

İlginç Market Simülatörü

Bir markette 1 adet kasa, **qServer** ve **qLottery** isimlerinde 2 adet kuyruk bulunmaktadır. Hizmet görececek sıradaki müşteri şu şekilde belirlenmektedir:

- **qServer** kuyruğunda eğer sıradaki müşterinin bekleme süresi **Wthreshold değerinden büyük veya eşitse** kuyruğun başındaki müşteri hizmet almaktadır. (müşteri kuyruktan çıkarılmaktadır)
- Küçükse bu müşteri **qLottery** kuyruğuna eklenmektedir. Ve rastgele bir sayı üretilerek sıradaki müşteri bu sayı ile bu kuyruktan seçilmektedir.
-

Bu marketin çalışmasını simüle eden bir Java programı yazmanız istenmektedir.

Detaylar

MM1 queue ile ilgili tanımlamalar

- Bir müşterinin kasada geçirdiği süreye hizmet süresi (**service time**) denilmektedir.
- Müşterinin servis alana kadar kasada bekleme süresi bekleme zamanı (**waiting time**) olarak isimlendirilmektedir.
- Müşterinin kasaya gelmesi/yada kasa dolu iken kuyruğa eklenmesi varış zamanı (**arrival time**)
 - **Arrival rate** (varış sıklığı) birim zamanda gelen müşteri sayısı iken (the number of arrivals per unit of time)
 - **Inter-arrival time** (geliş süreleri farkı) ise birbiri ardınca gelen iki varış süresi arasındaki zamandır. (the time between each arrival into the system and the next.)

Bir sonraki müşteri varış zamanı (**arrival time**) hesaplanması

Verilen müşteri varış sıklığı (**λ : lambda**: arrival rate) için aşağıdaki şekilde bir sonraki müşterinin geliş zamanı hesaplanabilir:

```
Random random = new Random();
double u = random. NextDouble();

double interArrivalTime =
    -Math.log(u)/(Double.parseDouble(lambda));

customerNext.arrivalTime
    = customerLast.arrivalTime + interArrivalTime;
```

Yine müşteri servis süresi (**service time**) hesaplama

Servis süresi yine beklenen ortalama müşteri servis süresi kullanılarak aşağıdaki şekilde exponansiyel dağılımdan oluşturulabilir:

Eğer ortalama servis süresi ($1/\mu$: μ) verildiğini varsayalım. Müşterinin rastgele servis süresi:

```
Random random = new Random();  
double u = random. NextDouble();  
customerX.serviceTime = -Math.log(u)/(Double.parseDouble(mu));
```

Bunu müşteri varlığında hesaplayarak atayabilirsiniz.

Simülasyonun yazılması

- Programı yazarken bir döngünün içerisinde simülasyon zamanı ilerlerken iki farklı event olabilir.
 - Customer Arrival: ya müşteri gelebilir
 - Customer Departure: Yada müşteri hizmet alıp kasadan çıkabilir.
- Simülasyon zamanını bu iki değere göre ilerletmeniz gerekmektedir.

Queueların yazılması

- Size verilen dosyalardaki **QServer circular array kullanarak queue implement etmesi gerekmektedir**. Bunu tamamlamanız gerekiyor.
- QLotteryde ekleme ve indeksle çıkarma (rastgele sayı ile çıkarıldığı için) işlemlerinin verimli olması istenmektedir. Bunun implementasyonu tamamen size bırakıldı.

Değerlendirme

- %30 puan QServer.java
 - %15 Dequeue
 - %15 Enqueue
- %30 QLottery
 - Çıkarma ve ekleme için metotlar size bırakıldı
- %40 CrazyMarket2021.java
 - %30 simulateMarket
 - Müşteri varış zamanları/çıkış zamanlarının hesaplanması
 - %10 diğer kısımlar
 - İstatistiklerin oluşturulması vb diğer kısımlar
- **-%20 Verilen Class ve metod prototiplerinin değiştirilmesi (yeni metodlar ekleyebilirsiniz)**
- **-%10 diğer kodlama standartları**
- **-%50 Compile olmayan kod**