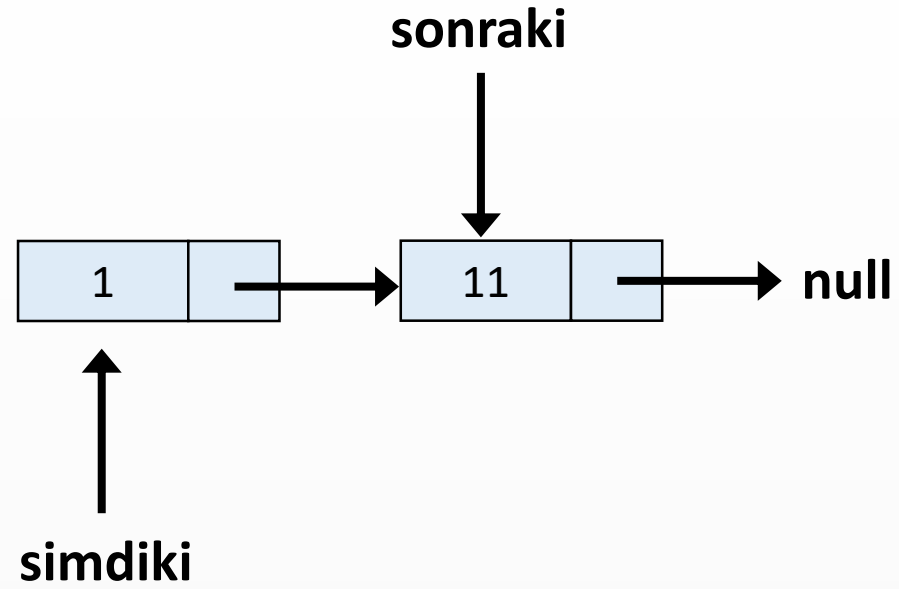
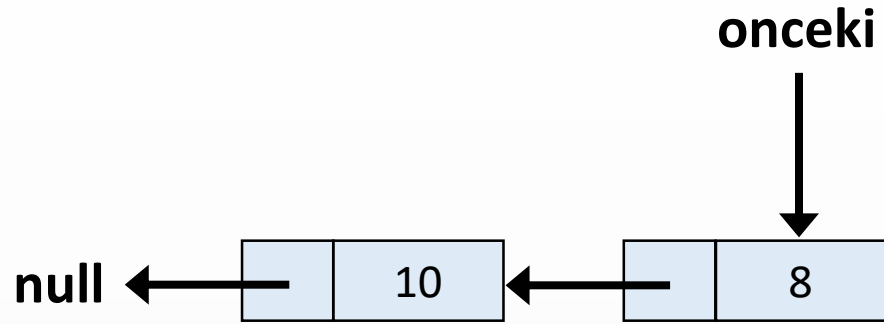
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```

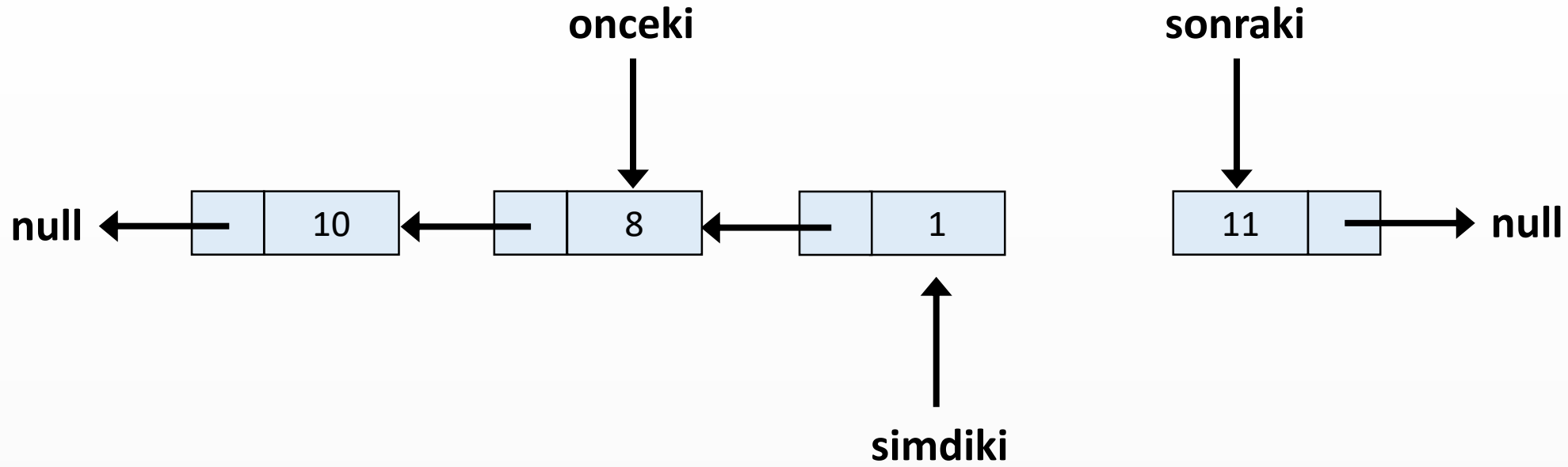
```
Dugum simdiki = ilk;
Dugum onceki = null;
Dugum sonraki = null;
while(simdiki != null) {
    sonraki = simdiki.sonraki;
    simdiki.sonraki = onceki;
    onceki = simdiki;
    simdiki = sonraki;
}
return onceki;
```



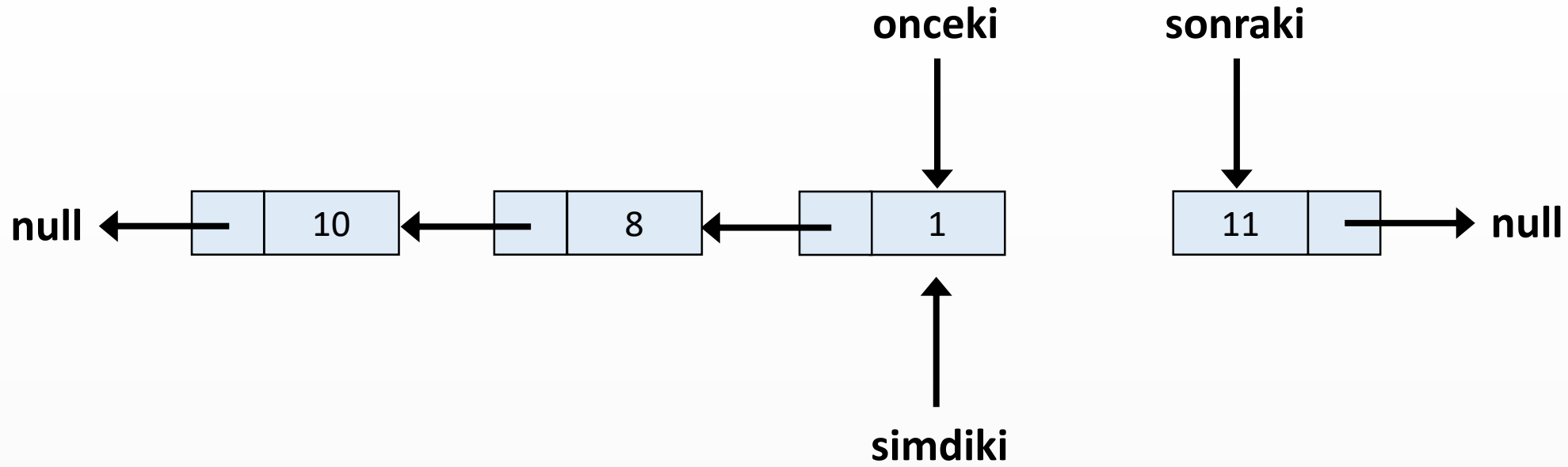
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
→ while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



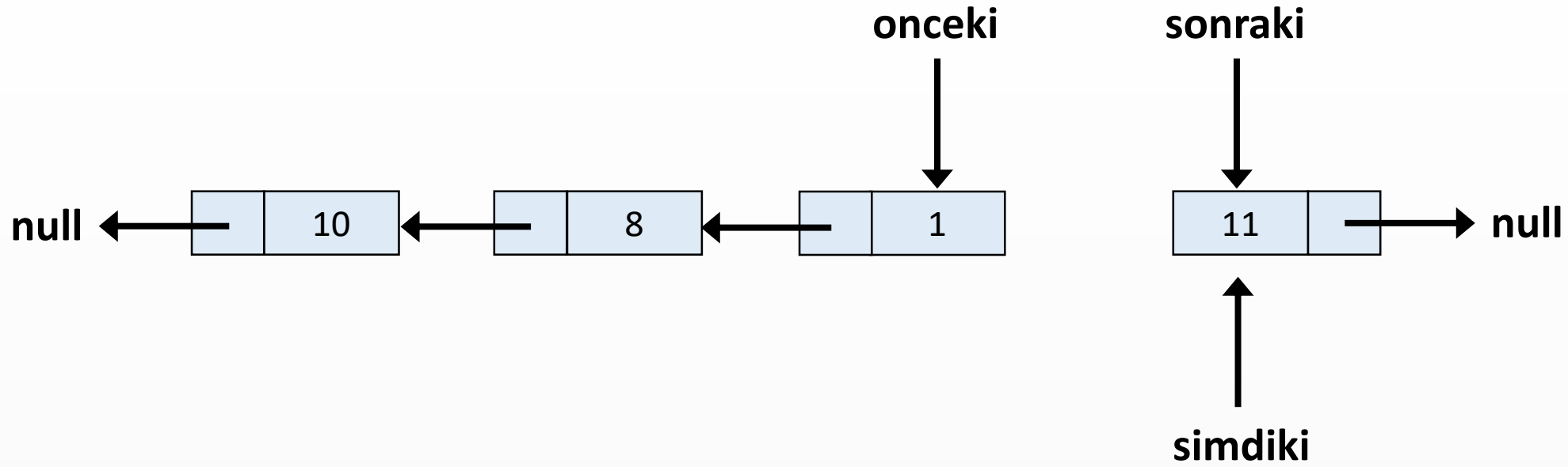
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



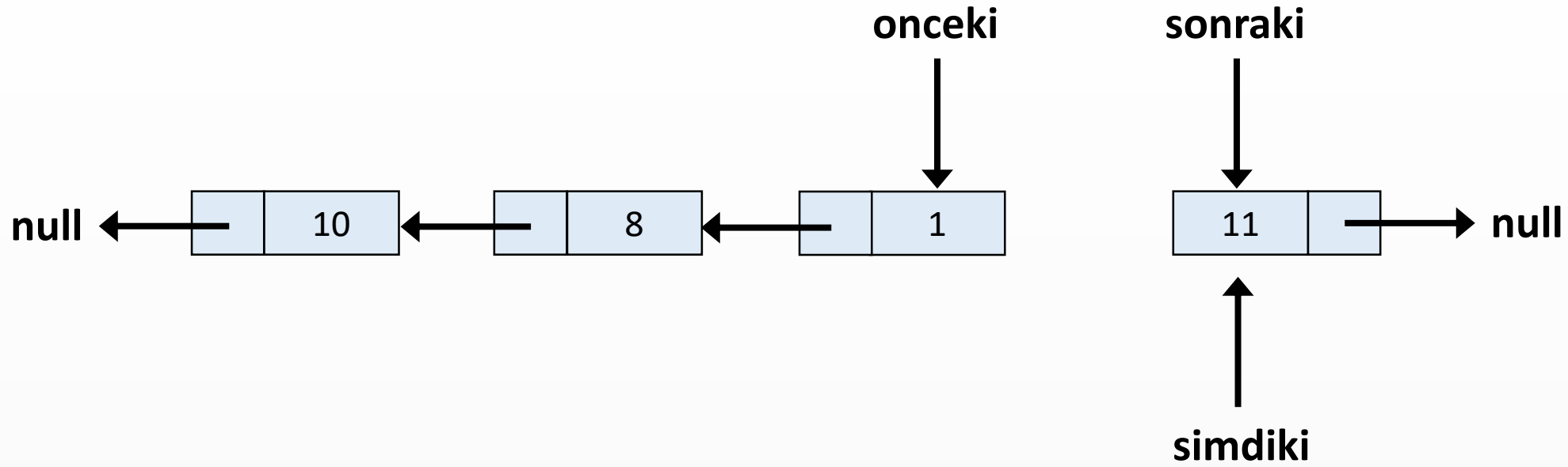
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



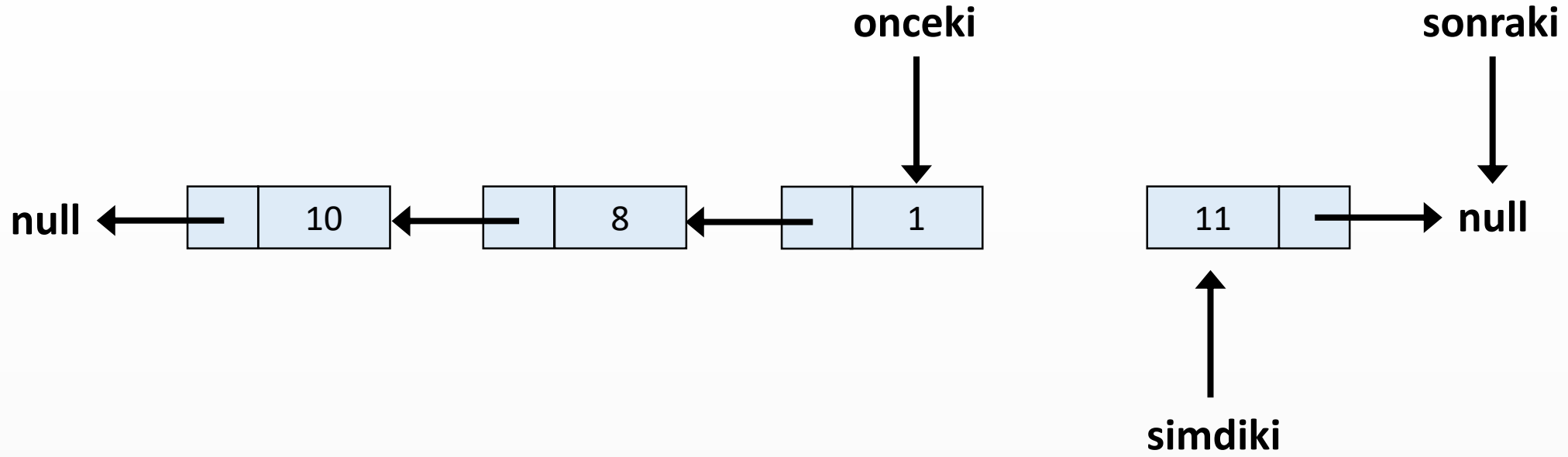
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



