LİNKED LİST 2 ÖDEVİ

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

namespace linked_list2
{
    public class NotNode
    {
        public int OgrenciNo;
        public string DersKodu;
        public string HarfNotu;
        public NotNode SonrakiOgrenci;
        public NotNode SonrakiDers;
    }
}
```

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

namespace linked_list2
{
    public class OgrenciHeadNode
    {
        public int OgrenciNo;
        public NotNode IlkDers;
    }
}
```

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

namespace linked_list2
{
    public class DersHeadNode
    {
        public string DersKodu;
        public NotNode IlkOgrenci;
    }
}
```

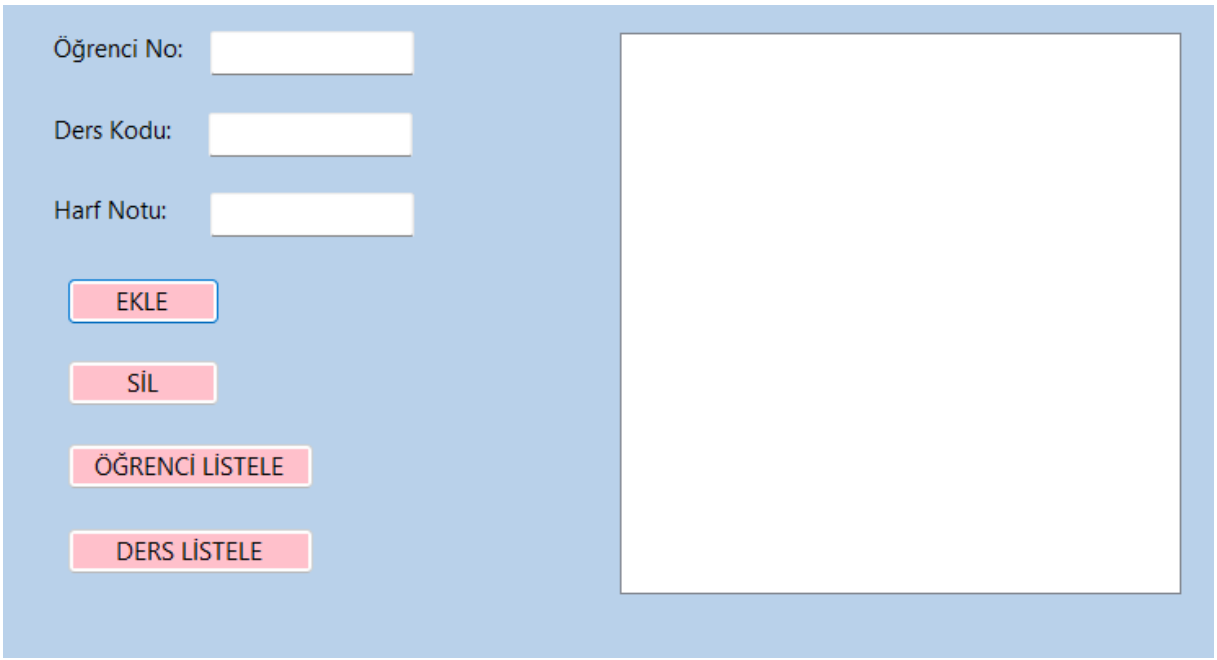
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

namespace linked_list2
{
    public class VeriYapisi
    {
        public List<OgrenciHeadNode> Ogrenciler;
        public List<DersHeadNode> Dersler;

        public VeriYapisi()
        {
            Ogrenciler = new List<OgrenciHeadNode>();
            Dersler = new List<DersHeadNode>();
        }

        public string Ekle(int ogrenciNo, string dersKodu, string harfNotu)
        {
            OgrenciHeadNode oHead = Ogrenciler.Find(o => o.OgrenciNo ==
ogrenciNo);
            if (oHead == null)
            {
                oHead = new OgrenciHeadNode { OgrenciNo = ogrenciNo };
                Ogrenciler.Add(oHead);
            }

            DersHeadNode dHead = Dersler.Find(d => d.DersKodu == dersKodu);
            if (dHead == null)
            {
                dHead = new DersHeadNode { DersKodu = dersKodu };
                Dersler.Add(dHead);
            }

            NotNode newNode = new NotNode
            {
                OgrenciNo = ogrenciNo,
                DersKodu = dersKodu,
                HarfNotu = harfNotu
            };

            NotNode prevDers = null;
            NotNode currentDers = oHead.IlkDers;
            while (currentDers != null && string.Compare(currentDers.DersKodu,
dersKodu) < 0)
            {
                prevDers = currentDers;
                currentDers = currentDers.SonrakiDers;
            }

            if (currentDers != null && currentDers.DersKodu == dersKodu)
                return "Hata: Kayıt zaten var.";

            newNode.SonrakiDers = currentDers;
            if (prevDers == null)
                oHead.IlkDers = newNode;
            else
                prevDers.SonrakiDers = newNode;

            NotNode prevOgr = null;
            NotNode currentOgr = dHead.IlkOgrenci;
            while (currentOgr != null && currentOgr.OgrenciNo < ogrenciNo)
            {
                prevOgr = currentOgr;
                currentOgr = currentOgr.SonrakiOgrenci;
```

```
            }

            newNode.SonrakiOgrenci = currentOgr;
            if (prevOgr == null)
                dHead.IlkOgrenci = newNode;
            else
                prevOgr.SonrakiOgrenci = newNode;

            return $"Eklendi: {ogrenciNo} - {dersKodu}";
        }

        public string Sil(int ogrenciNo, string dersKodu)
        {
            OgrenciHeadNode oHead = Ogrenciler.Find(o => o.OgrenciNo ==
ogrenciNo);
            DersHeadNode dHead = Dersler.Find(d => d.DersKodu == dersKodu);

            if (oHead == null || dHead == null)
                return "Hata: Öğrenci veya ders bulunamadı.";

            NotNode prevDers = null;
            NotNode currentDers = oHead.IlkDers;
            while (currentDers != null && currentDers.DersKodu != dersKodu)
            {
                prevDers = currentDers;
                currentDers = currentDers.SonrakiDers;
            }

            if (currentDers == null)
                return "Hata: Öğrenci bu dersi almıyor.";

            if (prevDers == null)
                oHead.IlkDers = currentDers.SonrakiDers;
            else
                prevDers.SonrakiDers = currentDers.SonrakiDers;

            NotNode prevOgr = null;
            NotNode currentOgr = dHead.IlkOgrenci;
