

2019 Ödev

2. Bir ba lı listede veri yapıları dersini alan ö renci bilgileri tutulacaktır. Ö rencilerin numarası(int), adı(String), vize(int) ve genel(int) sınav bilgilerini tutan bir veri yapısı tanımlayarak a a ıdaki i lemleri yapan kodları yazınız?

Ekleme

- Ba tan ekleme
- Sona ekleme
- Araya ekleme

Silme

- Ba tan silme
- Sondan silme
- stenen elemanı silme

Arama

- Vizesi en yüksek olan ö rencinin bilgilerini gösteren metodu yazınız?
- Bütün ö renci bilgilerini gösteren bir metot yazınız?
- Ba lı liste üzerinde dola arak ö renci ortalamalarını bulan ve ortalamaya göre geçenleri ve kalanları ayrı ba lı listelere yazan bir metot yazınız?

Veri yapıları dersinin bilgilerini içeren VER YAPISI:

```
Public class Eleman{
    Int onumara;
    String adi;
    Int vize;
    Int genelsinav;
    Eleman once;
    Eleman ileri;
    Eleman geri;
    Public Eleman(int onumara,int vize,int genelsinav,String adi){
        This.onumara=onumara;
        This.vize=vize;
        This.genelsinav=genelsinav;
        This.adi=adi;
        Ileri=null;
        Geri=null;
    }
}
```

Veri yapısını bir insanın özellikleri gibi düşünebiliriz. Bliste ise başı ve sonu olan bir yapı. Bu Şekilde ikisinin farkını daha kolay anlayabiliriz.

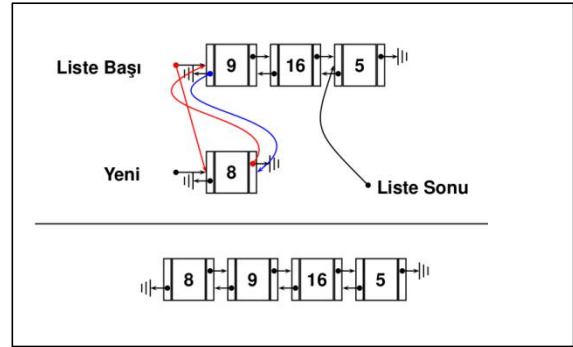
Veri yapıları dersinin bilgilerini içeren Ba lı Liste Tanımı

```
Public class BListe{
    Eleman bas;
    Eleman son;
    Public BListe(){
        Bas=null;
        Son=null;
    }
}
```

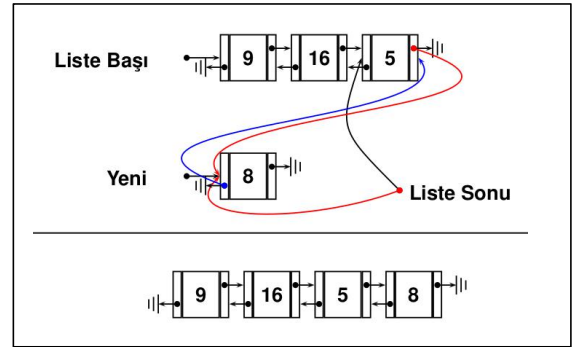
```
}  
}
```

Ekleme

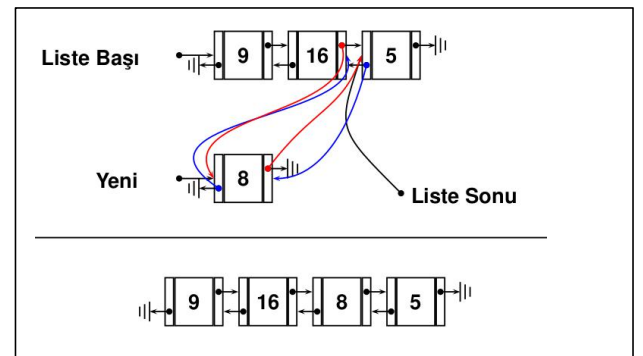
```
Void listeBasinaEkle(Eleman yeni){  
If(bas==null){  
Bas=yeni;  
}  
Else {  
Bas.geri=yeni;  
Yeni.ileri=bas;  
Bas=yeni; }  
}
```



```
Void listeSonunaEkle(Eleman yeni){  
If(son==null){  
Son=yeni; }  
Else {  
Son.ileri=yeni;  
Yeni.geri=son;  
Son=yeni; }  
}
```



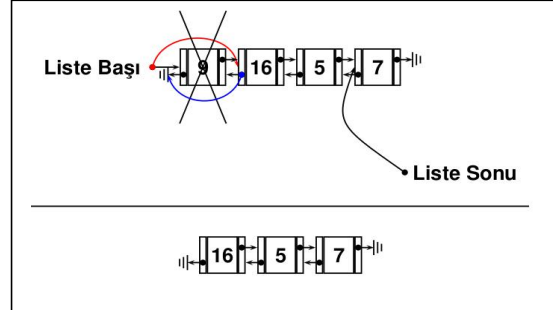
```
Void arayaEkle(Eleman yeni, Eleman once){  
Yeni.ileri=once.ileri;  
Yeni.geri=once;  
Once.ileri.geri=yeni;  
Once.ileri=yeni;
```



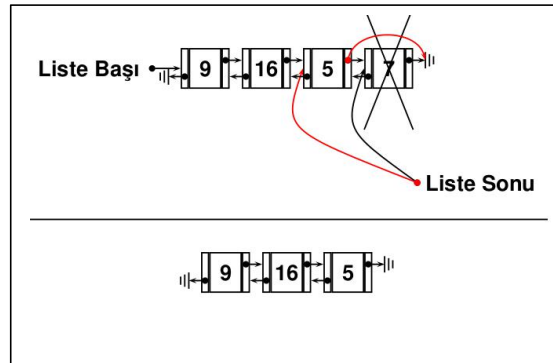
Listenin ortasına eleman eklemek için yeni elemandan önce gelen elemanın adresi bilinmelidir.

