

# Nesne Tabanlı Programlama

Bu çalışma materyalinin içerisinde uzun çalışma projeleri bulunmaktadır. Bu materyalin amacı öğrencilerin öğrendiği konuları pekiştirmek ve Otomasyon projelerinin rahatlıkla yapılabilmesini kanıtlamaktır.

İçerisinde var olan konular.

- Değişkenler ve Temel Veri Türleri
- Aritmetiksel Operatörler
- İşlem Önceliği
- Karar İfadeleri
- Mantıksal Operatörler
- Döngüler
- Hata Ayıklama
- Sınıflar ve Nesneler
- Tek Boyutlu Diziler
- Çok Boyutlu Diziler
- ArrayList Koleksiyonu
- List Koleksiyonu
- Dictionary Koleksiyonu
- Formlar
- MenuStrip Kontrolü
- İletişim Kutuları (Dialog Boxes)
- Veri Doğrulama
- Veri Bağlama

---

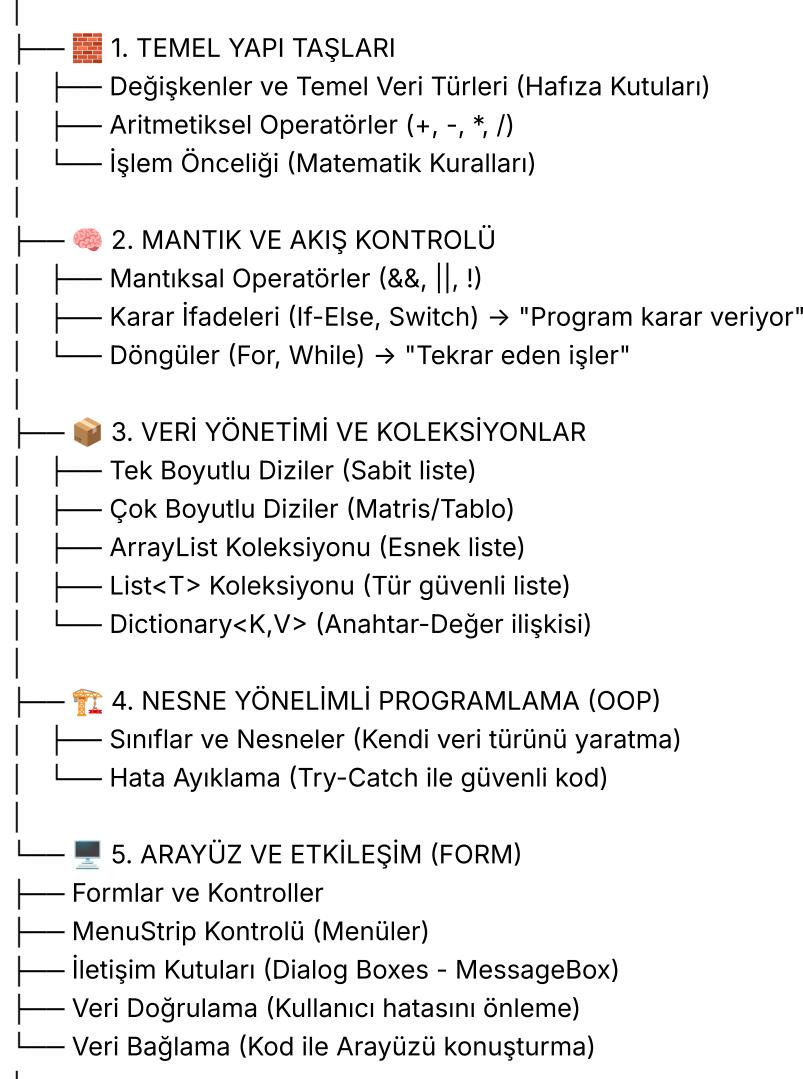
## Proje-1 Çalışma Materyali: CineTech Sinema Otomasyonu

### **Merhaba Geleceğin Yazılımcısı!**

Bu proje, şimdije kadar öğrendiğin değişkenlerden nesne tabanlı programlamaya kadar olan bilgileri birleştirerek gerçek bir otomasyon sistemi kurmayı sağlayacak. Amacımız, bir sinema salonundaki bilet satış, koltuk seçimi ve büfe işlemlerini yöneten profesyonel bir Windows Form uygulaması geliştirmektir.

