

# VERİ YAPILARI DERSİ DÖNEM PROJESİ

## PROJENİN AMACI

Öğrenci bilgilerinin olduğu çift bağlantılı (doubly linked) bir liste yapısı oluşturup öğrenci numarası yardımıyla bu listede gerekli fonksiyonlar sayesinde istenilen işlemleri listeye uygulamak.

## İŞLEM BASAMAKLARI

- ✓ **DUGUM** yapısı: Öğrenci bilgilerini depolamak için kullanılır. İçerisinde öğrenci numarası, ad, soyad, bölüm ve sınıf gibi veri alanları bulunur. **Void\*** işaretçiler aracılığıyla her bir öğrenci verisinin esnek bir şekilde tutulması sağlanmıştır.
- ✓ **head** ve **tail** : Bağlı listedeki başlangıç ve son düğümü işaret eden işaretçilerdir.
- ✓ **Fonksiyonlar**
  - **createLinkList()**: Bağlı liste oluşturma işlevini gerçekleştirir. Yeni bir bağlı liste oluşturur ve başlangıç ve son düğümleri NULL olarak ayarlar.
  - **addNode()**: Kullanıcıdan alınan öğrenci bilgilerini yeni bir düğüm olarak bağlı liste sonuna ekler.
  - **ogrenciBul()**: Girilen öğrenci numarasına göre öğrenciyi bulur ve bilgilerini ekrana yazdırır.
  - **listeyiGoster()**: Bağlı listedeki öğrencilerin bilgilerini ekrana yazdırır.
  - **ogrenciSil()**: Girilen öğrenci numarasına göre öğrenciyi listeden siler.
  - **listeyiSil()**: Tüm bağlı listeyi bellekten temizler.
- ✓ **Program İşleyişi**
  - Menü tabanlı bir kullanıcı arayüzü sunar. Kullanıcı seçim yapabilir:
    - **1** seçeneği, yeni bir öğrenci eklemeyi sağlar. Kullanıcıdan öğrenci bilgileri alınır ve bağlı liste sonuna eklenir.
    - **2** seçeneği, öğrenci aramayı sağlar. Girilen öğrenci numarasına göre öğrenciyi bulur ve bilgilerini ekrana yazar.
    - **3** seçeneği, listedeki öğrencilerin bilgilerini ekrana yazdırır.
    - **4** seçeneği, belirtilen öğrenciyi listeden siler.
    - **5** seçeneği, tüm bağlı listeyi siler.
    - **0** seçeneği, programdan çıkışı sağlar.

## Ekran Çıktıları:

```
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
1
Ogrenci numarasini giriniz:12
Ogrenci sinifini giriniz:1
Ogrenci adini giriniz:samet
Ogrenci soyadini giriniz:ucar
Ogrenci bolumunu giriniz:bm
Ogrenci eklendi
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
3
#####
Ogrenci NO:12
Adi:samet
Soyadi:ucar
Bolum:bm
Sinif:1
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
5
liste silindi
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
3
Liste bos!
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
|
```

listeye 1 öğrenci eklenir (ekleme fonksiyonu)

liste ekrana bastırılır (listeyi görüntüle fonksiyonu)

liste silinir (listeyi sil fonksiyonu)

liste ekrana bastırılır (listeyi görüntüle fonksiyonu)

sonuç olarak liste boşalacağından program doğru çalışıyor.

```
1
Ogrenci numarasini giriniz:12
Ogrenci sinifini giriniz:2
Ogrenci adini giriniz:samet
Ogrenci soyadini giriniz:ucar
Ogrenci bolumunu giriniz:bm
Ogrenci eklendi
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
1
Ogrenci numarasini giriniz:13
Ogrenci sinifini giriniz:1
Ogrenci adini giriniz:enes
Ogrenci soyadini giriniz:coban
Ogrenci bolumunu giriniz:ee
Ogrenci eklendi
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
2
Aranacak ogrenci numarasini giriniz:13
Ogrenci bulundu:
Adi:enes
Soyadi:coban
Bolum:ee
Sinif:1
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
4
Silinecek ogrencinin numarasini giriniz:13
Istediginiz ogrenci silindi
-----
1-Ogrenci Ekle
2-Ogrenci ara.
3-Listeyi Goruntule
4-Ogrenci Sil
5-Listeyi Sil
-----
3
#####
Ogrenci NO:12
Adi:samet
Soyadi:ucar
Bolum:bm
Sinif:2
-----
```

listeye 2 öğrenci eklenir (ekleme fonksiyonu)

ikinci öğrenci arama fonksiyonunda numarası girilerek bulunur ve ekrana yazılır.

ikinci öğrenci numarası sayesinde listeden silinir(silme fonksiyonu)

listede sadece 1 öğrenci kaldığını listeyi görüntüle fonksiyonu ile de görebildiğimizden program doğru çalışıyor.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct dugum
{
    void* ogrenci_No;
    void* Ad;
    void* Soyad;
    void* Bolum;
    void* Sinif;
    struct dugum* onceki;
    struct dugum* sonraki;
}DUGUM;
```

DUGUM \*head = NULL; // head, bagli listenin basini (ilk dugumu) isaret eder

DUGUM \*tail = NULL; //tail bagli listenin sonunu (son dugumu) isaret eder.Baslangicta NULL olarak ayarlanir cunku herhangi bir dugum yoktur.

