**PROGRAMLAMA LABARATUVARI 3.PROJE**

**170201043-İLKER TINKIR, 170201075-ALİ EKEN**

**ÖZET:**

Bu projede bizimle paylaşılan isterler doğrultusunda aralarında bir hiyerarşi bulunan robot türleri nesneleri tarafından çözülmesi beklenen; gezgin robotun hareketi, manipulator robotun yük taşıması, hibrit robotun hareketinden sonra yükü farklı bir noktaya taşıması şeklindeki üç problemi çözen kodu yazdık.

**GİRİŞ:**

Oluşturduğumuz projeyi temel olarak hem kod yazma kolaylığı açısından hem de okunabilirlik açısından daha kolay anlaşılabilir hale getirmek için pek çok farklı sınıfa ve arayüze böldük. Projede belirtildiği üzere hibrit robot nesnesinde çoklu kalıtım özelliği gerektiği için pek çok özellik ve fonksiyon barındıran robot, gezgin robot, manipulator robot kısımlarını interface olarak tanımladık. Interfaceler yapıları gereği metodlar dışında sabit ve değişken içeremedikleri için bu özellikleri gettter ve setter metodlar yardımıyla daha alt sınıflarda almak yoluna gittik. Hibrit robot nesnesi bunun dışında tekerlekli robot, paletli robot, spider robot nesnelerinden de özellikler barındırdığı ve proje içinde bu kısımların sınıf olarak tanımlanması gerekmesi sebebiyle bu sınıfların kendilerine has yani üst sınıf ve interfacelerden kalıtılmayan özelliklerini içeren “IGezginPaletli”, “IGezginTekerlekli”, “IGezginSpider” adlı ayrı interfaceler oluşturduk ve bu interfaceleri hibrit robot sınıfına kalıttık. Ancak seri ve paralel robot nesnelerinin yukarıda sayılan diğer alt sınıflar gibi kendilerine has yani üst sınıflardan

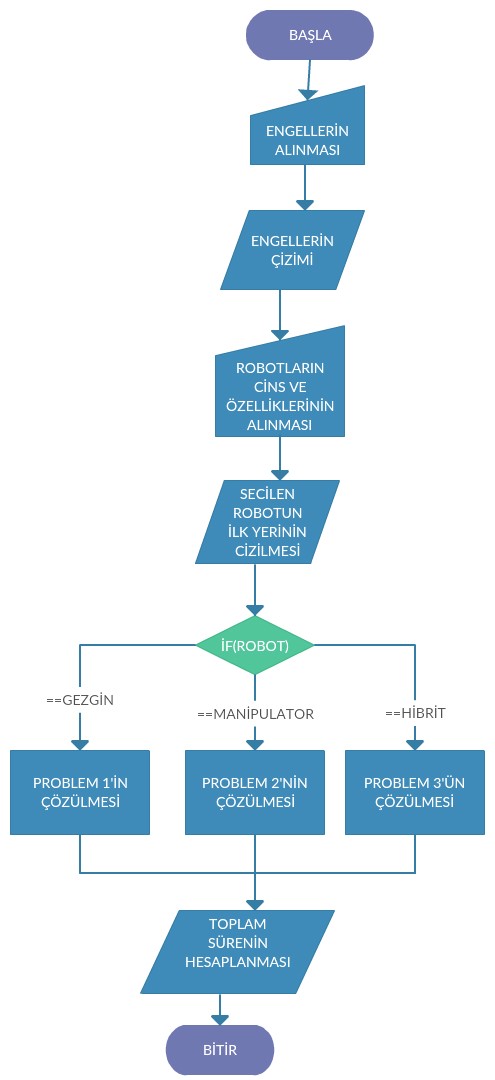
kalıtılmayan özellikleri olmadığı için bu adda interfaceler oluşturma gereği duymadık.

Hibrit robot nesnesi dışında diğer gezgin robot interfacesinin alt sınıfları sınıf olarak tanımlanmıştır. Projede problemlerin çözülmesi için birden fazla robotun aynı zamanda oluşturulması ve istenildiği zaman istenilen indexteki robot nesnesine ulaşılmak istendiği için Main fonksiyonu içerisinde robot interfacesi tipinde bir ArrayList oluşturduk. Bu ArrayList robot tipinde olmasına rağmen gezgin, manipulator, hibrit robotların alt sınıflarının nesnelerini içeren bir yapıda tanımlandı. Bu yaptığımız hareket ile nesneye yönelimli programlamanın Polymorphism özelliğini kullanmış olduk.

Verilen problemlerin çözümlerinin görselleştirilmesi amacıyla çizim yapılacak Engeller1, Zemin, RobotMoving, RobotilkYer, YukMoving adlarında ayrı sınıflar oluşturduk. Bu sınıflar Javanın JComponent sınıfına extends edilmiştir. Robot, yuk ve engel çizimlerinin yapılması için bahsedilen JComponent sınıfının “paint” metodu override edilerek uygun biçimde yeniden yazılmıştır.

