

PROJE 2: Multithread Projesi

Barış Arslan 180202112 – Melih Yeşilyurt 180202060

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

barisaarslan98@hotmail.com – melih58yesilyurt@outlook.com

1-) Özet

Geliştirdiğimiz proje başlatıldığında gelen ara yüzde başlat butonuna basıldıktan sonra başlar. 500 ms zaman aralıklarıyla 1-10 arasında rastgele sayıda müşteri üretilir. Sonrasında ise bu yolcular 1. Asansör yardımıyla 200 ms aralıklarla katlara dağıtılmaya başlar. Asansör sayısı asansör kapasitesinin 2 katını aştığı durumda yeni asansör devreye girer. Aynı şekilde kuyruklardaki toplam yolcu sayısı asansör kapasitesinin altına indiğinde asansörlerden biri pasif hale geçer. Her 1000 ms aralıklarla da 1-5 arasında rastgele sayıda müşteri, AVM den çıkış yapmak üzere yine rastgele bir kattan (1-4), asansör kuyruğuna alınır. Bu işlemler programı kapatana kadar devam eder.

2-) Giriş

Projenin girişinde kullanıcıdan herhangi bir giriş değeri istenmiyor. Başlat butonuna basılınca proje başlar. Başlangıçta Enterance, Exit, ControlTask ve 1. Asansör threadi çalışmaya başlar. Enterance threadi sayesinde 500 ms zaman aralıklarıyla 1-10 arasında rastgele sayıda kişi Alışveriş merkezine giriş yapar. Bu kişiler kuyruğa alınır. Exit Threadi ise diğer katlarda boşta olan 1-5 arasında rastgele sayıda insanları bulundukları

katlardaki kuyruklara 1000ms aralıklarla yerleştirir. Control Threadi ise katlardaki kuyrukları kontrol eder. Eğer kuyruklardan biri 20'yi geçtiyse diğer asansörü aktif hale getirir. Eğer bütün kuyruklardaki toplam insan sayısı asansör kapasitesinin altına indiyse bir asansörü kapatır. Asansör threadleri ise asansör sayısı kadardır maksimum 5 tane olabilir. Kuyruklardan asansör isteği gelirse o kuyruğa gider ve o kuyruktaki insanları gitmek istedikleri yerlere bırakır.

3-) Yöntem

Java dilinde bulunan JavaFx uygulamasıyla ara yüzü tanımladık. Bulunmayan fonksiyonlar ekibimiz tarafından yapıldı. Proje yapılırken birçok problemle karşılaşıldı. Problemlerin çözümünde ekip halinde çözüm aranıldı, aynı problem veya benzeri bir problemle karşılaşmış arkadaşlarımızla görüş alışverişi yapıldı, internette daha önce bu problemlerle karşılaşan insanların bu problemlerin üstesinden nasıl geldiği araştırıldı ve farklı bakış açıları kazanıldı. Bulunan birçok farklı çözümünden problemlerimizin çözümlerine uygun olanları seçildi.

Proje ilk açıldığında kullanıcının karşısına proje uygulama ara yüzü geliyor.

Proje 3 farklı tür nesne bulunur. Bu nesneler elevator, floor ve persongroup olarak geçer. Elevatorın içinde asansör hakkında bilgiler tutulur ve buradan yeni asansörler üretilir. Floor ise proje ilk çalıştığında 5 adet üretilir ve her kat için 1 tane vardır. İçinde kat ile ilgili bilgileri tutar. PersonGroup ise her bir insan için üretilir. İçinde insanın hangi gruba ait olduğunu nereye gittiğini ve nerede bulunduğunu tutan değişkenler vardır.

Proje her bir thread için 4 farklı task bulunmaktadır. Bunlar sırasıyla EntranceTask, ElevatorTask, ControlTask, ve ExitTask'dır.

Projede ilk önce 4 thread çalıştırılır. Bunlar giriş, çıkış, kontrol ve asansör threadi. Giriş threadinde sonsuza dek 500 ms zaman aralıklarıyla 1-10 arasında rastgele sayıda müşteri üretilir. Bu müşteriler giriş katının kuyruğuna eklenir.

Çıkış threadin de ise her 1000 ms aralıklarla da 1-5 arasında rastgele sayıda müşteri, AVM den çıkış yapmak üzere yine rastgele bir kattan (1-4), asansör kuyruğuna alınır. Bu müşterilerin yeni hedefi ise zemin kat olarak belirlenir.

