

# ROBOT PROJESİ

KOCAELİ ÜNİVERSİTESİ  
Bilgisayar Mühendisliği Bölümü  
Mutluhan Boz  
mutluhanb2@gmail.com

## Özet

*Programda kullanıcı 6 farklı robot nesnesinden istediğini oluşturup oluşturduğu robot nesnesini 20x20 ızgara üzerinde dilediği kordinatlara yerleştirmektedir.*

*Yerleştirme işleminden sonra robotların özelliklerine göre proje dökümanında istenen 3 problemten ilgili olanı çözülmektedir.*

*Çözülen problemler ana hatlarıyla robotun hareket etmesi ve robotun üzerine yüklenen yükü verilen noktaya taşınması üzerine kuruludur.*

*Problemlerin çözümü sırasında robotun ve/veya yükün hareketi 20x20 ızgara üzerinde görselleştirilmektedir.*

*Problemler sonucunda yapılan işlemler sırasında geçen süre ekrana yazdırılmaktadır.*

## 1. Giriş

Robot Simülasyon programı kullanıcıya sunulan robot tiplerinden bir tanesini oluşturma ve oluşturulan robota göre ilgili problemi çözme imkanı sağlar. Robotların oluşturulmasında class yapıları ve kalıtım kullanıldığından robotların ortak özellikleri üst sınıflarından miras alınarak oluşturulmaktadır.

Problemin çözümünde inheritance / abstraction / polymorphism gibi OOP konseptleri kullanılmıştır.

İlgili problemlerin çözümü sırasında problem çözüm ortamı ve bileşenler (robot / yük / ızgara) grafik kütüphanesi kullanılarak görselleştirilmiştir.

## 2. Temel Bilgiler

Program Java dili (JDK 9) kullanılarak geliştirilmiştir. Arayüz tasarımında Swing kütüphanesinden yararlanılmıştır.

Tüm robot sınıflarının atası olarak “robot” sınıfı oluşturulmuş, ardından bu sınıf abstract class ve interface yapıları ile gezgin ve manipülatör robot sınıflarına ayrılmıştır. Hibrit robot sınıfı için ise manipülatör robot interfacei implement edilip gezgin robot sınıfı extend edilmesi ile multiple inheritance gerçekleştirilmiştir.

Problemler görselleştirilirken robotlar yeşil, engeller kırmızı, yükler ise siyah kareler ile modellenmiştir.

## 3. Yöntem

### 3.1 Genel Yapı

Program ilk olarak kullanıcıdan kaç adet robot oluşturulacağı bilgisini aldıktan sonra kullanıcıya girilen değer kadar robot yaratma imkanı sunmaktadır. Oluşturulan robotlar tüm özelliklerinin okunmasının ardından tüm sınıfların atası olan Object sınıfı tipinden bir array listte tutulmaktadır.

Ardından kullanıcıdan kaçınıcı robotu

seçmek istediği bilgisi okunur okunan değere göre listeden ilgili sıradaki nesne çağırılır, instanceof operatörü ile çağırılan objenin hangi robot sınıfına ait olduğu belirlenir ve ilgili sınıfa cast edilmesinin ardından ilgili problemin çözümüne geçilir. Problem

gerektiriyorsa engell miktarı ve kordinatları kullanıcıdan okunur ve robot ızgarada belirtilen yere yerleştirilir.

Ardından yön bilgisi bilgisi kullanıcıdan okunur ve vector içerisinde tutulur. İlerletme işlemi sırasında bu vector içerisinde elemanlar ikişerli olarak okunarak ilerletme veya taşıma işlemi yapılır.

Son adımda süre matematiksel işlemler yapılarak hesaplanır ve kullanıcıya bildirilir.

### 3.2 Class Yapıları

Tüm robot nesneleri için robot sınıfı ata sınıftır. Kordinat bilgileri ve motor sayısı gibi ortak değişkenler bu sınıfta bulunur.

Gezgin robotlar için gezgin robot soyut sınıf oluşturulmuş ve gezgin robotlara özgü değişkenler burada tanımlanmıştır.

Tekerlekli, Paletli ve spider tipi robot sınıfları gezgin robot soyut sınıfından türetilmiştir.

Manipülator robotlar için ek bir soyut sınıf daha tanımlanmış ancak ilgili method ve değişkenleri hibrid robot sınıfı düşünülerek oluşturulan ek bir manipülator robot interfacei içerisinde tanımlanmıştır.

Gezgin robot soyut sınıfı ve manipülator robot interfacei yardımı ile hibrit robot sınıfı oluşturulmuştur.

