

Bağlı Liste (Linked List) Ödevi

AD: ESMA

SOYAD: BALIKÇI

NUMARA: 24080410019

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

namespace bagli_liste
{
    internal class Program
    {
        //Node Sınıfı: Her bir düğüm bir veriye ve bir sonraki düğüme işaret eder
        //veri yapısı tanımladık
        public class Node //Düğüm tanımlama
        {
            public int data; //verimiz
            public Node Next; //Node türünde sonraki pointerı

            public Node(int _data) // gönderilen değeri node a dönüştürür
            {
                data = _data;
                Next = null;
            }
        }

        //LinkedList sınıfı

        public class BagliList
        {
            private Node head;
            private Node tail;

            public BagliList()
            {
                head = null;
                tail = null;
            }

            public void BasaEkle(int value)
            {
                Node newNode = new Node(value);

                if (head == null) //içi boşsa
                {
                    head = newNode;
                    tail = newNode;
                }
                else
                {
                    newNode.Next = head;
                    head = newNode;
                }
                Console.WriteLine($"{value} başa eklendi.");
            }
        }
    }
}
```

```

//sona eleman ekleme
public void sonaEkle(int value)
{
    Node newNode = new Node(value);
    if (head == null) // içinde başka eleman yoksa
    {
        head = newNode;
        tail = newNode;
        Console.WriteLine($"{value} sona eklendi");
        return;
    }
    tail.Next = newNode;
    tail = newNode;
    Console.WriteLine($"{value} sona eklendi.");
}

//Eleman arama
public void ara(int value)
{
    Node current = head;
    while (current != null)
    {
        if (current.data == value)
        {
            Console.WriteLine($"{value} listede bulunuyor.");
            return;
        }
        current = current.Next;
    }
    Console.WriteLine($"{value} listede bulunmuyor.");
}

//yazdırma
public void display()
{
    if (head == null)
    {
        Console.WriteLine("Liste boş");
        return;
    }
    Node current = head;
    Console.Write("Liste: ");
    while (current != null)
    {
        Console.Write(current.data + "---->");
        current = current.Next;
    }
    Console.WriteLine();
}

//belirli bir değerin sonrasına eleman ekleme
public void BelirliDegerSonrasiEkle(int arananDeger, int yeniDeger)
{
    Node current = head;
    while (current != null)
    {
        if (current.data == arananDeger)
        {
            Node newNode = new Node(yeniDeger);
            newNode.Next = current.Next;
            current.Next = newNode;
        }
    }
}

```

```

        // Eğer son elemandan sonra ekleme yapıldıysa tail'i
        güncelle
        if (newNode.Next == null)
        {
            tail = newNode;
        }

        Console.WriteLine($"{yeniDeger}, {arananDeger} değerinin
        sonrasına eklendi");
        return;
    }
    current = current.Next;
}
Console.WriteLine($"{arananDeger} değeri listede bulunamadı,
ekleme yapılamadı");
}
// Eleman Silme
public void silme(int value)
{
    if (head==null)
    {
        Console.WriteLine("Liste boş silinecek eleman yok");
        return;
    }
    Node current = head;
    Node previous = null;
    if (current!=null && current.data == value) //ilk eleman boş
    değilse ve silinecek değere eşitse
    {
        head = current.Next; //headi bir sonrakine kaydır

        if (head== null) // liste tek elemanlıysa head nullsa til da
        null yapılır
        {
            tail = null;
        }
        Console.WriteLine($"{value} silindi.");
        return;
    }
    while(current!=null&& current.data != value) //mevcut eleman boş
    değilse ve verisi aranan değere eşit değilse
    {
        previous = current; //önceki düğüm şimdiki olur
        current = current.Next; //sonraki düğüm şimdiki olur
    }
    previous.Next= current.Next;

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

    if(current.Next == null) //sonraki nullsa son elemandır
    {
        tail = previous;
    }
    Console.WriteLine($"{value} listeden silindi");
}
}

static void Main(string[] args)

```

```

{
    BagliList liste = new BagliList();
    int secim;

    do
    {
        Console.WriteLine("\n--- Bağlı Liste İşlemleri ---");
        Console.WriteLine("1. Başa eleman ekle");
        Console.WriteLine("2. Sona eleman ekle");
        Console.WriteLine("3. Belirli değerin sonrasına ekle");
        Console.WriteLine("4. Eleman ara");
        Console.WriteLine("5. Listeyi görüntüle");
        Console.WriteLine("6. Eleman silme");
        Console.WriteLine("0. Çıkış");
        Console.Write("Seçiminizi yapın: ");

        if (int.TryParse(Console.ReadLine(), out secim))
        {
            switch (secim)
            {
                case 1:
                    Console.Write("Başa eklenecek değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out int
basaDeger))
                    {
                        liste.BasaEkle(basaDeger);
                    }
                    else
                    {
                        Console.WriteLine("Geçersiz değer girdiniz!");
                    }
                    break;

                case 2:
                    Console.Write("Sona eklenecek değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out int
sonaDeger))
                    {
                        liste.sonaEkle(sonaDeger);
                    }
                    else
                    {
                        Console.WriteLine("Geçersiz değer girdiniz!");
                    }
                    break;

                case 3:
                    Console.Write("Aranan değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out int
arananDeger))
                    {
                        Console.Write("Eklenecek yeni değeri girin: ");
                        if (int.TryParse(Console.ReadLine(), out int
yeniDeger))
                        {
                            liste.BelirliDegerSonrasiEkle(arananDeger,
yeniDeger);
                        }
                        else
                        {
                            Console.WriteLine("Geçersiz değer
girdiniz!");
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        else
        {
            Console.WriteLine("Geçersiz değer girdiniz!");
        }
        break;

    case 4:
        Console.Write("Aranacak değeri girin: ");
        if (int.TryParse(Console.ReadLine(), out int
aranacakDeger))
        {
            liste.ara(aranacakDeger);
        }
        else
        {
            Console.WriteLine("Geçersiz değer girdiniz!");
        }
        break;

    case 5:
        liste.display();
        break;
    case 6:
        Console.Write("Değer giriniz: ");
        liste.silme(int.Parse(Console.ReadLine()));
        break;

    case 0:
        Console.WriteLine("Programdan çıkılıyor...");
        break;

    default:
        Console.WriteLine("Geçersiz seçim! Lütfen 0-5
arasında bir değer girin.");
        break;
    }
}
else
{
    Console.WriteLine("Geçersiz giriş! Lütfen sayısal bir değer
girin.");
    secim = -1; // Döngünün devam etmesi için
}
} while (secim != 0);
}
}
}

```