

Nesne Yönelimli Programlama Proje Analizi

Projem Zincir Otel Yönetim Sistemi yapmaktı. Bu projeyi gerçekleştirirken Eclipse IDE kullandım.

TestClass'ı dışında 7 Class Tanımladım. Bu classlar: müşteriBilgiler, müşteriMilliyeti, odaBilgileri, otelBilgi, otelGiderler, personelBilgisi, Vatandaslik classlarıdır. Bu program bir Zincir otelin müşteri kayıt işlemleri, gelir-gider işlemleri, personel işlemleri, otel işlemleri ve oda işlemlerini gerçekleştirmek üzerine tasarlanmıştır.

Classların Görevlerinden ve fonksiyonlarından Kısaca Bahsedelim:

musteriBilgiler Classı: Müşteri bilgilerinin alınması, gerekli depolamanın yapılması ve giriş çıkış işlemlerini gerçekleştirmek üzere tasarlanmıştır.

1. **Constructor** ile müşteriden Müşteri No, Ad, Soyad, Şehir, Cinsiyet ve Medeni Durum bilgileri alındı.
2. **Constructor** overload edildi.
3. Müşteri kaydı için Static bir arraylist oluşturuldu.
4. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
5. **musteriEkle** fonksiyonu ile alınan veriler arraylistte veriler tutuldu.
6. **musteriSil** fonksiyonu ile arraylistten verilen index karşılığı giriş yapılmış müşterinin bilgileri sildildi.
7. **musteriSayisi** fonksiyonu ile kayıtlı bulunan müşterilerin sayısı arraylistten çekildi.
8. **cinsiyeteGoreDagilim** fonksiyonu ile otelde olan müşterilerin cinsiyet dağılımları hesaplandı

personelBilgisi Classı: Personel bilgilerinin alınması, gerekli depolamanın yapılması ve işe giriş çıkış işlemleri ve maaş işlemlerini gerçekleştirmek üzere tasarlanmıştır.

1. **Constructor** ile personele ait Ad, Soyad, Doğum Tarihi, Maaş, Cinsiyet ve Unvan bilgileri alındı.
2. **Constructor** overload edildi.
3. Personel bilgileri kaydı için Static bir arraylist ve Maaşların saklanması için bir dizi oluşturuldu.
4. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
5. **personelEkle** fonksiyonu ile alınan veriler arraylistte kaydedildi.
6. **personelSil** fonksiyonu ile arraylistten verilen index karşılığı giriş yapılmış müşterinin bilgileri sildildi.
7. **toplamPersonel** fonksiyonu ile otelerde kayıtlı bulunan tüm personellerin sayısı arraylistten çekildi.
8. **toplamMaas** fonksiyonu ile otelerde kayıtlı bulunan tüm personellerin toplam maaşı hesaplanarak arraylistten çekildi
9. **ortalamaMaas** fonksiyonu ile otelerde bulunan tüm personellerin maaşlarının ortalaması hesaplandı

otelBilgi Classı: Otel bilgilerinin alınması, gerekli depolamanın yapılması, zincir otel giriş çıkış işlemlerini gerçekleştirmek üzere tasarlanmıştır.

1. **Constructor** ile kayıt edilecek otele ait Otel adı, Aktivlik durumu, Toplam oda sayısı, Otelin bulunduğu bölge, Otelin yıldız sayısı ve Oda türleri bilgileri alındı.
2. **Constructor** overload edildi.
3. Otel bilgileri kayıdı için Static bir arraylist oluşturuldu.
4. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
5. **otelEkle** fonksiyonu ile alınan veriler arraylistte kaydedildi.
6. **otelSil** fonksiyonu ile arraylistten verilen index karşılığı giriş yapılmış otelin bilgileri sildindi.

odaBilgileri Classı: Bu Class otelBilgi classının alt classıdır. Bu class mevcut oteldeki oda bilgilerini almak, depolamak, düzenlemek ve fiyatı hakkında gerekli bilgileri sunmak üzere tasarlanmıştır

1. **Constructor** ile otele ait Otel adı, Odaların büyüklüğü, Oda türü, Odanın Cinsine göre Fiyatı, 2 adet Ekstra(girilmesi zorunlu değil) alındı.
2. **Constructor** overload edildi.
3. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
4. **odaEkle** fonksiyonu ile alınan veriler arraylistte kaydedildi.
5. **odaSil** fonksiyonu ile arraylistten verilen index karşılığı giriş yapılmış odanın bilgileri sildindi.
6. **odaSayisi** fonksiyonu ile o otelde bulunan toplam oda sayısı hesaplandı.
7. **ortalamaFiyat** fonksiyonu ile o otelde bulunan odaların ortalama fiyatı hesaplandı.

otelGiderler Classı: Bu Class otelBilgi classının alt classıdır. Bu class mevcut oteldeki gider bilgilerini almak, depolamak, düzenlemek ve bunlar hakkında gerekli bilgiyi sunmak üzere tasarlanmıştır.

1. **Constructor** ile otele ait Giderin adı, Giderin türü, Komisyon oranı, Toplam tutar ve Ödemenin nasıl yapılacağı bilgileri alındı. Ödemeler dizisinde ise ödemelerin listesi tutuldu.
2. **Constructor** overload edildi.
3. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
4. **giderEkle** fonksiyonu ile alınan veriler arraylistte kaydedildi.
5. **giderSil** fonksiyonu ile arraylistten verilen index karşılığı giriş yapılmış olan gider bilgileri sildindi. Ve
 - a. yeni gider miktarı hesaplandı
6. **giderToplami** fonksiyonu ile o otelin toplam gideri hesaplandı.
7. **setYildizSayisi** fonksiyonu ile otelin Yıldız sayısı override edildi.

MusteriMilliyeti Classı: Bu Class Polymorphism için kullanılmıştır. Bir müşteriye ait Milliyet ve İkamet bilgilerini almak, saklamak ve yazdırmak için geliştirilmiştir

1. **Constructor** ile müşteriye ait Milliyet ve Nerede İkamet ettiği bilgileri alındı
2. **get** ve **set** metotları ile alınan bilgiler private tutularak kapsülasyon gerçekleştirildi.
3. **bilgiler** fonksiyonu ile alınan veriler yazdırıldı.

Vatandaslik Classı: Bu Class Polymorphism için kullanılmıştır ve MusteriMilliyeti Classının alt classıdır. Bir Müşteriye ait Milliyet, İkamet ve Sabıka durumunu almak ve yazdırmak için Geliştirilmiştir.

1. **Constructor** ile müşteriye ait Milliyet ve İkamet bilgileri alındı (superclass'tan). Bunlara ek olarak Sabıka kaydı bilgileri alındı.
2. **bilgiler** fonksiyonu ile müşteriye ait veriler yazdırıldı.