



**BİTLİS
EREN
ÜNİVERSİTESİ**

AD: NATHANAELLE

SOYAD : BOPTI NGAH BONG

ÖĞRENCİ_NUMERASI: 24080410150

FAKÜLTE : MÜHENDİS-MİMARLIK FAKÜLTESİ

BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ

ÖDEV : Bir linked List örneği yapmanız istenmektedir.

Node tanımlama, yapıcı metod, Linked Listin;

başına sonuna ve istenen değerden sonrasın ve öncesine ekleme,

baştan-sonran ve istenen değer öncesini, sonrasını ve istenen değeri silme,

arama,

listeleme,

kullanıcıdan değer alarak ekleme fonksiyonlarının olduğu

bir öğrenci ad, soyad, numara bilgisinin olduğu bir Linked List kodu yazınız.

```

using System;
using static Sınıf;
class Sınıf
{
    public class Student
    {
        public string öğrenci_ad;
        public string öğrenci_soyad;
        public int öğrenci_numera;
        public Student Next;

        public Student(string student_name, string student_last_name, int
student_number)
        {
            öğrenci_ad = student_name;
            öğrenci_soyad = student_last_name;
            öğrenci_numera = student_number;
            Next = null;
        }
    }

    public class bağlıList
    {
        private Student head;
        private Student tail;
        public bağlıList()
        {
            head = null;
            tail = null;
        }

        // Başa eleman ekleme
        public void BasaEkle(string student_name, string student_last_name, int
student_number)
        {
            Student newStudent = new Student(student_name, student_last_name,
student_number);
            newStudent.Next = head;
            head = newStudent;
            Console.WriteLine($"{student_name} , {student_last_name},
{student_number} başa eklendi.");
        }

        // sona eleman ekleme
        public void SonaEkle(string student_name, string student_last_name, int
student_number)
        {
            Student newStudent = new Student(student_name, student_last_name,
student_number);
            if (head == null)
            {
                head = newStudent;
                tail = newStudent;
                Console.WriteLine($"{student_name} , {student_last_name},
{student_number} sona eklendi");
                return;
            }

```

```

        Student current = head;
        while (current.Next != null)
        {
            current = current.Next;
        }

        current.Next = newStudent;
        tail = newStudent;
        Console.WriteLine($"{student_name} ,{student_last_name},
{student_number} sona eklendi. ");
    }

    // Belirli bir değerin sonrasına eleman ekleme
    public void sonrasınaEkle(int yeniDegeri, string student_name, string
student_last_name, int student_number)
    {
        Student current = head;

        while (current != null && current.öğrenci_numera != yeniDegeri)
        {
            current = current.Next;
        }

        if (current == null)
        {
            Console.WriteLine($"{yeniDegeri} listede bulunamadı ");
            return;
        }

        // yeni elemanı mevcut elemandan sonrasına ekle
        Student newStudent = new Student(student_name, student_last_name,
student_number);
        newStudent.Next = current.Next;
        current.Next = newStudent;
        Console.WriteLine($"{yeniDegeri},{student_number}'nin sonrasına eklendi
");
    }

    // ilk elemanı silme
    public void bastanSil()
    {
        if (head == null)
        {
            Console.WriteLine("Liste boş, silecek eleman yok ");
            return;
        }
        head = head.Next;
        Console.WriteLine("ilk eleman silindi");
    }

    // son elemanı silme
    public void sonndanSil()
    {
        if (head == null)
        {
            Console.WriteLine("Liste boş , silinecek eleman yok");
            return;
        }
    }

