

## Sınıf yapısı içinde Çiftli Linked List

```
class Ciftli
{
    public int data;
    public Ciftli next;
    public Ciftli prev;
    public void Print()
    {
        Console.WriteLine(" [" + data + "] - ");
        if (next != null) next.Print();
        else Console.WriteLine();
    }
    public void SatirSatirOku()
    {
        if (prev != null)
        {
            Console.ReadLine();
            Console.WriteLine(" [" + data + "] - ");
            prev.SatirSatirOku();
        }
        else Console.WriteLine();
    }
    public void DataSil(int dataveri)
    {
        if (next != null)
        {
            if (data == dataveri)
            {
                Console.WriteLine("Data Değer " + dataveri + " olduğu için SİL : " + data);
                //burada silinen kayıt olduğu için önceki kaydın bir sonraki
                değeri silinen kaydın sonraki değeri atanıyor
                prev.next = next;
                //burada silinen kayıt olduğu için sonraki kaydın bir önceki
                değeri silinen kaydın önceki değeri atanıyor
                next.prev = prev;
            }
            //Metoda yeniden yönlendirme yapılıyor
            next.DataSil(dataveri);
        }
    }
    public Ciftli KayitSiral(Ciftli baslik)
    {
        Ciftli c = baslik, yenisira = null;
        while (c != null)
        {
            Ciftli gecici = c.prev;
            c.next = yenisira;
            yenisira = c;
            c = gecici;
        }
        return yenisira;
    }
}
```

Burada yazılan kodu bilgisayarda denemenizi tavsiye ediyorum sadece buradaki kodlara bakarak anlamak pek mümkün olmayacaktır.

Console uygulaması içinde Main Alanına yazılacak kodlar

```
//Denemeler için rastgele 0dan 9a kadar değer alacak 15 elemanlı
çiftli linked list oluşturuyoruz.
Random rnd = new Random();
Ciftli baslik = null;
Ciftli qc = null;
Ciftli onceki = null;
Ciftli x = null; //Örnek sorudaki x rastgele bir elemanı işaret
edecek
int rastgele = rnd.Next(15); //rastgele olacak elemanın for içinde
kaçıncı olduğu belirleniyor
Console.WriteLine("Rastgele Sıra : "+rastgele);

for (int i = 0; i < 15; i++)
{
    qc = new Ciftli();
    qc.data = rnd.Next(9);
    qc.next = null;
    if (baslik == null)
    {
        baslik = qc;
        onceki = baslik;
        qc.prev = null;
        Console.WriteLine("Sıra " + i + ": " + qc.data);
    }
    else
    {
        baslik = qc;
        qc.prev = onceki;
        qc.prev.next = baslik;
        onceki = baslik;
        Console.WriteLine("Sıra " + i + ": " + qc.data);
    }
    //Rastgele bir eleman işaret etmek için rastgele tutulan değere
eşit olduğunda x değerine mevcut qc verisi atanıyor.
    if(rastgele == i)
    {
        x = qc;
        Console.WriteLine("X Data : " + x.data);
    }
}
```

```
//Ekran Çıktısını kontrol etmek için rastgele bir noktayı işaret eden x değerinin önceki ve sonraki kaydı ekrana bastırılıyor.
if(x.prev != null)
    Console.WriteLine(" X Data Önceki: " + x.prev.data);
if(x.next != null)
    Console.WriteLine(" X Data Sonraki : " + x.next.data);

//Bu şekilde Sondan başa doğru okuma yapıyoruz.
Console.WriteLine(" Başlık Data: " + baslik.data);
Console.WriteLine(" Başlık Prev Data: " + baslik.prev.data);
Console.WriteLine(" Başlık Prev -> Prev Data: " +
baslik.prev.prev.data);
Console.WriteLine(" Başlık Prev -> Prev -> Prev Data: " +
baslik.prev.prev.prev.data);

//Sıralamayı tersine çeviriyoruz.
Ciftli c = baslik, yenisira = null;

//Tersten Sıralama işlemi yaptırıyoruz bunun için recursive yapıda Çiftli Sınıf içinde bir metod tanımladık işlem orada yapılıp yeni sıra geriye dönüyor.
yenisira = baslik.KayitSiralama(baslik);
//yenisira.Print();

//Yeniden sıraladığımız için baştan sona doğru okuma yapıyoruz.
Console.WriteLine(" Başlık Data: " + yenisira.data);
Console.WriteLine(" Başlık Next Data: " + yenisira.next.data);
Console.WriteLine(" Başlık Next -> Next Data: " +
yenisira.next.next.data);
Console.WriteLine(" Başlık Next -> Next -> Next Data: " +
yenisira.next.next.next.data);

//Kullanıcının belirleyeceği değerin olduğu data kayıtlarınız listeden çıkartıyoruz
Ciftli Yazdir = yenisira;
```

```
//Kullanıcıya data değeri kaç olan kayıtları silmek istediğini
soruyoruz.
Console.WriteLine("Silinmesini istediğiniz data değerini giriniz:");
int cikar1 = Convert.ToInt32(Console.ReadLine());

//Sınıf içinde tanımlanan metod Recursive yapıda data değeri
belirtilen kayıtları silme
Yazdir.DataSil(cikar1);

//Sınıf içinde istenilen data değerli kayıtları sildikten sonra yine
sınıf içinde tanımladığımız Print metodu ile son halini ekrana
basıyoruz.
Yazdir.Print();
Console.WriteLine("Sonraki işleme geçmek için Entera basın");
Console.ReadLine();

//Yazdır Ciftli linked list tekrar eski haline getiriliyor aşağıdaki
kod çalışması için
Yazdir = baslik;
//Recursive yapı olmadan istenilen data değerinde olan kayıtları
silme
int say = 1;
while (Yazdir != null)
{
    if (Yazdir.data == cikar1)
    {
        Console.WriteLine("SİL:" + say + " . eleman : " +
Yazdir.data);
        baslik.prev.next = Yazdir.next;
        baslik.next.prev = Yazdir.prev;
    }
    if (Yazdir.next != null)
    {
        Yazdir = Yazdir.next;
        baslik = Yazdir;
    }
    else
        Yazdir = null;
    say++;
}
```

```
//Sıralamayı yeniden baştan sona doğru olacak şekilde yapıyoruz.  
//Rekursiz yapı olmadan doğrudan burada ve tek while ile  
c = baslik;  
yenisira = null;  
while (c != null)  
{  
    Ciftli gecici = c.prev;  
    c.next = yenisira;  
    yenisira = c;  
    c = gecici;  
}  
baslik = yenisira;  
say = 1;  
  
//istenilen kayıtlar çıkartılmış olarak baştan sona ekrana  
bastıtıyoruz.  
Ciftli Goster = baslik;  
while (Goster != null)  
{  
    Console.WriteLine(say + " . eleman : " + Goster.data);  
    if (Goster.next != null)  
        Goster = Goster.next;  
    else  
        Goster = null;  
    say++;  
}
```

```
//Soru: x çiftli linked list'in bir bloğuna bakmaktadır. bu çiftli linked listin eleman sayısını bulma
```

```
Ciftli ileriye = baslik;
```

```
Ciftli geriye = baslik;
```

```
//-1 ile başlatmamızın nedeni kesişen bir değerin olması nedeniyle onu iki defa sayacağı için 0 yerine -1 dedik
```

```
int adet = -1;
```

```
//Aynı çiftli linked list işleminde üst sıralarda yapılan silme işlemleri sonucunda listede kalan eleman sayısını hesap edecektir.
```

```
while (ileriye != null)
```

```
{
```

```
    adet++;
```

```
    if (ileriye.next != null)
```

```
        ileriye = ileriye.next;
```

```
    else
```

```
        ileriye = null;
```

```
}
```

```
while (geriye != null)
```

```
{
```

```
    adet++;
```

```
    if (geriye.prev != null)
```

```
        geriye = geriye.prev;
```

```
    else
```

```
        geriye = null;
```

```
}
```

```
Console.WriteLine("X'in rastgele bir elemanını işaret ettiği çiftli linked list Eleman Sayısı : " + adet);
```

