**ASANSÖRLERDEKİ TALEP YOĞUNLUĞUNUN MULTITHREAD İLE KONTROLÜ PROJESİ**

Derya ÖZGÜ, Mehmet KIYAK

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[180201115@kocaeli.edu.tr](mailto:180201115@kocaeli.edu.tr) , [180201135@kocaeli.edu.tr](mailto:180201135@kocaeli.edu.tr)

Özet

*Projede bir alışveriş merkezinde ki asansörlerin talep yoğunluğunun multithread ile kontrolü yapılmaktadır. Alışveriş merkezinde beş adet asansör bulunmaktadır. Birinci asansör sürekli çalışmakta olup diğer asansörler ise yoğunluğa göre aktif veya pasif durumu geçmektedir.*

*Program kullanıcı tarafından durdurulmaz ise veya hata olmadığı sürece sürekli alışveriş merkezine birilerinin girip çıkması şeklinde olup asansörler sürekli çalışarak hizmet vermektedir. Anlık olarak ekrana asansörlere ve katlara ait bilgiler yazılmaktadır.*

# Giriş

Program ilk çalışma esnasında giriş ve çıkış sınıflarından üretilen objeler ile çalışmaya başlamaktadır. Giriş thread sürekli olarak 500ms (milisaniye) de bir alışveriş merkezine kişi ve bu kişilerin gideceği hedef katları üretmektedir.

Çıkış thread ise sürekli olarak her 1000ms (milisaniye) de bir çıkışa gidecek olan kişi ve bu kişilerin hangi katlardan ineceğini rastgele olarak üretmektedir. Ve belirtilen sıraya göre asansörler hizmet vermektedir.

# Temel Bilgiler

Program Java programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak “NetBeans IDE 8.2” kullanılmıştır. Akış ve UML diyagramı için [draw.io](http://draw.io) online diyagram yapma programı kullanılmıştır

# Tasarım

Asansörlerdeki Talep Yoğunluğunun Multithread İle Kontrolü programının programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

# Algoritma

Program ilk çalışma esnasında giriş ve çıkış sınıflarından üretilen objeler ile çalışmaya başlamaktadır.

Giriş thread sürekli olarak 500ms (milisaniye) de bir alışveriş merkezine kişi ve bu kişilerin gideceği hedef katları üretmektedir.

Çıkış thread ise sürekli olarak her 1000ms (milisaniye) de bir çıkışa gidecek olan kişi ve bu kişilerin hangi katlardan ineceğini rastgele olarak üretmektedir.

Sonrasında kontrol sınıfından beş adet kontrol objesi üreterek katlardaki kişi sayısını ve asansörlerdeki yoğunluğa göre asansörleri aktif veya pasif duruma getirmektedir. Anlık olarak katlardaki ve asansörlerdeki durum bilgilerini ekrana yazmaktadır.

# Kullanılan Sınıflar ve Fonksiyonlar

* + 1. **Main.java :** Programın başlamasını sağlayan sınıftır. Giriş ve çıkış threadleri başta olmak üzere kontrol ve asansör threadlerimizi oluşturup daha sonra çalıştırmaktadır. Yapılan işlemleri ekrana yazmaktadır.
    2. **Giris.java:** Bu sınıfiçerisinde giriş threadimizi çalıştırıyor ve her 500ms’de bir yukarıya çıkacak kişi ve hangi kata çıkacağını rastgele olarak üreterek main sınıfında yer alan yön sınıfından türettiğimiz yukarı dizisi içerisine eklemektedir.   
       1. **run:** run metodunu sınıfımızı extends ettiğimiz Thread sınıfından override ederek giriş sınıfı içerisinde kendimize göre restore edip her 500ms de bir oluşturulan kişi ve hedef kat değerini dizilerimize aktarma işlemi yaptırmasını sağladık.
    3. **Cikis.java:** Bu sınıfiçerisinde çıkış threadimizi çalıştırıyor ve her 1000ms’de bir aşağıya inecek kişi ve hangi kattan ineceğini rastgele olarak üreterek main sınıfında yer alan yön sınıfından türettiğimiz aşağı dizisi içerisine eklemektedir.  
       1. **run:** run metodunu sınıfımızı extends ettiğimiz Thread sınıfından override ederek çıkış sınıfı içerisinde kendimize göre restore edip her 1000ms de bir oluşturulan kişi ve inilecek kat değerini dizilerimize aktarma işlemi yaptırmasını sağladık.
    4. **Asansor.java:** Main sınıfı içerisinde bu sınıftan türeterek beş adet asansör üretiyoruz. Daha sonra bunları main içerisinde asansör sınıfından türeterek belirttiğimiz asansör dizisinde gerekli objeleri tutuyoruz.   
         
       Bu sınıf içerisinde asansörlerimizin durumlarına göre her 200ms de bir yukarı veya aşağı katlara kişileri götürmesini sağlayan run metodu çalışmaktadır.  
       1. **run:** Asansörümüzün durumuna göre gideceği kata göre yukarı veya aşağı inmesi işlemini yapmakta olup ayrıca katlarda ki kişilerin genel toplam değerlerini güncellemektedir. Bu işlemi her 200ms de bir içerisinden üretilen her asansöre göre yapmaktadır.
    5. **Kontrol.java:** Main içerisinde bu sınıftan beş adet obje üretip yine main içerisinde bu sınıftan türeterek oluşturduğumuz dizi içerisine ekliyoruz. Bu sınıf içerisinde her 150ms de bir asansörlerimiz katlara göre yoğunluğunu kontrol ederek asansörlerimizin durumlarını aktif veya pasif olarak değiştirmekteyiz.  
       1. **run:** Her 150ms’de bir katlara göre asansörlerimizin durumlarını güncelleyerek hizmetin doğruluğunu sağlamaktadır.
    6. **Yon.java:** Bu sınıfı yukarıya veya aşağıya gidecek kişilerin sayılarını ve hangi kata gideceğini bilgisini tutmak için oluşturduk.

