

## KUYRUKLAR

---

- Queues
- Circular queues
- Priority queues

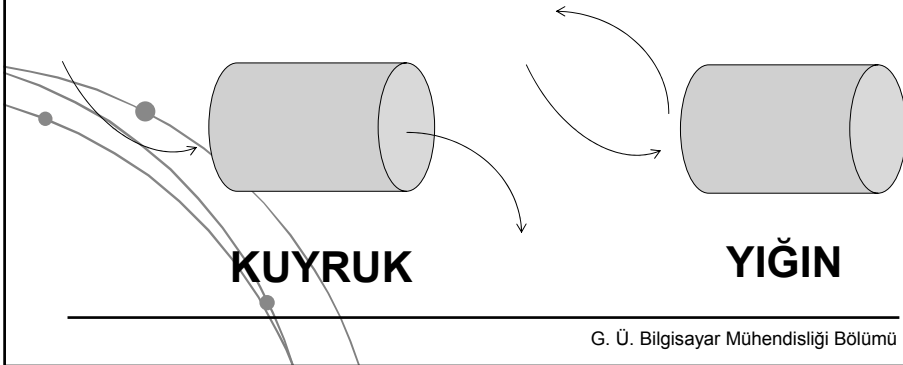
Yrd.Doç.Dr. M. Ali Akcayol

G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

---

enqueue → Queue → dequeue  
getFront



## KUYRUKLAR

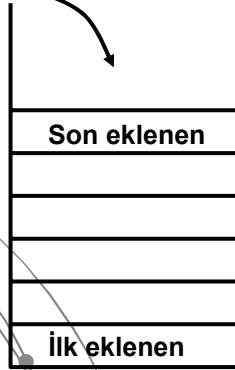
- Kuyruktaki elemanlardan sadece ilk eklenene erişim yapılır.
- Kuyruğa son eklenen eleman en son elde edilir.
- FIFO (First-in-First-out) veya LIFO (Last-in-Last-out) mantığıyla çalışır.
- İki tane temel işlem yapılabilir ;
  - enqueue, kuyruğun sonuna yeni bir eleman ekleme
  - dequeue, kuyruğun ilk elemanın alınması

G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

- Dizilerle veya bağlı listelerle yapılabilir.
- Dizilerde boyut değiştirme ve yeni eleman ekleme zorluğundan dolayı genellikle bağlı dizilerle yapılır.
- İlk elemanı gösteren bir node tanımlanır (headNode).

enqueue

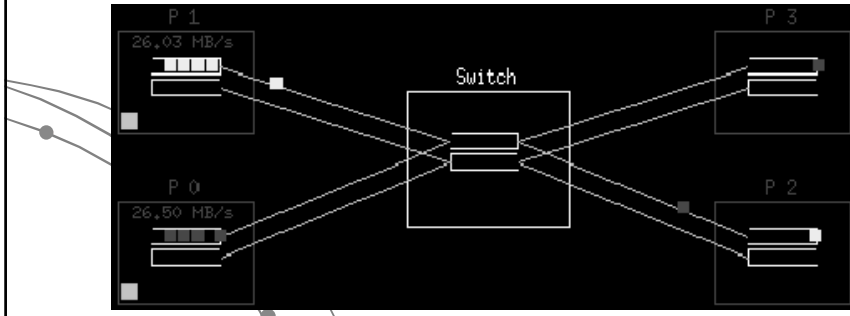


G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

### Örnek kullanım yerleri :

- Bilgisayarlarda kaynak paylaşımı sırasında kullanılır.
- Yazıcıya gönderilen işler kuyruğa konular ve sonra geliş sırasına göre işlem yapılır.
- Bilgisayar ağlarında paket bilgilerin gönderiminde kullanılır.



G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

### Queue oluşturma :

```
class queueNodeC
```

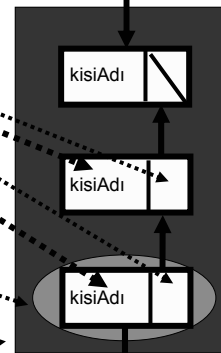
```
{  
    public string kisiAdi;  
    public queueNodeC sonraki;  
    public queueNodeC(string kisiAdi)  
        this.kisiAdi = kisiAdi;  
}
```

```
class queueC
```

```
{  
    public int size;  
    public queueNodeC headNode;  
    public queueC(string kisiAdi)  
    {  
        this.headNode = new queueNodeC(kisiAdi);  
        this.headNode.sonraki = headNode;  
        this.size = 0;  
    }  
}
```

```
queueC kisiKuyruk = new queueC("");
```

Son eleman



headNode

G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

Queue işlemleri :

