

BMU260 – Halı Yıkama Takip Sistemi – Ödev Raporu

Öğrenci Bilgileri

Ad Soyad: Ali Nebi ER

Bölüm: Bilgisayar Mühendisliği

Sınıf: 3. Sınıf

Ders: BMU260 - Nesne Tabanlı Programlama

Teslim Tarihi: 8 Nisan 2025

Proje Adı

Halı Yıkama Takip Sistemi

Geliştirme Ortamı

Projeyi Visual Studio 2022 kullanarak, C# diliyle ve Windows Forms teknolojisiyle geliştirdim. Arayüz işlemleri için hazır bileşenleri (TextBox, ListBox, Button vs.) kullanarak kullanıcı dostu bir yapı oluşturmaya çalıştım.

Kullanılan Sınıflar

1. Musteri

Her müşterinin adı, soyadı, telefon numarası ve adres bilgileri tutuluyor. Ayrıca her müşterinin birden fazla halısı olabileceği için `List<Hali>` tipinde bir halı listesi tanımladım.

2. Hali

Haliya ait metrekare, alınma tarihi, tahmini teslim tarihi gibi bilgiler tutuluyor. Ayrıca halının yıkama durumu (true/false) ve otomatik hesaplanan bir ücret özelliği (metrekare x 20 TL) bulunuyor.

Uygulama Özellikleri

Projede aşağıdaki işlemler kullanıcı arayüzü üzerinden yapılabiliyor:

- Yeni müşteri kaydı eklenebiliyor.
- Seçilen müşteriye halı eklenebiliyor.
- Halıların durumu "Teslim Edildi" olarak güncellenebiliyor.
- Sadece yıkamada olan halılar filtrelenebilir ve listelenebiliyor.
- Sadece teslim edilen halılar da ayrı şekilde gösterilebiliyor.

Nesne Tabanlı Programlama Özellikleri

Projede nesne tabanlı programlamanın temel prensiplerine dikkat ettim:

- **Kapsülleme:** Alanları `private` yapıp, `public` property'lerle erişimi kontrol ettim.
- **Sınıf ilişkisi:** Musteri sınıfı içinde halıları tutarak bire-çok ilişkisini kurdum.
- **ToString override:** ListBox gibi yerlerde anlamlı veri göstermek için `ToString()` metodunu ezdim.
- **Constructor kullanımı:** Formlar arası veri aktarımı yaparken constructor ile nesne gönderdim.

Karşılaştığım Hatalar ve Nasıl Çözdüm

- **CS0103 Hatası:** `lstMusteriler` adındaki kontrol, yanlışlıkla `FormDurumTakip` içinde kullanılmaya çalışılmış. Sonradan fark ettim ki bu kontrol sadece `Form2`'de var. Kodun yerini değiştirerek çözdüm.
- **CS0051 Hatası:** Formlar arası veri aktarırken `Musteri` sınıfının erişim düzeyi `public` değilmiş. Constructor dışarıdan erişilebilir olunca `Musteri` de `public` yapılmalıymış, bunu araştırarak öğrendim.
- **Tasarımda kaybolan kontroller:** Tasarım ekranında elle bazı kontrolleri tanımlamaya çalıştığımda, Visual Studio bu kontrolleri kaldırıyordu. `InitializeComponent()` dışında kontrol tanımlamamak gerektiğini anladım.

Öğrendiğim Şeyler ve Kodlardan Örnekler

1. ToString() override sayesinde GUI'de anlamlı veri gösterimi:

```
csharp
KopyalaDüzenle
public override string ToString()
{
    return $"{Ad} {Soyad} - {Telefon}";
}
```

2. Constructor ile formdan forma nesne taşıma:

```
csharp
KopyalaDüzenle
public FormDurumTakip(Musteri seciliMusteri)
{
    InitializeComponent();
    this.musteri = seciliMusteri;
}
```

3. Liste filtreleme ile yıkamadaki halıları ayrı göstermek:

```
csharp
KopyalaDüzenle
foreach (var hali in musteri.Halilar)
{
    if (!hali.TeslimEdildi)
        lstHalilar.Items.Add(hali);
}
```

Son olarak

Bu projeyi yaparken sadece C#'ta GUI geliştirmeyi değil, aynı zamanda nesne tabanlı yapının gerçek hayattaki bir problem üzerine nasıl uygulanabileceğini de öğrendim. Karşılaştığım hatalar beni durdurmadı, aksine her biri bana bir şey öğretti.

Halı yıkama gibi basit görünen bir işlemin bile yazılım desteğiyle nasıl daha sistematik ve takip edilebilir hale geleceğini görmek oldukça ilginçti. Kodları yazarken bol bol test yaptım, farklı senaryolar denedim.

Sonuç olarak, bu projeyi bitirdiğimde nesne tabanlı programlama konusuna çok daha hakim olduğumu hissettim.