

ISIM: ABDOU VALERIO FOMA KENFACK

NUMARA: 24080410152

BÖLÜM: Bilgisayar Mühendiliği

Linked list 2. Ödev

K adet ders için N adet öğrencinin numara ve harf ortalamalarını bağlı dizilerle gösteriniz.

Her bir node öğrenci numarası, dersin kodu, dersin harf ortalaması bilgilerini bulunduracak.

Her öğrencinin numarası headNode içindeki bilgi olacak.

Her dersin kodu headNode içindeki bilgi olacak.

Her bir node aynı derste bir sonraki öğrenciyi gösterecek.

Her bir node aynı kişinin diğer dersini gösterecek.

Program Java veya C# Windows application olarak hazırlanacak ve aşağıdaki işlemleri butonlarla yapacak.

- 1- Bir öğrenciyi yeni bir ders ekleme
- 2- Bir derse yeni bir öğrenci ekleme
- 3- Bir öğrencinin bir dersini silme
- 4- Bir derste bir öğrenciyi silme
- 5- Bir derste tüm öğrencileri numara sırasına göre sıralı listeleme
- 6- Bir öğrencinin aldığı tüm dersleri ders koduna göre sıralı listeleme

AÇIKLAMA

Şunları içeren bir Düğüm oluşturdum:

1. Öğrenci numarasını tutan bir tam sayı.
2. Ders numarasını ve ders harf ortalamasını bir sözlük.
3. Bir işaretçi.

NB: Ayrıca verilerimin bozulmasını önlemek için Nesne Yönelimli Programlama kurallarını da uyguladım

Form kısmıyla ilgili olarak 2 panel (panelMainMenu ve panelStuDersOperations) kullandım.

1. panelMainMenu, ana menüyü görüntülemek için kullanılır ve farklı işlemlere erişmek için düğmeler içerir.
2. panelStuDersOperations, tüm farklı işlemlerin gerçekleştirildiği yerdir. Kullanıcının seçtiği işleme bağlı olarak bazı kontroller (textbox, buttons, listbox...) görünür hale getirilebilir veya getirilemez.

KODU

Linklist.cs

```
namespace Ogrenci_Ders_Linklist
{
    public class Node
    {
        public int OgrenciNo { get; private set; }
        public Dictionary<int, string> Derskodu_HarfOrt { get; private set; }
        public Node Next { get; internal set; }

        public Node(int ogrenciNo, Dictionary<int, string> derskodu_harfOrt)
        {
            OgrenciNo = ogrenciNo;
            Derskodu_HarfOrt = derskodu_harfOrt;
            Next = null;
        }
    }

    public class BagliList
    {
        private Node headNode;
        private Node tailNode;

        public BagliList()
        {
            headNode = null;
            tailNode = null;
        }

        public void OgrenciDersEkler(int orgNum, int dersNum, string harf)
        {
            //Eğer öğrenci varsa
            Node current = headNode;
            while (current != null)
            {
                if (current.OgrenciNo == orgNum)
                {
                    //Öğrenciye ders ekliyor(ders yoksa)
                    if (!current.Derskodu_HarfOrt.ContainsKey(dersNum))
                    {
                        current.Derskodu_HarfOrt.Add(dersNum, harf);
                        MessageBox.Show($"Ders {dersNum} öğrenci {orgNum} için eklendi!");
                    }
                }
                else
                {
                    MessageBox.Show("Bu ders zaten öğrenciye kayıtlı!");
                }
            }
        }
    }
}
```

```

        return;
    }
    current = current.Next;
}

//Eğer öğrenci yoksa
var dict = new Dictionary<int, string>()
{
    { dersNum, harf }
};
Node newNode = new Node(orgNum, dict);
newNode.Next = headNode;
headNode = newNode;
if (tailNode == null)
{
    tailNode = newNode;
}
MessageBox.Show($"Yeni öğrenci {orgNum} ve ders {dersNum} eklendi!");
}

public void OgrenciDersSil(int orgNum, int dersNum)
{
    if (headNode == null)
    {
        MessageBox.Show("List boştur");
        return;
    }
    //Eğer aranan öğrenci head'de
    if (headNode.OgrenciNo == orgNum)
    {
        if (headNode.Derskodu_HarfOrt.ContainsKey(dersNum))
        {
            headNode.Derskodu_HarfOrt.Remove(dersNum);
            MessageBox.Show("İşlem Başarılı");
            //Ders olmayan öğrencileri silme
            if (headNode.Derskodu_HarfOrt.Count == 0)
            {
                headNode = headNode.Next;
                if (headNode == null)
                {
                    tailNode = null;
                }
                return;
            }
        }
        else
        {
            MessageBox.Show("Ders bulunamadı!");
            return;
        }
    }

    // Aranan öğrenciyi listede arama
    Node current = headNode;
    while (current.Next != null && current.Next.OgrenciNo != orgNum)
    {
        current = current.Next;
    }
    if (current.Next == null)
    {

