

# Araç Yönetim Sistemi

Ata GÜLALAN - Oğuzhan TÜRKER

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[xavanco@gmail.com](mailto:xavanco@gmail.com) oguzturker8@gmail.com

## Özet

*Araç Yönetim Sistemi, araç sınıflarının kullanıcının belirlediği özelliklerde oluşturulup, düzenlenmesini ve birbirleri arasındaki iletişimini, kolay ve anlaşılabilir bir arayüz ile kullanıcıya sunduğu bir hizmettir.*

*Kullanıcı istediği araç türünü oluşturup, onu düzenleyebilir. Oluşturulan sınıfları listeleyip, güncelleyebilir.*

## 1. Giriş

Araç Yönetim Sistemi, aralarında hiyerarşik bir bağ bulunan, birbirleri ile ilişkilendirilmiş nesnelerin, belli başlı yöntemler ışığında (Encapsulation, Inheritance, Polymorphism, Abstraction) düzenlenmesini sağlar.

Kullanıcı dostu arayüz ile birlikte, uygulama kullanıcısı kolay bir şekilde programı kullanabilir. Oluşturduğu sınıfları düzenleyebilir. Ve projenin bizlerden beklediği tüm istekleri karşılayabilir..

## 2. Temel Bilgiler

Proje gelişiminde Tümlşik Geliştirme Ortamı olarak “Jetbrains IntelliJ IDEA” kullanılmıştır.

## 3. Tasarım

Proje aşağıdaki başlıklar altında geliştirilmiştir.

### 3.1. Hiyerarşik Yapı

Araç Yönetim Sisteminde bulunan oluşturulabilir, düzenlenebilir temel beş sınıf bulunmaktadır. Beş temel sınıfın birbirleri ile olan ortak özelliklerini belirleyen üç sınıf (en az biri interface) vardır. Tüm sınıfların ortak özelliklerini içeren temel ata “Araç” sınıfı bulunmaktadır.

Üzerlerinde işlem yapılabilen “Otomobil, Bisiklet, Gemi, Uçak ve Uçan Gemi” sınıfları, bu sınıfların ortak özelliklerini oluşturan “Kara Taşıtları, Deniz Taşıtları (Interface) ve Hava Taşıtları” ve tüm sınıfların atası olan

“Araç” sınıfı aşağıda şekillendirilmiş hiyerarşik yapıyı (Resim 6.1) kullanmaktadır.

### 3.2. Yöntemler

Projede kullanılan fonksiyonlar aşağıda açıklanmıştır.

Tara;

SınıfID’si girilen araç türünü metin belgesinde bulup, o sınıfın mevcut bilgilerini arayüze değiştirilmek adına gösterir.

Edit;

Değiştirilmek istenilen sınıf Tara fonksiyonunda bulunduktan sonra değiştirilmek istenilen bilgiler arayüze yazılır. Güncelleme butonuna tıklanıldığında Edit fonksiyonu o sınıfın bulunduğu satırı yeni bilgilerle günceller.

Oku;

Veri girişi yapılırken belirlenen sınıf metin belgesine kayıt edilmeden önce o sınıfın sayısı hesaplanır ve bir sonraki SınıfID’si yeni veri girişe döndürülür.

KaraListe;

Metin belgesindeki tüm kara taşıt araçlarını bulup arayüze döndürür.

DenizListe;

Metin belgesindeki tüm deniz taşıt araçlarını bulup arayüze döndürür.

HavaListe;

Metin belgesindeki tüm hava taşıt araçlarını bulup arayüze döndürür.

### 3.3. Sınıfların Özellikleri

Araç (Interface) ;

Tüm sınıfların atasıdır.

Ortak Get ve Set methodları burada bodysiz tanımlanmıştır.

Kara Taşıtları (Abstract Class) ;

Otomobil ve bisiklet sınıflarının atasıdır.

Otomobil ve bisiklet sınıflarının ortak değişkenleri ve fonksiyonların bodyleri burada tanımlanmıştır.

