



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
OBJECT ORIENTED PROGRAMMING



ÖDEV RAPORU

Öğrenci Adı, Soyadı, No : İsmail Mert ÖZTÜRK - 402506

Ders Adı: Object Oriented Programming

Ders Sorumlusu: Dr. Öğr. Üyesi Beste Üstübioğlu

Teslim Tarihi : 23.12.2022

Savunma Tarihi : 26.12.2022

İçerik :

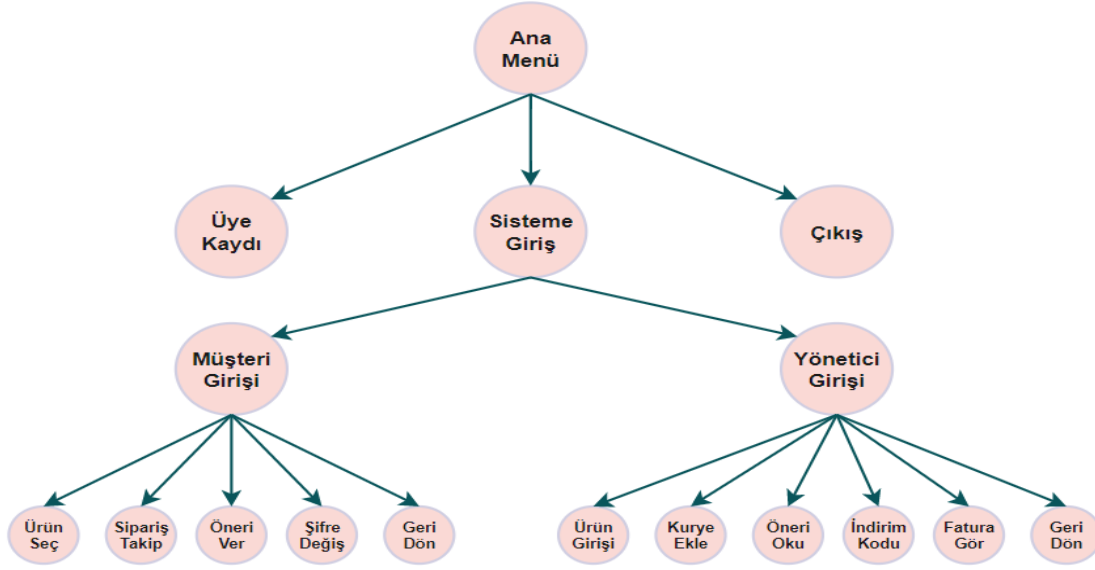
Bu dosya, 2022-2023 eğitim öğretim yılı güz dönemi Object Oriented Programming dersine ait dönem ödevi raporunu içermektedir.

Rapor :

Ödevde yapılması gerekenler ödev dokümanında belirtilmiştir. Bu özelliklerin programa nasıl uyarlandığı aşağıda sırayla açıklanmıştır.

Proje içeriğinde pek çok txt dosyası bulunmaktadır. Bu dosyaların isimleri ve bulundukları konumlar tek bir noktadan ayarlayabilmek mümkündür. Gerekli include işlemlerinin ardından txt dosyalarının temel konumları ve isimleri statik olarak tanımlanmıştır. Uygulama başlatılmadan önce ayarlanması gerekmektedir. Eğer dosyaların temel yolunu gösteren rootPath değişkeni özel olarak ayarlanacak ise belirtilecek yolun mevcut olması gerekmektedir. Aksi halde dosya yolu bulunamayacak ve uygulama işleyişi bozulacaktır. Varsayılan dosya yolu proje dizini olarak belirlenmiştir.

Uygulamada yönetici sınıfına ait kullanıcı oluşturma menüsü olmadığından yönetici girişi yapmak için 123456 şifresini girmek gerekmektedir. Veya bir kere giriş yaptıktan sonra admin.txt dosyasından bu şifreyi manuel olarak değiştirerek istenildiği şekilde kullanılabilir. Ödev dokümanında bunun ile ilgili bir kısıtlama yapılmadığından böyle bir yol tercih edilmiştir.



