

BAĞLI LİSTELER (LINKED LISTS)

Aynı kümeye ait verilerin bellek üzerinde bir gösterici (pointer) yardımı ile birbirine bağlanması sonucu oluşturulan veri yapısıdır. Bu veri yapısındaki bütün elemanların ortak noktası kendi içlerinde bağlantı bilgisi içeren kendi türünden bir ya da daha fazla göstericiye sahip olmasıdır. Veri yapısı, kümedeki herhangi bir elemanın, kendisinden önce ve kendisinden sonra hangi elemanın bulunduğu bilgisi bu göstericilere değer vererek kurulur. Bağlı listeler dinamik veri yapılarıdır. Bellek üzerinde tanımlanan elemanlar göstericiler yardımıyla birbirlerine bağlanarak bağlı liste yapısı oluşturulur. Listeye istenirse dizilerde olduğu gibi eleman ekleme ve silme işlemleri uygulanabilir.

BAĞLI LİSTELER (LINKED LISTS) (DEVAM)

Bağlı liste kullanımının dizi kullanımından farkı programcının, başta kaç adet elemanla işlem yapılacağını bilmek zorunda olmamasıdır. Listenin boyutu dizilerde olduğu gibi önceden belirlenmez ve dinamik olarak yaratılır. Kaç adet elemanla çalışacağımızı bilmediğimiz durumlarda bağlı liste kullanmak bize büyük avantajlar sağlar çünkü sadece birkaç elemanın kullanılacağı bir diziyi, ne kadarlık bir bellek alanı ayıracağımızı bilmeden -örneğin 1000 elemanı barındıracak şekilde- yaratmak bize bellek kullanım miktarı açısından büyük çaplı bir kayba mal olacaktır.

BAĞLI LİSTELER (LINKED LISTS) (DEVAM)

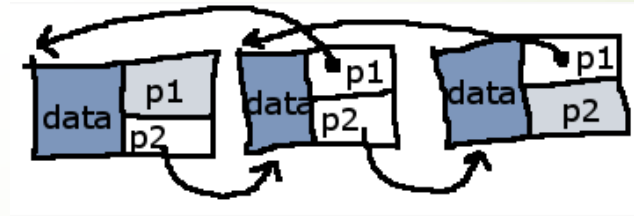
Tek Yönlü Bağlı Listeler : Listedeki elemanlar arasında sadece tek bir bağ vardır. Bu tür bağlı listelerde hareket yönü sadece listenin başından sonuna doğrudur. (Şekil - 1)

Çift Yönlü Bağlı Listeler : Listedeki elemanlar arasında iki yönlü bağ vardır. Elemanın bağlantı bilgisi bölümünde iki gösterici bulunur. Bu göstericinin biri kendisinden sonra gelen elemanı diğeri ise kendisinden önce gelen elemanın adres bilgisini tutar. Bu sayede listenin hem başından sonuna hem de listenin sonundan başına doğru hareket edilebilir. Bu yöntem daha esnek bir yapıya sahip olduğundan bazı problemlerin çözümünde daha işlevsel olabilmektedir. (Şekil - 2)

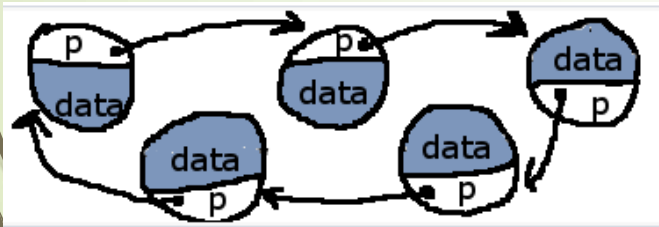
Dairesel Bağlı Listeler : Listedeki elemanlar arasında tek yönlü bağ vardır. Tek yönlü bağlı listelerden tek farkı ise son elemanın göstericisi ilk listenin ilk elemanının adresini göstermesidir. Bu sayede eğer listedeki elemanlardan birinin adresini biliyorsak listedeki bütün elemanlara erişebiliriz. (Şekil - 3)



Tek Yönlü Bağlı Listeler



Çift Yönlü Bağlı Listeler



Dairesel Bağlı Listeler

BAĞLI LİSTELER (LINKED LISTS) (KOD)

```
public class Dugum
{
    public Dugum next;
    public Object Veri;
}

public class LinkedList
{
    private Dugum head;

    public void dugumYazdir()
    {
        Dugum aktif = head;
        while (aktif != null)
        {
            Console.WriteLine(aktif.Veri);
            aktif = aktif.next;
        }
    }
}
```

BAĞLI LİSTELER (LINKED LISTS) (KOD)

```
public void OneEkle(Object Veri)
{
    Dugum eklenecek = new Dugum();

    eklenecek.Veris = Veri;
    eklenecek.next = head;

    head = eklenecek;
}
```

```
public void SonaEkle(Object Veri)
{
    if (head == null)
    {
        head = new Dugum();

        head.Veris = Veri;
        head.next = null;
    }
    else
    {
        Dugum eklenecek = new Dugum();
        eklenecek.Veris = Veri;

        Dugum aktif = head;
        while (aktif.next != null)
        {
            aktif = aktif.next;
        }

        aktif.next = eklenecek;
    }
}
```

BAĞLI LİSTELER (LINKED LISTS) (KOD)

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Once Ekle:");
        LinkedList Listel = new LinkedList();

        Listel.OneEkle("Veri");
        Listel.OneEkle("Yapilari");
        Listel.OneEkle("Dersi");
        Listel.dugumYazdir();

        Console.WriteLine();

        Console.WriteLine("Sonrasına ekle:");
        LinkedList Liste2 = new LinkedList();

        Liste2.OneEkle("Veri");
        Liste2.OneEkle("Yapilari");
        Liste2.OneEkle("Dersi");
        Liste2.dugumYazdir();

        Console.ReadLine();
    }
}
```