Deniz Taşıtları (Interface) ;  
Gemi ve uçan gemi sınıflarının atasıdır.  
Ortak fonksiyonlar bodysiz tanımlanmıştır.

Hava Taşıtları (Class) ;  
Uçak ve uçan gemi sınıflarının atasıdır.  
Uçak ve uçan gemi sınıflarının ortak değişkenleri ve fonksiyonların bodyleri tanımlanmıştır.

Otomobil (Class) ;  
Kara taşıtlarından miras aldığı fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

Bisiklet (Class) ;  
Kara taşıtlarından miras aldığı fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

‘Yakıt Türü’ değişkeni “YOK” olarak tanımlanmıştır.

Gemi (Class) ;  
Deniz taşıtları interface olduğundan, gemi sınıfı içerisinde miras aldığı bodysiz fonksiyonlar ve geminin değişkenleri tanımlanmıştır  
‘Yakıt türü’ değişkeni “YOK” olarak tanımlanmıştır.  
‘Teker Sayısı’ değişkeni “YOK” olarak tanımlanmıştır.

Uçak (Class) ;  
Hava taşıtlarından miras aldığı fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.  
Yakıt türü değişkeni “YOK” olarak tanımlanmıştır.

Uçan Gemi (Class) ;  
Hava ve deniz taşıtlarından miras aldığı fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.  
Yakıt türü değişkeni “YOK” olarak tanımlanmıştır.

### 3.4. Sınıf ID

Otomobil, bisiklet, gemi, uçak ve uçan gemi sınıfları aşağıda gösterilen anahtar kelimeler ile listelendirilmişlerdir.

1. Otomobil : “OTO\_ID” ;
2. Bisiklet : “BİS\_ID” ;
3. Gemi : “GEM\_ID” ;
4. Uçak : “UCK\_ID” ;
5. Uçan Gemi : “UGM\_ID” ;

### 3.5. Arayüz

Arayüz veri girişi (Resim 6.2), veri listeleme ve veri güncelleme işlemlerinin her biri ve seçim ekranı olmak üzere toplam 4 çerçeve (Frame) ve 4 panel içermektedir. Her yapılan işlem sonrası ilgili çerçeve kapanarak bir sonraki işlemin çerçevesi açılmaktadır. Buton, yazı alanı gibi arayüzde bulunan elemanlar o işlemin paneline eklenmiştir. O işlemin paneli de kendi çerçevesine eklenilip arayüz tasarlanmıştır.

## 4. Karşılaşılan Sorunlar ve Çözüm Yöntemleri

### 1. Sınıf Bilgilerinin Arayüze Aktarımı

Kayıt edilen nesnelerin filtreleneip arayüzde gösterilmesi gerekiyordu. Fakat okunan bilgilerin dinamik bir şekilde ekranda gözükmesi için gerekli Swing veri tipini bilmiyorduk. Okunan bilgileri TextArea’ya gönderdik. Ardından dinamik olması adına TextArea’yu ScrollPane’e gönderdik.

### 2. Sınıf ID Hesaplanması

Sınıf ID her oluşturulan nesnenin yapıcısında 1 arttırılıyordu. Fakat programı tekrar çalıştırdığımızda bu ID sıfırlanıyordu. Bunu çözmek adına program çalıştırıldığında her sınıf oluşturulduktan sonra Sınıf ID 1 arttırdık. Ardından metin dosyasındaki o sınıf türündeki tüm nesneleri okuyarak toplam nesne sayısını fonksiyon ile bulduk. Bulduğumuz sayıyı da yapıcı içinde artan Sınıf ID’sine ekledik.

