



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**BM208 VERİ YAPILARI VE ALGORİTMALAR**  
**DERSİ**  
**UYGULAMA FÖYÜ - 2.GRUP**



**5. Deney Föyü: Çift Yönlü Bağlı Listeler**

**Adı Soyadı:**

**Öğrenci No:**

**19.03.2019**

**1. Amaç:**

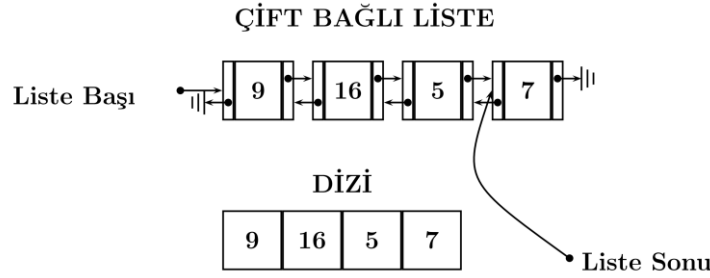
Çift yönlü bağlı listelerin tanımı, bağlı listeye ekleme, silme ve güncelleme işlemleri

**2. Uygulama Dersinden Önce Yapılması Gerekenler**

Çift yönlü bağlı listelerin önerilen kitaplardan incelenmesi gerekmektedir. Çift yönlü bağlı listede doğrusal listede olduğu gibi herhangi bir yerinden eleman eklenmesi ve çıkarılması nasıl yapılır? Ayrıca pointer ile düğüm hafızada nasıl tutulur? Bu konuların araştırılması gerekmektedir.

**3. Uygulamaya Hazırlık:**

Bağlı liste veri yapısı, dizi veri yapısına göre ekleme ve silme işlemlerinde sabit zaman harcama gibi önemli üstünlükler taşımasına rağmen önemli bir dezavantajı da içerisinde barındırmaktadır: bağlı listede bulunduğumuz yerden sadece ileri doğru gidebiliriz. Bu dezavantajı, her elemana bir önceki elemanın gösteren yeni bir işaretçi ekleyerek ortadan kaldırılabılır. Her elemanın bir değil iki işaretçisi olan bu bağlı liste yapısı, çift bağlı liste olarak adlandırılır. Çift bağlı listede her elemanın bir sonraki elemanı işaret eden ileri işaretçisinin yanında bir önceki elemanı işaret eden geri işaretçisi de vardır.



**Şekil 1- Çift yönlü bağlı liste örneklenmesi**

**4. Örnek Uygulama Kodu: Sıralı Eleman Ekleme**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <iostream>
4 // #include "Header2.h"
5 using namespace std;
6 struct dugum{
7     int veri;
8     dugum *birsonraki;
9     dugum *bironceki;
10 };
11 //Çift Yonlu Bagli Listeye Sıralı Eleman Ekleme
12 dugum * siralielemanekle(dugum * kontrol, int deger) {
13     //İki Yonlu Bagli liste bossa. Tek Yonlu Bagli listeden farklı olarak
14     kontrol->birsonraki = NULL; ifadesi NULL'u gösteriyordu, birde
15     yeni eklenen kontrol->bironceki=NULL'u gösterecek.
16     *
17     if (kontrol == NULL) {
18         kontrol = (dugum *)malloc(sizeof(dugum));
19         kontrol->birsonraki = NULL;
20         kontrol->bironceki = NULL;
21         kontrol->veri = deger;
22         return kontrol;
23     }
24     //İki Yonlu Bagli liste dolu ve eklenecek veri ilk elemandan kucukse
25     Bu durumda "kok" degisiyor. Burada farklı olarak ilerle diye bir isaretci
26     kullanildi, bu isaretci liste sonuna gitti ve yeni eklenen dugumun adresine
27     kuyruktaki yani en son dugumun adresine esitledi.*/
28     if (kontrol->veri > deger) {
29         dugum *gecici = (dugum *)malloc(sizeof(dugum));
30         gecici->veri = deger;
31         gecici->birsonraki = kontrol;
32         kontrol->bironceki = gecici;
33         return gecici;
34     }
35     //Eğer eklenecek eleman arada biryerde ise ilerle pointeri tanimlanir.
36     Bu pointer dugumlerde gezerek yeni eklenecek elemanın yerini bulacak.*/
37     dugum *ilerle = kontrol;
38     while (ilerle->birsonraki != NULL && ilerle->birsonraki->veri < deger) {
39         ilerle = ilerle->birsonraki;
40     }
41     ilerle->birsonraki = gecici;
42     gecici->bironceki = ilerle;
```

