ASANSÖRLERDEKİ TALEP YOĞUNLUĞUNUN MULTITHREAD İLE KONTROLÜ

Şevki Karagöl
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
170201009

Mustafa Yiğit
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
180201108

ÖZET- Bu projede, projeyi yapan kişiler için thread'lerin çalışma mantığını anlaması, bunun ardından bu yapıları kullanarak proje gerçekleştirimi ile beraber projeyi yapan kişilerin thread ve multithread kavramlarının yapısını anlaması ve çözüm sağlayabilmesi amaçlanmaktadır. Bu amaç doğrultusunda öğrencilerden, talep yoğunluğuna göre bir alışveriş merkezine ait asansörlerin kullanımına ilişkin bir uygulama geliştirilmesi istenmiştir.

Bu amaçlar ve isterler doğrultusunda multithread yapısı kullanılarak talep yoğunluğuna göre aksiyon alan bir asansör uygulaması geliştirilmiştir.

Anahtar kelimeler-

Asansör, thread, multithread, talep yoğunluğu, eş zamanlı çalışma ve program

I.GİRİŞ

Bu projede istenilen koşul aralıklarında çalışan thread'ler doğrultusunda bir çalışma düzeni oluşturulmuştur. Eş zamanlı çalışan thread'ler ve bu thread'lerin içinde bulunduğu sınıflara ait fonksiyonlarla beraber çalışan bir konsol uygulaması yazılmıştır. Oluşan bu uygulamada thread'ler çeşitli işlemler yapar ve bu işlemler doğrultusunda program çalışır.Bu işlemlerin durdurulmasının ardından proje sonlandırılmış olur.

Bu projede thread ,multithread,arraylist yapılarının ve random sınıfının bir arada kullanımına yönelik bir çalışma gerçekleştirilmiştir. Aynı zamanda öğrencilerin, proje isterlerlerinin çözümününe yönelik oluşturduğu algoritmalar ide aracılığıyla bilgisayar ortamına aktarılmıştır. Proje için tercih edilen programlama dili "Java olmuştur.

II.TEMEL BİLGİLER

Bu proje Java Programlama dili ile geliştirilmiş olup, geliştirme ortamı olarak "Netbeans IDE 8.2" kullanılmıştır. İlk etapta proje için bir yol haritası çıkarılarak ön hazırlık sürecine girilmiştir.Bu aşamada projenin isterlerine yönelik araştırmalar gerçekleştirilmesi adına grup içerisinde bir iş bölümü yapılmış olup elde edilen veriler doğrultusunda projenin ana hatları ortaya çıkarılmış ve büyük ölçüde karşılaşılabilecek problemler saptanıp çözümlendirildikten sonra IDE ortamında projenin ilk adımları atılmıştır.

Yapılan ön hazırlık sürecinde Java Programlama dilinde thread'ler arasındaki bağlantının yapılması, random kütüphanesi kullanılarak proje isterlerine yönelik ran--dom sayı üretimi, oluşturulan thread'lerin multithread yapısı ile Netbeans IDE ortamında çalıştırılması, thread'lerin eş zamanlı çalışmasından kaynaklanan senkronizasyon sorunu gibi problemler üzerinde durulmuştur. Bu problemler aşılıp projeye şekil verme aşamasına gidilmiştir. Proje ön hazırlık süreciyle birlikte yaklaşık bir haftalık bir süreçte tamamlanmıştır.

III.YÖNTEM

Bu proje gerekli thread'lerin oluşturulduğu ve çalıştırıldığı main sınıfı, asansörlerin hareket etmelerini sağlayan asansör thread sınıfları, alışveriş merkezine belirli aralıklarla müşteri giriş ve çıkışı olmasını sağlayan thread sınıfları, talep yoğunluğuna göre asansörlerin aktiflik durumlarını değiştiren kontrol thread sınıfı ve alışveriş merkezindeki kuyrukların arraylist şekinde tutulmasını sağlayan kontrol sınıfından oluşmaktadır. Bu başlık altında, kullanılan sınıfların içerikleri detaylı bir şekilde anlatılmıştır.

Bu projede kullanılan sınıflar:

1)Asansör Sınıfları

-AsansorThread1

-AsansorThread2

- AsansorThread3

- AsansorThread4

- AsansorThread5

2)LoginThread

3)ExitThread

4)KontrolThread

5)Katlar

6)Main

Bu başlık altında sınıflar ayrı ayrı ele alınacak ve detayları anlatılacaktır.

1)Asansor Sınıfları

Tüm asansörlerin çalışması için gerekli run() fonksiyonlarının ve random sayıda oluşturulan müşterilerin alışveriş merkezi içerisinde hareketini sağlayan fonksiyonların bulunduğu sınıflardır. KontrolThread sınıfından alınan bilgiler doğrultusunda asansörlerin aktiflik durumları bu sınıflarda değiştirilmiştir. Thread sınıfından kalıtım alırlar ve main sınıfından çağrılırlar.

2)LoginThread

Bu sınıf içerisinde alışveriş merkezine 500 ms aralıklarla müşteri girmesini ve bu müşterilerin belirli katlara gitmek için kuyruğa eklenmesini sağlayan thread bulunmaktadır. Bu sınıf, Thread sınıfından kalıtım alır ve main sınıfından çağrılır.

