

BLM 112- Programlama Dilleri

II

Hafta 6: Bağlı Listeler (Kısım-3)

Assist.Prof.Dr. Ümit ATİLA

Uygulama-1

- Dışardan aldığı isimleri alfabetik sıra ile listeleyen, listeye yeni düğüm ekleyen, belirtilen düğümü silen ve düğümdeki en uzun isimi bulan tek bağlı liste uygulaması.

Düğüm Yapısı

```
5 struct personel
6 {
7     char    adi[21];
8     struct personel *sonraki;
9 };
10
11 //artık struct personel yerine dugum kullanılacak
12 typedef struct personel dugum;
13 // *head dugum tipinde bir işaretçidir.
14 //Herzaman listenin başını gösterecek
15 dugum *head, *oncekiDugum,*yeni,*silinecek;
```

Kayit Arama

```
18 void ara(char *aranan) //listede arama yapar
19 {
20     dugum *p;
21     p = head;
22     oncekiDugum = head;
23     while(p->sonraki!=NULL)
24     {
25         p= p->sonraki;
26         if (strcmp(p->adi, aranan) >= 0) break;
27         oncekiDugum = p;
28     }
29 }
```

Yeni Kayit Ekleme

```
30 void ekle(char *s)
31 {
32     yeni = (dugum *) malloc(sizeof(dugum));
33     strcpy(yeni->adi, s);
34     ara(yeni->adi);
35     yeni->sonraki = oncekiDugum->sonraki; /* listeye ekle */
36     oncekiDugum->sonraki = yeni;
37 }
```

Kayit Silme

```
38 void sil(char *s)
39 {
40     ara(s);
41     silinecek = oncekiDugum->sonraki;
42     oncekiDugum->sonraki = silinecek->sonraki;
43     free(silinecek);
44 }
```

Listeleme

```
45 void listele(void)
46 {
47     dugum *p;
48     p = head;
49     p = p->sonraki;
50     while (p!= NULL)
51     {
52         printf("%s \n",p->adi);
53         p = p->sonraki;
54     }
55 }
```

En Uzun Kaydi Bulma

```
56 void enUzunBul(void)
57 {
58     dugum *p, *enuzun;
59     p = head;
60     p= p->sonraki;
61     enuzun=p;
62     while (p != NULL)
63     {
64         if (strlen(p->adi) >= strlen(enuzun->adi))
65         {
66             enuzun = p;
67         }
68         p = p->sonraki;
69     }
70     printf("\nEn uzun :%s  Uzunluk:%d",enuzun->adi,strlen(enuzun->adi));
71     getchar();
72 }
```


Main Fonksiyonu

```
74 void main()
75 {
76     char    sec;
77     char    s[21];
78     head = (dugum *) malloc(sizeof(dugum));
79     strcpy(head->adi," listenin basi");
80     head->sonraki = NULL;
81     do
82     {
83         system("cls");
84         listele();
85         printf("\n\n1 - Ekle\n2 - Sil\n3 - En Uzun isim\n4 - Cikis\n\nSec :");
86         sec = getche();
87         switch (sec)
88         {
89             case '1':printf("\nAdi :"); gets(s);
90                 ekle(s);break;
91             case '2':printf("\nAdi ");  gets(s);
92                 sil(s);break;
93             case '3':enUzunBul();break;
94             case '4':exit(0);break;
95         }
96     }
97     while (1);
98 }
```

Uygulama-2

- Öğrencilere ait numara, ad, vize ve final bilgilerini tutan bir tek bağlı doğrusal liste uygulaması için:
 - Kayıt ekleme silme
 - Öğrencileri başarı notlarını hesaplayarak (Vize %40, Final %60) listeleme
 - Sınıfın en yüksek başarı notuna sahip öğrenci bilgilerini listeleme
 - Sınıfın başarı notu ortalamasını hesaplama

Düğüm Yapısı

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct ogrenci{
5     int no;
6     char adi[40];
7     int vize,final;
8     struct ogrenci *sonraki;
9 };
10 typedef struct ogrenci dugum;
11 dugum *head,*yeniDugum,*oncekiDugum,*enBasarili;
12
```

Liste Oluşturma

```
13 void ogrenciListeOlustur()
14 {
15     int n,k;
16     printf("Kac elamanli liste olusturacaksin");
17     scanf("%d",&n);
18     for(k=0;k<n;k++)
19     {
20         if(k==0) //ilk düğüm ekleniyor
21         {
22             yeniDugum = (dugum *)malloc(sizeof(dugum));
23             head = yeniDugum;
24         }
25         else
26         {
27             yeniDugum->sonraki = (dugum *) malloc(sizeof(dugum));
28             yeniDugum = yeniDugum->sonraki;
29         }
30         printf("Ogrenci No:");scanf("%d",&yeniDugum->no);
31         printf("Ogrenci Adi:");scanf("%s",&yeniDugum->adi);
32         printf("Ogrenci Vize:");scanf("%d",&yeniDugum->vize);
33         printf("Ogrenci Final:");scanf("%d",&yeniDugum->final);
34     }
35     yeniDugum->sonraki = NULL;
36 }
```

Öğrenci Kaydı Listeleme