**Hazır mısın? Adım adım başlıyoruz!**

## BAŞLANGIÇ NOKTASI



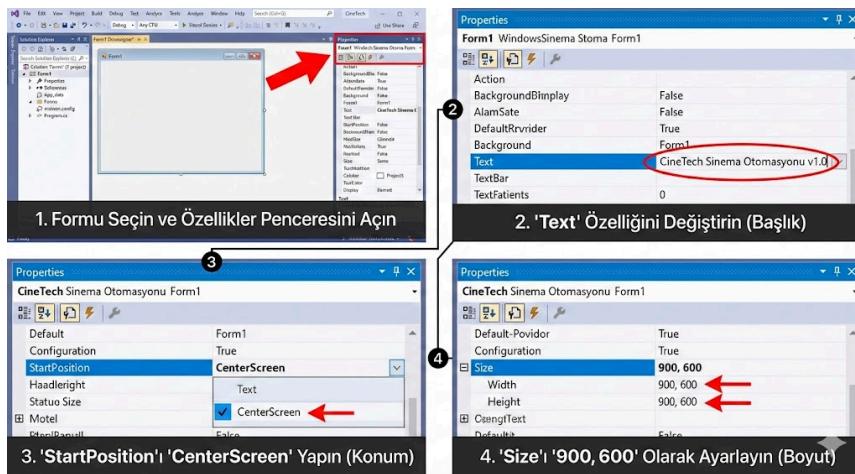
## PROJE: SİNEMA OTOMASYONU

### ADIM 1: Arayüz (Form) Tasarımı

Once projenin dış görünüşünü tasarlayalım. Visual Studio'da yeni bir **Windows Forms App (.NET Framework)** projesi aç ve formu aşağıdaki yönergelere göre dizayn et.

#### 1. Form Ayarları

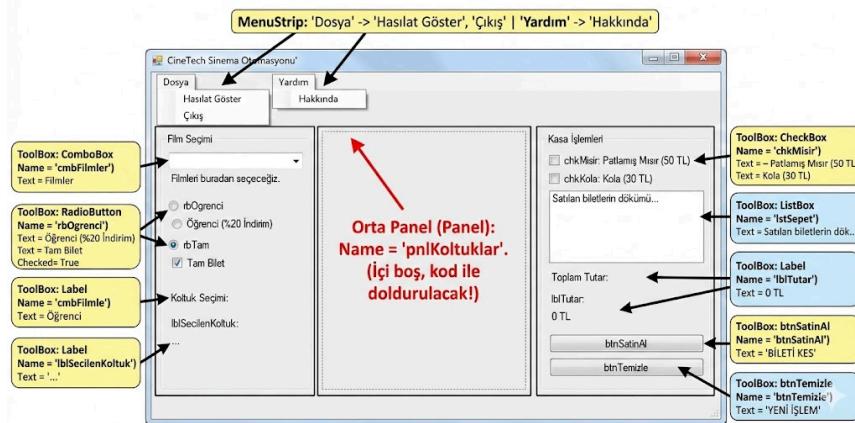
- Form Text:** "CineTech Cinema Otomasyonu v1.0"
- StartPosition:** CenterScreen (Ekranın ortasında açılsın)
- Size:** 900, 600 (Geniş bir alan)



## 2. Araçları (Toolbox) Yerleştirme

Formun üzerine şu kontrolleri sürükle ve "Properties" penceresinden **Name** (İsim) kısımlarını parantez içindeki gibi değiştir. Bu isimlendirmeler kod yazarken çok önemli!

### CineTech Otomasyonu: Araçları (Toolbox) Yerleştirme ve İsimlendirme Rehberi



- Üst Kısmı:**

- **MenuStrip**: İçine "Dosya" → "Hasılat Göster", "Çıkış" ve "Yardım" → "Hakkında" menülerini ekle.

- Sol Panel (Film Seçimi - GroupBox):**

- **ComboBox** (**cmbFilmler**): Filmleri buradan seçeceğiz.
- **RadioButton** (**rbOgrenci**): Text = "Öğrenci (%20 İndirim)"
- **RadioButton** (**rbTam**): Text = "Tam Bilet" (Checked = True yap)
- **Label**: "Koltuk Seçimi:" yazısı.
- **Label** (**lblSecilenKoltuk**): Başlangıçta "..." yazsın.

