



VERİ YAPILARI ÖDEVİ

Ödev 1



Ceyda ÖZGENÇ

192802281

Örgün Öğretim 2. Sınıf

23.04.2021

Veri Yapıları

Yusuf ÖZÇEVİK

Öğretmenimiz tarafından verilen aşağıdaki yapıyı kullanarak C++ programlama dili ile geliştirdiğim programda kullanıcıdan öğrenci sayısı ve test sayısı sordurma, daha sonra gerekli hafıza bloklarını tahsis etmekteyim. Bu verilerin saklanması için dinamik dizi yapısından faydalanmaktayım.

```
struct Ogrenci
{
    string isim,harfnotu;
    int ogrencino,*testnotlari;
    double ortalama;
};
```

Öğretmenimiz tarafından belirlenen programda olması gereken fonksiyonlara uygun bir şekilde hazırladığım programda:

1. int main():

Bu bölümde öncelikle sistemden dinamik bellek alanı istedim. Daha sonra kullanıcı tarafından öğrenci sayısı ve test sayısını isteyerek veriAl fonksiyonunu çağırdım ve veriAl fonkiyonu sonlandığında öğrenci tarafından programdan çıkış ve kullanıcı bilgilerini görme seçeneklerini sundum. Bu seçeneklerden bilgi gösterme seçildiğinde göster fonksiyonuna, çıkış seçildiğinde silme fonksiyonuna yönlendirdim. Çünkü çıkma işlemi gerçekleşeceğinde dinamik belleklerin iade edilmesinin uygun olacağını düşündüm.

```
int main(){
    int *o_no,*notlar,o_sayi,t_sayi,secim;
    double *ort;
    string *ad,*harfnotu;
    harfnotu=new string[20];
    ad=new string[20];
    o_no=new int[20];
    notlar=new int[200];
    ort=new double[20];
    cout<<"Ogrenci sayisini giriniz:";
    cin>>o_sayi;
    cout<<"Test sayisini giriniz:";
    cin>>t_sayi;
    cout<<endl;
    veriAl(o_sayi,t_sayi,&ad[0],&o_no[0],&notlar[0],&harfnotu[0],&ort[0]);
    cout<<endl;
    cout<<"Ogrenci bilgilerine bakmak icin:1"<<endl<<"Cikmak icin:2"<<endl<<"Seciminiz:";
    cin>>secim;
    switch (secim){
        case 1:
            goster(o_sayi,t_sayi,&ad[0],&o_no[0],&notlar[0],&harfnotu[0],&ort[0]);
        case 2:
            sil(o_sayi,t_sayi,&ad[0],&o_no[0],&notlar[0],&harfnotu[0],&ort[0]);
    }
    return 0;
}
```

2. void veriAl():

Main tarafından çağırılan bu bölümde mainden dinamik belleklerin bellek adresleri, öğrenci sayısı ve test sayısı aktarılmıştır. Dinamik dizilerin değişkeni değil adreslerini

aktarmayı tercih etmemin nedeni kalıcı olarak değişkenlerin içine verileri yazmaktır.

Öğrenci sayısı kadar bir döngü oluşturularak kullanıcıdan bilgileri dinamik belleğe aktardım ve bu döngü içinde test sayısı kadar bir döngü açarak her öğrenci için belirlenen test sayısı kadar kullanıcıdan test notları aldım. Öğrenci sayısı için oluşturduğum döngü içinde notHesapla fonksiyonunu oluşturdum çünkü her öğrenci için bir not hesaplama döndürmesi gerekiyordu. Ayrıca bu bölümde notlar için bir ofset hesaplattım çünkü not dizisine diğerleri gibi bir değil birden çok verilmesi gerekiyordu.

```
void veriAl(int o_sayi,int t_sayi,string *ad,int *o_no,int *notlar,string *harfnotu,double *ort){
    int offset=0;
    for(int i=0;i<o_sayi;i++){
        cout <<i+1<< " ". ogrencinin adi:";
        cin>>*(ad+i);
        cout<<i+1<< " ". ogrencinin numarasi:";
        cin>>*(o_no+i);
        for(int x=0;x<t_sayi;x++){
            cout <<i+1<< " ". ogrencinin "<<x+1<< " ". test notu:";
            cin>>*(notlar+x+offset);
        }
        notHesapla(&notlar[offset],&harfnotu[i],&ort[i],t_sayi);
        offset+=3;
    }
}
```

