**YZM3017-YAZILIM TASARIMI VE MİMARİSİ 2020 GÜZ DÖNEMİ UZEM**

**FİNAL SINAVI SORULARI**

**Yazılı Sınav Süresi:** 120 dk dır.

**Sınav başlangıç ve bitiş saatleri**:16:00-18:00 (25 Ocak 2021).

**Açıklamalar**

**1)**Yazılı sınavınızın cevapları için bir ana klasör oluşturunuz. Klasörün adı: Soyadınız\_Öğrenci\_Numaranız (örneğin Kahraman\_1519939)

**2)**Birinci adımda oluşturduğunuz ana klasörün altında iki klasör oluşturunuz. Bu klasörlerin adları: Cevap-1 ve Cevap-2

**3)**Cevap-1 klasörü altında C1-Kod klasörü oluşturunuz.

a.       Soru 1 için hazırladığınız cevabın çalıştırılabilir kodunu C1-Kod klasörüne yükleyiniz.

b.       Ayrıca Soru-1 için hazırladığınız cevabı tıpkı soru dokümanında size verdiğimiz gibi sunmanız gerekmektedir. Cevabı Word/pdf formatlarından birinde hazırlayarak (soruda verdiğimiz gibi) Cevap-1 klasörüne yükleyiniz. UML diyagramı (sınıf diyagramı) ve kodlar aynı dokümanda (tek belge içinde) verilecek. Çok sayıda belge/doküman yüklemeyiniz.

**4)**Cevap-1’i yüklemek için yaptığınız işlemlerin aynısınıCevap-2 için de yapınız.

***Not: Cevaplarınız için yukarıda tarif edilen klasörlerin oluşturulmuş bir örneğini yükledim. Onu kullanabilirsiniz. Örnekte ana klasörün adı “Kahraman\_1519939” dır. Klasör adını kendinizinki ile değiştirmeyi unutmayınız.***

**5)**Cevaplarınızı sisteme sıkıştırılmış (winzip/winrar şeklinde) bir dosya olarak yükleyiniz. 50MB dan fazla boyuta sahip dosya yükleyemezsiniz. Bu nedenle resim formatında vs. yüksek boyutlu dokümanlarınızı düşük boyutlara indirgemeniz gerekebilir.

**6)**Kodlama çevresi: C# ya da Java (sadece birini yükleyiniz)

**7)**Sorular aşağıda verilmektedir. Sorular eşit puanlıdır.

*Not: Soruların anlaşılıp çözülebilmesi için azami 1 saat yeterlidir. Geri kalan süre panik yapmadan ve rahat bir şekilde sınavınızı tamamlamanız içindir. UML diyagramını elinizle (düzgün bir şekilde) kağıda çizip bilgisayara aktarıp ekran görüntüsü vs. yöntemler kullanarak ekleyebilirsiniz. Cevaplarınızı başkalarıyla paylaşmayınız ve sadece kendi çabanızla soruları cevaplayınız. Başarılar dilerim.*

**Ders Sorumlusu:** Prof. Dr. Hamdi Tolga KAHRAMAN

**SORULAR**

**Soru-1)**

Bir açık artırma uygulaması geliştirmeniz isteniyor. Bu uygulamada ürünler açık artırma ile satılmaktadır. Ürünün fiyatı alıcılar tarafından artırılarak belirlenmektedir. Açık artırmada olan ürünün fiyat değişiminin **önceden teklif yapan kullanıcılara** bildirilmesi istenilmektedir. Yani ürüne bir alıcı fiyat verdiği (**yeni bir fiyat eski fiyattan daha büyük olmalıdır**) anda tüm alıcılara yeni fiyat bilgisi mail olarak gönderilmelidir. **(UML sınıf diyagramı çizilecek)**

NOT: Her kullanıcıya bir fiyat değişimi aşamasında bir mail gönderildiğinden emin olunuz.

