Asansörlerdeki Talep Yoğunluğunun Multithread ile kontrolü

Ertuğrul Demir

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
170201052
ertugrulbusiness@gmail.com

ÖZET

Multitread kullanarak asansör simüle etmek. Simülasyonda 4 çeşit thread vardır.

1. Login Thread
   a. 1 adet bulunur.
   b. Görevi
      i. avmye giren insanları ürermek.
      ii. Bu insanların avm kat kuyruğuna girmesini sağlamak.

2. Exit Thread
   a. 1 adet bulunur.
   b. Görevi
      i. avmden çıkan isteyen insanları üretir
      ii. .

3. Controller Thread
   a. 1 adet bulunur.
   b. Görevi yoğunluğa göre aktif Elivator Thread sayısını ayarlar.

4. Elevator Thread
   a. 5 adettir.
   b. Görevi
      i. Kat Kuyruğunda bekleyenleri kata ulaştırır.
      ii. Çıkış kuyruğunda bekleyenleri çıkışa ulaştırır.

Projenin amacı, threadler kullanılarak oluşturulan insan yoğunluğunun tekrardan threadler vasıtasıyla simüle edilen ortamın kurallarına uygun olarak bu yoğunluğun kontrol edilmesidir.

1.GİRİŞ

Projenin başlatıldığı anda arka planda threadler oluşturulup işlem yapmaya başlamışlardır.

Yapılan işlemlerin gözlemlerini ekran üzerindeki butona basmak vasıtasıyla ekrana yansıtılabilir.

Bu threadler simülasyon için verilen kurallara uygun olarak görevlendirilimişlerdir.

Login Thread'i verilen bir kısa gecikme ardından rasgele sayıda insan yoğunluğu oluşturur. Oluşturulan bu youğunluk rastgele bir kata hedeflenir
.
Hedeflendirmenin ardından yoğunluk bu hedef için oluşturulmuş kuyruğa gönderilir.

Exit Thread'i kısa bir gecikmenin ardından rastgele katlarda avmden çıkmak isteyen insan yoğunlukları oluşturur. Oluşan bu yoğunluk kendi katında daha önceden oluşturulmuş ve zemin kata gitmek üzere hedeflenmiş kuyruğa ynlendirilir.

Controller Thread, zemin kattaki ve diğer katlardaki kuyrukların yoğunluğuna bakar. Bu yoğunlukları gidermekle görevli Elevator Threadlerin sayısını belirler. Yoğunluğun giderilmesinde Elevator Threadleri kontrol etmekle rol oynar..

Elivator Threadleri kuyruktaki insan yoğunluğunu hedefledikleri noktalara ulaştırarak gidermekle görevlidirler.Katlar arasında kısa bir gecikmeyle hareket ederler.

.

2.YÖNTEM

Simülasyon, verilen kurallara göre simüle edilmesi için kullanılanlar

- Yazılım dili olarak Java kullanılmıştır.
- IDE olarak Netbeans kullanılmıştır.
- Threadler kullandıkları otrak kaynaklarda problem çıkartmamaları için senkronize edilmiştir.
- Threadler java concurrency paketi içerisindeki thread sınıfından kalıtım olarak alınmıştır.
- Konsol üzerine yazı yazmak için Java standart output kullanılmıştır.
- Ekrana görselleri yansıtmak için Java Swing kütüphanesinden faydalanılmıştır.
- Görselleri java JFrame sınıfının override edilerek ekrana verilmiştir.
- Veriler, java sınıfları üzerinde tutulmuştur.
- Yapılan eylemler program içerisinde tanımlanmış özel threadler ile sağlanmıştır.

Bütün proje Nesne yönelimli programlamadan yararlanılarak geliştirilmiştir.

3.YALANCI KOD

3.1 Login Thread

1. Basla
2. Kisi Yogunluğu sayısı oluştur();
3. Yoguluğa rastgele hedef ver();
4. Hedefi Zemin kat kuyruğuna gönder();
5. Adım 2'ye dön.

3.2 Exit Thread

1. Basla
2. rastgele kat seç();
3. Çıkmak isteyen Kisi Yogunluğu sayısı oluştur();
4. Hedefi çıkış isteği kuyruğuna gönder();
5. Adım 2'ye dön.

3.3 Control Thread

1. Basla
2. Kuyrukdaki yolcu sayısı sınırı aştı mı?
   a. Aştıysa, thread ekle.
3. Kuyrukdaki yolcu sayısı sınırın altında mı?;
   a. Altındaysa thred çıkar.
4. Adım 2'ye dön.

3.4 Asansör Thread

1. Basla
2. Zemin kattan queue'sinden yoğunluk al
3. Kat_çık
4. Katı kontrol et
   a. Hedef kat ise
      i. Youğunluk bırak
5. Kat_in
6. Katı kontrol et
   a. Katın queuesi doluysa ve yer varsa
      i. Kat queue'sinden eleman al
7. Zemin kata in
8. Elemanları boşalt.
   a. Çıkış yapanların sayısını arttır.
9. Adım 2'ye dön.

4.YAKLAŞIM

Proje deneme yanılma ve bu tecrübeler üzerinden eldeki ürünü geliştirme stratejisini izlemiştir.

Denemeler sırasında karşılaşılan hatalar ilgili teknolejiyi sunan firmaların dökümantasyonları aracılığıyla çözülmeye çalışılmıştır.

Dökümantasyonlardaki alternatif yöntemler projeye uygulanmaya çalışılmıştır.