```

```

        MessageBox.Show($"Öğrenci numara {orgNum} bulunmadı");
        return;
    }
    // It will make no sense starting from head because we tested it above
    Node current1 = current.Next;
    if (current1.Derskodu_HarfOrt.ContainsKey(dersNum))
    {
        current1.Derskodu_HarfOrt.Remove(dersNum);
        MessageBox.Show("İşlem Başarılı");
        // Eğer aranan öğrenci başka ders yoksa, silecek
        if (current1.Derskodu_HarfOrt.Count == 0)
        {
            current.Next = current1.Next;
            if (current1 == tailNode)
            {
                tailNode = current;
            }
        }
    }
    else
    {
        MessageBox.Show("Ders bulunamadı!");
    }
}

public List<string> OgrenciDersListing(int orgNum = -1, int dersNum = -1)
{
    List<string> sonucListesi = new List<string>();

    if (headNode == null)
    {
        MessageBox.Show("Liste boştur!");
        return sonucListesi;
    }
    Node current = headNode;
    if (orgNum != -1)
    {
        while (current != null && current.OgrenciNo != orgNum)
        {
            current = current.Next;
        }
        if (current == null)
        {
            MessageBox.Show($"Öğrenci numarası {orgNum} bulunamadı");
            return sonucListesi;
        }
        foreach (var item in current.Derskodu_HarfOrt)
        {
            sonucListesi.Add($"Ders: {item.Key}, Harf: {item.Value}");
        }
        return sonucListesi;
    }
    else if (dersNum != -1)
    {
        while (current != null)
        {
            if (current.Derskodu_HarfOrt.ContainsKey(dersNum))
            {
                sonucListesi.Add(current.OgrenciNo.ToString());
            }
            current = current.Next;
        }
    }
}

```

```

        }
        if (sonucListesi.Count == 0)
        {
            MessageBox.Show($"Ders kodu {dersNum} bulunmadı");
        }
        return sonucListesi;
    }
    else { return sonucListesi; }
}
}

}
Form.cs
using System.Drawing.Text;

namespace Ogrenci_Ders_Linklist
{
    public partial class Form1 : Form
    {
        private BagliList OgrenciDersListesi;
        public Form1()
        {
            OgrenciDersListesi = new BagliList();
            InitializeComponent();
            showMainMenu();
        }
        private void showMainMenu()
        {
            panelStuDersOperations.Visible = false;
        }

        private void buttonAddStudent_Click(object sender, EventArgs e)
        {
            listBox1.Visible = false;
            textBoxHarfOrt.Enabled = true;
            textBoxOrgNum.Enabled = true;
            textBoxDersKodu.Enabled = true;
            buttonDel.Visible = false;
            buttonQuery.Visible = false;
            buttonAdd.Visible = true;
            panelStuDersOperations.Visible = true;
        }

        private void buttonDelStu_Click(object sender, EventArgs e)
        {
            listBox1.Visible = false;
            textBoxHarfOrt.Enabled = false;
            textBoxOrgNum.Enabled = true;
            textBoxDersKodu.Enabled = true;
            buttonDel.Visible = true;
            buttonQuery.Visible = false;
            buttonAdd.Visible = false;
            panelStuDersOperations.Visible = true;
        }

        private void buttonStuListing_Click(object sender, EventArgs e)
        {
            listBox1.Visible = false;

```

```

        textBoxDersKodu.Enabled = true;
        textBoxOrgNum.Enabled = false;
        textBoxHarfOrt.Enabled = false;
        buttonDel.Visible = false;
        buttonQuery.Visible = true;
        buttonAdd.Visible = false;
        panelStuDersOperations.Visible = true;
    }

    private void buttonCourseListing_Click(object sender, EventArgs e)
    {
        listBox1.Visible = false;
        textBoxDersKodu.Enabled = false;
        textBoxOrgNum.Enabled = true;
        textBoxHarfOrt.Enabled = false;
        buttonDel.Visible = false;
        buttonQuery.Visible = true;
        buttonAdd.Visible = false;
        panelStuDersOperations.Visible = true;
    }

    private void buttonBack_Click(object sender, EventArgs e)
    {
        showMainMenu();
    }

    private void buttonAdd_Click(object sender, EventArgs e)
    {
        int orgNum;
        int derskodu;
        if (!int.TryParse(textBoxOrgNum.Text, out orgNum))
        {
            MessageBox.Show("Lütfen geçerli sayısal değerler girin!");
        }
        if (!int.TryParse(textBoxDersKodu.Text, out derskodu))
        {
            MessageBox.Show("Lütfen geçerli sayısal değerler girin!");
        }
        OgrenciDersListesi.OgrenciDersEkler(orgNum, derskodu,
        textBoxHarfOrt.Text);
        textBoxOrgNum.Clear();
        textBoxDersKodu.Clear();
        textBoxHarfOrt.Clear();
    }

    private void buttonDel_Click(object sender, EventArgs e)
    {
        int orgNum;
        int derskodu;
        if (!int.TryParse(textBoxOrgNum.Text, out orgNum))
        {
            MessageBox.Show("Lütfen geçerli sayısal değerler girin!");
        }
        if (!int.TryParse(textBoxDersKodu.Text, out derskodu))
        {
            MessageBox.Show("Lütfen geçerli sayısal değerler girin!");
        }
        OgrenciDersListesi.OgrenciDersSil(orgNum, derskodu);
        textBoxOrgNum.Clear();
    }