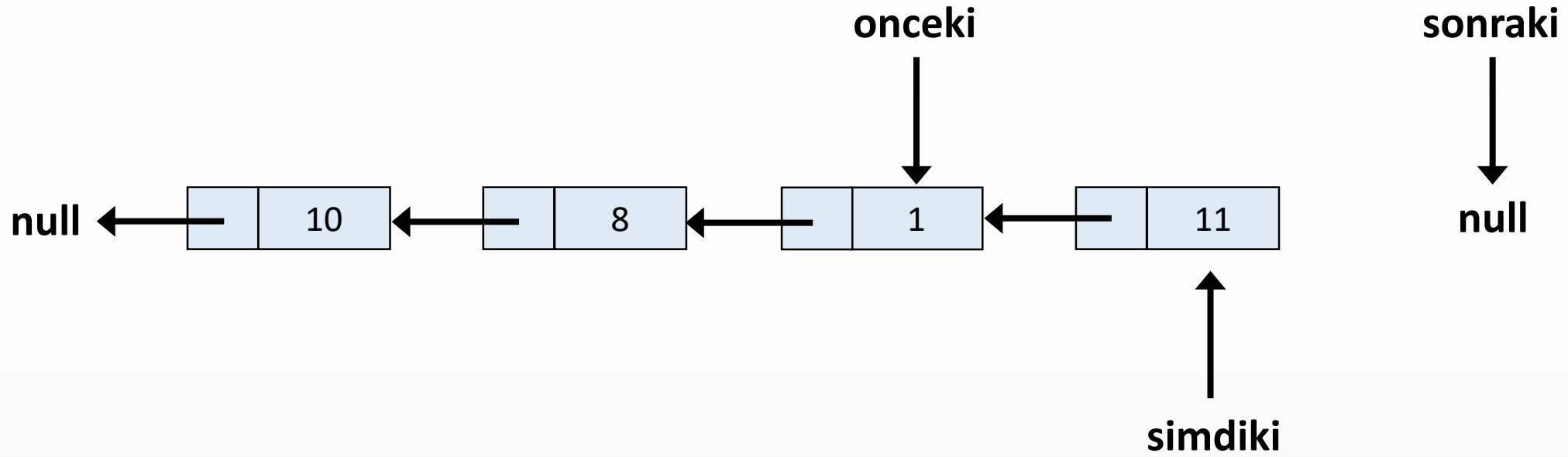
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



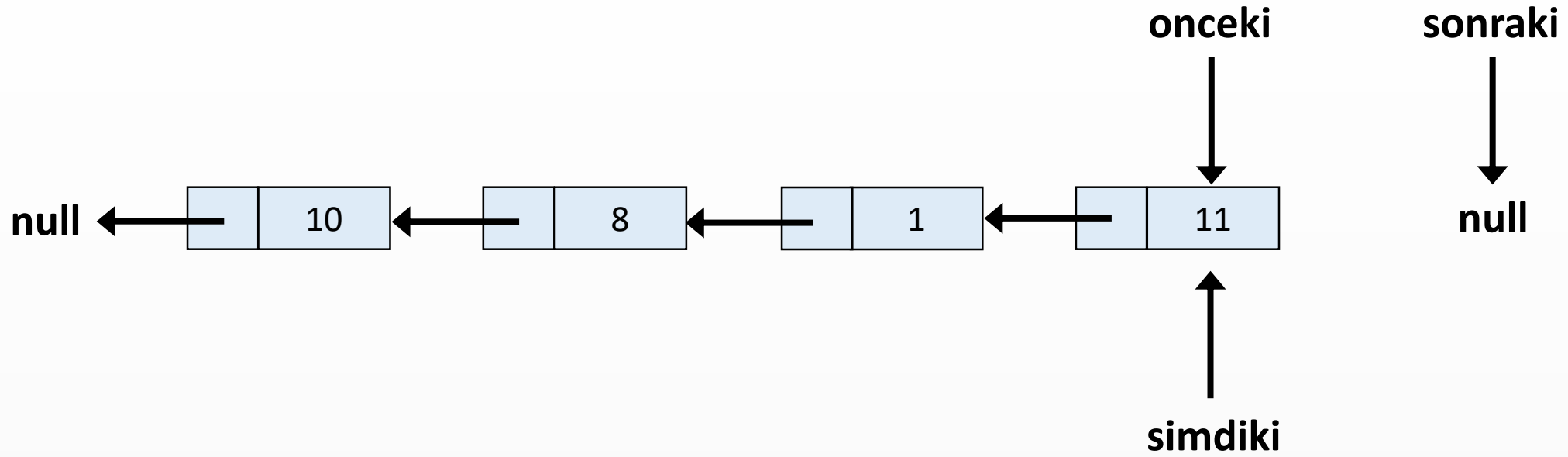
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
→ while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



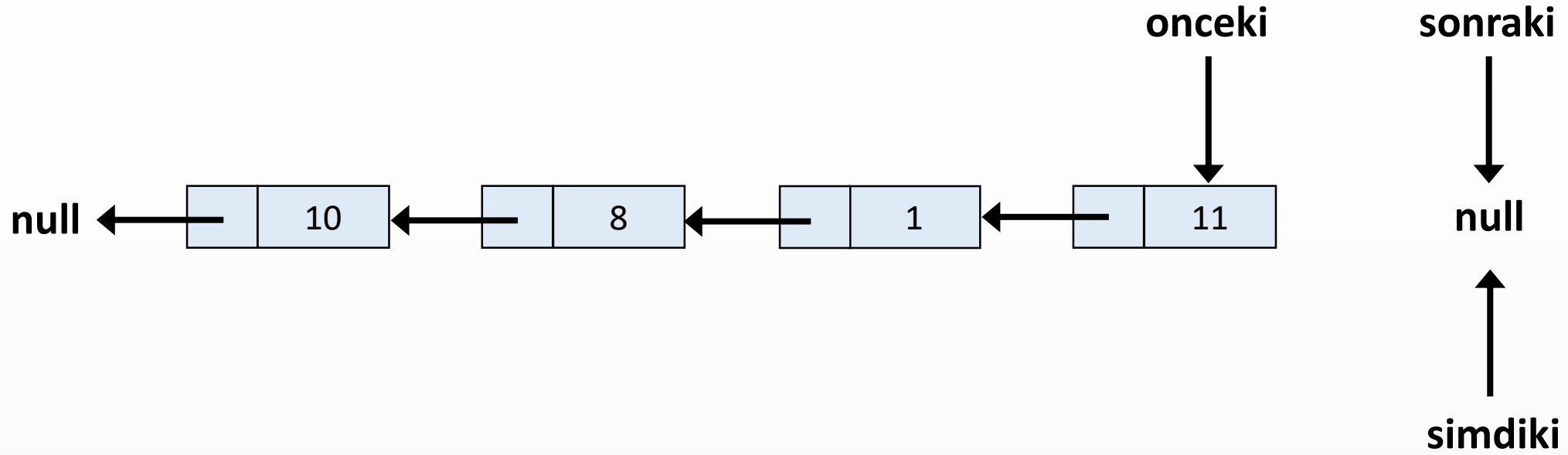
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```

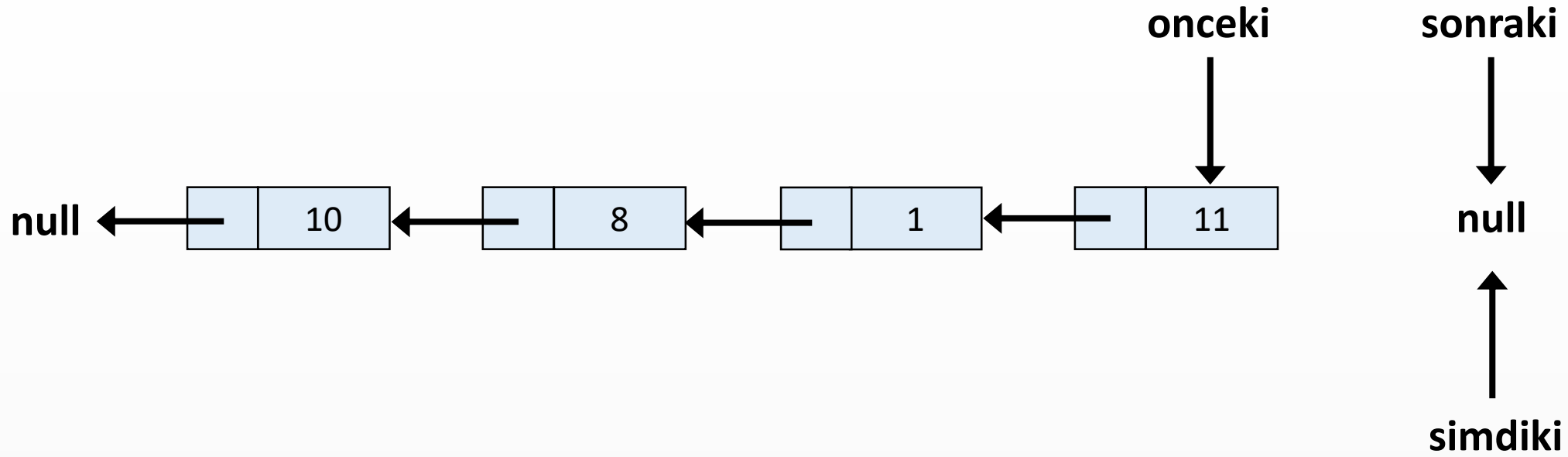
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



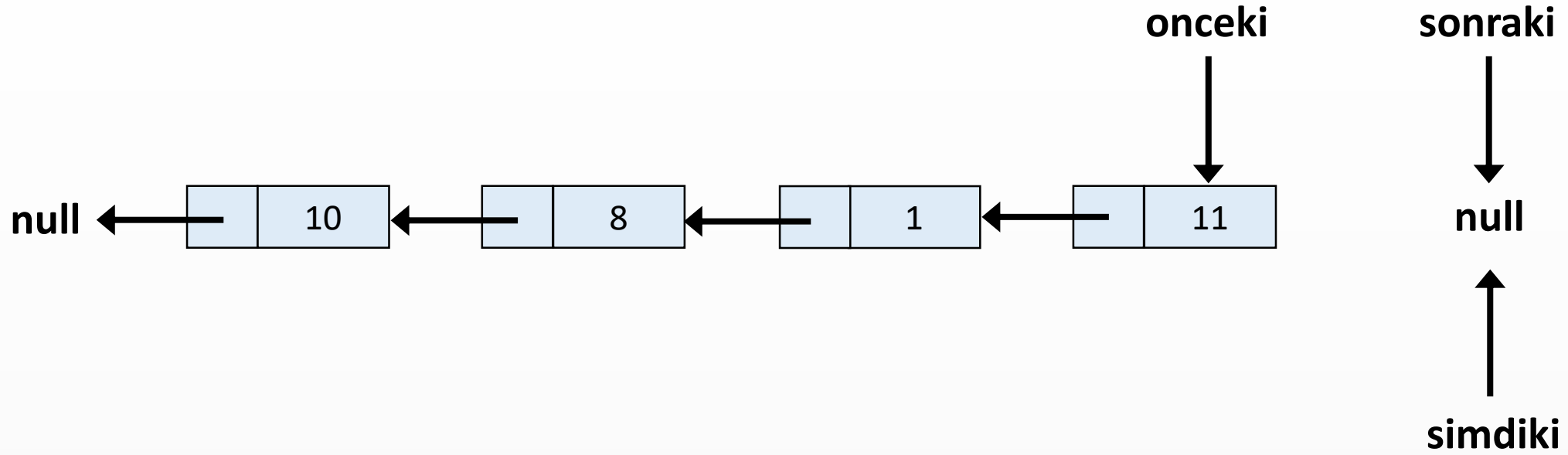
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



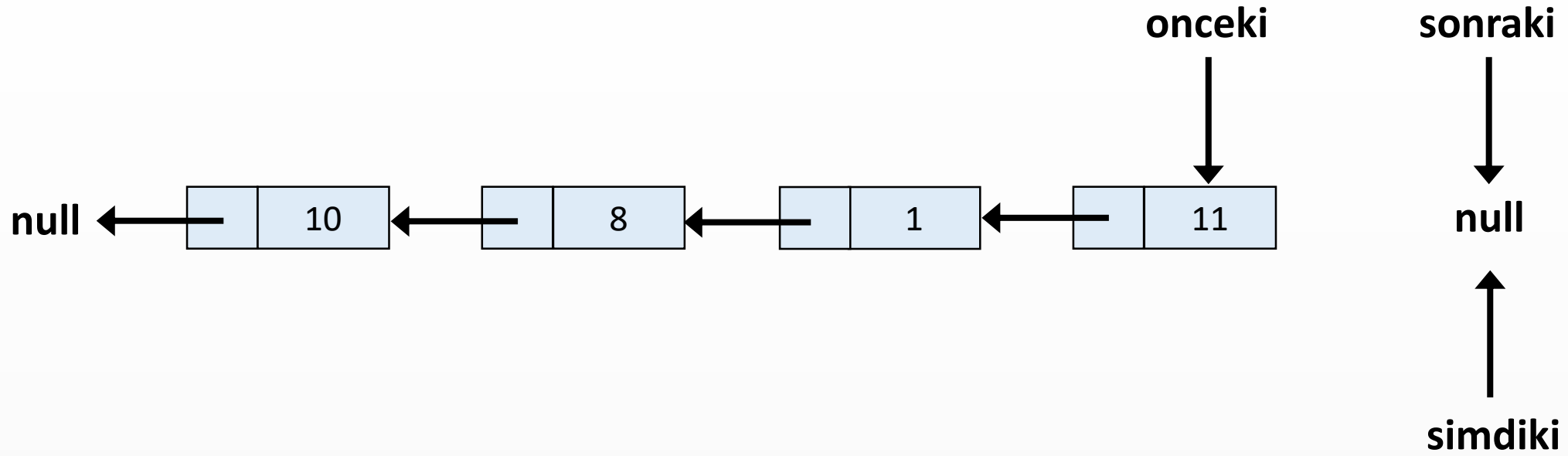
```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
→ while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
→ return onceki;
```



```
Dugum simdiki = ilk;  
Dugum onceki = null;  
Dugum sonraki = null;  
while(simdiki != null) {  
    sonraki = simdiki.sonraki;  
    simdiki.sonraki = onceki;  
    onceki = simdiki;  
    simdiki = sonraki;  
}  
return onceki;
```



Listede Orta Elemanı Bulma

```
int ortaElemaniBul() {  
    Dugum yavas = bas;  
    Dugum hizli = bas;  
  
    while (hizli != null && hizli.sonraki != null) {  
        yavas = yavas.sonraki;    // Yavaş pointer bir adım ilerler  
        hizli = hizli.sonraki.sonraki; // Hızlı pointer iki adım ilerler  
    }  
  
    System.out.println("Orta Eleman: " + yavas.veri);  
    return yavas.veri;  
}
```



