**ARAÇ YÖNETİM SİSTEMİ**

Sema Demir Emre Sualp

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[sema.demir.1997@gmail.com](mailto:sema.demir.1997@gmail.com)

[esualp95@gmail.com](mailto:esualp95@gmail.com)

**Özet**

Nesneye yönelik programlama mantığı daha önceki projelerde oluşmuştu ama tam anlamıyla oturmamıştı. Hocalarımızın bunun farkına varmasıyla verdikleri bu proje, dilin tüm özelliklerini kullandıran açık ve kompleks yapıda olmayan bir proje olduğu için mantık tamamiyle oturdu. Javanın şu özelliğini kullanmayı bilmiyorum diyebileceğimiz bir nokta kalmadı.Projeyi oluştururken en çok düşündüğümüz kısım uçan gemi oldu. Hem deniz hem hava sınıfına ait özellikleri taşıyor olması fakat çoklu kalıtımı interface tanımlayarak yapabiliyor olmak bizi tatmin etmedi. Bu sebeple bir nevi yalancı bir çoklu kalıtım kullanmaktansa başka bir sınıf tanımlayıp kara ve havayı da oradan extends ettik. Bu şekilde olunca uçan gemi her ikisinin de özelliğini taşımış oldu.Fonksiyon tanımlarken hızlanıp yavaşlama konusu da kafamızı karıştırmıştı. E-destekte yapılan açıklama sonucunda fonksiyonlarımızı yazdık. Seri no'ları da tanımlamayı tam anlamıyla yaptıktan sonra metin işlemlerini de öğrendik. Böylece projemizi tamamlamış olduk.

**1.Giriş**

Yaptığımız bu projede bizden bir firmaya araç yönetim sistemi tasarlamamız istenmiştir. Firmanın istekleri arasında 3 tür araç bulunması vardır. Bu araçlar kara, hava, deniz sınıflarına aittir. Kara taşıtlarını da motorlu ve motorsuz olarak ayırmamız istenmiştir. Proje nesne tabanlı bir proje olduğu için kullanacağımız nesneye yönelik programlama dilinin tüm özelliklerini öğrenebilmemiz ve kullanabilmemiz adına Encapsulation, Inheritance, Polymorphism, Abstraction özelliklerinin hepsinin kullanımı istenmiştir. Bu taşıtların kendine has özellikleri olabileceği gibi ortak özellikleri de bulunmaktadır. Bu özellikleri her seferinde tekrar yazmaktansa kalıtım (inheritance) özelliğini kullandık. Inheritance ile polymorphism'i de birbirinden ayırmadık. Kapsülleme özelliğini kullanarak bazı özellikleri sınırladık böylece yanlış kullanımı engellemiş olduk. Bazı sınıflarımızı, özelliklerimizi abstract tanımlayarak abstaction özelliğini de kullandık.Tercih ettiğimiz dil derste de gördüğümüz ve daha hakim olduğumuzu düşündüğümüz Java dili oldu. Gerekli constructor ve isterleri de tanımladıktan sonra fonksiyonlarımızı da hazırladık ve en sonunda hepsini bir txt dosyasına kaydettik. Araç kaydedip sildiğimizde seri no'ların güncellenmesine dikkat ettik ve uçan gemi sınıfı için fazladan bir sınıf daha oluşturduk. Bu oluşturduğumuz sınıfa x adını verdik. Sınıfları kara ve x olarak ayırıp hava, deniz ve uçan gemiyi bu sınıftan extend ettik. Bu sayede uçan gemimiz hem havanın hem de denizin özelliklerini extend etmiş oldu.

**2.Temel Bilgiler**

Projede kullandığımız dil Java dilidir. Eclipse 3.8.1 idesinde projemizi geliştirdik. Dosyalama işlemleri için farklı kütüphane kullandık. 2 Hafta süren bir proje hazırlık, tamamlama, debug etme sürecimiz oldu ve bu süreçler sonucunda rapor hazırlanıp teslimi yapıldı.

**3.Diğer Bölümler**

**3.1 Arac sınıfı**

Aracların kullanacağı ortak özellikleri burada tanımladık. Set get metodlarımızı burada çağırdık. Hızlan, yavaşla, dur fonksiyonlarımızı burada tanımladık.

*3.1.1 Arac.txt sınıfı*

Dosya işlemlerini yapabilmek için bu sınıfı tasarladık. Tüm dosyalama işlemlerini bu sınıfta gerçekleştirdik.

*3.1.2 Menu sınıfı*

Yapmak istediğimiz işlemleri 1-Veri girişi 2-Veri Listeleme 3- Veri güncelleme olarak ayırdık.

Veri girişi işlemleri için menu1 isimli başka bir fonksiyon oluşturduk ve bu fonksiyon altında 5 çeşit araçtan seçmek istediğimizi seçme ve bir önceki menüye dönmeyle çıkış seçeneği olarak 7 farklı seçenek sunduk. Hangi tür araçsa ona göre id verdik. Listele isimli fonksiyonda da araç özelliklerini teker teker listeledik.Listelemek istediğimiz 3 tür için menu2 fonksiyonunu oluşturduk. Giris fonksiyonumuzla da özellikleri girdik.Güncellemek istediğimiz aracın id'sini girerek seçim yapıp aracı güncelledik. Guncelle ve menu3 sınıflarını bu iş için oluşturduk.

*3.1.3 Main sınıfı*

anaMenu'yü çalıştırarak projemizi çalıştırmış olduk.

**3.2 Kara sınıfı**

Public olarak tanımladık. Hızlan, yavaşla,dur fonksiyonlarımızı override ettik. Arac sınıfından extends ettik. Otomobil, bisiklet gibi kara taşıtlarının extends edileceği sınıfı tanımlamış olduk.

*3.2.1 Motorlu Sınıfı*

Motorlu olduğu için yakıt türü ekledik. Kara sınıfından extends ettik.

*3.2.2 Motorsuz Sınıfı*

Kara sınıfından extends ettik.

**3.3 havaDeniz sınıfı**

hava, deniz, ucanGemi sınıflarının kalıtım yapabilmesi için böyle bir ortak sınıf oluşturduk. Bu sınıfta hepsinin ortak Hızlan, yavasla, dur fonksiyonlarını override ettik. Abstract olarak tanımladık ve arac sınıfından extends ettik

*3.3.1 Hava sınıfı*

Public olarak tanımlayıp havaDeniz sınıfından extends ettik. Ucak sınıfının extends edebilmesi için tasarladık

*3.3.2 Deniz sınıfı*

Public olarak tanımlayıp havaDeniz sınıfından extends ettik. Altına gemi sınıfı tanımladık ve gemiye de deniz sınıfını extends ettik.

*3.3.3 Ucan gemi sınıfı*

Hızlan, yavasla, dur fonksiyonlarını override ettik. HavaDeniz sınıfından extends ettik.

**4.Sonuçlar**

Projede verilen isterleri tam olarak yerine getirdiğimizi düşünüyoruz ve projeden tam puan almayı hedefliyoruz. Proje sayesinde java dilinde proje üretmeyi daha rahat yapacağımızı ve javanın özelliklerini öğrenip kullandığımız için severek yapacağımızı düşünüyoruz.

**5.Kaynakça**

[1] <http://kod5.org/javada-yapilandiricilar/>

[2] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri>

[3] http://selimkaratas.com.tr/wp/javada-metotlar.html