Öncelikle projede bir menü hiyerarşisi istenmektedir. Hiyerarşi yukarıdaki diyagramda gösterilmiştir. Bu hiyerarşiyi sağlamak için ‘n-ary tree’ yapısı kullanılmıştır. Bu yapı, MenuTree ve MenuNode sınıfları tarafından sağlanmaktadır. MenuTree genel ağaç yapısını oluştururken, her bir düğüm MenuNode sınıfı ile oluşturulmuştur.

MenuNode sınıfı programın işlevlerini gerçekleştiren actionHandler adlı Action sınıfından veya Action sınıfından türetilmiş bir veri yapısı içerir. Action sınıfı virtual fonksiyon içeren bir sınıftır ve her düğüm için ortak yapıları içerir. Kendine has özelliği olan her bir düğümün Action sınıfından türetilmiş bir actionHandler veri yapısı vardır ve yeri geldiğinde bu veri yapısının fonksiyonlarını yürürlüğe sokar.

Gelecek zamanlarda programa yeni bir işlev eklenmek istendiği takdirde bunu gerçekleştirmek oldukça kolaydır. MenuTree sınıfında initSubmenus() fonksiyonu içerisine üst düğüm belirtilecek şekilde yeni bir düğüm eklenebilir. Bu işlemin ardından eklenecek özelliğin amacını yerine getirecek bir Action sınıfı yazılarak sonuca ulaşılabilir.

Her bir düğüm için spesifik özellikler içeren Action sınıfları ve fonksiyonları main.cpp dosyası içerisinde fonksiyon gövdeleri üzerinde açıklanmıştır. Fakat programdaki önemi açısından Action sınıfında olan setApplicationTime() ve incrementTime() fonksiyonlarını ayrıyeten açıklamak gereklidir.

Action sınıfı içerisindeki setApplicationTime() fonksiyonu uygulama başlatıldığında başlangıç zamanını belirleyen fonksiyondur. Projemizde uygulama içerisindeki zaman yerel saatten bağımsız ilerlemektedir. Bu tercihin en önemli nedeni projenin önemli noktalarından olan kurye takip sisteminin çalışmasını daha net bir şekilde analiz edebilmektir. Uygulamanın başlangıç zamanı uygulama üzerinde gerçekleştirilen son aktif değişimin zamanı olarak belirlenir. Bu değişim doğal olarak bir müşterinin ürün siparişi yaptığı zamandır. Yapılan son siparişin teslim zamanı esas alınır ve uygulama zamanı buradan başlar. Eğer daha önce hiç sipariş verilmemiş ise uygulama varsayılan zaman olan 10:00’ dan başlar. incrementTime() fonksiyonu ise uygulama içerisinde menüler arasında geçiş yapıldıkça uygulama zamanını 00:01 (1 dakika) arttırmaktadır.

Action sınıflarının miras aldığı bir diğer sınıf olan CommonActions bütün Action sınıflarının ortak kullandığı fonksiyonları içerisinde barındırır. Action sınıflarının sıkça kullandığı kullanıcıdan giriş alma fonksiyonları ve alınan bu girişin belli kuralları sağlayıp sağlamadığını test etmek için gerekli olan Validator sınıfına ait bir nesne içerir. Validator sınıfı şifre, e-posta, doğum tarihi vb. uygulama akışı içerisinde kontrole ihtiyaç duyulan her kullanıcı girişini kontrol eden sınıftır.

CommonActions sınıfı, ayrıyeten uygulamanın çalışmasında kilit rol oynayan txt dosyaları ile iletişimi sağlayan sınıftır. Dosya okuma/yazma gibi işlemleri bu sınıf gerçekleştirir. Bu işlemleri gerçekleştiren fonksiyonlarda sıkça function overloading kavramı kullanılmıştır.

Uygulama içerisinde gerekli yönergeleri ve uyarıları kullanıcıya ulaştıran ve bunları tek bir noktadan yönetmeyi sağlayan sınıf Buffer sınıfıdır. Bir çeşit kullanıcı arayüzü etkisi yaratmak amacı ile çeşitli fonksiyonlar kullanılarak terminal ekranı üzerinde daha okunabilir ve anlaşılır çıktılar elde etmek amaçlanmıştır. Buffer sınıfı elemanları ve fonksiyonları statik tanımlanmıştır. Bu sayede bir Buffer nesnesi oluşturmaya gerek kalmadan programın her yerinde bu sınıf ve özellikleri kullanılabilir.

