

## JAVA-15 2. MARATON OOP

### CASE : RESTORAN REZERVASYON SİSTEMİ

#### Sınav Genel Hatırlatma

- ✓ Değişken isimlendirme kurallarına uyunuz.
- ✓ Değişken isimleriniz, metod isimleriniz anlaşılır olmalı.
- ✓ Anlaşılması zor olabilecek her satır koda javadoc şeklinde yorum eklemeyi unutmayınız.
- ✓ Ekstra bir metod, sınıf tanımlayıp kullandıysanız lütfen program içinde bunları neden kullandığınızı detaylı şekilde anlatınız.
- ✓ Süreye riayet etmeye çalışınız. Zamanınızı verimli kullanmaya özen gösteriniz.
- ✓ Sisteme yüklerken projeyi zipleyerek yükleyiniz.

#### ÖNEMLİ NOT:

- ✓ Bu projede ne kullanıyorsanız kullanma sebebini anlayarak kullanmaya çalışınız. Soruda sizce daha uygun çözümler varsa kendinizce değişiklikler yapabilirsiniz. Sonuç olarak bir restoran rezervasyon sistemi kurgulamanız gerekli. Güzel açıklamalarla kendi kurgunuzu da geliştirebilirsiniz. Önemli olan OOP'da gördüğümüz kavramları ve en azından aşağıda belirtilen metodları içermesidir. Daha fazla özellik eklemek isterseniz, neden eklediğinizi de açıklama olarak yazarak ekleyebilirsiniz. Gerek duyarsanız uml diagramlarını ve açıklamalarınızı readme.md dosyası içinde proje ile beraber yükleyebilirsiniz.

Bir restoran rezervasyon sisteminde aşağıda verilenleri yapmanız beklenmektedir.

#### 1. Sınıfların Tanımlanması (Toplam 50p)

##### a. Restoran Sınıfı (10p)

**Fields:** restoranID (String), adı (String), adresi (String), telefon (String), kapasite (int), durum (Enum: AÇIK, KAPALI)

**Metodlar:** constructor, getter ve setter metodları

##### b. Müşteri Sınıfı (10p)

**Fields:** müşteriID (String), isim (String), soyisim (String), telefon (String), eposta (String)

**Metodlar:** constructor, getter ve setter metodları

##### c. Rezervasyon Sınıfı (10p)

**Fields:** rezervasyonID (String), müşteriID (String), restoranID (String), rezervasyonTarihi (DateTime), kişiSayısı (int)

**Metodlar:** constructor, getter ve setter metodları

##### d. Yönetici Sınıfı (20p)

**Fields:** yöneticiID (String), isim (String), soyisim (String)

**Metodlar:** constructor, getter ve setter metodları, restoranEkle, restoranSil, restoranAra, restoranlarıListele, müşteriEkle, müşteriSil, müşteriAra, müşterileriListele, rezervasyonYap, rezervasyonİptalEt, rezervasyonlarıListele

**NOT: Sınıf isimlerini deęiřtirebilirsiniz, sayısını artırıp azaltabilirsiniz. Sınıfların içinde olması gereken fieldları da siz belirleyebilirsiniz.) Verileri tutmak için ArrayListlerden oluşan Database class'ı kullanabilirsiniz.**

**NOT: Burada interfacerler ve abstract classlar açıkça belirtilmemiřtir. Gerekli olduęunu düřündüğünüz interfacerleri veya abstract classları kullanabilirsiniz.**

## **2. İşlemler ve Metodlar (Toplam 30p)**

**rezervasyonYap, rezervasyonİptalEt, rezervasyonlarıListele:** Yönetici sınıfında, rezervasyon yönetimi için metodlar. (10p)

**restoranEkle, restoranSil, restoranAra, restoranlarıListele:** Yönetici sınıfında, restoran yönetimi için metodlar. (10p)

**müşteriEkle, müşteriSil, müşteriAra, müşterileriListele:** Yönetici sınıfında, müşteri yönetimi için metodlar. (10p)

**NOT: Bu metodlar dışında metodlar eklemek isterseniz ekleyerek programınızı genişletebilirsiniz.**

## **3. OOP Kavramları (Toplam 10p)**

**Kalıtım:** Mümkünse bazı sınıflar arasında ortak özellikler varsa, kalıtım kullanarak bu özellikleri taşıyan bir üst sınıf oluşturulabilir. (2p)

**Encapsulation:** Tüm sınıflarda private field'lar kullanılarak ve public getter/setter metodları ile kapsülleme yapılmalıdır. (2p)

**Constructor:** Tüm sınıflarda uygun constructor metodları tanımlanmalıdır. (2p)

**Interface:** Ortak metodları tanımlamak için interface'ler kullanılabilir. (2p)

**Final, Static, Enum:** Uygun yerlerde bu kavramlar kullanılmalıdır. (2p)

**NOT: Bu kavramları mutlaka kullanmaya çalışınız. Bu kavramları kullandığınız kısımlarda amacınızın ne olduğunu da yorum ekleyerek açıklayınız.**

## **4. Hata Yönetimi (Toplam 5p)**

Rezervasyon kapasitenin üzerindeyse, restoran veya müşteri bulunamazsa uyarı mesajları verilmelidir. (5p)

**NOT: Bunlar dışında uyarı mesajları eklenmesi gerekli ise ekleyiniz.**

## **5. Kullanıcı Menüsü (Toplam 5p)**

Kullanıcıya kolay kullanım sunmak için, işlemleri seçebileceęi basit ve anlaşılır bir menü tasarlanmalıdır. İşlemler numaralandırılarak ve açıklayıcı bir şekilde sunulmalıdır. (5p)