```

```

    }
    if (head.Next == null)
    {
        head = null;
        Console.WriteLine("son eleman silindi ");
        return;
    }
    Student current = head;
    while (current.Next != null && current.Next.Next != null)
    {
        current = current.Next;
    }
    current.Next = null;
    Console.WriteLine("son eleman silindi. ");
}
// Belirli bir elemanı silme
public void Remove(int yeniDegeri)
{
    if (head == null)
    {
        Console.WriteLine("Liste boş, silecek eleman yok");
        return;
    }

    if (head.ogrenci_numera == yeniDegeri)
    {
        head = head.Next;
        Console.WriteLine($"{yeniDegeri} listeden silindi ");
        return;
    }
    Student current = head;
    while (current.Next != null && current.Next.ogrenci_numera !=
yeniDegeri)
    {
        current = current.Next;
    }
    if (current.Next == null)
    {
        Console.WriteLine($"{yeniDegeri}Listede bulunamadı ");
        return;
    }
    current.Next = current.Next.Next;
    Console.WriteLine($"{yeniDegeri} listeden silindi ");
}

// Listeyi yazdırma
public void Display()
{
    if (head == null)
    {
        Console.WriteLine("Liste boş. ");
        return;
    }
    Student current = head;
    Console.Write("Liste: ");
    return;
}

```

```

// Eleman arama
public void ara(int student_number)
{
    Student current = head;
    while (current.ogrenci_numera == student_number)
    {
        Console.WriteLine($"{student_number} listede bulunuyor ");
        return;
    }
    current = current.Next;
}
static void Main(string[] args)
{
    bagliList listim = new bagliList();
    bool running = true;
    while (running)
    {
        Console.WriteLine("\n ---- LinkedList İşlemleri-----");
        Console.WriteLine("1. Listeyi Göster");
        Console.WriteLine("2. Başa Eleman Ekle");
        Console.WriteLine("3. Sona Eleman Ekle");
        Console.WriteLine("4. Belirli bir Değerin sonrasına Eleman ekle ");
        Console.WriteLine("5. İlk Elemanı Sil");
        Console.WriteLine("6. Son Elemanı Sil");
        Console.WriteLine("7. Eleman sil");
        Console.WriteLine("8. Eleman Ara");
        Console.WriteLine("9. Çıkış");
        Console.WriteLine("Bir işlem seçin (1-9) : ");

        string input = Console.ReadLine();
        switch (input)
        {
            case "1":
                listim.Display();
                break;

            case "2":
                Console.Write("Başa eklenecek değeri girin : ");
                int BasaEkleValue = Convert.ToInt32(Console.ReadLine());
                Console.Write("1. İsim girin : ");
                string isim1 = Console.ReadLine();
                Console.Write("2. İsim girin : ");
                string isim2 = Console.ReadLine();

                listim.BasaEkle(isim1, isim2, BasaEkleValue);
                break;

            case "3":
                Console.Write("Sona eklenecek değeri girin : ");
                int DegeriSonaEkle = Convert.ToInt32(Console.ReadLine());
                Console.Write("1. İsim girin : ");
                string Sonaisim1 = Console.ReadLine();
                Console.Write("2. İsim girin : ");
                string Sonaisim2 = Console.ReadLine();
                listim.SonaEkle(Sonaisim1, Sonaisim2, DegeriSonaEkle);
                break;
        }
    }
}

```

```

    case "4":
        Console.WriteLine("Sonrasına eklemek istediğiniz değeri girin :");

        int student_no = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Yeni değeri girin : ");
        int yeniDegeri = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("1. İsim girin : ");
        string Sonraisim1 = Console.ReadLine();
        Console.WriteLine("2. İsim girin : ");
        string Sonraisim2 = Console.ReadLine();
        listim.sonrasınaEkle(yeniDegeri, Sonraisim1, Sonraisim2,
student_no);

        break;

    case "5":
        Console.WriteLine("Silecek elemanı girin: ");
        int removeValue = Convert.ToInt32(Console.ReadLine());
        listim.Remove(removeValue);
        break;

    case "6":
        Console.WriteLine("Aranacak elemanı girin : ");
        int DegerAra = Convert.ToInt32(Console.ReadLine());
        listim.ara(DegerAra);
        break;

    case "7":
        running = false;
        Console.WriteLine("Çıkılıyor....");
        break;

    default:
        Console.WriteLine("Geçersiz seçim, lütfen tekrar deneyim.");
        break;
    }
}
}
}
}
}
}
}

```