

PROJE 1: AVM Asansör Simülasyonu

Erdem Nayın 180202050

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

erdem_nayin@hotmail.com – 180202050@kocaeli.edu.tr

1-)Özet

Daha önceden kıstasları belirlenen AVM asansör simülasyonu geliştirildi. Program geliştirilirken thread kavramının uygun yerlerde kullanılması, aynı işi yapan veya aynı bellek alanlarına erişerek işlem yapan farklı threadlerin birbiriyle çalışmalarını senkronize etmek, bir thread içinden birden fazla threadin kontrolünü sağlamak, bunun yanında geliştirilen uygulamanın dinamik bir şekilde geliştirilmesini sağlayarak kodlama ve algoritmik düşünme becerisinin geliştirilmesi amaçlandı.

2-)Giriş

Programı çalıştırmaya başlayınca şu anda çalışmakta olan threadler ve yaptıkları işlemler ve durumları kendi renkleriyle kendi çıktılarını görebilirsiniz. AVMExit ve AVMEntrance threadleri her işlem yaptığında ekranda yaptığı işlemi belirtir. Elevator'ler değiştirdikleri her katta, insanları ilgili katlara bırakırken, içine alacak insanları seçerken ekrana çıktı olarak o anki durumunu yazdırır.

3-)Yöntem

Proje java programlama dili ve İntellij kullanılarak yapıldı.

Java dilinde bulunan hazır kütüphanelerden ve fonksiyonlardan yararlanıldı. Bulunmayan fonksiyonlar ekibimiz tarafından yapıldı.

Proje yapılırken birçok problemle karşılaşıldı. Problemlerin çözümünde, 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.

Projeye ilk olarak, AVM'ye giriş ve çıkış yapacak olan insanların temsili için grup içindeki insan sayısını ve gitmek istedikleri katı belirten veri yapıları bulunan Group classı oluşturularak başlandı.

Daha sonra AVM'de bulunan katların temsil edildiği ve içinde, o katta bulunan insanların sayısını ve o kattan asansör sırasına giren ve AVM'den çıkmak isteyen insanların tutulduğu Queue yapısı ve bu insanların toplam sayısının tutulduğu veri yapıları bulunan Floor classı oluşturuldu.

Daha sonra terminalde hangi threadin çalıştığının anlaşılmasını kolaylaştırılmak için, içinde ASCII Escape karakterleriyle oluşturulmuş renk kodları bulunan Color classı oluşturuldu.

Daha sonra giriş katına 500ms aralıklarla 1-10 arasında insan sayısı üreten ve 1-4 arasında hedef kat üretip Group yapısında saklayan ve giriş katının Queue yapısına bu grupları ekleyen AVMEntrance threadi oluşturuldu.

Daha sonra AVM'den çıkış yapmak isteyen insanların maksimum 5 kişi üreterek katların Queue yapılarına alınmasını sağlayan AVMEExit threadi oluşturuldu. Kişi üretme sayısı katta bulunan kişiler, katın queue'sunda bulunan kişilerin ilgili mantık işlemlerinden geçirilmesiyle belirlendi.

Daha sonra AVM'de bulunan asansörleri temsil eden Elevator classı oluşturuldu. Bu class durmadan çalışacak şekilde tasarlandı. İçine yazılan fonksiyonlarla durdurma ve devam ettirme işlemlerinin yapılabilmesi sağlandı. Bu class özünde 4 ana fonksiyona sahiptir ve bu fonksiyonlar da 2 ana fonksiyonda birleştirilmiştir. Yani asansörler temelde 2 iş yaparlar : Yolcuları giriş kattan kapasiteni aşmayacak şekilde al ve grupları sırayla hedef katlarına git, içindeki yolcuları hedef katlarına bırak ve bıraktıktan sonra giriş katı hariç diğer katların sıralarını kontrol et ve yine kapasiteni aşmayacak şekilde bu katlara giderek insanları içine al ve giriş katına indirip AVM'den çıkart. Bu fonksiyonların yanında asansörün

durumunu kontrol eden ve ilgili durumlara göre değiştiren, asansörün katlar arasında geçiş yapmasını sağlayan ve her kat geçişinin 200ms sürmesini sağlayan fonksiyonlar da yazıldı.

Daha sonra tüm asansörleri yaratan ve tüm katların sıralarında bekleyen insanların toplam sayısını hesaplayıp bu sayı 20'yi aşarsa bir asansörü aktive edip bu sayı 10'dan aşağı inerse bir asansörün çalışmasını durdurmasının kontrolünü yapan ElevatorController classı oluşturuldu.

Terminalde çıktıların düzgünce okunup yorumlanabilmesi için ilgili classların toString fonksiyonları override edilerek düzenlendi ve her bir thread oluşturulurken constructor'una ilgili Color classından renkler verildi ve böylece programımızın yapımı tamamlanmış oldu.

4-)Sonuç

Sonuçta Java dili kullanılarak thread kavramı kullanılarak eş zamanlı işlemlerin yapılmasının sağlanması, threadlerin birbiriyle senkronize çalışması, threadlerin bekletilip devam ettirilmesi, ve algoritmik düşünme yetisinin gelişmesi sağlandı.