- Orta Panel (Salon Düzeni - Panel):**

- Bir **Panel** ekle (**pnlKoltuklar**). İçini boş bırak, buraya kod ile sihir yapacağız! (Döngüler konusu).

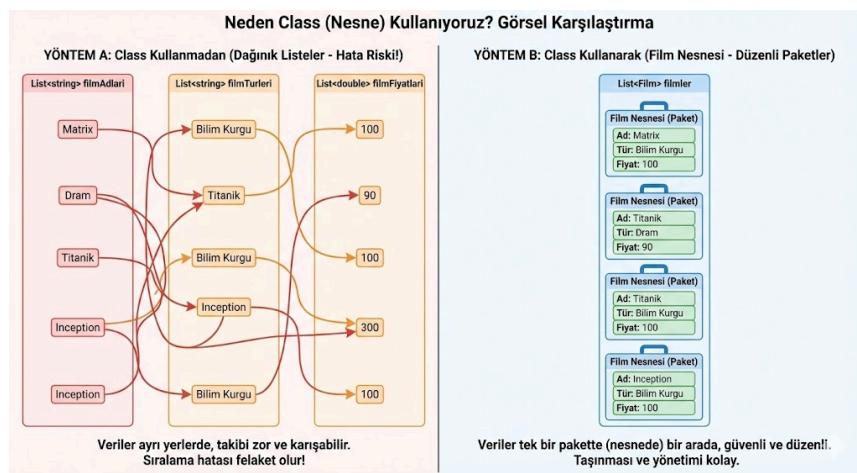
- Sağ Panel (Kasa İşlemleri - GroupBox):**

- **CheckBox** (**chkMisir**): Text = "Patlamış Mısır (50 TL)"
- **CheckBox** (**chkKola**): Text = "Kola (30 TL)"
- **ListBox** (**IstSepet**): Satılan biletlerin dökümü buraya gelecek.
- **Label**: "Toplam Tutar:"
- **Label** (**IbtTutar**): Başlangıçta "0 TL" yazsın.
- **Button** (**btnSatinAl**): Text = "BİLETİ KES"
- **Button** (**btnTemizle**): Text = "YENİ İŞLEM"

## ADIM 2: Kodlama - Temel Yapı ve Sınıflar

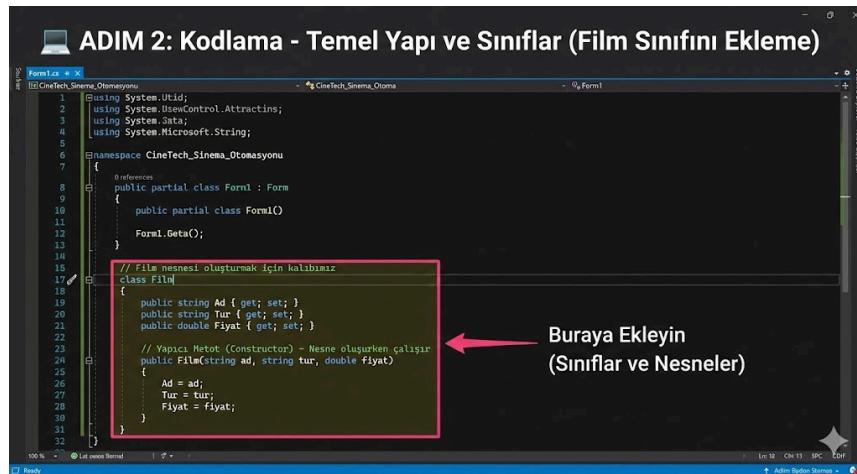
Kod ekranına geç (**F7** tuşu). İlk işimiz, filmleri temsil edecek bir "Kalıp" (Class) oluşturmak. Bu, **Sınıflar ve Nesneler** konusunu pekiştirecek.

- Peki neden class kullanıyoruz?



- Dilersen Film classını `Filmler.cs` olarak yeni bir öğe olarak da açabiliyorsun istersen aynı dosyanın içerisinde de tanımlayabilirsin.