            while (currentOgr != null && currentOgr.OgrenciNo != ogrenciNo)
            {
                prevOgr = currentOgr;
                currentOgr = currentOgr.SonrakiOgrenci;
            }

            if (currentOgr == null)
                return "Hata: Derste bu öğrenci yok (Veri tutarsızlığı).";

            if (prevOgr == null)
                dHead.IlkOgrenci = currentOgr.SonrakiOgrenci;
            else
                prevOgr.SonrakiOgrenci = currentOgr.SonrakiOgrenci;

            return $"Silindi: {ogrenciNo} - {dersKodu}";
        }

        public List<string> DersiAlanOgrencileriListele(string dersKodu)
        {
            List<string> sonuc = new List<string>();
            DersHeadNode dHead = Dersler.Find(d => d.DersKodu == dersKodu);
            if (dHead != null)
            {
                NotNode current = dHead.IlkOgrenci;
                while (current != null)
```

```
                {
                    sonuc.Add($"Öğrenci No: {current.OgrenciNo}, Not:
{current.HarfNotu}");
                    current = current.SonrakiOgrenci;
                }
            }
            return sonuc;
        }

        public List<string> OgrencininDersleriniListele(int ogrenciNo)
        {
            List<string> sonuc = new List<string>();
            OgrenciHeadNode oHead = Ogrenciler.Find(o => o.OgrenciNo ==
ogrenciNo);
            if (oHead != null)
            {
                NotNode current = oHead.IlkDers;
                while (current != null)
                {
                    sonuc.Add($"Ders Kodu: {current.DersKodu}, Not:
{current.HarfNotu}");
                    current = current.SonrakiDers;
                }
            }
            return sonuc;
        }
    }
}

namespace linked_list2
{
    public partial class Form1 : Form
    {
        private VeriYapisi sistem = new VeriYapisi();
        public Form1()
        {
            InitializeComponent();
        }

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

                string sonuc = sistem.Ekle(ogrenciNo, dersKodu, harfNotu);
                MessageBox.Show(sonuc, "İşlem Sonucu", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata: " + ex.Message, "Hata",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnSil_Click(object sender, EventArgs e)
        {
            try
            {
                int ogrenciNo = int.Parse(txtOgrenciNo.Text);
```

```
                string dersKodu = txtDersKodu.Text;

                string sonuc = sistem.Sil(ogrenciNo, dersKodu);
                MessageBox.Show(sonuc, "İşlem Sonucu", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata: " + ex.Message, "Hata",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnOgrenciListele_Click(object sender, EventArgs e)
        {
            lstSonuc.Items.Clear();
            try
            {
                int ogrenciNo = int.Parse(txtOgrenciNo.Text);
                List<string> liste =
sistem.OgrencininDersleriniListele(ogrenciNo);

                if (liste.Count == 0)
                    lstSonuc.Items.Add("Kayıt bulunamadı.");
                else
                    foreach (string s in liste)
                        lstSonuc.Items.Add(s);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata: " + ex.Message, "Hata",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnDersListele_Click(object sender, EventArgs e)
        {
            lstSonuc.Items.Clear();
            try
            {
                string dersKodu = txtDersKodu.Text;
                List<string> liste =
sistem.DersiAlanOgrencileriListele(dersKodu);

                if (liste.Count == 0)
                    lstSonuc.Items.Add("Kayıt bulunamadı.");
                else
                    foreach (string s in liste)
                        lstSonuc.Items.Add(s);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata: " + ex.Message, "Hata",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```


Öğrenci No:
Ders Kodu:
Harf Notu:
EKLE
SİL
ÖĞRENCİ LİSTELE
DERS LİSTELE