Asansör threadi bir for döngüsü içerisinde çalışır. Bu for döngüsünün içerisinde sürekli bir yeni kişi ekler. Asansör kapasitesi dolduğunda ise durdurur. `for (PersonGroup personGroup: personGroups)`

Bu süreç başlamadan önce asansörü meşguliyet dizisinde 1 hale getiriyor. `this.elevator.setBusy(new AtomicInteger(1));` Grubun içeriğini arayüze yazdırır ve grubun üyelerinin daha önceden bulunduğu kuyruktan onları siler.

```
floors.get(personGroup.getStartFloor()).getQueue().remove(personGroup);
```

Bu olaydan sonra asansörün meşguliyetini 0 hale getirir, yani asansör müsait olur.

```
this.elevator.setBusy(new AtomicInteger(0));
```

Asansör hedeflediği yerle, başta bulunduğu yer kadar 200 ms bekler.

```
Thread.sleep(200 * Math.abs(personGroup.getEndFloor() - personGroup.getStartFloor() + 1));
```

Daha sonrasında ise içindeki insanların konum ve hedef bilgilerini günceller.

```
personGroup.setStartFloor(personGroup.getEndFloor());  
personGroup.setEndFloor(0);
```

Kontrol threadi sonsuza dek çalışır. İlk önce ara yüzü günceller. İçerisindeki For döngüsü sayesinde katlar arasında dolaşır. `for (Floor floor : floors)`

Kattan alacağı kişi sayısını belirler ve ilk boştaki asansörün id'sini çağırır.

```
Elevator elevator = this.findNextAvailableElevator();  
int groupSize = Math.min(floor.getQueue().size(), 10);
```

Elde ettiği verileri asansör threadine gönderir.

```
elevator.getExecutorService().submit(new ElevatorTask(floors, elevator, groups));
```

Kontrol threadi kuyrukları kontrol eder eğer kuyruklardan biri 20'yi geçtiyse yeni bir asansör threadi oluşturur. `if ((floor.getQueue().size() > 20) && (this.elevators.size() < 5))`

Eğer bütün kuyrukların toplamı 10'un altına indiyse bir asansör threadi kapatır. `if ((allCountQueue < 10) && (this.elevators.size() > 1))`

Eğer bütün kuyrukların toplamı 10'un altına indiyse bir asansör threadi

kapatır. `if ((allCountQueue < 10) && (this.elevators.size() > 1))`

Projede threadlerden ayrı 2 class bulunmaktadır. Bu classlardan biri main classıdır. Bu class proje ekranının açılmasını sağlamaktadır. Bu class

2. önemli olan class ise ShoppingMall isimli classtır. Bu class doğrudan ara yüze bağlıdır. Ara yüzde gözükten verilerin oraya aktarılmasını sağlar ve threadlerin oluşturulduğu yerdir.

Proje de ara yüzden kaynaklı sorunlar çıkıyor bu sebeple grupları bastırmamayı tercih ettik. Ara yüz her zaman güncel olmuyor hatta bazen kodun patlamasına sebep oluyor. Bu sorunu ne kadar araştırsak da çözemedik.

4-) Sonuç

Sonuçta Java dili kullanılarak, Thread, Multithread, Ara yüz, gibi konularda program yazılarak Form yapıları düşünce tarzının ve algoritmik düşünme yetisinin gelişmesi sağlandı.

5-) Deneysel Sonuçlar

Mail Elevator

AVM

0. Floor:	Queue: 111	Groups : 0	
1. Floor: all : 1	Queue: 1	Groups : 0	
2. Floor: all : 1	Queue: 0	Groups : 0	
3. Floor: all : 3	Queue: 3	Groups : 0	
4. Floor: all : 2	Queue: 0	Groups : 0	
Exit Count : 0			

Elevator-1:

Active:	True
Floor:	4
Destination:	4
Direction:	Up
Capacity:	10
Count Inside:	0
Inside:	(0), (2), (3), (8), (9), (13), (14)
Mode:	Working

Elevator-2:

Active:	True
Floor:	0
Destination:	0
Direction:	Up
Capacity:	10
Count Inside:	0
Inside:	0
Mode:	Working

Start

Elevator-4:

Active:	False
Floor:	0
Destination:	0
Direction:	Up
Capacity:	10
Count Inside:	0
Inside:	0
Mode:	Idle

Elevator-3:

Active:	False
Floor:	0
Destination:	0
Direction:	Up
Capacity:	10
Count Inside:	0
Inside:	0
Mode:	Idle

Elevator-5:

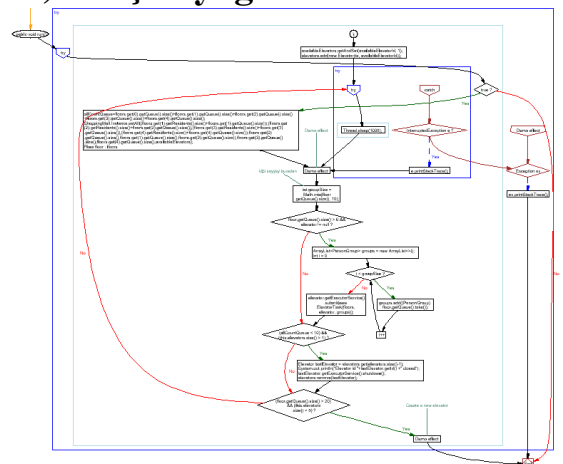
Active:	False
Floor:	0
Destination:	0
Direction:	Up
Capacity:	10
Count Inside:	0
Inside:	0
Mode:	Idle

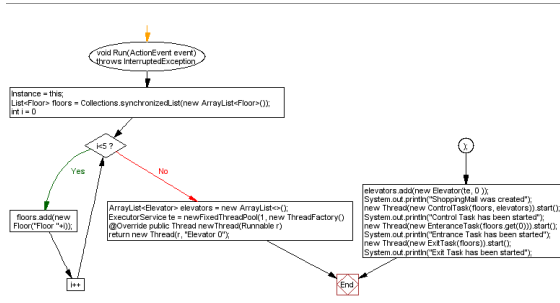
```

Group size: 3 and id: #341 has arrived to shopping mall
Group size: 3 and id: #342 has arrived to shopping mall
1 group [9] was loaded to the Floor 0 's queue
Elevator id: 0 Person Group id 47 reached to the 3 from 0
Group size: 8 and id: #343 has arrived to shopping mall
Group size: 8 and id: #344 has arrived to shopping mall
Group size: 8 and id: #345 has arrived to shopping mall
Group size: 8 and id: #346 has arrived to shopping mall
Group size: 8 and id: #347 has arrived to shopping mall
Group size: 8 and id: #348 has arrived to shopping mall
Group size: 8 and id: #349 has arrived to shopping mall
Group size: 8 and id: #350 has arrived to shopping mall
Group size: 4 and id: #351 has arrived to shopping mall
Group size: 4 and id: #352 has arrived to shopping mall
Group size: 4 and id: #353 has arrived to shopping mall
Group size: 4 and id: #354 has arrived to shopping mall
Elevator id: 0 Person Group id 48 reached to the 1 from 0
3 group [35, 37, 40] was loaded to the Floor 4 's queue
Elevator id: 1 Person Group id 51 reached to the 1 from 0
Group size: 5 and id: #355 has arrived to shopping mall
Group size: 5 and id: #356 has arrived to shopping mall
Group size: 5 and id: #357 has arrived to shopping mall
Group size: 5 and id: #358 has arrived to shopping mall
Group size: 5 and id: #359 has arrived to shopping mall
Elevator id: 0 Person Group id 49 reached to the 4 from 0
Elevator id: 1 Person Group id 52 reached to the 3 from 0
Group size: 6 and id: #360 has arrived to shopping mall
Group size: 6 and id: #361 has arrived to shopping mall
Group size: 6 and id: #362 has arrived to shopping mall
Group size: 6 and id: #363 has arrived to shopping mall
Group size: 6 and id: #364 has arrived to shopping mall
Group size: 6 and id: #365 has arrived to shopping mall
3 group [43, 46, 49] was loaded to the Floor 4 's queue
Group size: 6 and id: #366 has arrived to shopping mall
Group size: 6 and id: #367 has arrived to shopping mall
Group size: 6 and id: #368 has arrived to shopping mall
Group size: 6 and id: #369 has arrived to shopping mall
Group size: 6 and id: #370 has arrived to shopping mall
Group size: 6 and id: #371 has arrived to shopping mall
Elevator id: 1 Person Group id 53 reached to the 3 from 0
Elevator id: 0 Person Group id 50 reached to the 1 from 0
Group size: 6 and id: #372 has arrived to shopping mall

```

6-) Akış Diyagramı





7-) Kaynakça

- <https://www.youtube.com/watch?v=TCd8QIS-2KI>
- <https://www.youtube.com/watch?v=L95658yXRgI>
- <https://stackoverflow.com/question/s/28493053/java-manipulating-non-static-variable-from-thread>
- <http://yazilimnotlari.com/makale-1051-c-sharp-temelleri-3-array-ve-arraylist-dizi-ve-dizi-listeleri>
- <https://stackoverflow.com/question/s/7968631/how-can-i-check-if-a-queue-is-empty>