Listeyi Kopyalama

```
TekYonluBagliListe kopyala() {  
    TekYonluBagliListe kopyaListe = new TekYonluBagliListe();  
    Dugum gecici = bas;  
  
    while (gecici != null) {  
        kopyaListe.sonaEkle(gecici.veri);  
        gecici = gecici.sonraki;  
    }  
  
    kopyaListe.listeyiYazdir();  
    return kopyaListe;  
}
```




Listeden Tekrarlayanları Kaldırma

```
void tekrarlayanlariKaldir() {  
    TekYonluBagliListe kume = new TekYonluBagliListe();  
    Dugum onceki = null;  
    Dugum gecici = bas;  
    while (gecici != null) {  
        int veri = gecici.veri;  
        if (kume.elemanAra(veri)) { // Tekrarlayan elemanı listeden çıkar  
            onceki.sonraki = gecici.sonraki;  
        } else {  
            kume.sonaEkle(veri);  
            onceki = gecici;  
        }  
        gecici = gecici.sonraki;  
    }  
}
```



İki Sıralı Listeyi Birleştirme

```
TekYonluBagliListe birlestir(liste1, liste2) {  
    TekYonluBagliListe birlesmisListe = new TekYonluBagliListe();  
    Dugum gecici1 = liste1.bas;  
    Dugum gecici2 = liste2.bas;  
  
    while (gecici1 != null && gecici2 != null) {  
        if (gecici1.veri < gecici2.veri) {  
            birlesmisListe.sonaEkle(gecici1.veri);  
            gecici1 = gecici1.sonraki;  
        } else {  
            birlesmisListe.sonaEkle(gecici2.veri);  
            gecici2 = gecici2.sonraki;  
        }  
    }  
}
```



İki Sıralı Listeyi Birleştirme

```
while (gecici1 != null) {  
    birlesmisListe.sonaEkle(gecici1.veri);  
    gecici1 = gecici1.sonraki;  
}  
  
while (gecici2 != null) {  
    birlesmisListe.sonaEkle(gecici2.veri);  
    gecici2 = gecici2.sonraki;  
}  
  
return birlesmisListe;  
}
```



Tek Yönlü Bağlı Listeyi Sıralama

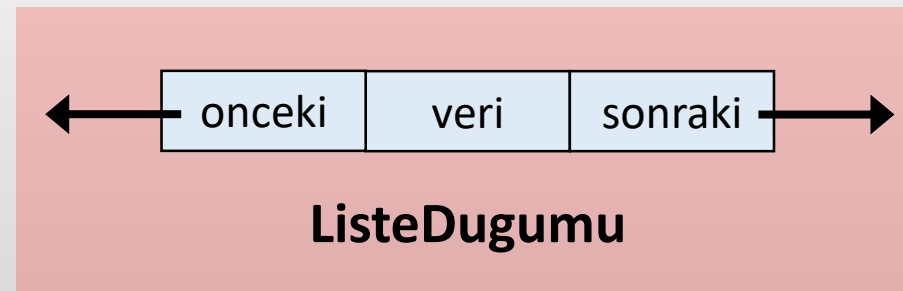
```
void sirala() {  
    Dugum suanki = bas;  
    while (suanki != null) {  
        Dugum diger = suanki.sonraki;  
        while (diger != null) {  
            if (suanki.veri > diger.veri) { // Elemanları değiştir  
                int gecici = suanki.veri;  
                suanki.veri = diger.veri;  
                diger.veri = gecici;  
            }  
            diger = diger.sonraki;  
        }  
        suanki = suanki.sonraki;  
    }  
}
```





Çift Yönlü Bağlı Liste (Doubly Linked List)

- Çift yönlü bağlı listeler, verilerin birbirine çift yönlü bağlandığı veri yapısıdır.
- Her düğümün iki işaretçisi vardır:
 - biri önceki düğümü
 - diğeri ise sonraki düğümü gösterir.
- Verileri her iki yönde dolaşmayı kolaylaştırır.





Çift Yönlü Bağlı Liste Karşılaştırması

- Tek yönlü bağlı listelerde sadece bir sonraki düğümü işaretlerken, çift yönlü bağlı listeler hem bir önceki hem de bir sonraki düğümü işaretler.
- Verilere hem ileri hem de geri yönde erişim sağlar.
- Temel İşlemler
 - **Ekleme:** Düğüm ekleme işlemi her iki yönde yapılır.
 - **Silme:** Bir düğümün hem önceki hem de sonraki düğümlerle bağlantısı kesilir.
 - **Arama:** İleri veya geri yönde dolaşarak veri aranır.



Avantajlar ve Dezavantajlar

- Avantajlar:
 - İleri ve geri yönde dolaşma kolaylığı
- Dezavantajlar:
 - Daha fazla bellek kullanımı
 - Kod karmaşıklığı

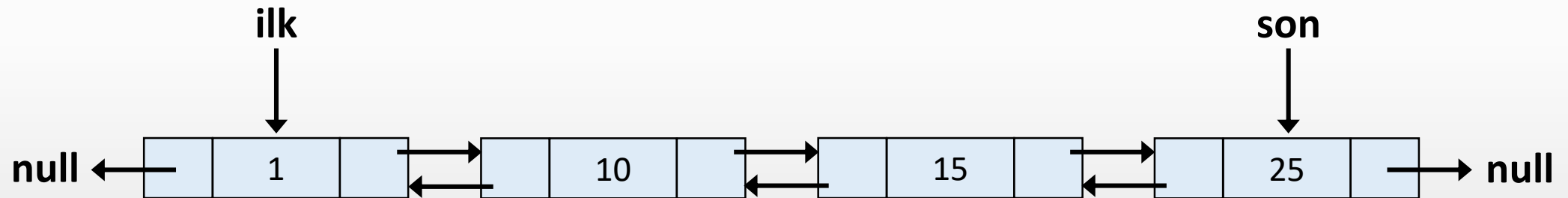


Çift Yönlü Bağlı Liste (Doubly Linked List)

```
class Dugum {  
    int veri;  
    Dugum onceki;  
    Dugum sonraki;  
  
    Dugum(int veri) {  
        this.veri = veri;  
        this.onceki = null;  
        this.sonraki = null;  
    }  
}
```



Çift Yönlü Bağlı Liste (Doubly Linked List)





Listenin Başına Düğüm Ekleme

```
public void basaEkle(int veri) {  
    Dugum yeniDugum = new Dugum(veri);  
    if (bas == null) {  
        bas = yeniDugum;  
        son = yeniDugum;  
    } else {  
        yeniDugum.sonraki = bas;  
        bas.onceki = yeniDugum;  
        bas = yeniDugum;  
    }  
}
```



Listenin Sonuna Düğüm Ekleme

```
public void sonaEkle(int veri) {  
    Dugum yeniDugum = new Dugum(veri);  
    if (bas == null) {  
        bas = yeniDugum;  
        son = yeniDugum;  
    } else {  
        yeniDugum.onceki = son;  
        son.sonraki = yeniDugum;  
        son = yeniDugum;  
    }  
}
```



Belirli Bir Konuma Eleman Ekleme

```
void konumaEkle(int veri, int konum) {  
    Dugum yeniDugum = new Dugum(veri);  
    if (konum <= 1) {  
        yeniDugum.sonraki = bas;  
        bas.onceki = yeniDugum;  
        bas = yeniDugum;  
    } else {  
        Dugum gecici = bas;  
        int adim = 1;  
        while (gecici != null && adim < konum - 1) {  
            gecici = gecici.sonraki;  
            adim++;  
        }  
    }
```



Belirli Bir Konuma Eleman Ekleme

```
if (gecici == null) {  
    System.out.println("Geçersiz konum. Eleman eklenemedi.");  
} else {  
    yeniDugum.sonraki = gecici.sonraki;  
    yeniDugum.onceki = gecici;  
    if (gecici.sonraki != null) {  
        gecici.sonraki.onceki = yeniDugum;  
    }  
    gecici.sonraki = yeniDugum;  
}  
}  
}
```



Çift Yönlü Bağlı Listenin Uzunluğunu Bulma

```
int listeUzunlugu() {  
    Dugum gecici = bas;  
    int uzunluk = 0;  
    while (gecici != null) {  
        uzunluk++;  
        gecici = gecici.sonraki;  
    }  
    System.out.println(uzunluk);  
    return uzunluk;  
}
```



Listenin Başındaki Elemanı Silme

```
void ilkElemaniSil() {  
    if (bas == null) {  
        System.out.println("Liste boş. Silme işlemi yapılamaz.");  
        return;  
    }  
    if (bas.sonraki != null) {  
        bas = bas.sonraki;  
        bas.onceki = null;  
    } else {  
        bas = null;  
    }  
}
```




Listenin Sonundaki Elemanı Silme

```
void sonElemaniSil() {  
    if (bas.sonraki == null) {  
        bas = null;  
    } else {  
        Dugum gecici = bas;  
        while (gecici.sonraki != null) {  
            gecici = gecici.sonraki;  
        }  
        gecici.onceki.sonraki = null;  
    }  
}
```



Listede Belirli Bir Elemanı Silme

```
void elemaniSil(int veri) {  
    if (bas.veri == veri) {  
        bas = bas.sonraki;  
        if (bas != null) {  
            bas.onceki = null;  
        }  
        return;  
    }  
}
```

```
Dugum gecici = bas;  
while (gecici != null && gecici.veri != veri) {  
    gecici = gecici.sonraki;  
}
```



Listede Belirli Bir Elemanı Silme

```
if (gecici == null) {  
    System.out.println("Belirtilen eleman listede bulunamadı.");  
    return;  
}
```

```
if (gecici.sonraki != null) {  
    gecici.onceki.sonraki = gecici.sonraki;  
    gecici.sonraki.onceki = gecici.onceki;  
} else {  
    gecici.onceki.sonraki = null;  
}  
}
```



Listeyi İleri Yönde Yazdırmak

```
void listeyiIleriYazdir() {  
    Dugum gecici = bas;  
  
    System.out.print("İleri Yönde: ");  
    while (gecici != null) {  
        System.out.print(gecici.veri + " <-> ");  
        gecici = gecici.sonraki;  
    }  
    System.out.println("null");  
}
```



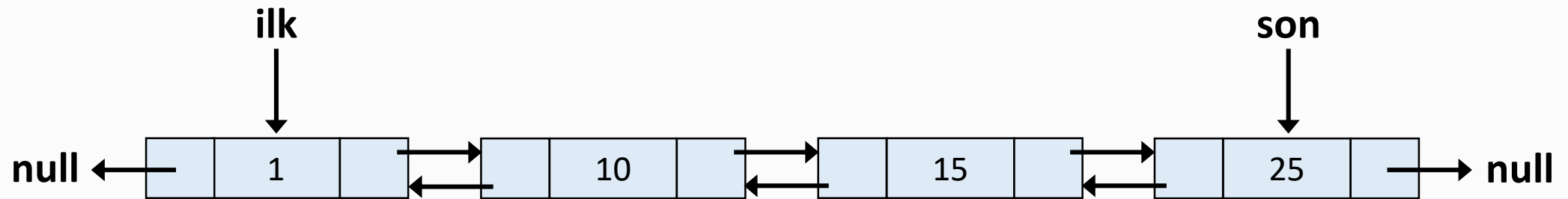
Listeyi Geri Yönde Yazdırmak

```
void listeyiGeriYazdir() {  
    Dugum gecici = bas;  
    // En son elemana ilerle  
    while (gecici.sonraki != null) {  
        gecici = gecici.sonraki;  
    }  
    System.out.print("Geri Yönde: ");  
    while (gecici != null) {  
        System.out.print(gecici.veri + " <-> ");  
        gecici = gecici.onceki;  
    }  
    System.out.println("null");  
}
```

Çift Yönlü Bağlı Liste Elemanlarını Geri Yönde Yazdırma



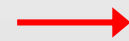
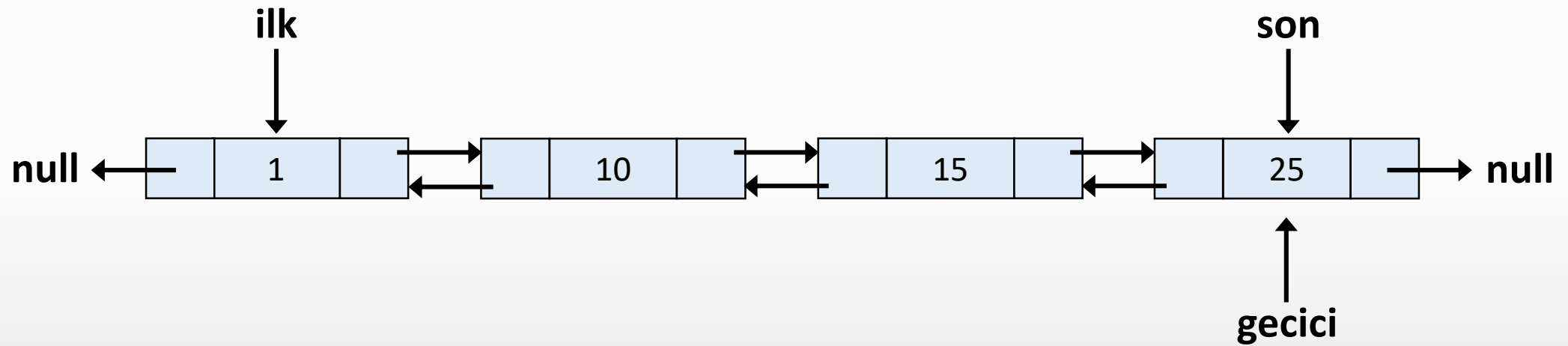
Çıktı:



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



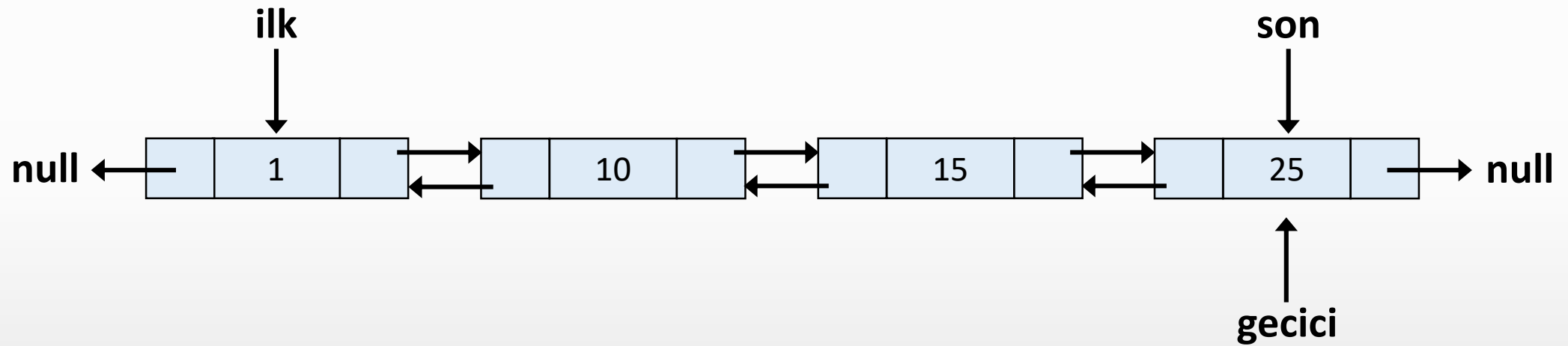
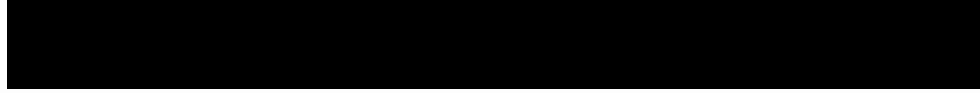
Çıktı:



```
Dugum gecici = son;
while(gecici != null) {
    System.out.print(gecici.veri + " --> ");
    gecici = gecici.onceki;
}
System.out.print("null");
```



