

# LINKED LİST 2. ÖDEVİ

ESMA BALIKÇI

24080410019

Oğrenci No:

Ders Kodu:

Harf Notu:

EKLE

SİL

ÖĞRENCİ LİSTELE

DERS LİSTELE

## --DERS BAŞLIKLARI--

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace linkedlist2odevi
{
    public class DersBaslikDugumu
    {
        public string DersKodu;
        public DersBaslikDugumu SonrakiDers;
        public NotDugumu IlkOgrenci;
        public DersBaslikDugumu(string dersKodu)
        {
            this.DersKodu = dersKodu;
            this.SonrakiDers = null;
            this.Ilkgrenci = null;
        }
    }
}
```

## --NOTLAR--

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace linkedlist2odevi
{
    public class NotDugumu
    {
        public int OgrenciNumarasi;
        public string DersKodu;
        public string HarfNotu;

        public NotDugumu OgrencininSonrakiDersi;

        public NotDugumu DerstekiSonrakiOgrenci;

        public NotDugumu(int ogrenciNo, string dersKodu, string harfNotu)
        {
            this.OgrenciNumarasi = ogrenciNo;
            this.DersKodu = dersKodu;
            this.HarfNotu = harfNotu;
            this.OgrencininSonrakiDersi = null;
            this.DerstekiSonrakiOgrenci = null;
        }
    }
}

```

## --ÖĞRENCİ BAŞLIKLARI--

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace linkedlist2odevi
{
    public class OgrenciBaslikDugumu
    {
        public int OgrenciNumarasi;
        public OgrenciBaslikDugumu SonrakiOgrenci;
        public NotDugumu IlkDers;

        public OgrenciBaslikDugumu(int ogrenciNo)
        {
            this.OgrenciNumarasi = ogrenciNo;
            this.SonrakiOgrenci = null;
            this.Ilkders = null;
        }
    }
}

```

## --NOT SİSTEMİ--

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace linkedlist2odevi

```

```

{
    public class NotSistemi
    {
        private OgrenciBaslikDugumu ogrenciListesiBasi = null;
        private DersBaslikDugumu dersListesiBasi = null;
        private OgrenciBaslikDugumu OgrenciBulVeyaOlustur(int ogrenciNo)
        {
            OgrenciBaslikDugumu temp = ogrenciListesiBasi;
            OgrenciBaslikDugumu onceki = null;

            while (temp != null && temp.OgrenciNumarasi < ogrenciNo)
            {
                onceki = temp;
                temp = temp.SonrakiOgrenci;
            }

            if (temp != null && temp.OgrenciNumarasi == ogrenciNo)
            {
                return temp;
            }

            OgrenciBaslikDugumu yeniOgrenciBaslik = new
OgrenciBaslikDugumu(ogrenciNo);

            if (onceki == null)
            {
                yeniOgrenciBaslik.SonrakiOgrenci = ogrenciListesiBasi;
                ogrenciListesiBasi = yeniOgrenciBaslik;
            }
            else
            {
                yeniOgrenciBaslik.SonrakiOgrenci = temp;
                onceki.SonrakiOgrenci = yeniOgrenciBaslik;
            }

            return yeniOgrenciBaslik;
        }

        private DersBaslikDugumu DersBulVeyaOlustur(string dersKodu)
        {
            DersBaslikDugumu temp = dersListesiBasi;
            DersBaslikDugumu onceki = null;

            while (temp != null && string.Compare(temp.DersKodu, dersKodu) < 0)
            {
                onceki = temp;
                temp = temp.SonrakiDers;
            }

            if (temp != null && temp.DersKodu == dersKodu)
            {
                return temp;
            }
            DersBaslikDugumu yeniDersBaslik = new DersBaslikDugumu(dersKodu);

            if (onceki == null)
            {
                yeniDersBaslik.SonrakiDers = dersListesiBasi;
                dersListesiBasi = yeniDersBaslik;
            }
            else
            {
                yeniDersBaslik.SonrakiDers = temp;
            }
        }
    }
}