### 3. Dosyaya Yazdırmadan Önce Fonksiyonlar

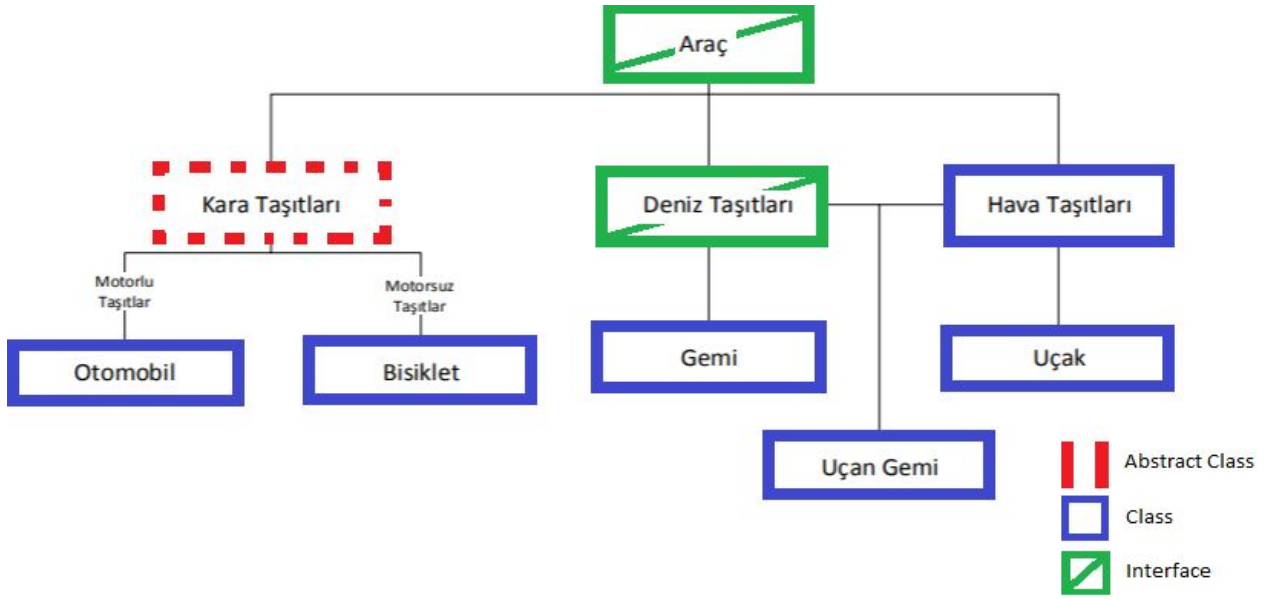
İlk hazırladığımız arayüz istenilen sınıfın butonuna tıklanılmasıyla o sınıfı oluşturup, girilen değerleri direk metin dosyasına yazdırıyordu. Yavaşlama, hızlanma ve durma fonksiyonlarının kayıt öncesi kullanımının istenilmesinin ardından, tekrar veri girişi arayüzü oluşturduk.

## 5. Kazanımlar

Araç Yönetim Sistemi’nin bize kazandırdıkları;

1. Interface & Abstract class kullanımı
2. Çoklu kalıtım
3. Swing kütüphanesi
4. Arayüz tasarımı
5. Dosya okuma işlemleri
6. Dosya satırı düzenleme