Kod sayfasının en altına, **namespace** parantezinin içine ama **Form** sınıfının dışına şu kodu ekle:

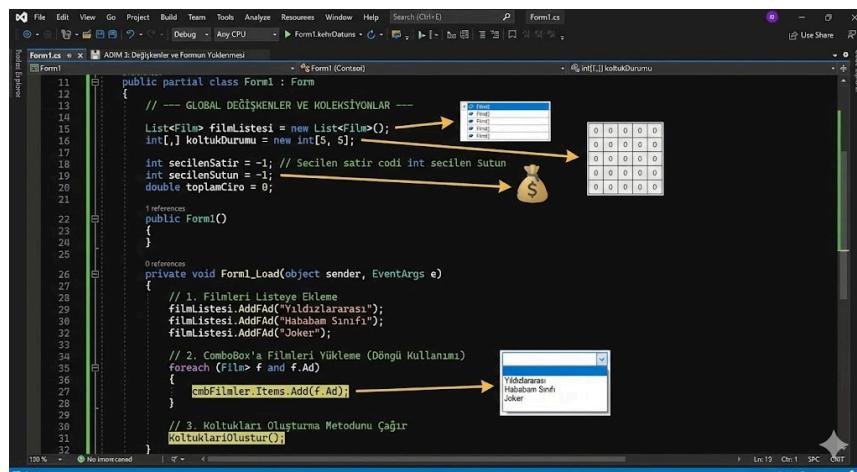


```
// Film nesnesi oluşturmak için kalıbımız
class Film
{
    public string Ad { get; set; }
    public string Tur { get; set; }
    public double Fiyat { get; set; }

    // Yapıçı Metot (Constructor) - Nesne oluşurken çalışır
    public Film(string ad, string tur, double fiyat)
    {
        Ad = ad;
        Tur = tur;
        Fiyat = fiyat;
    }
}
```

 ADIM 3: Değişkenler ve Formun Yüklenmesi

Şimdi formun hafızasını oluşturalım. `public partial class Form1 : Form` satırının hemen altına, global değişkenlerimizi tanımlayalım.



```

public partial class Form1 : Form
{
    // --- GLOBAL DEĞİŞKENLER VE KOLEKSİYONLAR ---

    // Filmleri tutacağımız List Koleksiyonu
    List<Film> filmListesi = new List<Film>();

    // Koltukların doluluk durumunu tutan Çok Boyutlu Dizi (5 satır, 5 süt
    un)
    // 0: Boş, 1: Dolu
    int[,] koltukDurumu = new int[5, 5];

    // Seçilen koltuğun koordinatlarını tutmak için
    int secilenSatir = -1;
    int secilenSutun = -1;

    double toplamCiro = 0; // Günlük kasa

    public Form1()
    {
        InitializeComponent();
    }

    // Form açıldığında (Load Event) çalışacak kodlar
    private void Form1_Load(object sender, EventArgs e)
    {
        // 1. Filmleri Listeye Ekleme
        filmListesi.Add(new Film("Yıldızlararası", "Bilim Kurgu", 100));
        filmListesi.Add(new Film("Hababam Sınıfı", "Komedi", 80));
        filmListesi.Add(new Film("Joker", "Dram", 90));

        // 2. ComboBox'a Filmleri Yükleme (Döngü Kullanımı)
        foreach (Film f in filmListesi)
        {
            cmbFilmler.Items.Add(f.Ad);
        }
        cmbFilmler.SelectedIndex = 0; // İlk filmi seçili yap

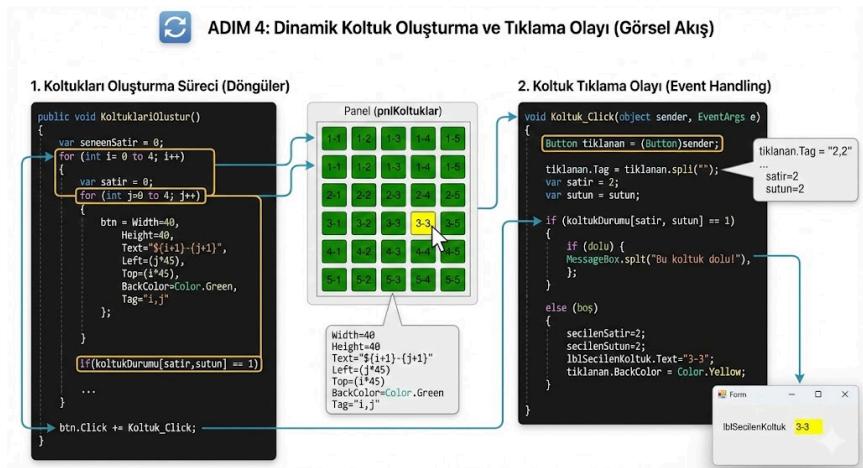
        // 3. Koltukları Oluşturma Metodunu Çağır
        KoltuklarıOlustur();
    }
}