3)ExitThread

Bu sınıf içerisinde alışveriş merkezinden 1000 ms aralıklarla belirli sayıda müşteri çıkması için müşterileri katlarda bulunan asansör kuyruklarına ekleyen thread bulunmaktadır. Bu sınıf, Thread sınıfından kalıtım alır ve main sınıfından çağrılır.

4)KontrolThread

Bu sınıf içerisinde katlarda bulunan müşterilerin asansörlere gönderdiği istek yoğunluğuna göre alışveriş merkezinde bulunan beş asansörden hangilerinin çalışıp, hangilerinin pasif durumda bekleyeceğine karar veren thread bulunur. Bu sınıf, Thread sınıfından kalıtım alır ve main sınıfından çağrılır.

5)Katlar

Bu sınıf içerisinde katlarda bulunan tüm müşterilerin, hangi durumda olduğunu bildiren(gezen ya da asansör bekleyen) Arraylist yapıları bulunur. Bu Arraylist yapılarında her index bir müşteriyi temsil eder ve her index'in değeri o müşterinin gitmek istediği kat numarasını tutar. Ayrıca bu sınıfta asansörlerin senkronizasyonunda kullandığımız anahtar objesi bulunmaktadır. Bu sınıf içerisinde bulunan tüm yapılar ve değişkenler proje içerisinde bulunan diğer sınıflardan ulaşılabilecek şekilde tanımlanmıştır.

6)Main

Tüm sınıfların çağrıldığı ve thread'lerinin çalıştırıldığı sınıftır. Anlık durumlar bir "While" döngüsü ile konsola bu sınıfta yazdırılır. Proje bu sınıf üzerinden çalıştırılmaktadır.

IV.KABA KOD

1)Program çalıştı.

2)Tüm thread'ler eş zamanlı olarak çalışmaya başladı.

3)LoginThread çalışarak giriş katın asansör kuyruğuna müşteriler eklendi.

4)Asansör thread'leriyle müşteriler git-mek istedikleri katlara çıkarıldı ve çıkış yapmak isteyenler giriş kata indirildi.

5)ExitThread ile alışveriş merkezinden çıkış yapmak isteyen müşteriler bulundukları katların asansör kuyruklarına eklendi.

6)KontrolThread sürekli olarak asansörlere gelen talebi değerlendirdi ve bunun sonucundan asansörlerin aktiflik ve pasiflik durumlarını değiştirdi.

7)Anlık olarak asansörlere, katlara ve müşteri durumlarına ait bilgiler konsol ekranına yazdırıldı.

****Madde 7'de bahsi geçen, anlık olarak güncellenen ve programın çalışması hakkında bilgi içeren çıktılar 200 ms aralıklarla yazdırılmıştır. Bunun sebebi, gecikme olmadığında yazdırma işlemi sürekli çalıştığından gereksiz ve kirli bilgi yazdırmasıdır.**

V.EKRAN GÖRÜNTÜLERİ

*ek1

```
0. kat --> Kuyruk : 14
1. kat --> Toplam : 0 Kuyruk : 0
2. kat --> Toplam : 1 Kuyruk : 0
3. kat --> Toplam : 0 Kuyruk : 0
4. kat --> Toplam : 4 Kuyruk : 0
Cikis Yapan Kisi Sayisi : 0
```

*ek2

```
Asansör 1
        Durum : true
        Bulundugu Kat : 3
        Hedef Kat : 0
        Yon : Asagi
        Kapasite : 10
        Icindeki Insan Sayisi : 0
        Icindekiler : []
Asansör 2
        Durum : true
        Bulundugu Kat : 1
        Hedef Kat : 2
        Yon : Yukari
        Kapasite : 10
        Icindeki Insan Sayisi : 10
        Icindekiler : [3, 3, 3, 3, 2, 2, 2, 2, 2, 2]
Asansör 3
        Durum : false
        Bulundugu Kat : 2
        Hedef Kat : 0
        Yon : Asagi
        Kapasite : 10
        Icindeki Insan Sayisi : 0
        Icindekiler : []
Asansör 4
        Durum : false
        Bulundugu Kat : 0
        Hedef Kat : 0
        Yon : Yukari
        Kapasite : 10
        Icindeki Insan Sayisi : 0
        Icindekiler : []
Asansör 5
        Durum : false
        Bulundugu Kat : 0
        Hedef Kat : 0
        Yon : Yukari
        Kapasite : 10
        Icindeki Insan Sayisi : 0
        Icindekiler : []
```

*ek3

```
Katlarda Asansor Kuyrugunda Bekleyenler
        0. Kat : [[10,2]]
        1. Kat : []
        2. Kat : [1,0]
        3. Kat : []
        4. Kat : []
```

VI.SONUÇ

Tasarlanan algoritmalar eş zamanlı çalışma mantığına göre birleştirilerek bir program geliştirilmiştir. İşletim Sistemleri dersinde temelleri atılan Thread yapısının, Java üzerinde nasıl çalıştırıldığı öğrenilmiştir.

VII.REFERANSLAR

[1]https://gelecegiyazanlar.turkcell.com.tr/

Java'ya dair bazı işlemlerin kullanımının hatırlanması.

[2] https://stackoverflow.com/

Proje yapım aşamasında karşılaşılan hataların sebeplerinin araştırılması.

[3] https://www.udemy.com/

Thread mantığının tekrar edilmesi ve eşzamanlı kullanılmasının öğrenilmesi.

[4] https://www.geeksforgeeks.org/

Çok boyutlu ArrayList'lerin tanımlanma biçimlerinin öğrenilmesi.