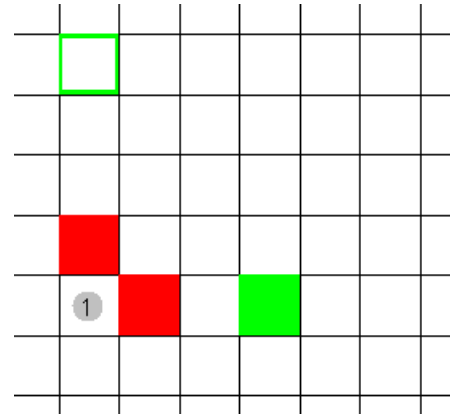
## 4.Deneysel Sonuçlar

### 4.1 Tekerlekli Robot 2 Engel

6 Motorlu ve gezinme hızı 5 m/s olan tekerlekli robot 5,5 kordinatlarına yerleştirilmiş ardından sırası ile 4 ileri , 3 sağ komutları girilmiştir. 5,8 ve 6,9 kordinatlarına kullanıcı tarafından engeller program akışının önceki adımlarında yerleştirilmiş durumdadır.

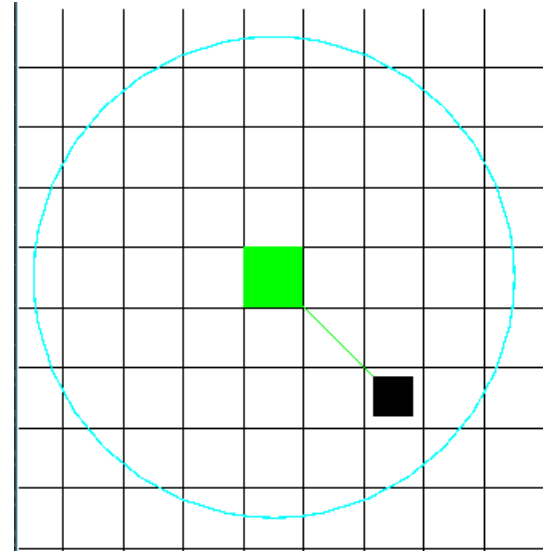
Robotun ilgili yönlere sırası ile ilerlediği engellerden doğru sürede geçtiği ve toplam süre hesaplamasını doğru yaptığı konsol üzerinde görülmüştür.

Izgara üzerinde durum görseldeki gibidir



### 4.2 Seri Robotun Yük Taşımaları

Kol uzunluğu 40 metre olan seri robotun yükü sırasıyla 2 ileri 2 sağ yönünde taşımaları sonucu konsolda doğru sürelerin yazdırıldığı görülmüştür.



### 4.3 DİĞER DURUMLAR

Spider robotların engele çarpması,  
Gezgin robotların ızgaradan çıkması,  
Manipülatör robotların kolunun  
yetişemeyeceği yere taşıma işlemi  
yapılması gibi kontrol gerektiren  
durumlarda gerekli uyarıların verilip  
işlemlerin kesilme işlemleri yaptırılmıştır.

## 5. SONUÇ

Programımız isterleri eksiklik olarak  
karşılacaktır. İstenilen robotlar  
oluşturulurken projede belirtilen sınıf içi  
hiyerarşilerin kontrol edildiği. Çoklu  
kalıtımın interface desteği ile sağlandığı  
ilgili problemleri doğru bir şekilde  
çözüldüğü ve görsel arayüzün doğru bir  
şekilde tasarlandığı gözlenmiştir.

Proje verildikten sonra girdiğim  
“Nesneye yönelik programlama”  
derslerinin sonucunda sınıf yapısının ve  
programın genelinin çeşitli OOP  
konseptlerinin etkin kullanımı ile daha  
düzenli ve toplu halde yazılabileceği  
tarafımda fark edilmiş olsa bile temel OOP  
konseptlerinin etkin kullanımının yazılım  
projelerinde sağladığı imkanlar proje  
genelinde kullanılmaya çalışılmıştır.

## 6.Kaba Kod

- 0-)Başla.
- 1-)Oluşturulacak robot sayısını oku.
- 2-)Okunan sayı kadar istenilen tipte robot  
oluştur ve özelliklerini oku
- 3-)Robotu seçtir.
- 4-)Robotun tipine göre ilgili probleme geç.

- 5-)Problem gerektiriyorsa engel sayısını ve
- 6-)Robotun yerleştirileceği kordinatı oku.
- 7-)Robotun / Yükün hareket talimatlarını  
oku.
- 8-)Robotun / yükün son konumunu göster.
- 9-)Hibrit robot ise ikinci aşamaya geç ve  
tekrar yön oku ve hesaplamaları tekrarla.

## 7.KAYNAKÇA

- I. Java Swing Draw Rectangle  
[http://www.java2s.com/Tutorial/Ja  
va/0261\\_\\_2D-  
Graphics/DrawRectangle.htm](http://www.java2s.com/Tutorial/Ja<br/>va/0261__2D-<br/>Graphics/DrawRectangle.htm)
- II. Java Swing Draw Line  
  
[https://stackoverflow.com/questions/1  
0767265/drawing-a-line-on-a-jframe](https://stackoverflow.com/questions/1<br/>0767265/drawing-a-line-on-a-jframe)
- III. Casting and runtime type checking  
  
[http://resources.esri.com/help/9.3/arcg  
isengine/java/doc/7c061b6b-7317-  
4d53-aa15-a9c0a43f278b.htm](http://resources.esri.com/help/9.3/arcg<br/>isengine/java/doc/7c061b6b-7317-<br/>4d53-aa15-a9c0a43f278b.htm)

## 8. UML Diyagramı