```
43     //eklenmek istenen deger dugum degerinden buyukse ve kok'den farklı iken.
44     Bu durumda listenin sonraki dugum arasına eleman ekleme. Burada ekleme
45     islemleri tek yonlu bagli liste ile aynı olmadıkça.*/
46     dugum * gecici = (dugum *)malloc(sizeof(dugum));
47     gecici->birsonraki = ilerle->birsonraki;
48     ilerle->birsonraki = gecici;
49     gecici->bironceki = ilerle;
50     //Bu kontrol iki yonlu bagli listenin sonuna eleman eklendiği için koyuldu.
51     En sona eleman eklenmede iki tane bag var. Bunlardan biri NULL yapıldı,
52     diğeri ise yine kendini gösterecek şekilde baglanir.*/
53     if (gecici->birsonraki != NULL)
54         gecici->birsonraki->bironceki = gecici;
55     gecici->veri = deger;
56     return kontrol;
57 }
58 //İki Yonlu Bagli Listeyi Yazdırma
59 Burada islemlerde Dairesel Listede olduğu gibidir.*/
60 void yazdir(dugum * kontrol) {
61     dugum * ilerle = kontrol;
62     int i = 0;
63     //While dongusune hic girmeyeceği için ilk dugum yazdirilir
64     ilerle bir sonraki dugume gecek ve program while dongusune girer.*/
65     cout << i << ". Elemanın Verisi:" << ilerle->veri << endl;
66     ilerle = ilerle->birsonraki;
67     //ilerle isaretçisi, kok isaretçisine esit olana kadar butun
68     listeyi dolasiir.*/
69     while (ilerle != NULL) {
70         cout << i << ". Elemanın Verisi:" << ilerle->veri << endl;
71         ilerle = ilerle->birsonraki;
72         i++;
73     }
74     getch();
75 }
76 int main() {
77     int secim;
78     dugum * kok;
79     cout << "Çift Yonlu Bagli Liste" << endl;
80     kok = NULL;
81     kok = siralielemanekle(kok, 100);
82     kok = siralielemanekle(kok, 10);
83     kok = siralielemanekle(kok, 1);
84     kok = siralielemanekle(kok, 150);
85     kok = siralielemanekle(kok, 50);
86     yazdir(kok);
87     getch();
88 }
```

- Uygulama derslerinde ilgili deney föyünü **mutlaka** yanınızda **bulundurunuz**. **Arkañ önlü çıktı alabilirsiniz**.
- Adınızı, Soyadınızı ve Öğrenci Numaranızı **mutlaka yazınız**.
- Uygulama sorusunun cevabı doğru ise kağıdın arka sayfasına **yazıp**, kodunuzu kısaca **açıklayınız**.

### 5. Uygulama Sorusu :

4. Bölümde verilen örnekteki struct yapısı ile oluşturulmuş tek yönlü dairesel listeyi ve eklenmiş olan elemanları kullanarak, **Eleman\_Sil()** fonksiyonu oluşturup aşağıdaki gibi bir ekran çıktısı veren programı yazınız.

**Örnek ekran çıktısı:**

```
Cift Yonlu Bagli Liste
1. Elemanin Verisi:1
2. Elemanin Verisi:10
3. Elemanin Verisi:50
4. Elemanin Verisi:100
5. Elemanin Verisi:150
sayi Bulunamadi
1. Elemanin Verisi:1
2. Elemanin Verisi:10
3. Elemanin Verisi:50
```

**CEVAP:**

Değerlendiren Öğretim Elemanı:		Çalışma Durumu:	0	1	2	3	4	5	Not:
-----------------------------------	--	--------------------	---	---	---	---	---	---	------