İstemci Kodu:

public static void main(String[] args) {  
  
 Kullanici sefa = new Kullanici("Sefa Aras", "sefa@ktu.edu");  
 Kullanici hakan = new Kullanici("Hakan Bozkurt", "hakan@ktu.edu");  
  
 Urun bilgisayar = new Urun("Bilgisayar", 100);  
  
 System.*out*.println("1. Teklif");  
 bilgisayar.teklifVer(sefa, 50);  
 System.*out*.println("2. Teklif");  
 bilgisayar.teklifVer(hakan, 150);  
 System.*out*.println("3. Teklif");  
 bilgisayar.teklifVer(sefa, 200);  
 System.*out*.println("4. Teklif");  
 bilgisayar.teklifVer(hakan, 250);  
  
}

Beklenen Çıktı:

1. Teklif

2. Teklif

Hakan Bozkurt kullanıcısına fiyat değişimi (150) mail olarak bildirildi.

3. Teklif

Hakan Bozkurt kullanıcısına fiyat değişimi (200) mail olarak bildirildi.

Sefa Aras kullanıcısına fiyat değişimi (200) mail olarak bildirildi.

4. Teklif

Hakan Bozkurt kullanıcısına fiyat değişimi (250) mail olarak bildirildi.

Sefa Aras kullanıcısına fiyat değişimi (250) mail olarak bildirildi.

**Soru-2)**

|  |  |
| --- | --- |
| **interface** Kullanici {  **public bool** HesapOku(string hesapNo);  }  **class** Yetkili : Kullanici {  **private string** no;  Yetkili (**string** No) {  No = no;  }  **public bool** HesapOku() {  //Hesap bilgileri Okundu  **return** true;  }  } | **class** DigerEleman : Kullanici {  **private string** no;  DigerEleman(**string** No) {  no = No;  }  **public bool** HesapOku() {  //Hesap bilgileri okundu  **return** true;  }  }  **public class** Test {  **public static void main**(**String**[] args) {  Yetkili y1 = **new** Yetkili("1234");  DigerEleman dE1 = **new** DigerEleman("1234");  y1.HesapOku();  dE1.HesapOku();  }  } |

* Müşterilere ait bilgilerin çekildiği bir uygulama hazırlanmıştır. Bu uygulamaya ait hali hazırdaki sınıf yapısı ve main kodu yukarıda verilmiştir. Sistemin **Yetkili** ve **Diğer eleman** adlarında iki kullanıcı sınıfı bulunmaktadır. Bu sınıflarda hesap bilgileri “*HesapOku*()” metodu ile okunmaktadır.
* *HesapOku* metodu private olarak verilmiş olan *no* bilgisine sahip müşterinin sadece hesap bilgilerini okuyabiliyor.
* Her iki kullanıcı sınıfında da hesap bilgileri okunabildiği için HesapOku metodu geriye **true** değeri döndürmektedir.
* Şu anki haliyle sadece hesap bilgileri okunabilen bu uygulamaya yeni bir özellik eklenmesi isteniyor. Bu yeni özellik ile hesap numarası (no) gönderilen müşterinin; hesap bilgileri dışındaki, kimlik bilgilerinin okunabilmesi sağlanacaktır. Bu bilgilerin sadece **Yetkili** türündeki kullanıcılar tarafından okunabilmesi istenmektedir. **DigerEleman** tipindeki kullanıcılara ise bu özellik eklenmeyecektir.

Müşteri kimlik bilgileri :

* Yetkili tarafından okunabileceği için yetkiliye ait ilgili metod geriye **true**;
* DigerEleman tarafından okunamayacağı için diğer elemana ait ilgili metod geriye **false** döndürecektir.

Bu yeni davranışın (metodun) eklenmesini visitor tasarım deseni kullanarak gerçekleştiriniz. **(UML sınıf diyagramı çizilecek)**

Not: **Visitor** tasarım deseni kullanılacaktır.

* Soyut visitor sınıfı adı : IVisitor
* Somut visitor adı: KimlikOkuyucu

olacaktır.