İsterlerde bahsedilen tekerlekli, paletli, spider robotlarının gezinme hızı; seri ve paralel robotların taşıma hızı özellikleri Mainde tanımlı ArrayListlere kaydedilmiş ve minimum hız değişkeni girilen verilere göre belirlenerek bu ArrayListlerin kendi içerlerinde karşılaştırma işlemleri yapılmıştır.

**AKIŞ ŞEMASI:**



**YÖNTEM:**

Bu projede karışıklıkları önlemek amacıyla çizim yapılan paint metodu içinde sadece diğer çizim fonksiyonlarını kullandık. Bunu uygulamamızın sebebi ise paint metodunun yapısal olarak içerisine gönderilen global değişkenleri algılayamaması ve bunlara göre işlem yapamamasıdır.

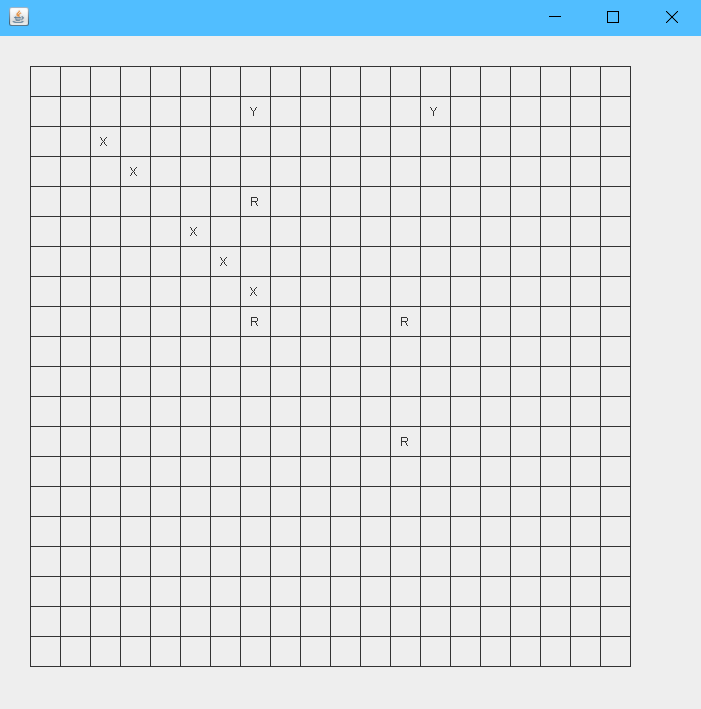
Bunun dışında hibrit robot sınıfının kendisi için gerekli bütün özellikleri içermesi amacıyla diğer alt sınıfların birer interface kopyalarını oluşturduk ve bu interfaceleri gezgin robot ve manipulator robot interfaceleri ile beraber hibrit robot sınıfına kalıttık.

Projede birden fazla robot olabileceği için bu robotlara indexleriyle ulaşmak amacıyla robot tipinde bir Arraylist oluşturduk ve farklı türlerdeki robotları robot tipindeki bu ArrayListe ekleyerek Polymorphsim yapısını kullanmış olduk.

**YAPILAN ARAŞTIRMALAR:**

Projede görselleştirilmesi gereken yerlerin çizimi için gerekli olan paint metodunu ve JComponent sınıfının özelliklerini çeşitli kaynaklardan araştırdık. Projede çoklu kalıtım gerektiği ve javanın çoklu kalıtımı desteklememesi sebebiyle çoklu kalıtımın nasıl yapılabileceği üzerine araştırmalar yaptık. Projeye başladığımızda daha interface yapısını okulda öğrenmemiş olduğumuz için interfacenin nasıl kullanıldığı üzerine araştırmalarda bulunduk. Polymorphsim yapısını projemizde kullanabilmek amacıyla internet üzerindeki faklı örnek kodları inceledik.

**ÖRNEK SONUÇLAR:**



Örnek hibrit robotun hareketi:

1)Engeller (3,3),(4,4),(6,6),(7,7),(8,8) koordinatlarına yerleştirildi.

2)Robot (13,13) koordinatına yerleştirildi.

3)Robot 4 birim yukarı,5 birim sola ve 4 birim yukarı hareket ettirildi,bu sırada (8,8) konumundaki engeli aşarak (8,5) konumuna getirildi.

4)Robot son konumunda sabitlendi ve yük taşıyabilecek duruma geldi.

5)Robot kolundaki yük 3 birim yukarı, 6 birim sağa taşındı.

6)Robotun bütün hareketi tamamlandı ve hareket boyunca geçen süre ekrana bastırıldı.

**KAYNAKÇA:**

1. <http://www.java2s.com/Tutorial/Java/0261__2D-Graphics/Drawaline.htm>
2. <https://creately.com/>
3. <https://www.tutorialspoint.com/java/java_interfaces.htm>
4. <https://docs.oracle.com/javase/tutorial/java/IandI/polymorphism.html>
5. <http://bilgisayarkavramlari.sadievrenseker.com/2011/03/22/java-dilinde-coklu-kalitim-miras/>