# Sonuçlar

Programımız proje kapsamında istenilen tüm isterleri yerine getirmektedir. Program da sadece bir hata mevcuttur hatanın nedenini çözemediğimizden tamamlayamadık.

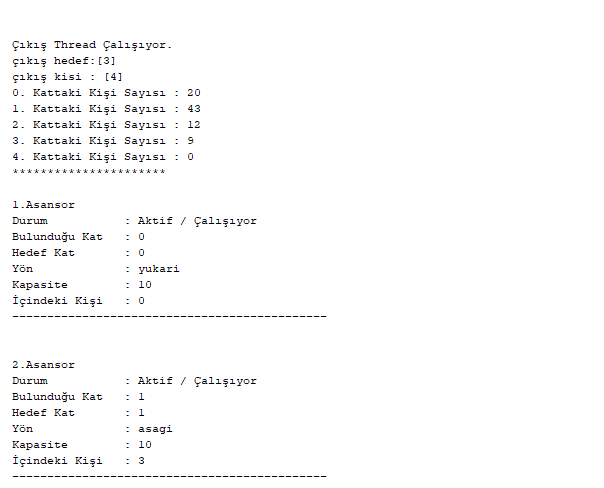
# Kaynakça

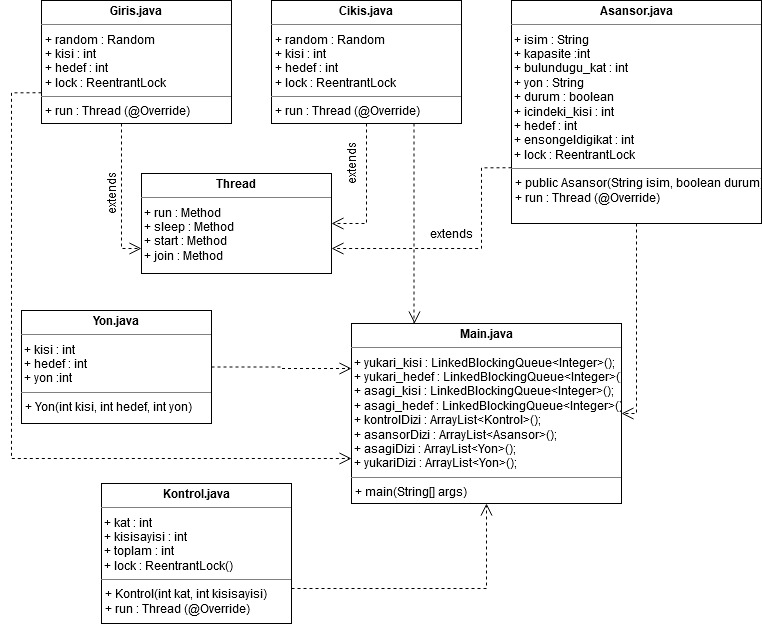
1. <https://www.udemy.com/course-dashboard-redirect/?course_id=1421116> kaynağından java diline ait gerekli terimler ve multithread işlemleri hakkında bilgiler edinildi.
2. <https://www.tutorialspoint.com/java/java_multithreading.htm> kaynağından karşılaştığımız bazı hatalara karşı çözümler elde edildi.

# 6. Denklemler

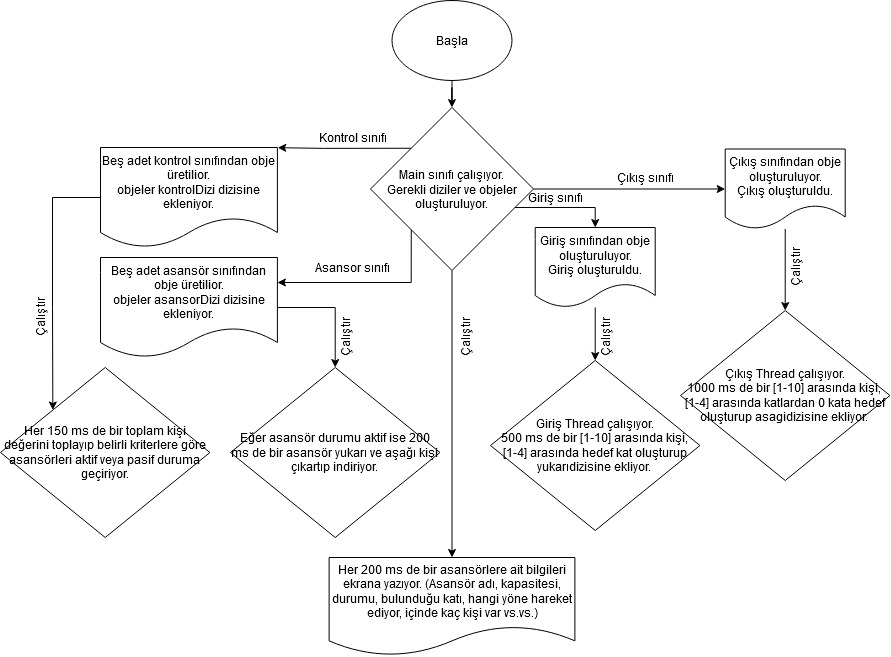
Program içerisinde matematiksel olarak çok büyük hesaplama bulunmadığı üzere küçük bazı denklemler kullanılmıştır.

# 7. Ekran Görüntüleri





**Resim.1 : UML Diyagramı**



**Resim.2 : Akış Diyagramı Diyagramı**