```
DUGUM* createLinkList()
{
    DUGUM* yeniListe=(DUGUM*)malloc(sizeof(DUGUM));
    yeniListe->onceki=NULL;
    yeniListe->sonraki=NULL;
    return yeniListe;
}

void addNode()
{
```

```
DUGUM* yeniListe=createLinkList();  
void* ogrenci_No=malloc(sizeof(int));  
void* Ad=malloc(sizeof(50));  
void* Soyad=malloc(sizeof(50));  
void* Bolum=malloc(sizeof(50));  
void* Sinif=malloc(sizeof(int));
```

```
printf("Ogrenci numarasini giriniz:");  
scanf("%d",(int*)&ogrenci_No);  
printf("Ogrenci sinifini giriniz:");  
scanf("%d",(int*)&Sinif);  
printf("Ogrenci adini giriniz:");  
scanf("%s",&Ad);  
printf("Ogrenci soyadini giriniz:");  
scanf("%s",&Soyad);  
printf("Ogrenci bolumunu giriniz:");  
scanf("%s",&Bolum);
```

```
yeniListe->Ad=Ad;  
yeniListe->Bolum=Bolum;  
yeniListe->Soyad=Soyad;  
yeniListe->ogrenci_No=ogrenci_No;  
yeniListe->Sinif=Sinif;
```

if (head == NULL) // Eger bagli liste bossa (head == NULL), head ve tail yeni dugumu isaret eder. İlk dugum eklenir.

```
{
```

```

    head = yeniListe;
    tail = yeniListe;
}
else// Eger bagli liste bos deđilse, yeni dugum sona eklenir.
{
    DUGUM* gecici=tail;

    gecici->sonraki=yeniListe;//son dugumun sonraki isaretcisi yeni dugumu
isaret eder

    yeniListe->onceki=gecici;//ayni mantikla dugumler karsilikli baglanır.
    tail=yeniListe;//tail artık son dugumu gostermis olur.
}
printf("Ogrenci eklendi\n");
}

void ogrenciBul(int aranaNo)
{
    DUGUM* gecici=head;
    if(gecici==NULL)
        printf("Ogrenci bulunamadi\n");
    while(gecici!=NULL)
    {
        if(*(int*)(gecici->ogrenci_No)==aranaNo)
        {
            printf("Ogrenci
bulundu:\nAdi:%s\nSoyadi:%s\nBolum:%s\nSinif:%d\n",(char*)gecici-
>Ad,(char*)gecici->Soyad,(char*)gecici->Bolum,*((int*)gecici->Sinif));

        }
    }
}

```

```

        gecici=gecici->sonraki;
    }

}

void listeyiGoster()
{
    if(head==NULL)
        printf("Liste bos!\n");
    else
    {
        DUGUM* gecici=head;
        while(gecici!=NULL)
        {
            printf("#####\nOgrenci
NO:%d\nAdi:%s\nSoyadi:%s\nBolum:%s\nSinif:%d\n",*((int*)gecici-
>ogrenci_No),(char*)gecici->Ad,(char*)gecici->Soyad,(char*)gecici-
>Bolum,*((int*)gecici->Sinif));

            gecici=gecici->sonraki;
        }
    }
}

void ogrenciSil(int sil)
{
    int sayac=0;

    DUGUM* gecici=head;

    if(gecici==NULL)//liste bossa

        printf("Liste bos!\n");

```

```
while(gecici!=NULL)
{
    if(*(int*)gecici->ogrenci_No == sil && gecici->onceki == NULL)//liste
doluysa ve silinicek numara listenin ilk dugumu ise
    {
        head=head->sonraki;//gecici listenin devamini tutsunki kaybetmeyelim
        free(gecici);//ilk dugum silinir
        sayac++;
        printf("Istediginiz ogrenci silindi\n");
    }
    else if(*(int*)gecici->ogrenci_No == sil) //liste dolu ve silinecek numara
listenin ilk dugumu degilse(ortada veya sonda ise)
    {
        gecici->onceki->sonraki=gecici->sonraki;
        free(gecici);
        sayac++;
        printf("Istediginiz ogrenci silindi\n");
    }
    gecici=gecici->sonraki;

}

if(sayac==0)//liste dolu ama silinecek ogrenci listede yok ise
    printf("Silme istediginiz ogrenci listede bulunamadi\n");
}

void listeyiSil()
{
    DUGUM* gecici=head;
```



```

DUGUM* temp;
while(gecici!=NULL)
{
    temp=gecici;
    gecici=gecici->sonraki;
    free(temp->Ad);
    free(temp->Bolum);
    free(temp->ogrenci_No);
    free(temp->Sinif);
    free(temp->Soyad);
    free(temp);
}
head=NULL;
tail=NULL;
}
int main()
{
    int sec,ara,sil;
    //DUGUM* liste=createLinkList();

    do{

        printf("-----\n1-Ogrenci Ekle\n2-Ogrenci ara.\n3-Listeyi
Goruntule\n4-Ogrenci Sil\n5-Listeyi Sil\n-----\n");
        scanf("%d",&sec);
        switch(sec)
        {

```

**case 1:**

**addNode();**

**break;**

**case 2:**

**printf("Aranacak ogrenci numarasini giriniz:");**

**scanf("%d",&ara);**

**ogrenciBul(ara);**

**break;**

**case 3:**

**listeyiGoster();**

**break;**

**case 4:**

**printf("Silinecek ogrencinin numarasini giriniz:");**

**scanf("%d",&sil);**

**ogrenciSil(sil);**

**break;**

**case 5:**

**listeyiSil();**

**printf("liste silindi\n");**

**break;**

**default:**

**sec=0;**

**printf("yanlis islem yaptiniz!");**

**break;**

**}**

**}while(sec!=0);**

```
return 0;
```

```
}
```