```
38 void ogrenciListele()  
39 {  
40     double donemNotu =0;  
41     dugum *p;  
42     p = head;  
43     while(p!=NULL)  
44     {  
45         printf("\n ***** \n");  
46         printf("Ogrenci No:%d\t",p->no);  
47         printf("Ogrenci Adi:%s\t",p->adi);  
48         printf("Ogrenci Vize:%d\t",p->vize);  
49         printf("Ogrenci Final:%d\t",p->final);  
50         donemNotu = p->vize*0.4+p->final*0.6;  
51         printf("Donem Notu : %.2f\t", donemNotu);  
52         printf("\n ***** \n");  
53         p = p->sonraki;  
54     }  
55 }
```

Öğrenci Kaydı Ekleme

```
57 void ogrenciEkle()  
58 {  
59     int kayıtNo;  
60     dugum *p, *q;  
61     yeniDugum = (dugum *) malloc(sizeof(dugum));  
62     printf("Oğrenci No:");scanf("%d",&yeniDugum->no);  
63     printf("Oğrenci Adi:");scanf("%s",yeniDugum->adi);  
64     printf("Oğrenci Vize:");scanf("%d",&yeniDugum->vize);  
65     printf("Oğrenci Final:");scanf("%d",&yeniDugum->final);  
66  
67     printf("Hangi kayıttan oncesine eklemek istiyorsunuz");  
68     printf("\nListe sonuna eklemek icin 0 gir");  
69     scanf("%d",&kayıtNo);  
70  
71     p = head;  
72     if(p->no == kayıtNo) //başa ekle  
73     {  
74         yeniDugum->sonraki = p;  
75         head = yeniDugum;  
76     }
```

Öğrenci Kaydı Ekleme

```
80     else
81     {
82         while(p->sonraki != NULL)
83         {
84             q=p;
85             p=p->sonraki;
86             if(p->no == kayitNo) break;
87         }
88         if(p->no == kayitNo) //araya ekleme
89         {
90             q->sonraki = yeni;
91             yeni->sonraki = p;
92         }
93         else if(p->sonraki == NULL)
94         {
95             p->sonraki = yeni;
96             yeni->sonraki = NULL;
97         }
98     }
99 }
```

Öğrenci Kaydı Silme

```
97 void ogrenciSil()  
98 {  
99     int kayitNo;  
100     dugum *p, *q;  
101  
102     printf("Silmek istediginiz ogrenci no gir");  
103     scanf("%d",&kayitNo);  
104  
105     p = head;  
106     if(p->no == kayitNo) //baştakini sil  
107     {  
108         head = p->sonraki;  
109         free(p);  
110     }
```


Öğrenci Kaydı Silme

```
115     else
116     {
117         while(p->sonraki !=NULL)
118         {
119             if(p->no == kayitNo) break;
120             else {
121                 q=p;
122                 p=p->sonraki;
123             }
124         }
125         if(p->no == kayitNo) //araya ekleme
126         {
127             q->sonraki = p->sonraki;
128             free(p);
129         }
130         else if(p->sonraki == NULL)
131         {
132             printf("Silinecek ogrenci no yok\n");
133         }
134     }
135 }
```

En Başarı Öğrenci Bulma

```
130 double basariNotuHesapla(int vize,int final)
131 {
132     return (0.4*vize)+(0.6*final);
133 }
134
135 void enBasariliOgrenci()
136 {
137     double basariNotu = 0;
138
139     dugum *p;
140     p = head;
141     enBasarili = head;
142
143     while(p->sonraki!=NULL)
144     {
145         p=p->sonraki;
146         if(basariNotuHesapla(p->vize,p->final)>basariNotuHesapla(enBasarili->vize,enBasarili->final))
147             enBasarili = p;
148     }
149     printf("En basarili ogrenci:\n");
150     printf("No:%d - Ad: %s Basari Notu:%.2f\n",enBasarili->no,
151         enBasarili->adi, basariNotuHesapla(enBasarili->vize,enBasarili->final));
152 }
```

Sınıf Ortalaması Hesaplama

```
154 void sinifBasariOrtalamasi()  
155 {  
156     double basariNotuOrtalamasi = 0;  
157     double toplam =0;  
158     int sayac = 0;  
159     dugum *p;  
160     p = head;  
161     if(p==NULL)  
162     {  
163         printf("Listede kayit yok!");  
164     }  
165     else  
166     {  
167         do  
168         {  
169             toplam += basariNotuHesapla(p->vize,p->final);  
170             p=p->sonraki;  
171             sayac++;  
172         }  
173         while(p!=NULL);  
174  
175         basariNotuOrtalamasi = toplam/sayac;  
176         printf("Basari Notu Ortalamasi: %.2f \n",basariNotuOrtalamasi);  
177     }  
178 }
```

Main Fonksiyonu

```
180 int main(void)
181 {
182     int secim=0;
183     printf("1-Liste Olustur \n2-Yeni Kayit Ekle \n3-Kayit Sil \n4-En Yuksek Basari Notu\n5-Sinif Ortalamasi\n");
184     while(1)
185     {
186         printf("Secim yap [1-5]?");
187         scanf("%d",&secim);
188         switch(secim)
189         {
190             case 1: ogrenciListeOlustur();
191                     ogrenciListele();break;
192             case 2: ogrenciEkle();ogrenciListele();break;
193             case 3: ogrenciSil();ogrenciListele();break;
194             case 4: enBasariliOgrenci();break;
195             case 5: sinifBasariOrtalamasi();break;
196         }
197     }
198 }
```