```

```

        onceki.SonrakiDers = yeniDersBaslik;
    }

    return yeniDersBaslik;
}
public bool NotEkle(int ogrenciNo, string dersKodu, string harfNotu)
{
    OgrenciBaslikDugumu ogrenciBaslik = OgrenciBulVeyaOlustur(ogrenciNo);
    DersBaslikDugumu dersBaslik = DersBulVeyaOlustur(dersKodu);
    NotDugumu yeniNot = new NotDugumu(ogrenciNo, dersKodu, harfNotu);

    NotDugumu tempYatay = ogrenciBaslik.Ilkders;
    NotDugumu oncekiYatay = null;
    while (tempYatay != null && string.Compare(tempYatay.DersKodu,
dersKodu)
        < 0)
    {
        oncekiYatay = tempYatay;
        tempYatay = tempYatay.OgrencininSonrakiDersi;
    }

    if (tempYatay != null && tempYatay.DersKodu == dersKodu)
    {
        return false;
    }

    if (oncekiYatay == null)
    {
        yeniNot.OgrencininSonrakiDersi = ogrenciBaslik.Ilkders;
        ogrenciBaslik.Ilkders = yeniNot;
    }
    else
    {
        yeniNot.OgrencininSonrakiDersi = tempYatay;
        oncekiYatay.OgrencininSonrakiDersi = yeniNot;
    }

    NotDugumu tempDikey = dersBaslik.IlkoGrenci;
    NotDugumu oncekiDikey = null;
    while (tempDikey != null && tempDikey.OgrenciNumarasi < ogrenciNo)
    {
        oncekiDikey = tempDikey;
        tempDikey = tempDikey.DerstekiSonrakiOgrenci;
    }

    if (tempDikey != null && tempDikey.OgrenciNumarasi == ogrenciNo)
    {
        return false;
    }

    if (oncekiDikey == null)
    {
        yeniNot.DerstekiSonrakiOgrenci = dersBaslik.IlkoGrenci;
        dersBaslik.IlkoGrenci = yeniNot;
    }
    else
    {
        yeniNot.DerstekiSonrakiOgrenci = tempDikey;
        oncekiDikey.DerstekiSonrakiOgrenci = yeniNot;
    }

    return true;
}

```

```

public bool NotSil(int ogrenciNo, string dersKodu)
{
    OgrenciBaslikDugumu ogrenciBaslik = OgrenciBulVeyaOlustur(ogrenciNo);
    if (ogrenciBaslik == null) return false;

    NotDugumu tempYatay = ogrenciBaslik.Ilkders;
    NotDugumu oncekiYatay = null;
    while (tempYatay != null && tempYatay.DersKodu != dersKodu)
    {
        oncekiYatay = tempYatay;
        tempYatay = tempYatay.OgrencininSonrakiDersi;
    }

    if (tempYatay == null) return false;

    if (oncekiYatay == null)
    {
        ogrenciBaslik.Ilkders = tempYatay.OgrencininSonrakiDersi;
    }
    else
    {
        oncekiYatay.OgrencininSonrakiDersi =
tempYatay.OgrencininSonrakiDersi;
    }

    DersBaslikDugumu dersBaslik = DersBulVeyaOlustur(dersKodu);
    if (dersBaslik == null) return false;

    NotDugumu tempDikey = dersBaslik.IlkoGrenci;
    NotDugumu oncekiDikey = null;
    while (tempDikey != null && tempDikey.OgrenciNumarasi != ogrenciNo)
    {
        oncekiDikey = tempDikey;
        tempDikey = tempDikey.DerstekiSonrakiOgrenci;
    }

    if (tempDikey == null) return false;

    if (oncekiDikey == null)
    {
        dersBaslik.IlkoGrenci = tempDikey.DerstekiSonrakiOgrenci;
    }
    else
    {
        oncekiDikey.DerstekiSonrakiOgrenci =
tempDikey.DerstekiSonrakiOgrenci;
    }

    return true;
}

public List<string> DerstekiOgrencileriListele(string dersKodu)
{
    List<string> sonuc = new List<string>();
    DersBaslikDugumu dersBaslik = DersBulVeyaOlustur(dersKodu);

    if (dersBaslik == null || dersBaslik.IlkoGrenci == null)
    {
        sonuc.Add("Bu dersi alan öğrenci bulunamadı.");
        return sonuc;
    }
}

