Kuyruk Uygulamaları

(Kuyruğu ters çevirme, kuyruğun ilk n elemanını ters çevirme, öncelikli kuyruk)

Kuyruk yapısı bağlı liste kullanılarak oluşturulmuştur. Gerekli tanımlamalar yapılmış, ekleme ve çıkarma metodları tanımlanmıştır.

```
class Eleman{
    Eleman ileri;
    int icerik;
    Eleman(int icerik){
        this.icerik=icerik;
        ileri=null;
    }
}

class Kuyruk {
    Eleman bas, son=null;

boolean KuyrukBos(){
    if(bas==null)
        return true;
    else
        return false;
}

public void KuyrugaEkle(int n){
    Eleman e=new Eleman(n);
    if(!KuyrukBos()){
        son.ileri=e;
        son=e;
    }
    else{
        bas=e;
        son=e;
    }
}
public void KuyrugaEkle(int n){
    Eleman e=new Eleman(n);
    if(!KuyrukBos()){
        son.ileri=e;
        son=e;
    }
    else{
        bas=e;
        son=e;
    }
}

public int KuyruktanCikar(){
    Eleman cikacak=bas;
    if(!KuyrukBos()){
        bas=bas.ileri;
        if(bas==null)
            son=null;
    }
    return cikacak.icerik;
}
```

Kuyruğu ters çevirmek için yığıt kullanılmıştır. Yığıta ait tanımlamalar ve metodlar yazılmıştır.

```
class Yigit {
Eleman ust=null;
boolean YigitBos(){
    if(ust==null)
        return true;
    else
        return false;
}

public void YigitaEkle(int n){
    Eleman e = new Eleman(n);
    e.ileri=ust;
    ust=e;
}

public int YigittanCikar(){
    Eleman cikacak=ust;
    if(!YigitBos())
        ust=ust.ileri;
    return cikacak.icerik;
}
```

Kuyruğa ilk eklenen veri ilk sırada çıkmaktadır. Son eklenen veriyi kuyruğun ilk elemanı yapmak için ters sırada işleyen yığıttan yararlanılmıştır. İlk olarak kuyruğun elemanları çıkarılarak yığıta eklenmiştir. Son durumda kuyruğun son elemanı yığıtın en üstündeki eleman olmuştur. Yığıttan çıkarma işlemi en üstteki elemandan gerçekleştirildiği için elemanlar ters kuyruğa yerleştirilecek sırada elde edilmiştir. Bu kez yığıttan elemanlar çıkarılarak boşalan kuyruğa eklenmiş ve ilk kuyruk ters çevrilmiştir.

```
public void KuyrukTersCevir(Kuyruk k){
    Yigit y = new Yigit();
    while(!k.KuyrukBos()){
        y.YigitaEkle(k.KuyruktanCikar());
    }

    while(!y.YigitBos()){
    k.KuyrugaEkle(y.YigittanCikar());
}
    System.out.println("Ters Kuyruk: ");
    k.KuyrukGöster();
}
```

Kuyruğun ilk n elemanını ters çevirmek için yine yığıt kullanılmıştır. İlk n eleman yığıta alınmış daha sonra tekrar çıkarılarak tersi sırada kuyruğa eklenmiştir. Bu şekilde ilk n eleman tersi sırada kuyruğun sonunda yer almıştır. Bunun için kuyruğun başında yer alan ilk n eleman dışındaki elemanlar kuyruktan çıkarılarak kuyruğun sonuna eklenmiştir. Sonuç olarak ilk n elemanın ters sıralandığı kuyruk yapısı elde edilmiştir.

```
public void KuyrukElemanTersCevir(Kuyruk k, int n){
    Yigit y = new Yigit();
    for (int i = 0; i < n; i++) {
        y.YigitaEkle(k.KuyruktanCikar());
    }

    while(!y.YigitBos()){
        k.KuyrugaEkle(y.YigittanCikar());
    }

    for (int i = 0; i < k.KuyrukElemanSayisi()-n; i++) {
        k.KuyrugaEkle(k.KuyruktanCikar());
    }

    System.out.println("Ters Kuyruk: ");
    k.KuyrukGöster();
}
```

Tamsayılar içeren bir kuyruk tanımlanarak metodlar çalıştırılmıştır.

```
    public static void main(String[] args) {
        Kuyruk k = new Kuyruk();

        for (int i = 1; i <= 10; i++) {
            k.KuyrugaEkle(i);
        }

        k.KuyrukGöster();
        k.KuyrukTersCevir(k);
        k.KuyrukElemanTersCevir(k, 4);
    }
}
```

Öncelikli kuyruk oluşturmak için Java hazır kuyruk sınıfı kullanılmıştır. String değerler içeren bir öncelikli kuyruk tanımlanmıştır. Artan öncelikli ve azalan öncelikli kuyruk yapıları oluşturulmuştur. Sıralama için Comparator sınıfından yararlanılmıştır.

```
import java.util.Comparator;
import java.util.PriorityQueue;

public class ÖncelikliKuyruk {
    public static void main(String[] args) {

PriorityQueue<String> pqueue1 = new PriorityQueue<String>();

PriorityQueue<String> pqueue2 = new PriorityQueue<String>(10, new Comparator<String>() {
    public int compare(String a, String b) {
        return b.compareTo(a);
    }
});

String[] eleman={"A", "G", "B", "L", "C", "Z", "K", "J", "E", "D"};

for (String e : eleman) {
    pqueue1.add(e);
    pqueue2.add(e);
}

System.out.println("Artan öncelikli kuyruk:");
while (pqueue1.size() != 0) {
    System.out.print(pqueue1.poll()+ " ");
}

        System.out.println("");

System.out.println("Azalan öncelikli kuyruk:");
while(pqueue2.size() != 0) {
    System.out.print(pqueue2.poll()+ " ");
}

        System.out.println("");

    }
}
```