Silme

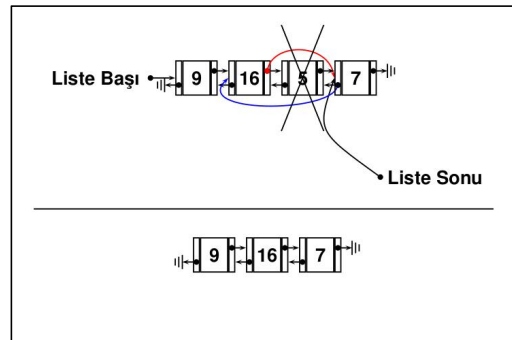
```
Void bastanSil(){  
    bas=bas.ileri;  
    If(bas==null){  
        Son=null; }  
    Else  
        Bas.geri=null;  
}
```



```
Void sondanSil(){  
    Son=son.geri;  
    If(son==null){  
        Bas=null; }  
    Else  
        Son.ileri=null;  
}
```



```
Void aradanSil(Eleman s){  
    s.ileri.geri=s.geri;  
    S.geri.ileri=s.ileri; }
```



Arama

Arama yaparken önce listenin ilk elemanından başlanır ve bir elemandan diğerine iaretçiler yardımıyla ilerleyerek her elemanın içeriğiyle aradığımız değer karşılaştırılır. Değerler aynı ise aranan eleman bulunmuştur demektir yoksa null değerini döndürebiliriz.

```
Eleman listeAra (int deger){  
    Eleman tmp;  
    Tmp=bas;
```

```
While(tmp!=null){  
If(tmp.icerik==deger)  
Return tmp;  
Tmp=tmp.ileri; }  
Return null;
```

-Vizesi en yüksek olan öğrencinin bilgilerini gösteren
metodu yazınız?

```
Public class Eleman{  
Int onumara;  
String adi;  
Int vize;  
Int genelsinav;  
Eleman once;  
Eleman ileri;  
Eleman geri;  
Public Eleman(int onumara,int vize,int genelsinav,String adi){  
This.onumara=onumara;  
This.vize=vize;  
This.genelsinav=genelsinav;  
This.adi=adi;  
Ileri=null;  
Geri=null;  
}  
Public class BListe{  
Eleman bas;  
Eleman son;  
Public BListe(){  
Bas=null;  
Son=null;  
}  
}  
Void listeBasinaEkle(Eleman yeni){  
If(bas==null){  
Bas=yeni;  
}  
Else {  
Bas.geri=yeni;  
Yeni.ileri=bas;  
Bas=yeni; }  
  
Eleman listeAra (){  
  
Eleman enyuksek;  
Eleman tmp;  
tmp=bas;  
Enyuksek=bas;
```

```

While(tmp!=null){
If(tmp.vize>enyuksek.vize)
Enyuksek=tmp;
Tmp=tmp.ileri;
Else{
tmp=tmp.ileri; }
}
Return enyuksek;
}

```

```

Void tumgoster(){
Eleman gez=bas;
While(gez!=null){
System.out.println(gez.onumara+gez.adi+gez.vize+gez.genelsinav);
Gez=gez.ileri;
}
}

```

```

Void gecenkalanbul(){
Eleman tmp=bas;
While(tmp!=null){
Double ortalama=((tmp.vize*0.4)+(tmp.genelsinav*0.6));
If(ortalama>50){
Gecenlist.ekle(tmp);
}
Else {
Kalanlist.ekle(tmp);
}
Tmp=tmp.ileri;
}
}

```

```

Public static void main(String[] args) {
Eleman e=new Eleman();
Bliste gecenlist=new bliste();
Bliste kalanlist=new bliste();
Bliste ilkliste=new bliste();
e.onumara=265;
e.adi="Neslihan";
e.vize=80;
e.genelsinav=80;
Ilkliste.listeBasinaEkle(e);
Eleman n=new Eleman();
n.onumara=245;
n.adi="Mehmet";
n.vize=50;
n.genelsinav=30;
Ilkliste.listeBasinaEkle(n);
Eleman s=new Eleman();
}
}

```

```
s.onumara=140;  
s.adi="Na me";  
s.vize=100;  
s.genelsinav=100;  
Ilkliste.listeBasinaEkle(s);  
Ilkliste.gecenkalanbul();
```

KISA BİR ÖZET !

Yeni düğüm oluşturulması ve bilgi ilavesi yapılandırıcılar kullanılarak şöyle yapılabilir;

```
newNode x = new node();  
newNode.element = new String("Bolu");
```

Liste başına bağltda şu kodlar sırasıyla kullanılmalıdır.

```
newNode.next = head;  
head = newNode;
```

Listeden eleman çıkarılması ise tersi mantıkla şöyle kodlanabilir;

```
node oldNode;  
oldNode=head;  
head=head.next;
```

Yeni düğüm

şöyle oluşturulabilir.

```
node newNode = new node();  
newNode.element(new String("Tokat"));
```