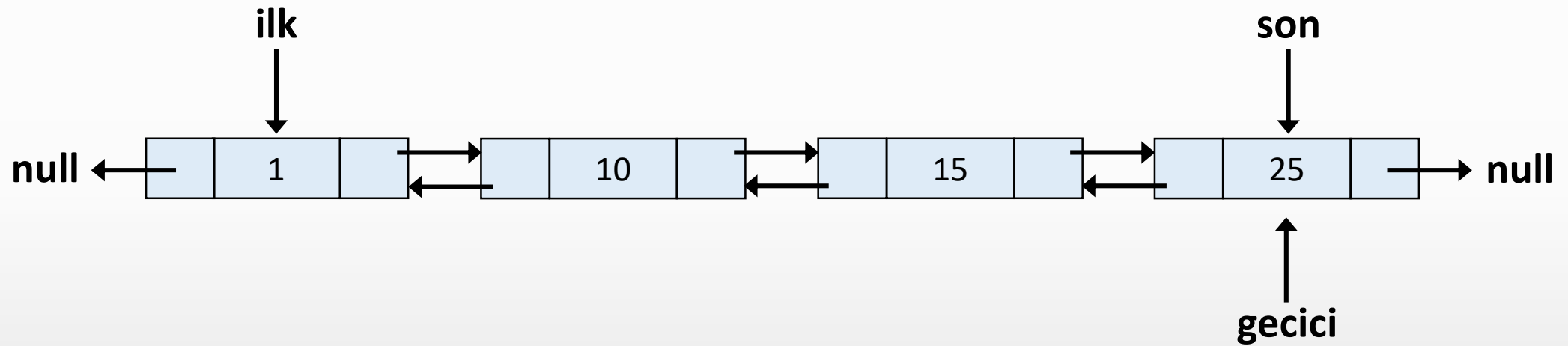
Çıktı:



```
→ Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



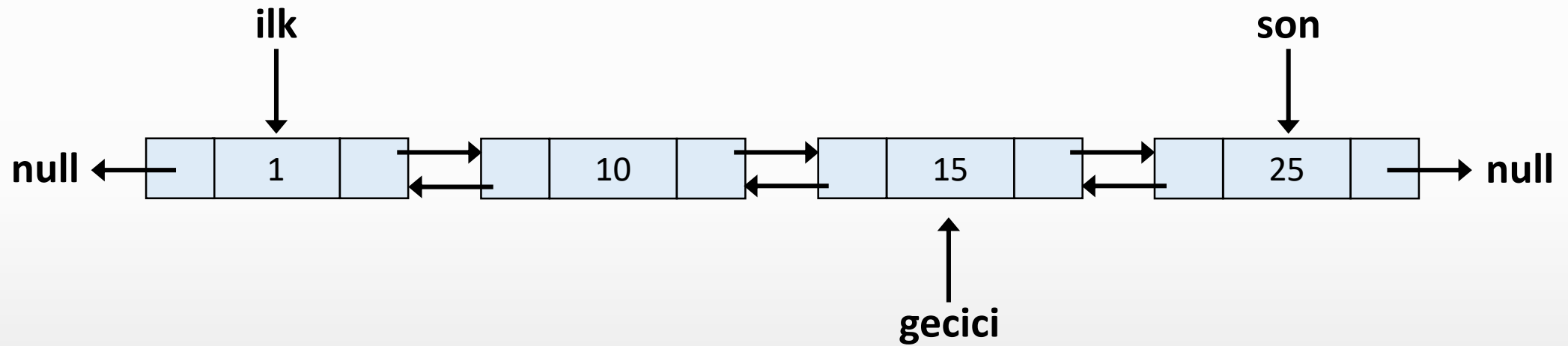

Çıktı: 25 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



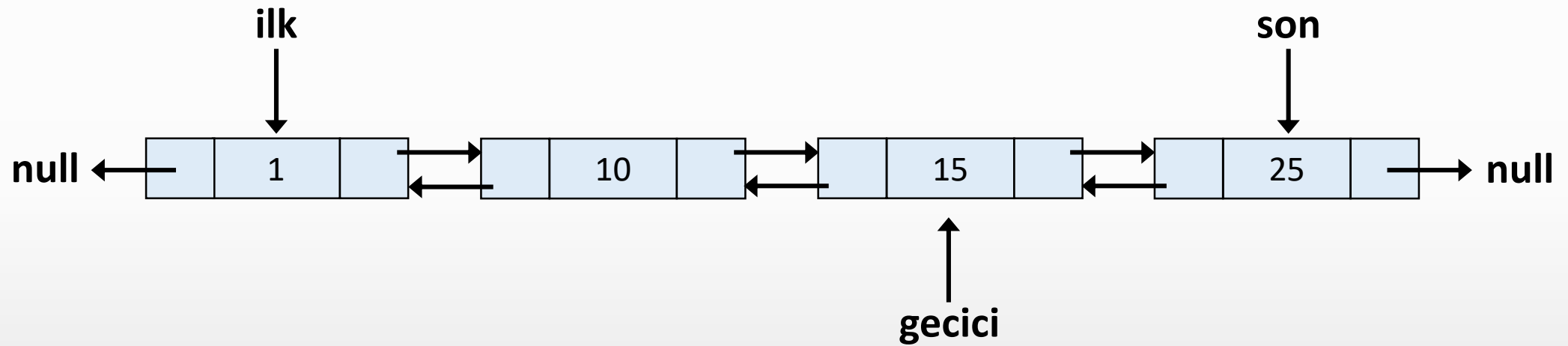
Çıktı: 25 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



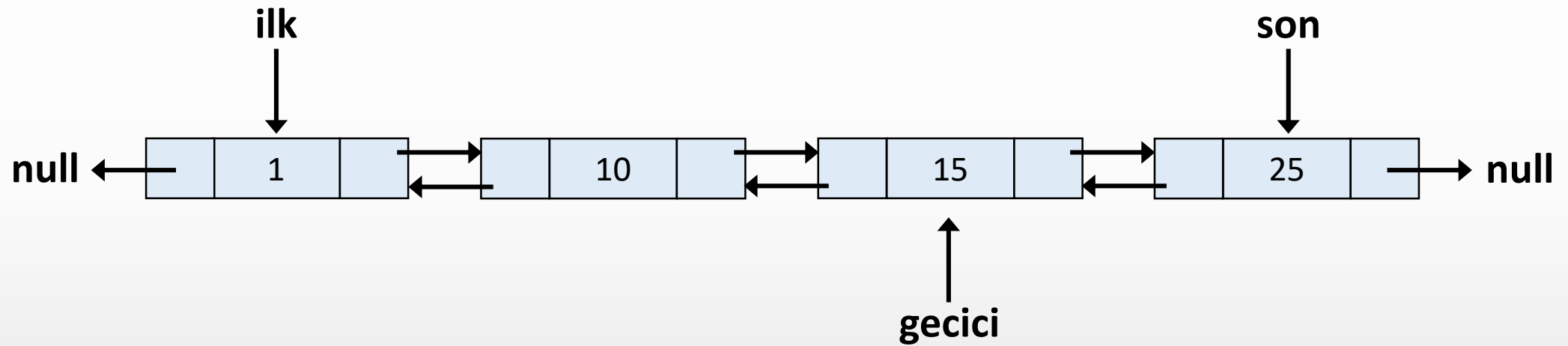
Çıktı: 25 -->



```
→ Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



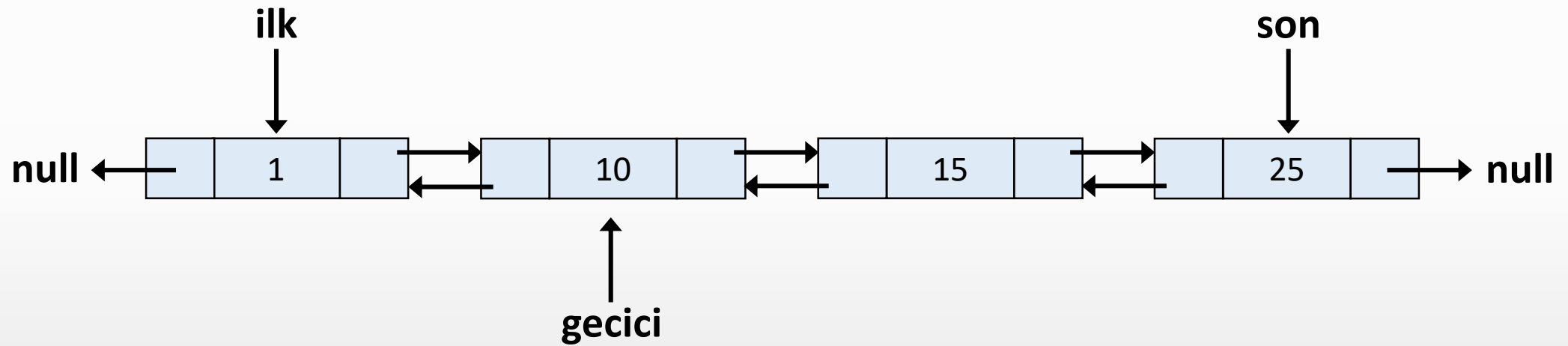
Çıktı: 25 --> 15 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



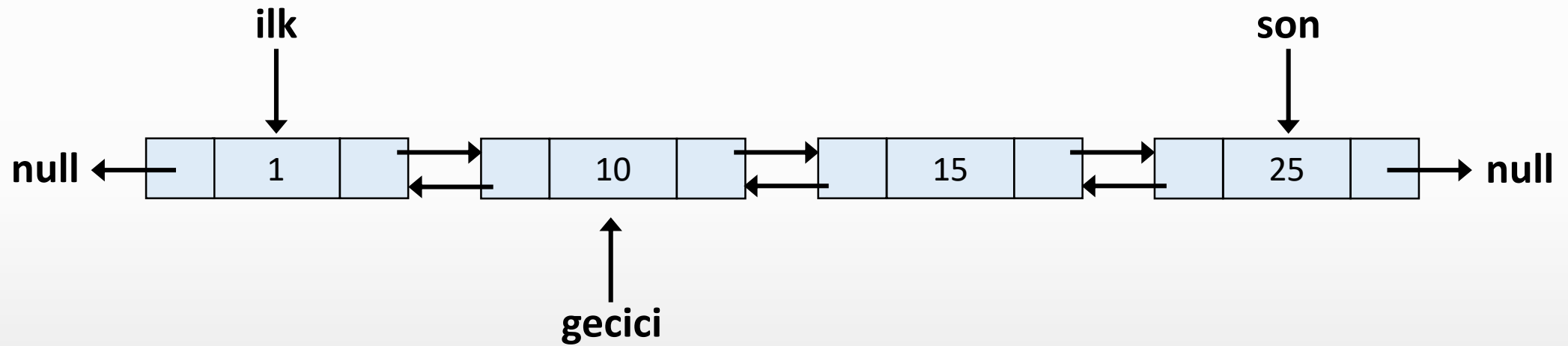
Çıktı: 25 --> 15 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