3. doubleHesapla():

veriAl fonksiyonundan çağırılan bu bölümde dinamik belleklerin adresini ve test sayısını aldım daha sonra not ortalamasını hesapladım. Sonrasında harf notunu belirleyerek gerekli diziye aktardım.

```
double notHesapla(int *testnot,string *harfnotu,double *ort,int t_sayi){
    double toplam=0;
    for(int y=0;y<t_sayi;y++){
        toplam+=*(testnot+y);
    }
    *(ort)=toplam/t_sayi;
    if(*(ort)>=0 and *(ort)<=49){*(harfnotu)="F";}
    else if(*(ort)<=64){*(harfnotu)="D";}
    else if(*(ort)<=74){*(harfnotu)="C";}
    else if(*(ort)<=84){*(harfnotu)="B";}
    else if(*(ort)<=100){*(harfnotu)="A";}
    cout<<"Ortalama is: "<<*(ort)<<" Harf notu: "<<*(harfnotu)<<endl<<endl;
}
```

4. void göster():

Main tarafından çağırılan bu fonksiyonda verileri struct'a aktararak ekranda gösterme işlemini gerçekleştirdim ve daha sonra yeniden kullanıcıya tarafından programdan çıkış ve kullanıcı bilgilerini görme seçeneklerini sundum.

```

void goster(int o_sayi,int t_sayi,string *ad,int *o_no,int *notlar,string *harfnotu,double *ort)
{
    int offset=0,secim;
    cout<<endl;
    for(int i=0;i<o_sayi;i++){
        Ogrenci *ogrenci= new Ogrenci;
        ogrenci->isim=(ad+i);
        ogrenci->ogrencino=(o_no+i);
        ogrenci->testnotlari=(notlar+offset); // "ogrenci" nesnesinin "testnotlari" degerine "notlar"
        ogrenci->harfnotu=(harfnotu+i); // "ogrenci" nesnesinin "harfnotu" degerine "harfnotu"
        ogrenci->ortalama=(ort+i); // "ogrenci" nesnesinin "ortalama" degerine "ort" dinamik bellek
        cout<<i+1<<" ". ogrencinin adi: "<<ogrenci->isim<<endl<<i+1<<" ". ogrencinin numarası: "<<ogrenci->ogrencino<<endl<<i+1<<" ". ogrencinin test notları: "<<ogrenci->testnotlari<<endl<<i+1<<" ". ogrencinin harf notu: "<<ogrenci->harfnotu<<endl<<i+1<<" ". ogrencinin ortalama: "<<ogrenci->ortalama<<endl<<i+1<<" ". test notu: "<<*(ogrenci->testnotlari+x) <<endl;
        for(int x=0;x<t_sayi;x++){ // Kullanıcıdan alınan test sayısı kadar döngü oluşturmak.
            cout <<i+1<<" ". ogrencinin "<<x+1<<" ". test notu: "<<*(ogrenci->testnotlari+x) <<endl;
        }
        cout<<"Ortalama is: "<<ogrenci->ortalama<<" Harf notu: "<<ogrenci->harfnotu<<endl<<endl;
        offset+=3;
    }
    cout<<"Ogrenci bilgilerine bakmak icin:1"<<endl<<"Cikmak icin:2"<<endl<<"Seciminiz:";
    cin>>secim;
    switch (secim){ //Seçilen işleme göre yönlendirme yapmak.
        case 1:
            goster(o_sayi,t_sayi,&ad[0],&o_no[0],&notlar[0],&harfnotu[0],&ort[0]); // "goster" fonks.
        case 2:
            sil(o_sayi,t_sayi,&ad[0],&o_no[0],&notlar[0],&harfnotu[0],&ort[0]); // "sil" fonksiyonu
    }
}

```

5. void sil():

main tarafından çağırdığım bu fonksiyonda dinamik dizilerin başlangıç adreslerinin fonksiyona yönlendirdim ve bu fonksiyonda dizilerin içeriğini döngü içinde sırayla sisteme geri iade ettim.

```

void sil(int o_sayi,int t_sayi,string *ad,int *o_no,int *notlar,string *harfnotu,double *ort){
    int offset=0;
    for(int i=0;i<o_sayi;i++){
        delete (ad+i), (o_no+i), (harfnotu+i), (ort+i);
        for(int x=0;x<t_sayi;x++){
            delete (notlar+x+offset);
        }
        offset+=3;
    }
    exit(0);
}

```