```

```

        textBoxDersKodu.Clear();
    }

    private void buttonQuery_Click(object sender, EventArgs e)
    {
        listBox1.Visible = true;
        listBox1.Items.Clear();
        List<string> sonuclar = new List<string>(); ;

        if (textBoxOrgNum.Enabled && !string.IsNullOrEmpty(textBoxOrgNum.Text))
        {
            // Bir öğrenci için dersleri listeleyin
            int orgNum = int.Parse(textBoxOrgNum.Text);
            sonuclar = OgrenciDersListesi.OgrenciDersListing(orgNum: orgNum);
        }
        else if (textBoxDersKodu.Enabled &&
!string.IsNullOrEmpty(textBoxDersKodu.Text))
        {
            // Bir dersteki öğrencileri listeleyin
            int dersKodu = int.Parse(textBoxDersKodu.Text);
            sonuclar = OgrenciDersListesi.OgrenciDersListing(dersNum: dersKodu);
        }
        else
        {
            MessageBox.Show("Lütfen geçerli sayısal değerler girin!");
        }
        listBox1.Items.AddRange(sonuclar.ToArray());
    }
}

```

ÇIKTI

Öğrenci Ders Yönetim Sistemi

Bir öğrenciye yeni bir ders ekleme

Bir öğrencinin bir dersini silme

Bir dersteki tüm öğrencileri numara sırasına göre sıralı listeleme

Bir öğrencinin aldığı tüm dersleri ders koduna göre sıralı listeleme

"Bir öğrenciye yeni bir ders ekleme" butona bastı

Form1

Geri

Öğrenci Numarası : 24081

Ders Kodu : 101

Harf ortalaması: AA

Ekler

Yeni öğrenci 24081 ve ders 101 eklendi!

OK

"Bir dersteki tüm öğrencileri numara sırasına göre sıralı listeleme" butona bastı

Form1

Geri

Öğrenci Numarası :