Vaar olan sınıflar override edilerek tekrardan isteklere cevap verilbilmesi için çalışmalar yürütülmüştür.

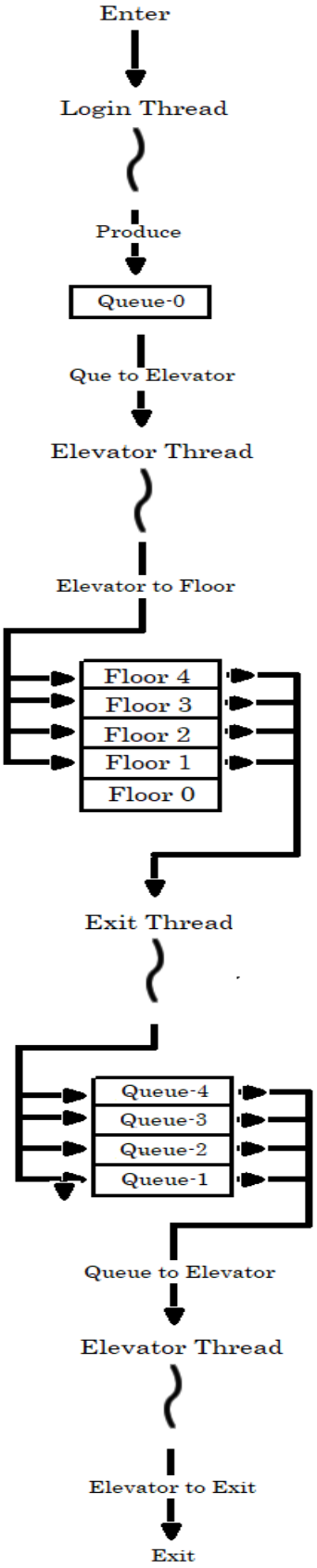
Çözülen ve elde edilen kodlar daha sonrasında gerekli isterleri karşılabildiklerinden emin olunmak amacıyla teste tabi tutulmuşlardır.

Kullanılan kodların eksik tarafları üzerine test senaryoları oluşturularak program sınanmıştır.

5.KAZANIMLAR

- Nesne yönlimli programlama
- Thread oluşturma
- Thread senkronizasyonu
- Programları test etme
- Test senaryosu oluşturma
- Sınıf oluşturma
- Kalıtım
- Kullanıcı arayüzü için Component kullanımı
- Ekrana görsel çıktı verme

6.PROJENİN GÖRSEL İFADESİ



7.EKSİKLİKLER

- Kullanıcı arayüzü tasarımı
- Çıktının otomatik olarak JFrame üzeinde gösterilmesinin sağlanamaması.
- Efektif olmayan Kullanıcı etileşimleri
- Asansör Thread'lerinin sağlıklı senkronize edilememesi.
- Threadlerin kullanıcı arayüzü ile efektif olmayan etkileşimleri.

8.SONUÇ

Program Kullanıcı arayüzü üzerinden kontrol edilebilir ve çıktıları konsoldan alınabilir.

Açılan ekran vasıtasıyla Threadlerin yaptıkları hareketler takip edilebilir.

Asansör Threadlerinin senkronizasyonundaki problemlerin sağlıklı olarak giderilememesi nedeniyle anlam verilemeyen yoğunluk hareketleri.

Projenin eksikleri ise

- Sağlıksız thread senkronizasyonu
- Kullanıcı arayüzünün görselliğinin kullanıcı dostu olmaması.
- Kullanıcı arayüzündeki verilerin arayüzünde gözükebilmesi için gerekli olan güncellemenin manuel olarak button ile sağlanması.

9.KAYNAKÇA

1) https://docs.oracle.com/javase/tutorial/
2) https://stackoverflow.com/
3) https://www.tutorialspoint.com/index.htm
4) https://www.w3schools.com/

10.DENEY GÖRSELLERİ


```
------------------------------------
Elevator_4 Activated
------------------------------------



Asansör Yukarı Yöne Hareketleniyor....


-->Totalde Çıkan eleman sayısı: 0
-->Totalde Çıkan eleman sayısı: 0


Asansör Yukarı Yöne Hareketleniyor....



Asansör Yukarı Yöne Hareketleniyor....


-->Totalde Çıkan eleman sayısı: 0
```


```
------------------------------------
Elevator_5 Activated
------------------------------------



Asansör Yukarı Yöne Hareketleniyor....



Asansör yukarı Çıkıyor...
Asansör yukarı Çıkıyor...
Asansör yukarı Çıkıyor...
Asansör yukarı Çıkıyor...
Asansör yukarı Çıkıyor...
Kat-1
Asansör yukarı Çıkıyor...
Kat-1


Asansör Aşağı Yöne Hareketleniyor....


çıkış talebinin olduğu en yüksek başlangıç katı:3
-->1.kattan ==>3.kata  2adet kat gecilecek
-->Asansor iniş için  başlangıç katına çıkıyor...
Kat-1
Asansör yukarı Çıkıyor...
Kat-1
Asansör yukarı Çıkıyor...
Kat-1
Asansör yukarı Çıkıyor...
Kat-2


Asansör Aşağı Yöne Hareketleniyor....


çıkış talebinin olduğu en yüksek başlangıç katı:3
-->2.kattan ==>3.kata  ladet kat gecilecek
-->Asansor iniş için  başlangıç katına çıkıyor...
Kat-2
```