- boş kuyruk
- eleman sayısı
- eleman ekleme
- eleman alma

size == 0

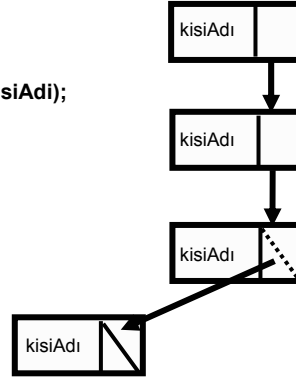
size

enqueue(kisiAdi)

dequeue()

```
public void enqueue(kisiAdi)
{
    queueNodeC yeniNode = new queueNodeC(kisiAdi);
    queueNodeC aktif = kisiKuyruk.headNode;
    while (aktif.sonraki != aktif)
    {
        aktif = aktif.sonraki;
    }
    aktif.sonraki = yeniNode;
    yeniNode.sonraki = yeniNode;
    kisiKuyruk.size++;
}
```

headNode

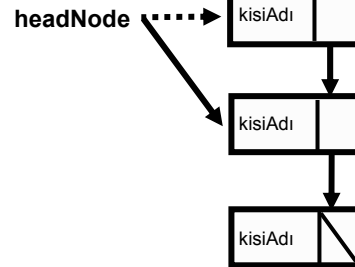


G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

Queue işlemleri (eleman alma)

```
public void dequeue()
{
    queueNodeC yeniNode = new queueNodeC("");
    yeniNode = kisiKuyruk.headNode;
    kisiKuyruk.headNode = kisiKuyruk.headNode.sonraki;
    kisiKuyruk.size--;
}
```

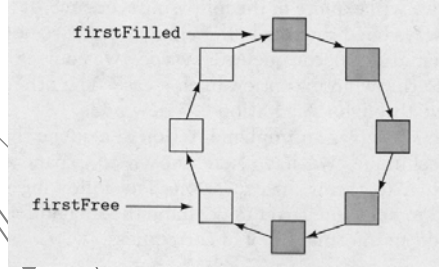
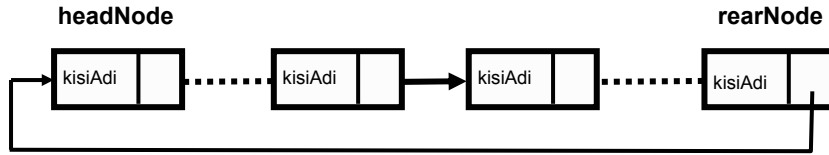


G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

### Circular Queues (Dairesel Kuyruklar)

- En son eleman kuyruğun ilk elemanını gösterir.

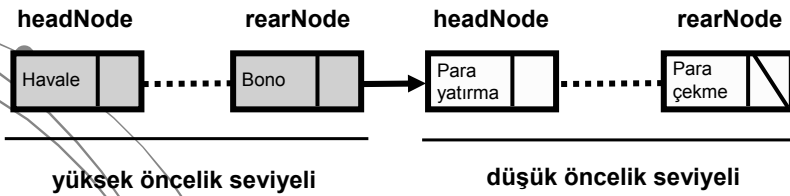


G. Ü. Bilgisayar Mühendisliği Bölümü

## KUYRUKLAR

### Priority Queues (Öncelikli Kuyruklar)

- Kuyruktaki işler kendi arasında önceliklerine göre seviyelendirilebilir.
- Her öncelik seviyesinin headNode ve rearNode'u vardır.
- Eleman alınırken en yüksek seviyeye ait elemanların ilk geleni öncelikli alınır.
- Yüksek öncelik seviyeli grubun son elemanı düşük öncelik seviyeli grubun ilk elemanından daha önceliklidir.



G. Ü. Bilgisayar Mühendisliği Bölümü

# KUYRUKLAR

## Uygulama programı (Stacks ve Queues):

Stack (Kitap Yığını)

Üst

Wireless Internet  
C# How to Program  
Algorithms  
Automata  
Computer Organization  
Data Structures

Kitap Adı (Push)  
Wireless Network

Push

Kitap Adı (Üst)  
Wireless Network

Pop

Max Size 10

Stack Size = 6

Alt

Queue (Banka Kuyruğu)

Ad - İş süresi - Bekleme süresi

Baş

Dilek - 5 - 0  
Mustafa - 3 - 5  
Mehmet - 8 - 8  
Fatma - 10 - 16  
Füsun - 2 - 26  
Fethi - 7 - 28

Kişi Adı (Enqueue)  
Fethi

İş süresi (dk) 7

Enqueue

Kişi Adı (Baştaki)

Dequeue

Max Size 10

Queue Size = 6

Son

Örnek Stack (Maze - Labirent)

Yrd.Doç.Dr. M. Ali Akcayol  
Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü

•G. Ü. Bilgisayar Mühendisliği Bölümü

# KUYRUKLAR

## Haftalık Ödev:

**Banka kuyruğu örneğindeki yapıyı öncelikli kuyruk yapısına dönüştürünüz. Banka işlemlerini kendi içerisinde üç öncelik grubuna ayırınız. Her yeni gelen kişiyi iş seçimine göre otomatik olarak ait olduğu grubun en sonuna ekleyiniz (enqueue). Her eleman alımında (dequeue) ise kuyruktaki en yüksek seviyeli grubun ilk elemanını alınız.**

G. Ü. Bilgisayar Mühendisliği Bölümü