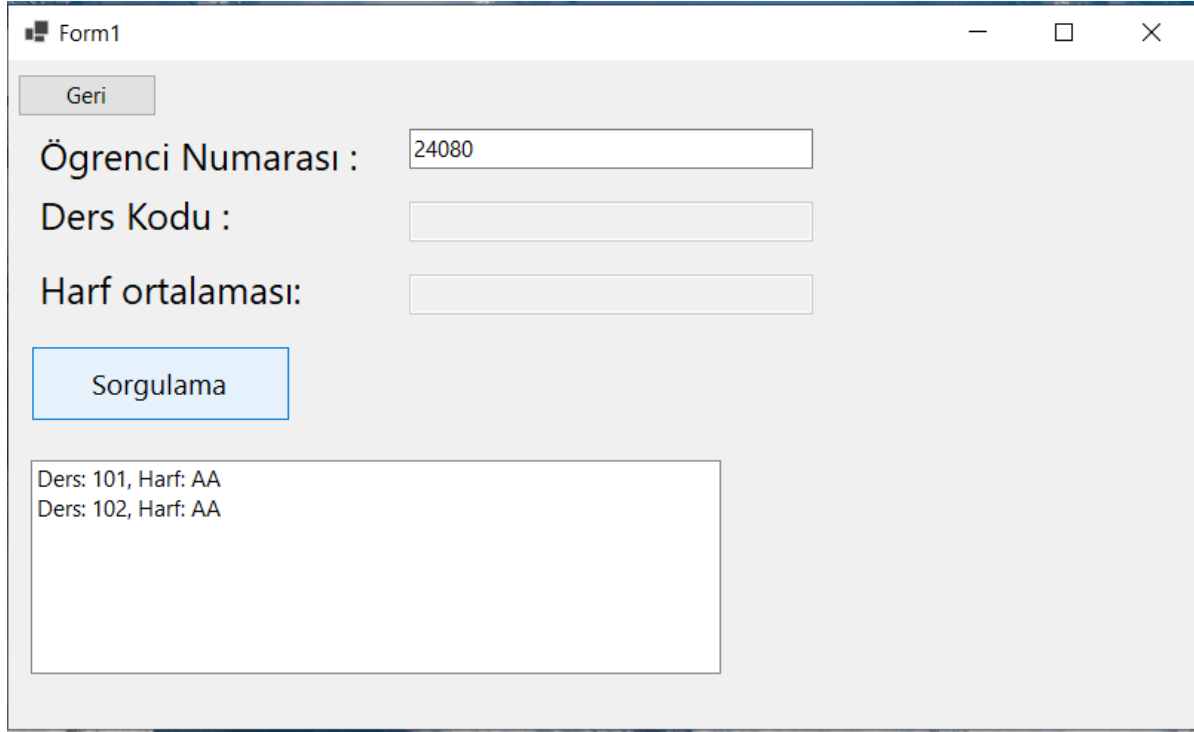
Ders Kodu : 101

Harf ortalaması:

Sorgulama

24081
24080

"Bir öğrencinin aldığı tüm dersleri ders koduna göre sıralı listeleme" butona bastı



Form1

Geri

Öğrenci Numarası : 24080

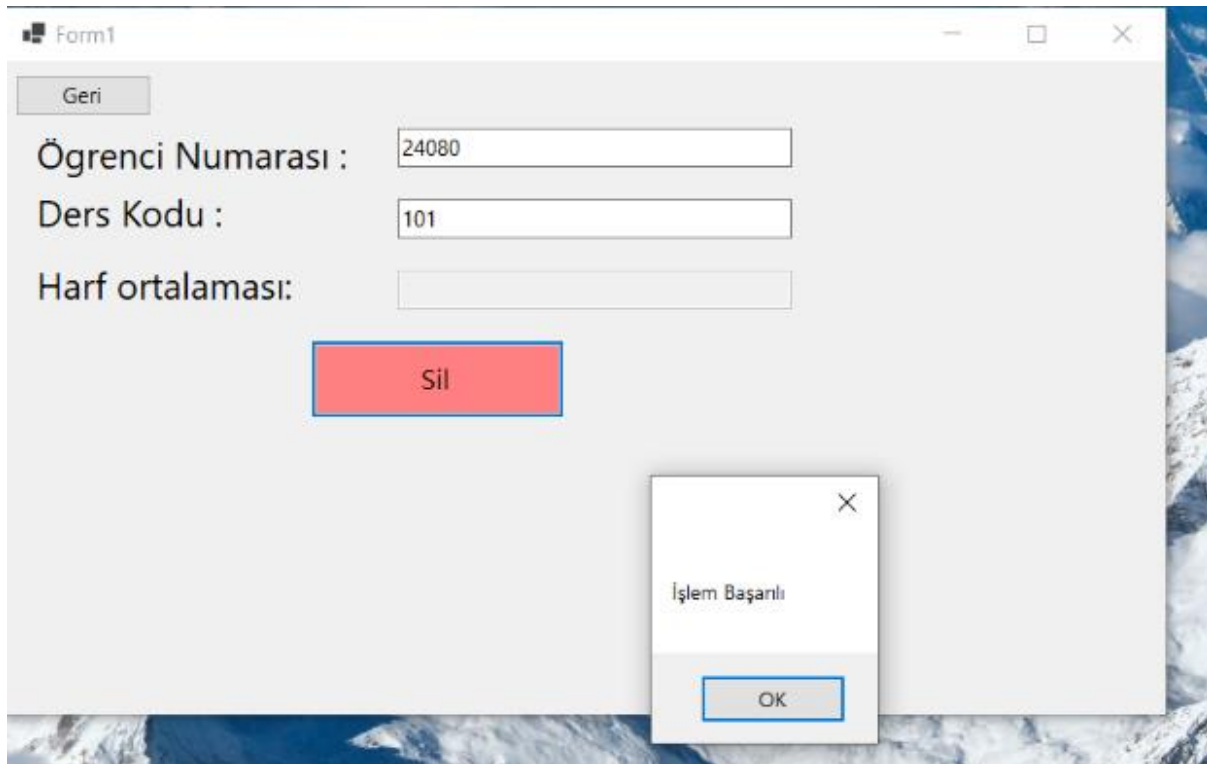
Ders Kodu :

Harf ortalaması:

Sorgulama

Ders: 101, Harf: AA
Ders: 102, Harf: AA

"Bir öğrencinin bir dersini silme " butona bastı



Form1

Geri

Öğrenci Numarası : 24080

Ders Kodu : 101

Harf ortalaması:

Sil

İşlem Başarılı

OK