Proje dökümanından bağımsız olarak uygulama işleyişine kolaylık sağlaması adına Invoice adında bir sınıf oluşturulmuştur. Bu sınıf müşterilerin yaptığı siparişlerin bilgilerini barındırırken set ve get fonksiyonları ile program içerisinde kullanım kolaylığı sağlar.

Uygulamada main fonksiyonunda çağrılan Application sınıfı uygulama başlatıcı sınıf olarak kullanılmıştır. Başlatıldığı anda uygulamanın temel yapısını tanımlar. Ağaç yapısı, menü yapısı tanımlanır ve uygulama zamanı ayarlanır. Ardından kullanıcıyı bir hoş geldin ekranı karşılar ve program akışı başlar.

```
int index = 0;
Courier available = couriers.at(0);

for (int i = 0; i < couriers.size(); i++) {
    if (couriers.at(i).getAvailableTime() < available.getAvailableTime()) {
        available = couriers.at(i);
        index = i;
    }
}
```

Projenin bir diğer önemli noktası ise yapılan sipariş için istenen şekilde kurye ataması yapmaktır. İstenilen algoritma yukarıda belirtilen kod kesiti ile gerçekleşmiştir. Kısaca açıklamak gerekirse couriers, Courier sınıfı ile temsil edilen mevcut kuryeleri tutan bir vektördür. Bu vektör içerisinde bütün kuryeler arasında son teslim edilen siparişin bitiş zamanları karşılaştırılır. Sipariş teslim zamanı en erken biten kurye mevcut yapılacak sipariş için seçilen kurye olacaktır.

```
Time appendix = addTimeByLocation();

if (available.getAvailableTime() > currentTime) {
    deliveryTime = available.getAvailableTime() + appendix;
    available.addOrder(invoiceID, deliveryTime);
}
else {
    deliveryTime = currentTime + appendix;
    available.addOrder(invoiceID, deliveryTime);
}
```

Ardından bu siparişin sipariş numarasını ve teslim zamanını seçilen kuryenin bilgilerine eklememiz gerekir. Teslim zamanını hesaplamak için addTimeByLocation() fonksiyonu devreye girer ve mevcut kullanıcının adresine bağlı teslimat zamanı hesaplanır. Ardından uygulama içerisindeki zaman ile seçilen kuryenin son siparişinin teslim zamanları karşılaştırılır. Eğer kuryenin boşa çıkacağı zaman uygulama zamanından büyük ise yeni siparişi teslim etmek için önce kuryenin boşa çıkması beklenmelidir. Yani teslim saati kuryenin boşa çıktığı zaman + adrese bağlı teslimat zamanıdır. Eğer uygulama zamanı kuryenin boşa çıkacağı zamandan küçük değil ise teslim saati uygulama zamanı + adrese bağlı teslimat zamanı olarak belirlenir. Burada zaman sınıfından türetilen nesneler üzerindeki aritmetik işlemleri gerçekleştirmek için Time sınıfı içerisinde operator overloading kavramı kullanılmıştır.

Nesne yönelimli programlama konseptini proje genelinde etkin bir şekilde kullanılmış olup ve program akışının tamamı oluşturulan sınıflar üzerinden ilerlemektedir. Proje raporunda istenilen miras ilişkileri eksiksiz bir şekilde tamamlanmış olup, set ve get fonksiyonları ile encapsulation kavramının kullanımı net bir şekilde hissettirilmiştir. Bunların yanında gerekli yerlerde operator ve function overloading amacına uygun bir şekilde kullanılmıştır.

Kaynak dosyası incelendiğinde mümkün olabilecek her yerde yorum satırlarına yer verildiği görülecektir. Modüler programlama ilkesini esas alan projemiz farklı amaçları olan sınıflardan bir bütün oluşturarak kullanıcı dostu bir yazılım elde etmeyi hedeflemiştir.