## 6. Resimler

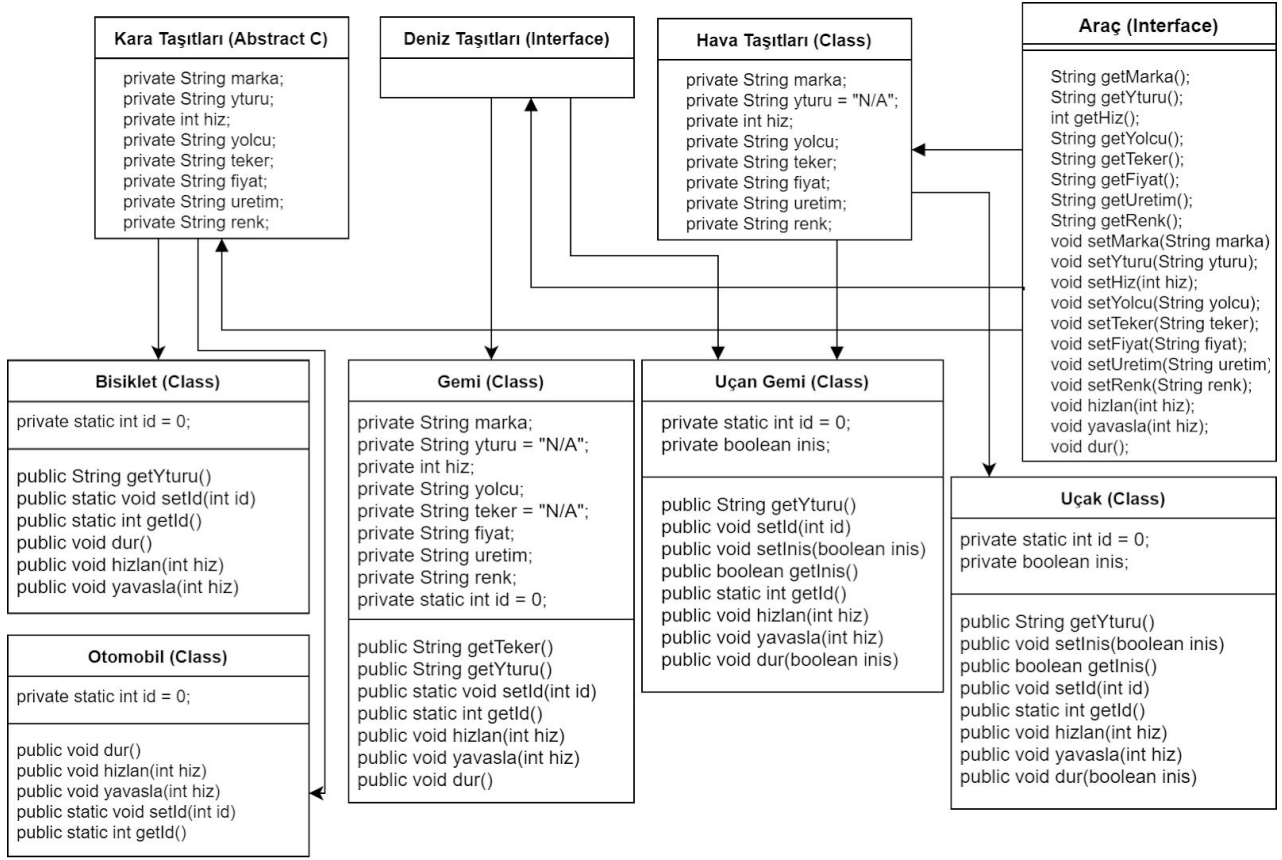


Şekil 1 : Sınıfların Hiyerarşisi

The screenshot shows a web application window titled 'Veri Girişi'. On the left, there are input fields for 'Marka', 'Yakıt Türü', 'Hız', 'Yolcu Sayısı', 'Teker Sayısı', 'Fiyat', 'Üretim Tarihi', and 'Renk'. In the center, there are five red buttons for selecting vehicle types: 'Otomobil', 'Bisiklet', 'Gemi', 'Uçak', and 'Uçan Gemi'. On the right, there is a 'Miktar' (Quantity) input field with the value '0', and four red buttons: 'Hızlan' (Accelerate), 'Yavaşla' (Decelerate), 'Dur' (Stop), and 'İniş(Hava A.)' (Landing (Air)). At the bottom right, there are two red buttons: 'Kayıt' (Register) and 'Geri' (Back).

Şekil 2 : Veri Girişi

## 7. UML Diyagramı



## 8. Kaynakça

- I. Java Swing JButton  
<https://stackoverflow.com/questions/423950/ro-unded-swing-jbutton-using-java>  
(Erişim Tarihi: 22.03.2018)
- II. Edit a spesific line from data in java.  
<https://www.youtube.com/watch?v=HFC-KspB914>  
(Erişim Tarihi: 23.03.2018)
- III. Center Graphics Drawstring  
<https://stackoverflow.com/questions/27706197/how-can-i-center-graphics-drawstring-in-java>  
(Erişim Tarihi: 23.03.2018)