```

```

    }

    NotDugumu temp = dersBaslik.IlkOgrenci;
    while (temp != null)
    {
        sonuc.Add($"Öğrenci No: {temp.OgrenciNumarasi} - Harf Notu: {temp.HarfNotu}");
        temp = temp.DerstekisiSonrakiOgrenci;
    }

    return sonuc;
}

public List<string> OgrencininDersleriniListele(int ogrenciNo)
{
    List<string> sonuc = new List<string>();
    OgrenciBaslikDugumu ogrenciBaslik = OgrenciBulVeyaOlustur(ogrenciNo);

    if (ogrenciBaslik == null || ogrenciBaslik.IlkDers == null)
    {
        sonuc.Add("Bu öğrencinin aldığı ders bulunamadı.");
        return sonuc;
    }

    NotDugumu temp = ogrenciBaslik.IlkDers;
    while (temp != null)
    {
        sonuc.Add($"Ders Kodu: {temp.DersKodu} - Harf Notu: {temp.HarfNotu}");
        temp = temp.OgrencininSonrakiDersi;
    }

    return sonuc;
}
}
}

```

## --FORM--

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace linkedlist2odevi
{
    public partial class Form1 : Form
    {
        private NotSistemi benimNotSistemim = new NotSistemi();
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        int ogrenciNo = Convert.ToInt32(txtOgrenciNo.Text);
        string dersKodu = txtDersKodu.Text;
        string harfNotu = txtHarfNotu.Text;

        if (string.IsNullOrEmpty(dersKodu)
||string.IsNullOrEmpty(harfNotu))
        {
            MessageBox.Show("Ders kodu ve harf notu boş olamaz!");
            return;
        }

        bool basarili = benimNotSistemim.NotEkle(ogrenciNo, dersKodu,
harfNotu);

        if (basarili)
        {
            MessageBox.Show("Not başarıyla eklendi.");
            // Temizlik
            txtOgrenciNo.Clear();
            txtDersKodu.Clear();
            txtHarfNotu.Clear();
        }
        else
        {
            MessageBox.Show("Hata: Bu not zaten kayıtlı olabilir.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata oluştu: " + ex.Message);
    }
}

private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        int ogrenciNo = Convert.ToInt32(txtOgrenciNo.Text);
        string dersKodu = txtDersKodu.Text;

        bool basarili = benimNotSistemim.NotSil(ogrenciNo, dersKodu);

        if (basarili)
        {
            MessageBox.Show("Kayıt başarıyla silindi.");
        }
        else
        {
            MessageBox.Show("Hata: Silinecek kayıt bulunamadı.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata oluştu: " + ex.Message);
    }
}

```

```

private void ogrenciListele_Click(object sender, EventArgs e)
{
    try
    {
        string dersKodu = txtDersKodu.Text;
        var sonuclar
=benimNotSistemim.DerstekiOgrencileriListele(dersKodu);

        lstSonuc.Items.Clear();
        foreach (string satir in sonuclar)
        {
            lstSonuc.Items.Add(satir);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata oluřtu: " + ex.Message);
    }
}

private void dersListele_Click(object sender, EventArgs e)
{
    try
    {
        int ogrenciNo = Convert.ToInt32(txtOgrenciNo.Text);
        var sonuclar
=benimNotSistemim.OgrencininDersleriniListele(ogrenciNo);

        lstSonuc.Items.Clear();
        foreach (string satir in sonuclar)
        {
            lstSonuc.Items.Add(satir);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata oluřtu: " + ex.Message);
    }
}
}
}
}

```