

CSE222 HW1 REPORT

131044042_AhmetEmin_Kaplan

Ödev pdf inde belirtilen hiyerarşiyi oluşturmak için öncelikle bütün classların ortak özelliklerinin sınırını belirleyeceğim interfacerleri oluşturdum .Daha sonra aralarında olması gereken "is a relation" ilişkisine göre tutor sınıfının, öğrenci sınıfından türemesi gerektiğini düşündüm.Bu durumda teacher,administrator ve student sınıfları belirlediğim abstract sınıftan türemesi durumunda object oriented programming felsefesine daha uygun olarak daha az kod yazacaktım.Yani bütün sınıflara ait ortak metodları bir kere abstract sınıfta implement ettim (getter/setter gibi yapıları).Herhangi bir sistem sınıfı oluşturmak yerine AbstractUser sınıfında bütün sistemin verilerini saklayabileceğim static arraylistler oluşturdum.Ve böylece Sistemi oluşturan verileri yani userlari ve course lari yazacağım metodlarla çevrelemiş olacaktım.Arraylistleri static olarak oluşturma sebebim bütün sisteme ait olması yani course objelerine administratorler,teacherlar , studentler ve tutorlar erişebilecekti.Ve 2 Admin tanımladığım durumda bu adminlerin üzerinde işlem yaptığı kurslar aynı kurslar olacaktı isin ozeti static memberlar butun objelere ait olacaktı.Pdfte belirtilen özelliklerin büyük bölümünü implement ettim.Administrator sınıfının objeleri pdfte belirtildiği gibi course ekleme , course silme,user ekleme ve user silme metodlarına sahip.Teacher sinifinin objeleri course lara öğrenci , doküman ve assignment ekleyip çıkarma metodlarına sahip.Student sinifinin objeleri sisteme kayıt olabilmek için istek gönderme metoduna sahip .Ayrıca polymorphic çağrıya imkan vermesi için showAllCourses() metodunu bütün userlar için override ettim.Pdf te belirtildiği gibi Studentler eski kursları görüntüleyemiyor.Diğer Userlar bütün kurslari görüntüleyebiliyor.Gerekli error-handling try-catch mekanizmalarini bir çok durumda kullanmaya özen gösterdim.

Test kodumdaki senaryoyu özetlemem gerekirse.Öncelikle bir Administrator objesi oluştuyorum ve bu objenin sahip olduğu constructor kendi instancelarını initialize ederken kendini bütün userlari kayıt altında tuttuğum static ArrayListe ekliyor.Oluşturmuş olduğum administrator objesiyle sisteme 2 farklı kurs ekliyorum ardından bir miktar student,teacher ve tutor ekliyorum.Ekleme işleminin başarıyla yapıldığını görmek için adminin sahip olduğu showAllCourses() ve showAllUsers() metodlariyla kayıtlı ders ve userları listeliyorum.Ardından adminin sahip olduğu ders ve user delete etme metodlarini kullanıp sonrasında tekrar mevcut ders ve userlari listeliyorum.Böylece delete metodlarinin basarili olduğunu gözlemliyorum.Oluşturmuş olduğum teacher objesinin derslere öğrenci,doküman,assignment ekleme metodlarinin başarı durumunu before-after şeklinde listeleme yaparak gözlemliyorum.Sonrasında studentler tarafından bazı assignmentlara submission yapıyorum ve gerekli testlerle metodlarin başarı durumunu gözlemliyorum.Bu mantıkla farklı user tipleri için implement ettiğim metodların test işlemlerini main metodumda yapmış oluyorum.Main metodumun son kısmında ise bütün user tiplerinin ortak metodu olan showAllCourses() metodunu polymorphic olarak çağırarak her bir override in sonucunu ekrana bastırıyorum.

Ödevde çok fazla detay olduğunu düşünüyorum bu kadar detaylı olması benim temel probleme odaklanmamı biraz geciktirdi.Ama bu ödev bana JUnit – Unit Test , Interface implementasyonu ve ArrayList yapısının kullanımı hakkında tecrübe kazandırdı.