5-)Deneysel Sonuçlar

```
0. floor => Queue : [[9, 3]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 9 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 17
```

```
mode : working
floor : 2
destination : 0
direction : down
capacity : 10
count inside : 0
inside queue : []
```

```
0. floor => Queue : [[9, 3]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 9 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 17
```

```
mode : working
floor : 1
destination : 3
direction : up
capacity : 10
count inside : 6
inside queue : [[6, 3]]
```

```
|
0. floor => Queue : [[9, 3]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 9 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 17
```

```
floor : 2
destination : 3
direction : up
capacity : 10
count inside : 0
inside queue : []
```

```
0. floor => Queue : [[9, 3]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 9 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 17
```

```
mode : working
floor : 2
destination : 0
direction : down
capacity : 10
count inside : 4
inside queue : [[4, 0]]
```

```
0. floor => Queue : [[9, 3]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 9 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 17
```

```
mode : working
floor : 1
destination : 0
direction : down
capacity : 10
count inside : 0
inside queue : []
```

```
AVMEntrance => [4, 4]
```

```

0. floor => Queue : [[4, 4]]
1. floor => all : 19 Queue Count : 0 Queue : []
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 15 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 21

mode : working
floor : 1
destination : 3
direction : up
capacity : 10
count inside : 9
inside queue : [[9, 3]]

AVMExit => [1, 0]
1. floor => all : 19 Queue Count : 1 Queue : [[1, 0]]

AVMEntrance => [1, 3]

0. floor => Queue : [[4, 4], [1, 3]]
1. floor => all : 19 Queue Count : 1 Queue : [[1, 0]]
2. floor => all : 5 Queue Count : 0 Queue : []
3. floor => all : 15 Queue Count : 0 Queue : []
4. floor => all : 11 Queue Count : 0 Queue : []
Exit Count : 21

mode : working
floor : 1
destination : 0
direction : down
capacity : 10
count inside : 0
inside queue : []

```

6-)Yalancı Kod

- Başla.
- Floor yapılarını yarat.
- AVMEntrance, AVMExit, ElevatorController threadlerini yarat ve başlat.
- AVMEntrance threadi 500ms aralıklarla giriş katına 1-10 aralığında insanı ve gidecekleri katı 1-4 arasında belli eden bir grup insanı ekle.
- ElevatorController threadi kendi içinde bir tanesi sürekli çalışmak kaydıyla 5 farklı Elevator oluştur. Tüm katlarda çıkışa gitmek veya üst katlara ulaşmak için sıra bekleyen toplam insan sayısının 20'yi aşması durumunda bir adet Elevator'u aktifleştir, bu sıralardaki toplam insan sayısı 10'un altına indiyse en son aktive edilen Elevator'un çalışmasını duraklat.
- Elevator threadleri çalışmaya başladığında giriş katında sırada bekleyen gruplardan kendi kapasitelerini aşmayacak şekilde insanları içine al. Daha sonra içindeki grupları sırayla ilgili hedef katlara giderek onları o kata naklet. Daha sonra giriş katı harici diğer katlarda aşağı inmek isteyen bir grup insan olup olmadığının kontrolünü yaparak eğer inmek isteyen grup varsa ilgili grubun bulunduğu kata giderek onları Elevator kapasitesini aşmayacak şekilde içine al. En sonunda da giriş katına inerek içindeki yolcuları boşalt.

- Elevator yapıları her kat çıkışında, inişinde, yolcu alışında, yolcu bırakışında, katları kontrol edişinde, giriş katından içine alabileceği insanları kontrol ederken katları ve kendi bilgilerini ekrana yazdır.
- AVMExit threadi giriş katı harici içerisinde insan bulunan katlardan 1 ila 5 arasında bir grup insanı o katın çıkmak isteyen insanları tutan sırasına ekle.
- Program IDE'den kapatıldığında programı sonlandır, sonlanmadıysa sonsuza kadar çalış.

7-)Kaynakça

- Sıfırdan İleri Seviyeye Uygulamalı Java Kursu - Emre Altunbilek - Udemy
- <https://ufukuzun.wordpress.com/2015/05/04/javada-multithreading-bolum-8-beklet-ve-bildir-wait-and-notify/>
- <https://www.geeksforgeeks.org/arrayblockingqueue-class-in-java/>
- <https://howtodoinjava.com/java/java-misc/java-tuples/>
- <https://www.techiedelight.com/iterate-through-queue-java/>
- <https://www.javatpoint.com/java-thread-interrupt-method>

- <https://medium.com/@yusufcanelik/javada-executorservice-896e85eee2bc>
- <https://www.baeldung.com/java-executor-service-tutorial>
- https://www3.ntu.edu.sg/home/ehchua/programming/java/j5e_multithreading.html
- <https://stackoverflow.com/questions/49946623/java-multithreading-run-threads-run-method-only-once-in-sequence>
- <https://crunchify.com/how-to-run-multiple-threads-concurrently-in-java-executorservice-approach/>
- <https://www.geeksforgeeks.org/killing-threads-in-java/#:~:text=Modern%20ways%20to%20suspend%2Fstop,will%20be%20set%20to%20true.>
- <https://stackoverflow.com/questions/1881714/how-to-start-stop-restart-a-thread-in-java#:~:text=Once%20a%20thread%20stops%20you,it%20to%20do%20work%20again.>
- <https://stackoverflow.com/questions/35461897/stop-and-restart-a-already-running-thread>

- <https://www.baeldung.com/java-concurrentmodificationexception>
- <https://www.lihaoyi.com/post/BuildeyourOwnCommandLinewithANSIEscapeCodes.html>
- <https://stackoverflow.com/questions/4842424/list-of-ansi-color-escape-sequences>