## Kırıkkale Üniversitesi Bilgisayar Mühendisliği 2. Sınıf Veri Yapıları Ders Notları

```
//Soru Baslik verisi Çiftli Linked Listin ilk elamanına bakıyor,  
//Kullanıcıdan alınan data değerine eşit olan kayıtlardan sonrasına  
yeni kayıt ekleme işlemi  
//Burada işler biraz karmaşık gelebilir elimizde 2 kutu olduğunu  
düşünün 1 ve 2 numaralı arasına bir kutu ekleyeceğiz  
//1 yerinde kalıyor sonrasına 2 numaralı yeni kutu geliyor 2  
numaralı kutu artık 3 oluyor yeni kutu 2 oluyor  
//bu mantıkta çiftli linked listin ilgili değerinin işaret ettiği  
önceki ve sonraki kayıtları güncellememiz gerekecek.  
Console.WriteLine("Hangi data değerinden sonra kayıt eklenmesi  
istediğiniz belirtin:");  
int ekledata = Convert.ToInt32(Console.ReadLine());  
Ciftli Ekleme = null;  
Ciftli Temp = baslik;  
while (Temp != null)  
{  
    //Temp.Print();  
    if (Temp.data == ekledata)  
    {  
        Ciftli yenikayit = new Ciftli();  
        yenikayit.data = rnd.Next(9);  
  
        //buralarda ilk ve son kayıt durumlarını da kontrol  
ettirmemiz gerekiyor  
        //kafa karışıklığı olmasın diye hoca derste anlatırken buna  
benzer şekilde yaptı  
  
        Temp.next.prev = yenikayit;  
        yenikayit.next = Temp.next;  
        yenikayit.prev = Temp;  
        Temp.next = yenikayit;  
        Console.WriteLine("Yeni Kayıt Eklendi");  
    }  
    if(Temp.next != null)  
        Ekleme = Temp.next;  
  
    if (Temp.next != null)  
    {  
        Temp = Temp.next;  
    }  
    else  
    {  
        Temp = null;  
    }  
}  
//Ekleme Kayıtlarını Yeniden Saydırıyoruz  
Ekleme = Ekleme.KayitSiralama(Ekleme);  
Ekleme.Print();  
Console.ReadLine();
```

Hazırlayan: Nurettin Dönmez kodlar kontrol edilmiştir ama insanlık hali gözden kaçan hatalı kod olması ihtimaline karşı kontrol etmeniz tavsiye edilir.