

ISIM: ABDOU VALERIO FOMA KENFACK

NUMARA: 24080410152

BÖLÜM: Bilgisayar Mühendiliği

## C# Programlama Dilinde İKİ YÖNLÜ BAĞLI LİSTE

örneği yapmanız istenmektedir.

Baş ekleme

Son ekleme

Araya herhangi bir veriden sonra ekleme

Araya herhangi bir veriden önce ekleme

Baştan silme

Sondan silme

Aradan arayarak silme

Arama

Listelem

Tümünü silme

Tüm linked listi bir diziye atma

## KODU

```
public class Node
{
    public int Data;
    public Node Next;
    public Node Previous;

    public Node(int data)
    {
        Data = data;
        Next = null;
        Previous = null;
    }
}

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

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

    public void BasaEkleme(int value)
    {
        Node newNode = new Node(value);
        newNode.Next = head;
        head = newNode;
    }
}
```

```

        Console.WriteLine($"{value} başa eklendi");
        if(tail == null)
        {
            tail = newNode;
        }
    }
    public void SonaEkleme(int value)
    {
        Node newNode = new Node(value);
        if (tail == null)
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail.Next = newNode;
            newNode.Previous = tail;
            tail = newNode;
        }
        Console.WriteLine($"{value} sona eklendi");
    }
    // Belirli bir değerin sonrasına eleman ekleme
    public void sonrasınaEkleme(int varolanDeger, int yeniDeger)
    {
        Node newNode = new Node(yeniDeger);
        if(head == null)
        {
            Console.WriteLine("Liste boştur");
            return;
        }
        Node current = head;
        // Not: Yeni değeri mevcut sayıdan önce koymak için
        // current.Data != varolanDeger yerine current.Next.Data != varolanDeger
        kullanın
        while (current != null && current.Data != varolanDeger)
        {
            current = current.Next;
        }
        if(current == null)
        {
            Console.WriteLine($"{varolanDeger} listede bulunmadı");
            return;
        }
        // Son eleman sonrasına ekleme
        if (current == tail)
        {
            SonaEkleme(yeniDeger);
            return;
        }
        Node current2 = current.Next;
        newNode.Next = current2;
        current.Next = newNode;
        newNode.Previous = current;
        current2.Previous = newNode;
        Console.WriteLine($"{yeniDeger}, {varolanDeger}'nin sonrasına eklendi");
    }
    // Baştan silme
    public void bastanSil()
    {

```

```

        if (head == null)
        {
            Console.WriteLine("Liste boş, silenecek eleman yok");
            return;
        }
        head = head.Next;
        if(head != null)
        {
            head.Previous = null;
        }
        else
        {
            tail = null;
        }
        Console.WriteLine("İlk eleman silindi");
    }
    // Sona sil
    public void sonaSil()
    {
        if (head == null)
        {
            Console.WriteLine("Liste boş, silenecek eleman yok");
            return;
        }
        tail = tail.Previous;
        if(tail != null)
        {
            tail.Next = null;
        }
        else
        {
            head = null;
        }
        Console.WriteLine("Son eleman silindi");
    }
    // Belirli bir elemanı silme
    public void BelirliElemanSil(int value)
    {
        if (head == null)
        {
            Console.WriteLine("Liste boş, silenecek eleman yok");
            return;
        }
        if(head.Data == value)
        {
            head = head.Next;
            if(head != null)
            {
                head.Previous = null;
            }
            else
            {
                tail = null;
            }
            Console.WriteLine($"{value} listeden silindi");
            return ;
        }
        if (tail.Data == value)
        {
            tail = tail.Previous;

```

```

        if(tail != null)
        {
            tail.Next = null;
        }
        else
        {
            head = null;
        }
        Console.WriteLine($"{value} listeden silindi");
        return;
    }
    Node current = head.Next;
    while(current != null && current.Data != value)
    {
        current = current.Next;
    }
    if (current == null)
    {
        Console.WriteLine($"{value} listede bulunmadı");
        return;
    }
    current.Previous.Next = current.Next;
    current.Next.Previous = current.Previous;
    Console.WriteLine($"{value} listeden silindi");
}
// Arama
public void Arama(int value)
{
    if (head == null)
    {
        Console.WriteLine("Liste boştur");
        return;
    }
    Node current = head;
    while(current != null)
    {
        if(current.Data == value)
        {
            Console.WriteLine($"{value} listede bulundu");
            return;
        }
        current = current.Next;
    }
    Console.WriteLine($"{value} listede bulunmadı");
}
// Tümünü silme
public void tumunuSilme()
{
    if (head == null)
    {
        Console.WriteLine("Liste zaten boş");
        return;
    }
    head = null;
    tail = null;
    Console.WriteLine("Tüm liste silindi");
}
// Tüm linked listi bir diziye atma
public void diziyeAtma()

```

```

{
    if (head == null)
    {
        Console.WriteLine("Liste boştur");
        return;
    }
    Node current = head;
    List<int> list = new List<int>();
    while (current != null)
    {
        list.Add(current.Data);
    }
    Console.WriteLine("Tüm linked listi bir diziye attı");
}
// Listeyi yazdırma
public void yazdirma()
{
    if (head == null)
    {
        Console.WriteLine("Liste boş, yazdıracak eleman yok");
        return;
    }
    Node current = head;
    Console.Write("Liste: ");
    while (current != null)
    {
        Console.Write(current.Data + " <--> ");
        current = current.Next;
    }
    Console.WriteLine();
}
}
static void Main()
{
    BagliList listim = new BagliList();
    listim.BasaEkleme(10);
    listim.BasaEkleme(5);
    listim.BasaEkleme(18);
    listim.yazdirma();
    listim.SonaEkleme(2);
    listim.yazdirma();
    listim.sonrasinaEkleme(5, 26);
    listim.yazdirma();
    listim.bastanSil();
    listim.sonaSil();
    listim.BelirliElemanSil(5);
    listim.yazdirma();
    listim.Arama(26);
    listim.tumunuSilme();
    listim.yazdirma();
}
}

```

# ÇIKTI

```
Microsoft Visual Studio Debug Console

18 basa eklendi
Liste: 18 <--> 5 <--> 10 <-->
2 sona eklendi
Liste: 18 <--> 5 <--> 10 <--> 2 <-->
26, 5'nin sonrasina eklendi
Liste: 18 <--> 5 <--> 26 <--> 10 <--> 2 <-->
Ilk eleman silindi
Son eleman silindi
5 listeden silindi
Liste: 26 <--> 10 <-->
26 listede bulundu
Tüm liste silindi
Liste bos, yazdiracak eleman yok

E:\DELL 5580\Studies\school_C#\Veri_yapilari_ödevi\abdou_valerio_foma_kenfack\Week_6\Iki_Yonlu_LinkedList\Iki_Yonlu_LinkedList\bin\Debug\net8.0\Iki_Yonlu_LinkedList.exe (process 15868) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```