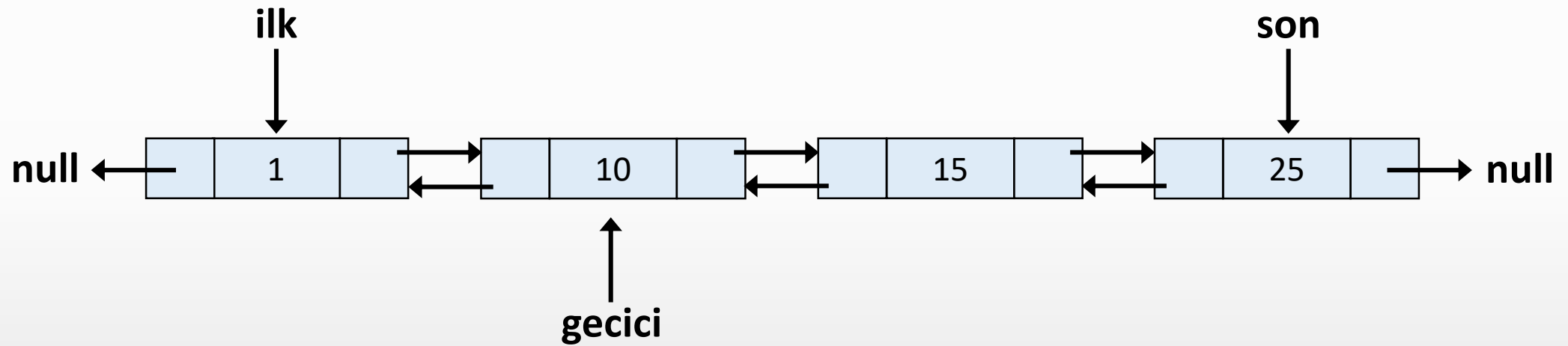
Çıktı: 25 --> 15 -->



```
Dugum gecici = son;  
→ while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



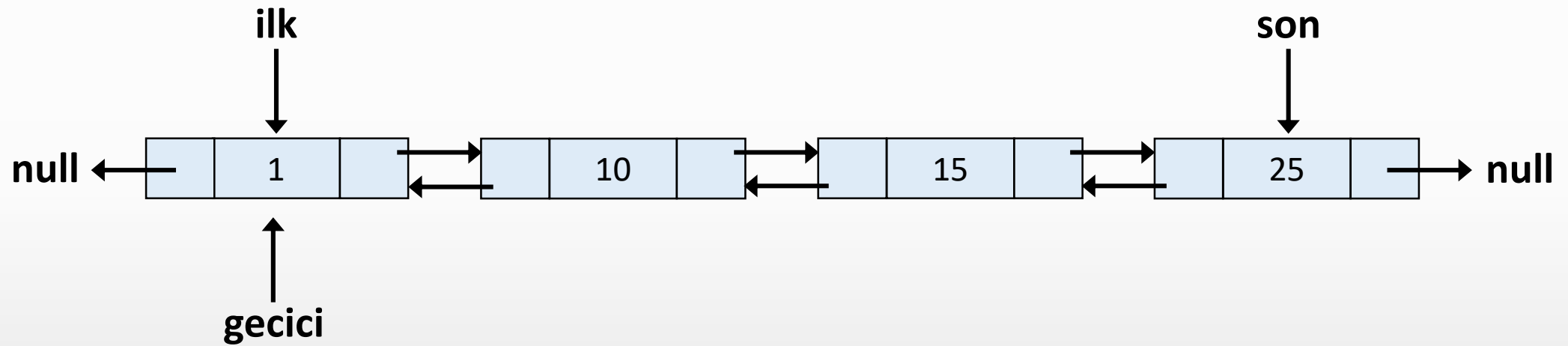
Çıktı: 25 --> 15 --> 10 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



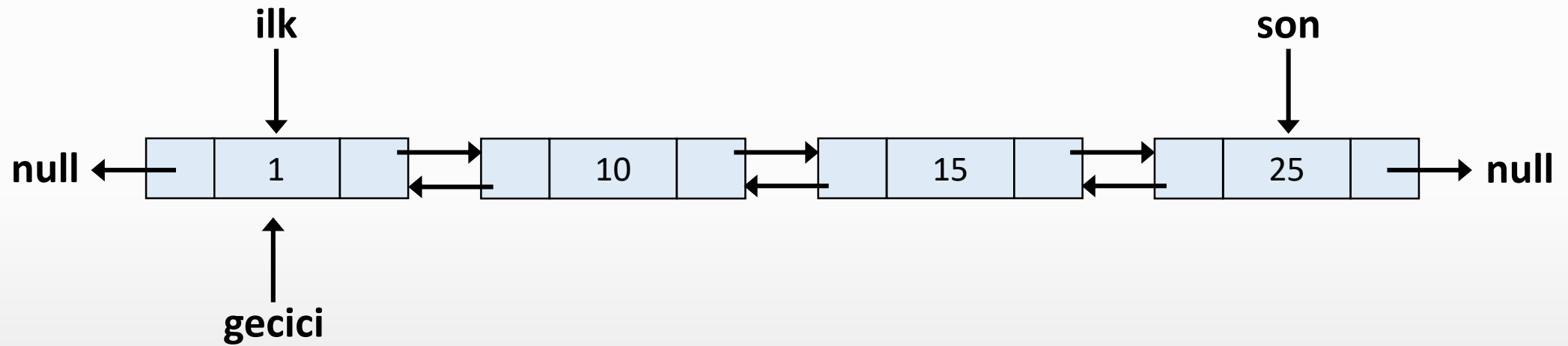
Çıktı: 25 --> 15 --> 10 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



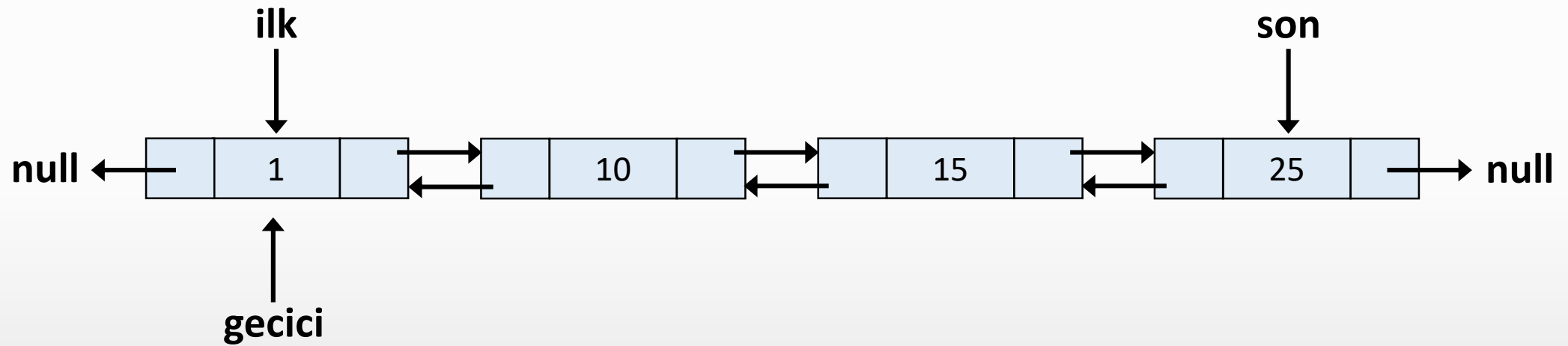

Çıktı: 25 --> 15 --> 10 -->



```
→ Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



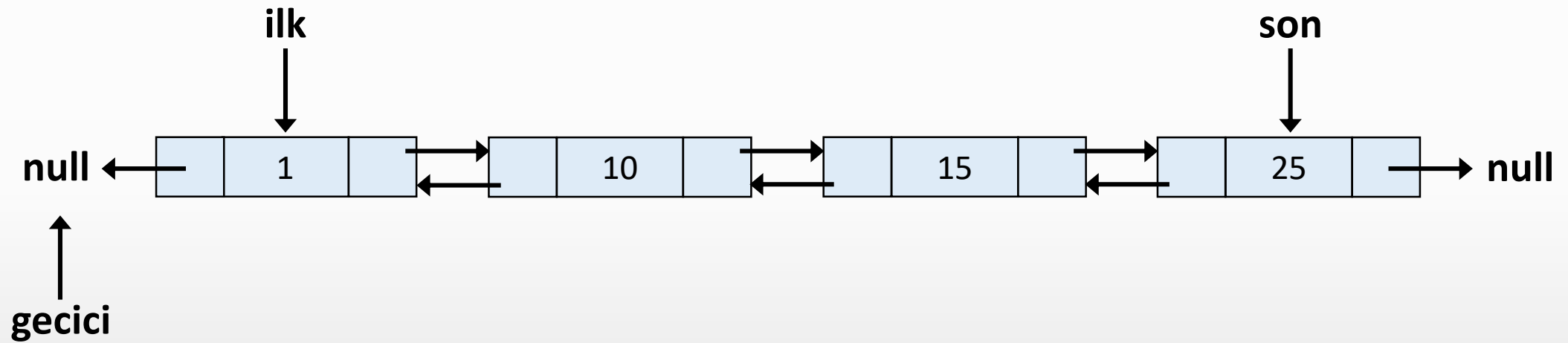
Çıktı: 25 --> 15 --> 10 --> 1 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



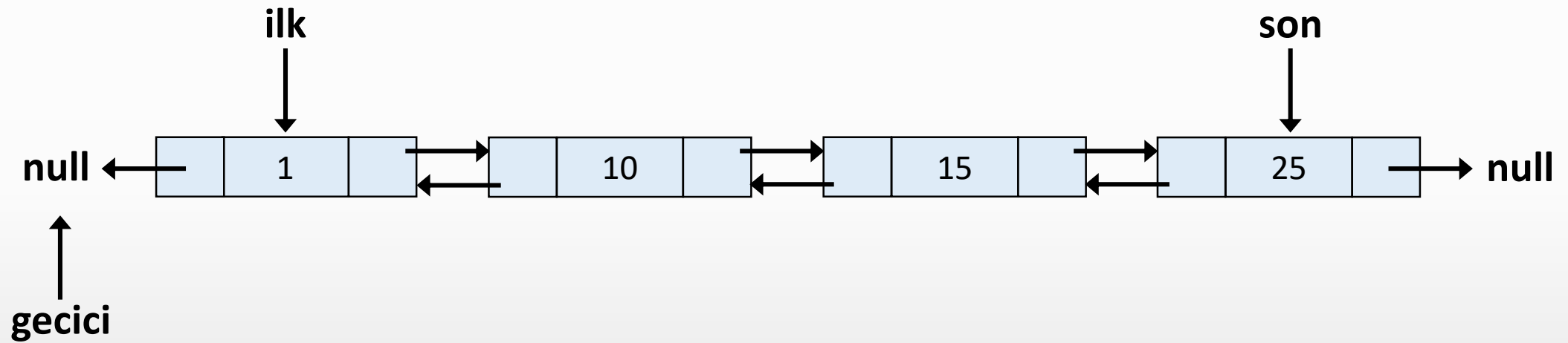
Çıktı: 25 --> 15 --> 10 --> 1 -->



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



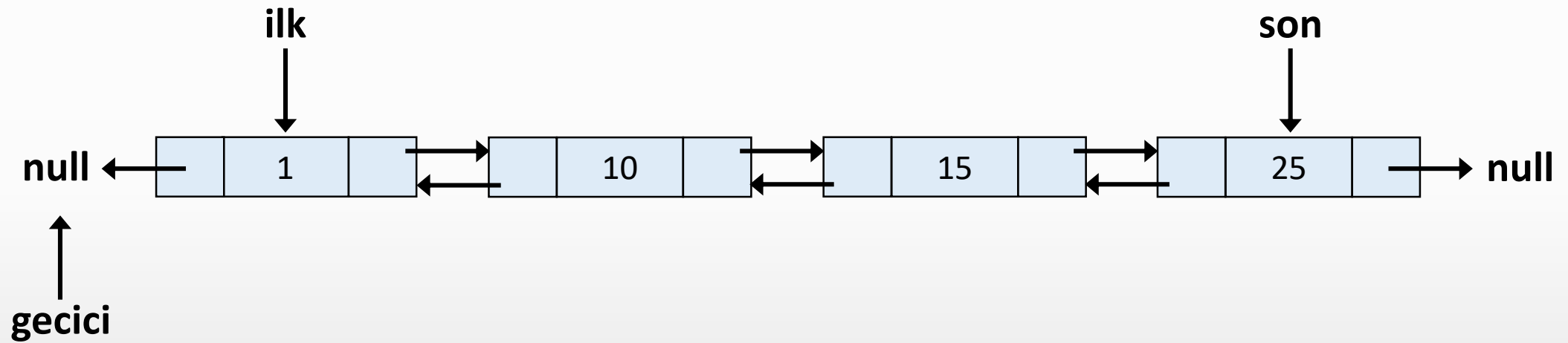
Çıktı: 25 --> 15 --> 10 --> 1 -->



```
→ Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
System.out.print("null");
```



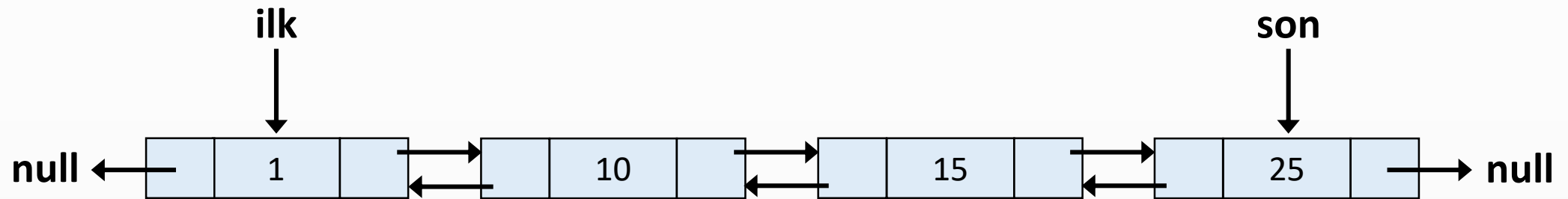
Çıktı: 25 --> 15 --> 10 --> 1 --> null



```
Dugum gecici = son;  
while(gecici != null) {  
    System.out.print(gecici.veri + " --> ");  
    gecici = gecici.onceki;  
}  
→ System.out.print("null");
```



Çıktı: 25 --> 15 --> 10 --> 1 --> null



```
Dugum gecici = son;
while(gecici != null) {
    System.out.print(gecici.veri + " --> ");
    gecici = gecici.onceki;
}
System.out.print("null");
```



İki Boyutlu Matrisi Bağlı Liste Yapısına Çevirme

```
class Dugum {  
    int veri;  
    Dugum sag;  
    Dugum asagi;  
};
```



İki Boyutlu Matrisi Bağlı Liste Yapısına Çevirme

```
Dugum olustur(int matris[][], int i, int j)
{
    // i veya j sınırları aşıldıysa dön
    if (i > matris.length - 1 || j > matris[0].length - 1)
        return null;
    // Geçerli i ve j için yeni bir düğüm oluştur ve
    // alt ve sağ işaretçilerini sırasıyla ayarla
    Dugum gecici = new Dugum();
    gecici.veri = matris[i][j];
    gecici.sag = olustur(matris, i, j + 1);
    gecici.asagi = olustur(matris, i + 1, j);
    return gecici;
}
```




İki Boyutlu Matrisi Bağlı Liste Yapısına Çevirme

```
void görüntule(Dugum baslangic)
{
    Dugum sag; // Sağa doğru hareket etmek için işaretçi
    Dugum asagi = baslangic; // Aşağı doğru hareket etmek için işaretçi
    while (asagi != null) { // Aşağıdaki düğüm NULL olana kadar dön
        sag = asagi;
        // Sağa doğru hareket ederken düğüm sağ NULL olana kadar dön
        while (sag != null) {
            System.out.print(sag.veri);
            sag = sag.sag;
        }
        asagi = asagi.asagi;
    }
}
```

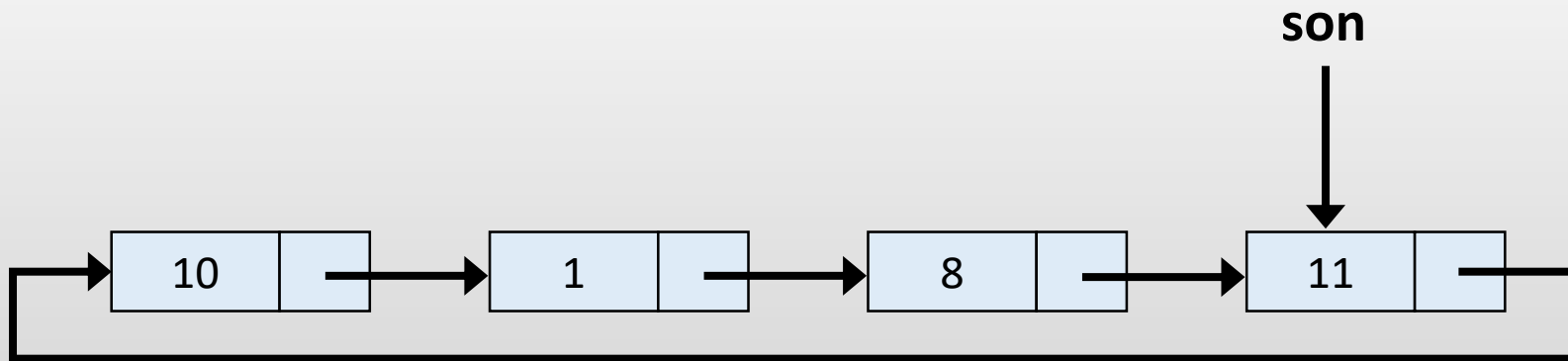
Dairesel Bağlı Liste (Circularly Linked List)





Dairesel Bağlı Liste (Circularly Linked List)

- Dairesel bağlı liste, bir bağlı listenin son düğümünün ilk düğümü işaret ettiği veri yapısını tanımlar.
- Her düğüm, bir sonraki düğümünü işaret eder.
- Son düğüm, ilk düğümü işaret ederek bir döngü oluşturur.
- Bağlı listeler son düğümde "null" işaretlerken, dairesel bağlı listelerde son düğüm ilk düğümü işaret eder.





