

Çift-Yönlü Bağlı Liste (Doubly Linked List)

Düğüm içerisinde 3 adet değişken bulunur. Bunlardan birisi yine veri için ayrılmıştır. Diğer iki değişkenler işaretçidir (pointer) ve birisi **kendinden önceki düğümü**, diğeri **kendinden sonraki düğümü** tutar.



LinkedList-çift yönlü

yselim

Düğüm Class'ı

```
public class Dugum
{
    protected object dObject;
    protected Dugum nOnceki;
    protected Dugum nSonraki;

    public object Deger
    {
        get { return dObject; }
        set { dObject = Deger;}
    }

    public Dugum Onceki
    {
        get { return nOnceki; }
        set { nOnceki = value;}
    }

    public Dugum Sonraki
    {
        get { return nSonraki;}
        set { nSonraki = value;}
    }

    // Yapıcı metot
    public Dugum(Dugum oncekiDugum, Dugum sonrakiDugum, object dugumObj)
    {
        nOnceki = oncekiDugum;
        nSonraki= sonrakiDugum;
        dObject = dugumObj;
    }
}
```

LinkedList-çift yönlü

yselim

Bağlı Liste Class'ı

```
public class bagliListe
{
    int dugumSayisi = 1;
    int suankidugumIndex = 0;
    Dugum suAnkiDugum;

    // Dugum sayısını döndürür
    public int Say
    {
        get { return dugumSayisi;}
    }

    // Su anki düğümü dondurur
    public Dugum SuAnkiDugum
    {
        get { return suAnkiDugum;}
    }

    // Su anki düğümün index numarasını dondurur
    public int SuAnkiDugumIndex
    {
        get { return suankidugumIndex;}
    }

    // Yapıcı metot
    public bagliListe(object obj)
    {
        suAnkiDugum = new Dugum(null, null, obj);
        suAnkiDugum.Sonraki = null;
        suAnkiDugum.Onceki = null;
    }
}
```

LinkedList-çift yönlü

yselim

Bağlı Liste Class'ı devam

```
// Yeni düğüm ekler ve eklenen düğümü o anki düğüm olarak belirler
public void DugumEkle(object obj)
{
    if(suAnkiDugum.Sonraki == null)    //listenin sonunda eleman yoksa sona ekler
    {
        suAnkiDugum = suAnkiDugum.Sonraki = new Dugum(suAnkiDugum, null,
obj);
    }
    else    // eleman varsa araya ekler
    {
        suAnkiDugum = suAnkiDugum.Sonraki = new Dugum(suAnkiDugum,
suAnkiDugum.Sonraki,obj);
    }
    dugumSayisi++;
    suankidugumIndex++;
}

// O anki düğümden sonraki düğüme gider
public void SonrakineGit()
{
    if(suAnkiDugum.Sonraki == null)    // ileride dugum yoksa istisnai durum
    {
        throw new Exception("Sonraki dugum bulunamadi!");
    }
    else // ileride dugum mevcutsa sonraki dugume gider
    {
        suAnkiDugum = suAnkiDugum.Sonraki;
        suankidugumIndex++;
    }
}

olusturulur
```

LinkedList-çift yönlü

ysevim

Bağlı Liste Class'ı devam

```
public void OncekineDon()  
{  
    if(suAnkiDugum.Onceki == null) // onde dugum yoksa istisnai durum olusturulur  
    {  
        throw new Exception("Onceki dugum bulunamadi!");  
    }  
    else // onde dugum mevcutsa onceki dugume gider  
    {  
        suAnkiDugum = suAnkiDugum.Onceki;  
        suankidugumIndex--;  
    }  
}  
  
// indeks numarası verilen düğüme git  
public void DugumeGit(int index)  
{  
    while(suankidugumIndex != index)  
    {  
        if(suankidugumIndex < index) SonrakineGit();  
        else if(suankidugumIndex > index) OncekineDon();  
    }  
}  
}
```

LinkedList-çift yönlü

yselim

Yazdır ve Main()

```
public void Yazdir()  
{  
    DugumeGit(0);  
    while(suankidugumIndex<Say)  
    {  
        Console.Write(suAnkiDugum.Deger+"-");  
        SonrakineGit();  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        bagliListe b1 = new bagliListe("Istanbul");  
        b1.DugumEkle("kocaeli");  
        b1.DugumEkle("manisa");  
        b1.DugumEkle(100);  
        b1.DugumEkle(23.45f);  
        b1.DugumEkle(true);  
        b1.Yazdir();  
    }  
}
```