Dairesel Bağlı Liste (Circularly Linked List)

```
public class DaireselBagliListe {  
    // Dairesel bağlı listenin düğümünü temsil eden iç içe sınıf  
    class Dugum {  
        int veri;  
        Dugum sonraki;  
  
        Dugum(int veri) {  
            this.veri = veri;  
            this.sonraki = null;  
        }  
    }  
  
    private Dugum bas;  
    private Dugum son;  
}
```

Dairesel Baęlı Listeyi Yazdırma





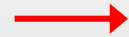
Dairesel Bağlı Listeyi Yazdırma

son
↓
null

```
→ if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



son
↓
null



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```

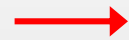
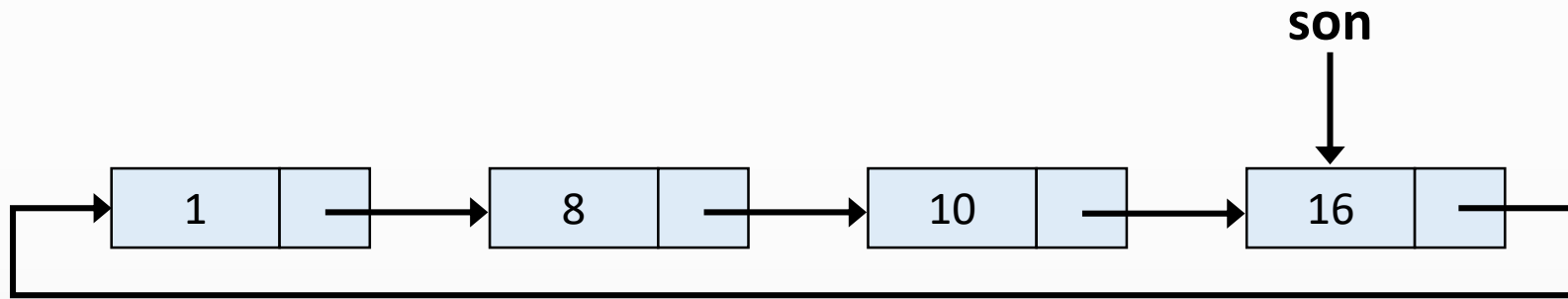


son
↓
null

```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



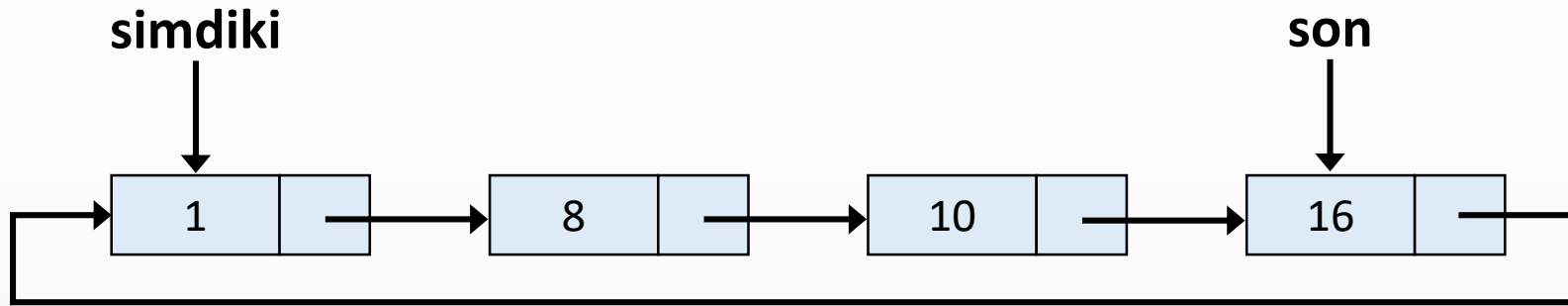

Çıktı:



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



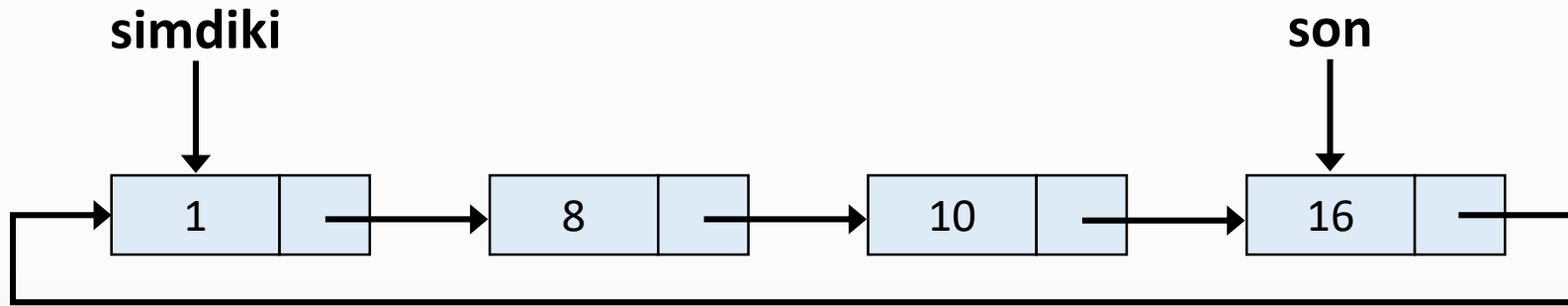
Çıktı:



```
if(son == null) {  
    return;  
}  
→ Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



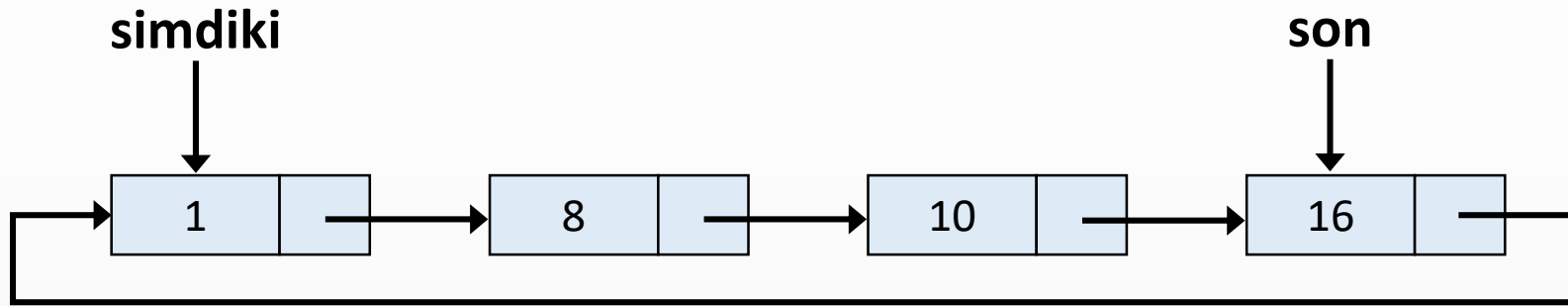
Çıktı:



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
→ while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



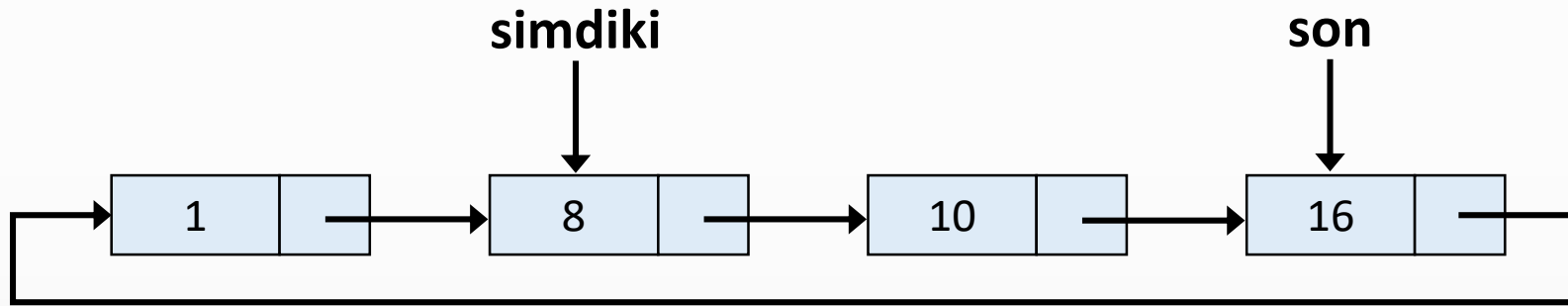
Çıktı: 1



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



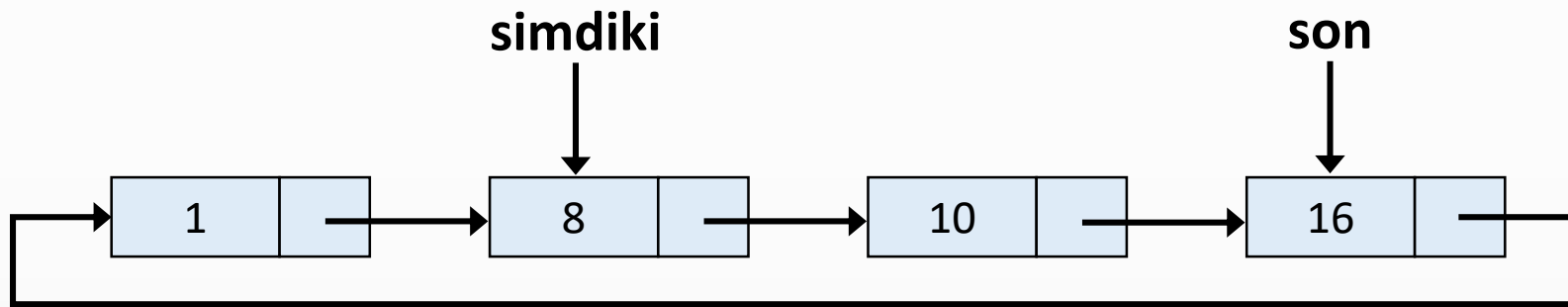
Çıktı: 1



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



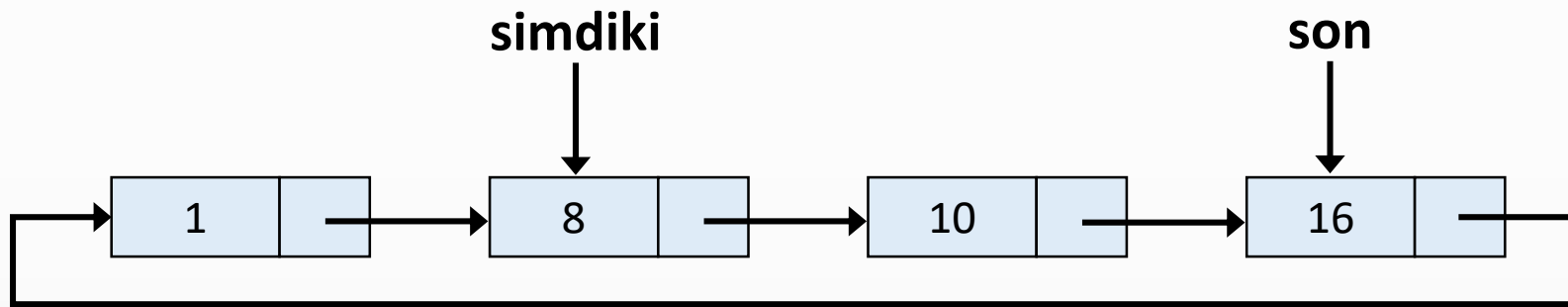
Çıktı: 1



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
→ while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



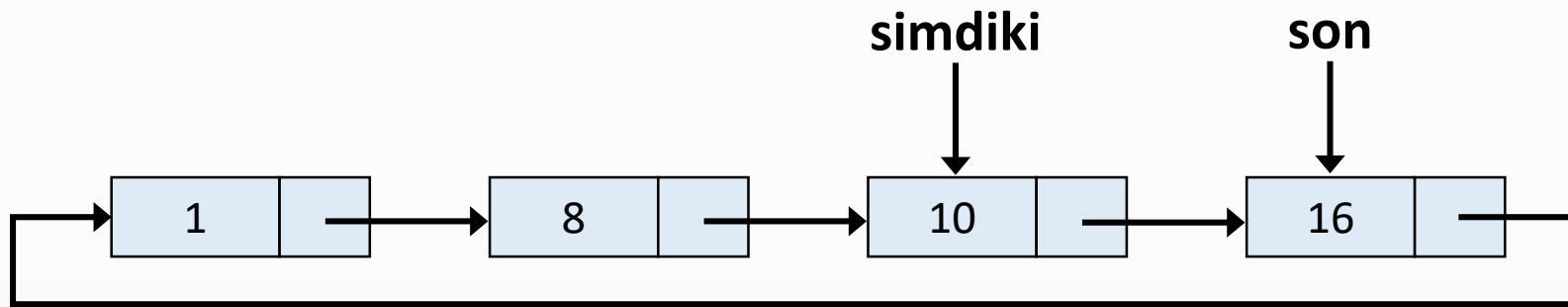
Çıktı: 1 8



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



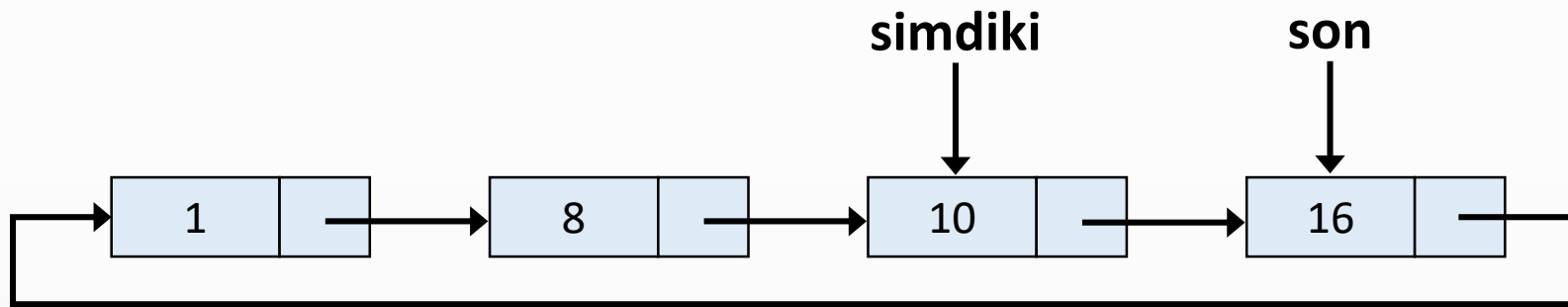
Çıktı: 1 8



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



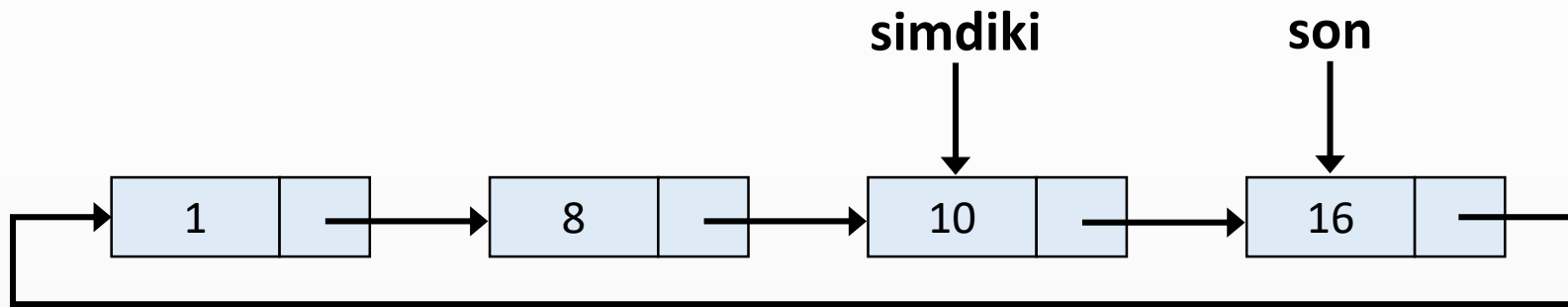

Çıktı: 1 8



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
→ while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



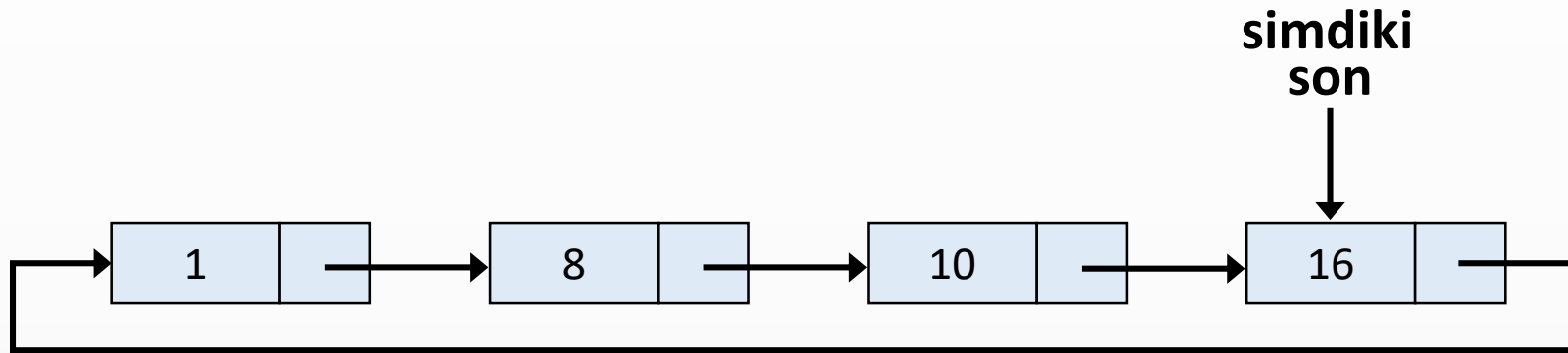
Çıktı: 1 8 10



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



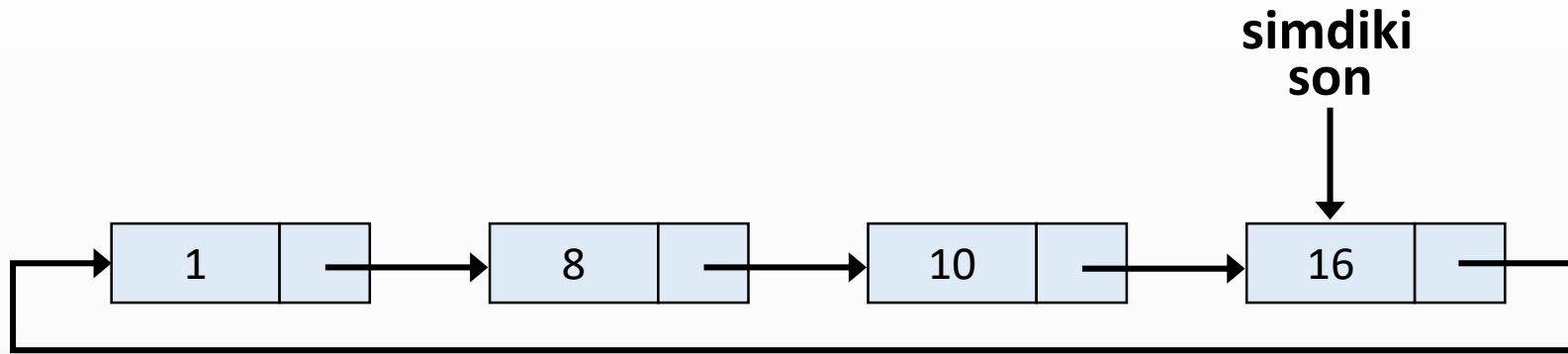
Çıktı: 1 8 10



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



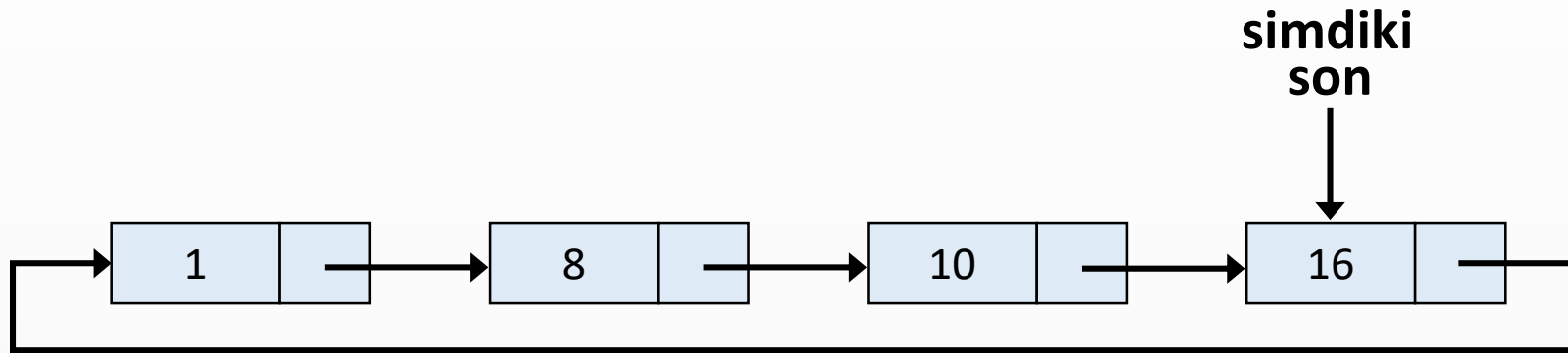
Çıktı: 1 8 10



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
→ while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```



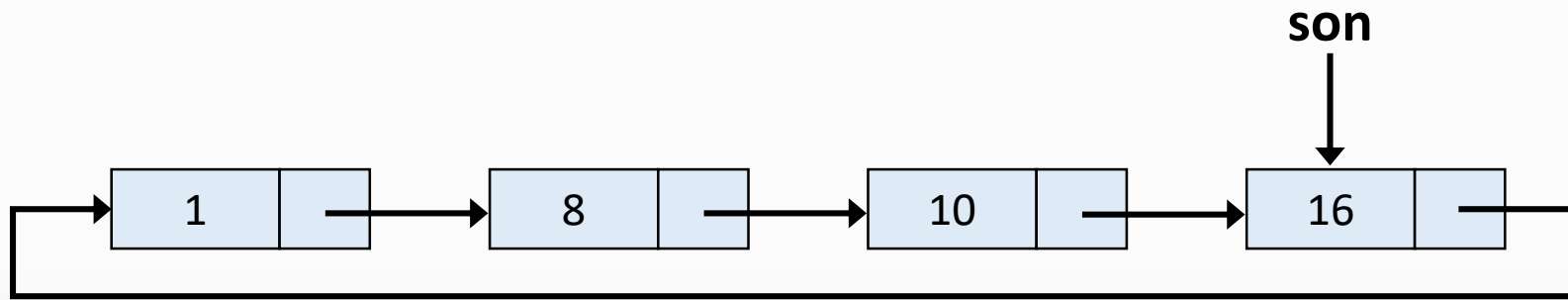
Çıktı: 1 8 10 16



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
→ System.out.print(simdiki.veri + " ");
```



Çıktı: 1 8 10 16



```
if(son == null) {  
    return;  
}  
Dugum simdiki = son.sonraki;  
while(simdiki != son) {  
    System.out.print(simdiki.veri + " ");  
    simdiki = simdiki.sonraki;  
}  
System.out.print(simdiki.veri + " ");
```

Dairesel Baęlı Listenin Sonuna Ekleme





Dairesel Bağlı Listenin Sonuna Ekleme

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```

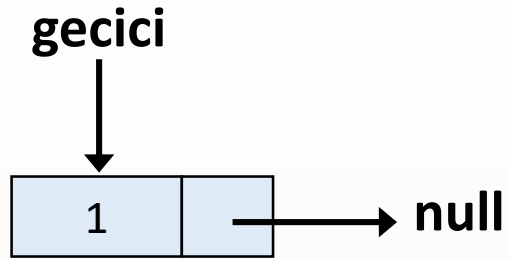



son
↓
null

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



son
↓
null



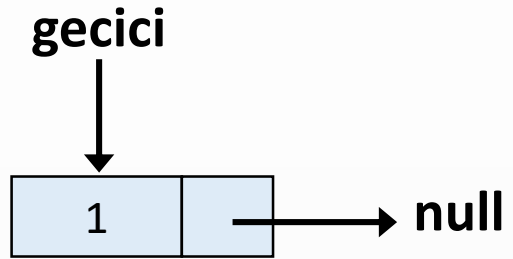
uzunluk = 0
deger = 1

→

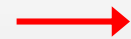
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



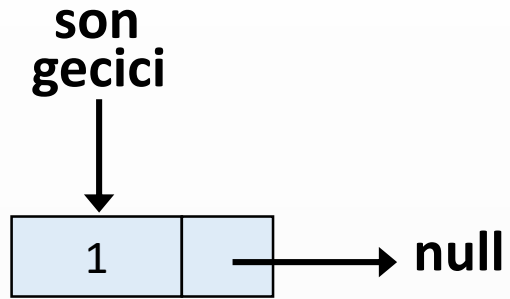
son
↓
null



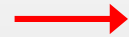
uzunluk = 0
deger = 1



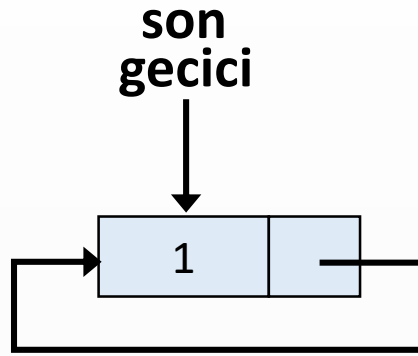
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



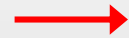
uzunluk = 0
deger = 1



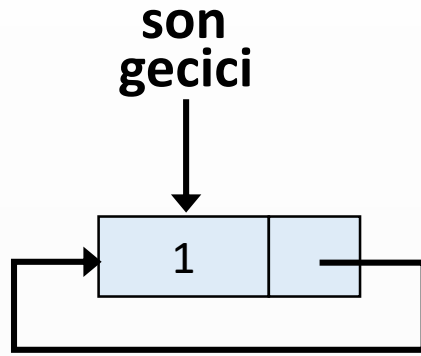
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 0
deger = 1

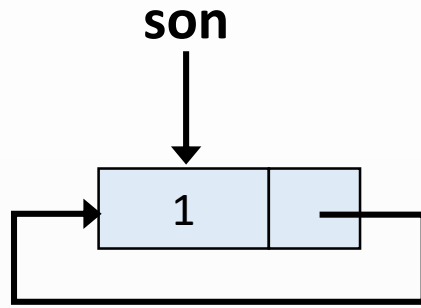


```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



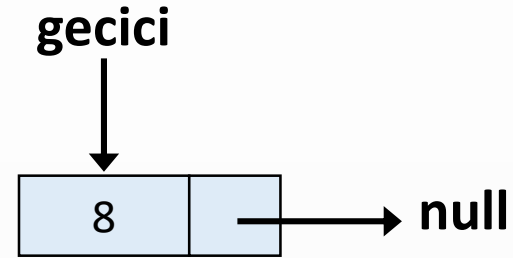
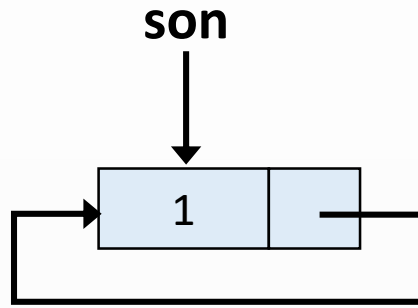
uzunluk = 1
deger = 1

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 1

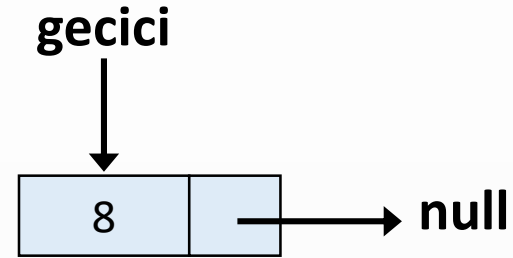
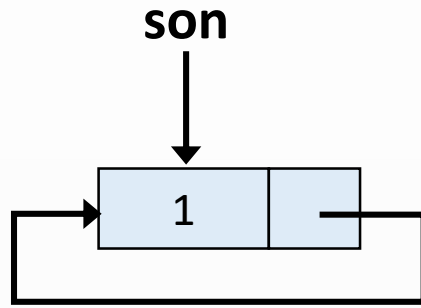
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 1
deger = 8

→

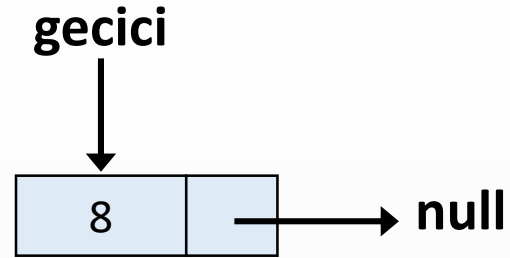
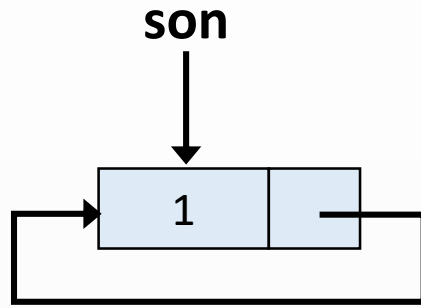
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```

uzunluk = 1
deger = 8

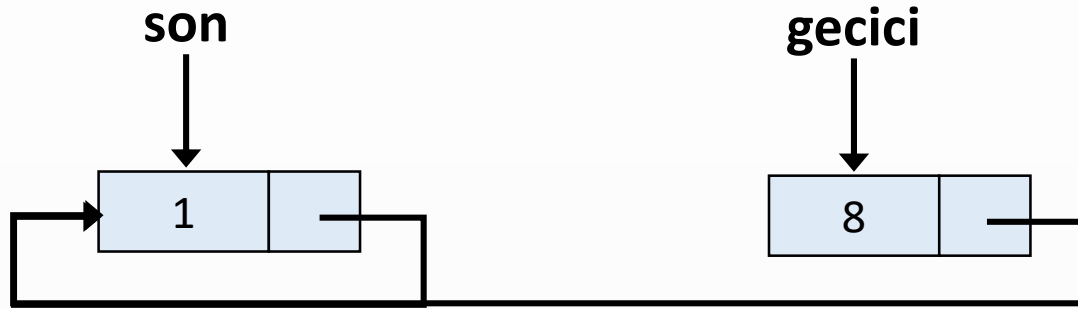
→

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```

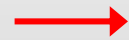


uzunluk = 1
deger = 8

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
→ else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 1
deger = 8



```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 1
deger = 8



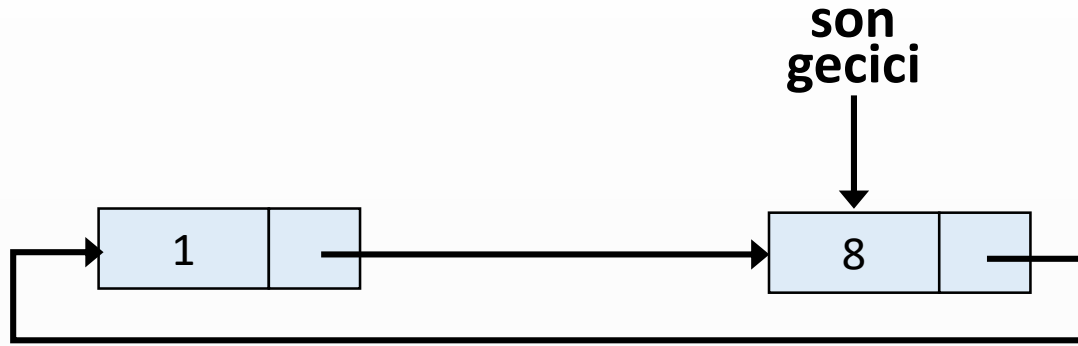
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 1
deger = 8



```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



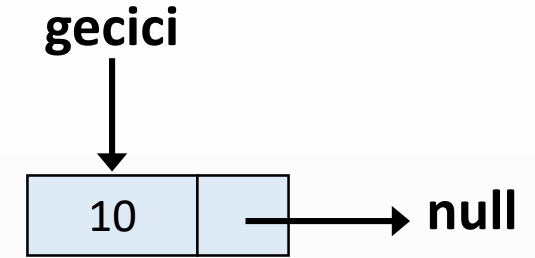
uzunluk = 2
deger = 8

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
→ uzunluk++;
```



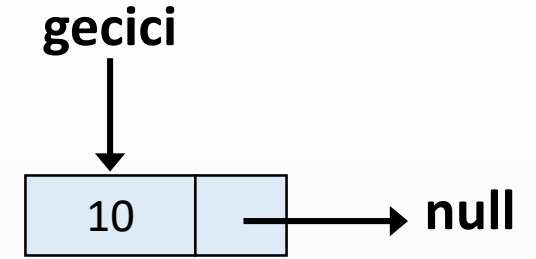
uzunluk = 2

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



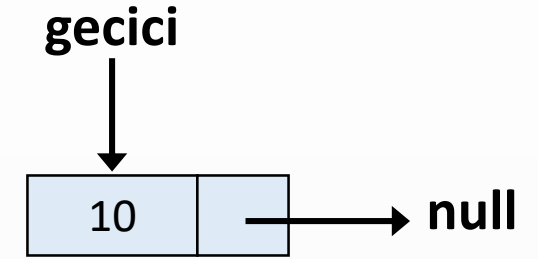
uzunluk = 2
deger = 10

```
→ Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```

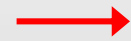



uzunluk = 2
deger = 10

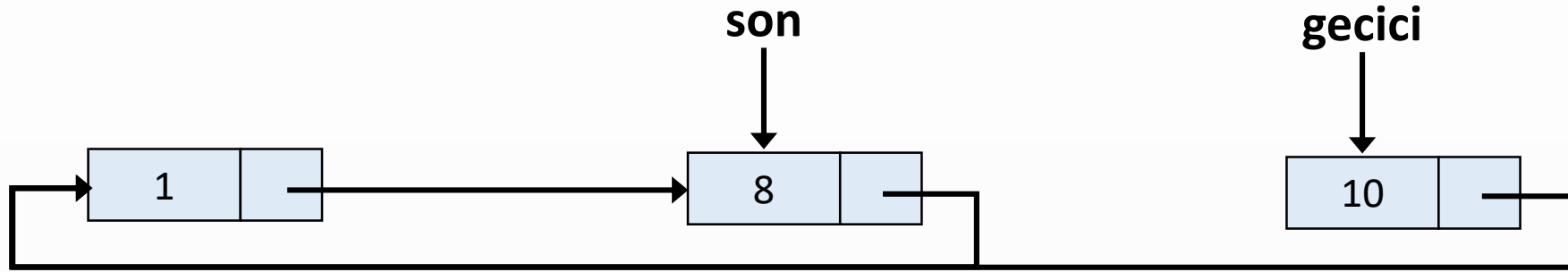
```
→ Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



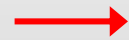
uzunluk = 2
deger = 10



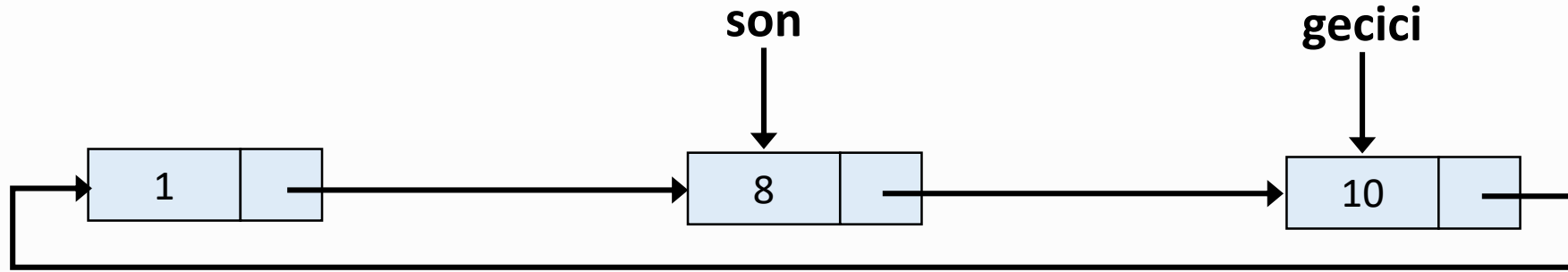
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 2
deger = 10



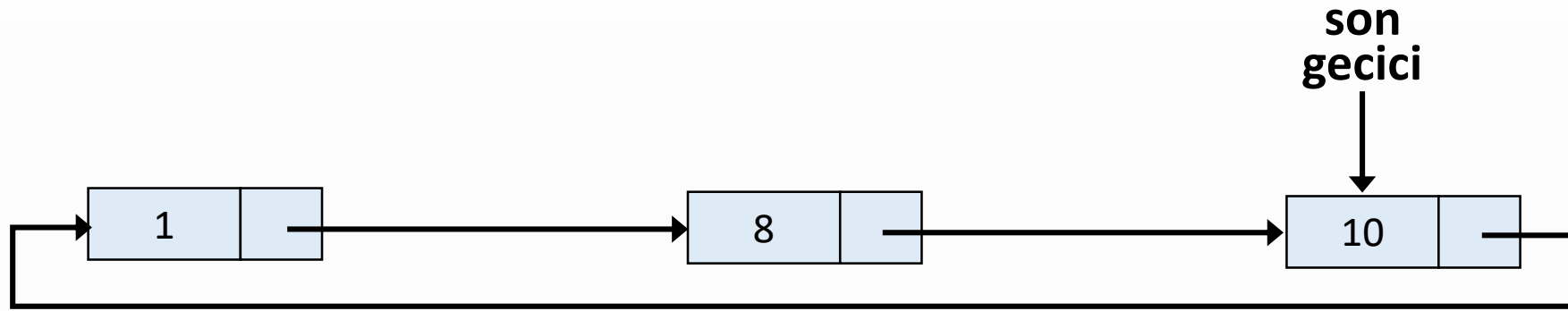
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



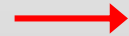
uzunluk = 2
deger = 10



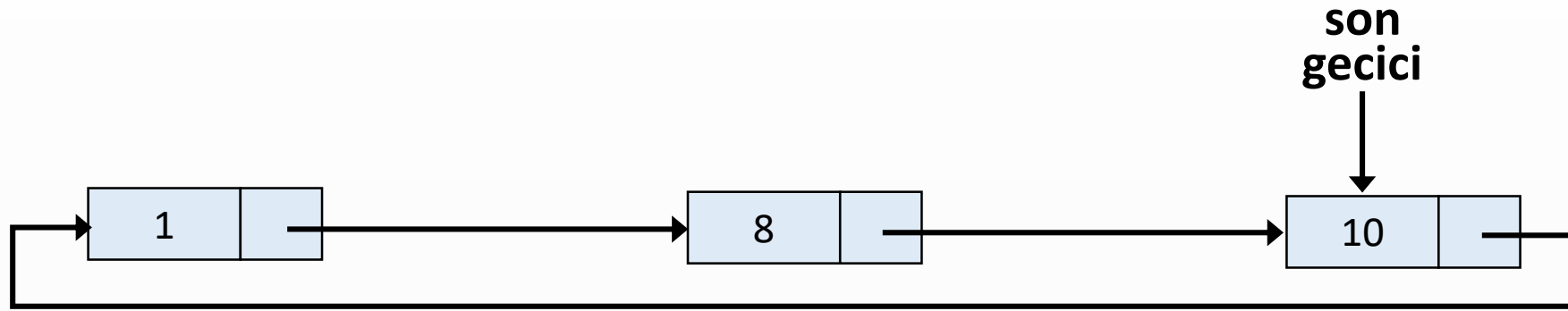
```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 2
deger = 10

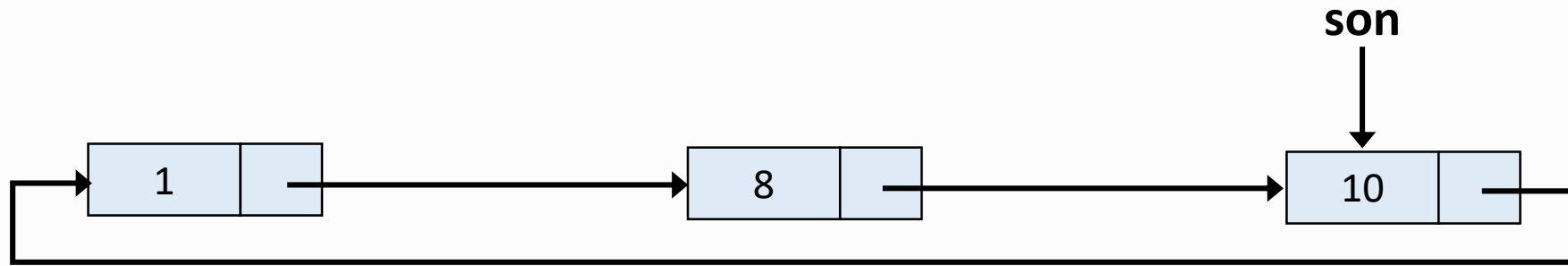


```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



uzunluk = 3
deger = 10

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
→ uzunluk++;
```



uzunluk = 3

```
Dugum gecici = new Dugum(deger);  
if(son == null) {  
    son = gecici;  
    son.sonraki = son;  
}  
else {  
    gecici.sonraki = son.sonraki;  
    son.sonraki = gecici;  
    son = gecici;  
}  
uzunluk++;
```



Soru

Tek Yönlü Bağlı Listede (Singly Linked List) bir düğümün önceki düğümüne erişim mümkün müdür?

- A) Evet
- B) Hayır
- C) Sadece son düğümde
- D) Bağlı liste türüne bağlı



Soru

Tek yönlü bağlı listenin aşağıdakilerden hangisi avantajı değildir?

- A. Kolay uygulanabilir.
- B. Ekleme ve silme işlemleri hızlıdır.
- C. Bellekte verimli bir şekilde kullanabilirler.
- D. Eleman arama işlemi hızlıdır.



Soru

Bağlı listenin dairesel bir yapıya sahip olduğu durumda, liste üzerinde döngü oluşturmak nasıl engellenir?

- a) Her düğümün bir düğüm sayacı bulundurulur.
- b) Listenin son düğümü boş bir düğümle işaretlenir.
- c) Tekrarlayan düğümler kullanılır.
- d) Döngüler engellenemez.



Soru

Bir çift yönlü bağlı listedeki herhangi bir düğümünü verilen bir düğümün hemen önüne eklemek için tipik bir zaman karmaşıklığı nedir?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$



Soru

Bağlı listenin başına bir düğüm eklemek için tipik bir zaman karmaşıklığı nedir?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$



Soru

Bağlı listenin sonundan bir düğümü çıkarmak için tipik bir zaman karmaşıklığı nedir?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$

Ödev



- Tek yönlü bağlı listeyi çift yönlü bağlı listeye $O(n)$ karmaşıklığında dönüştüren kod parçasını yazınız. Algoritmanızın çalışma mantığını kısaca anlatınız.
- sercan.kulcu@giresun.edu.tr
- Son tarih: 26 Ekim 2023 saat 23:59'a kadar
- Konu: «BİLM-201 Ödev 1»

Merhaba Hocam,

Ben xxxx numaralı 'yım. Ekte ödevimi gönderiyorum.

Saygılarımla,

İyi çalışmalar dilerim.

Ekler: odev1.java



SON