```



## ADIM 4: Dinamik Koltuk Oluşturma (Zor Bölüm!)

Burası projenin en havalı kısmı. Tek tek buton sürüklemek yerine, **Döngüler** kullanarak 25 tane koltuğu kodla oluşturacağız. Bu kodu `Form1` sınıfının içine ekle.



```

private void KoltuklariOlustur()
{
    // Önce paneldeki eski koltukları temizle
    pnlKoltuklar.Controls.Clear();

    // İç İçe Döngüler (Nested Loops) ile 5x5 matris tarama
    for (int i = 0; i < 5; i++) // Satır (i)
    {
        for (int j = 0; j < 5; j++) // Sütun (j)
        {
            Button btn = new Button();
            btn.Width = 40;
            btn.Height = 40;
            btn.Text = $"{i + 1}-{j + 1}"; // Örn: 1-1, 1-2
            btn.Left = (j * 45); // Yan yana dizilim
            btn.Top = (i * 45); // Alt alta dizilim
            btn.BackColor = Color.Green; // Başlangıçta hepsi boş (Yeşil)

            // Hangi butona tıklandığını anlamak için Tag özelliğine koord
            inat gömüyoruz
            btn.Tag = i + "," + j;

            // Butona tıklandığında çalışacak olayı (Event) bağlıyoruz
            btn.Click += Koltuk_Click;

            pnlKoltuklar.Controls.Add(btn);
        }
    }

    // Herhangi bir koltuk butonuna tıklandığında burası çalışır
    private void Koltuk_Click(object sender, EventArgs e)
    {
        Button tiklanan = (Button)sender;
        tiklanan.Tag = tiklanan.Split(",");
        var satir = 2;
        var sutun = 2;

        if (koltukDurumu[satir, sutun] == 1)
        {
            if (dolu)
            {
                MessageBox.Show("Bu koltuk dolu!");
            }
        }
        else (bos)
        {
            secilenSatir = 2;
            secilenSutun = 2;
            lblSecilenKoltuk.Text = "3-3";
            tiklanan.BackColor = Color.Yellow;
        }
    }
}

```

```

// Tag içindeki "2,3" gibi veriyi parçalayıp koordinatı alıyoruz
string[] koordinat = tiklanan.Tag.ToString().Split(',');
int satir = int.Parse(koordinat[0]);
int sutun = int.Parse(koordinat[1]);

// Veri Doğrulama ve Karar İfadesi
if (koltukDurumu[satir, sutun] == 1)
{
    MessageBox.Show("Bu koltuk dolu! Lütfen başka seçiniz.");
    return;
}

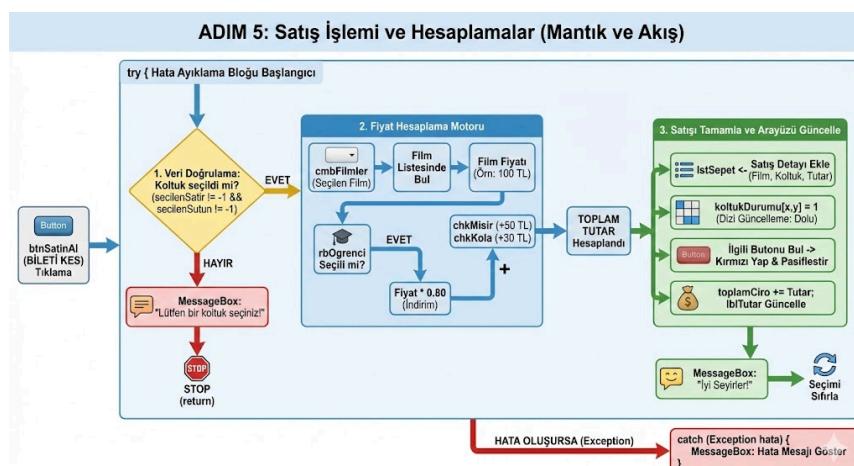
// Seçimi kaydet ve rengi değiştir
secilenSatir = satir;
secilenSutun = sutun;
lblSecilenKoltuk.Text = tiklanan.Text;

// Görsel geri bildirim (Sarı = Seçili)
tiklanan.BackColor = Color.Yellow;
}

```

## ADIM 5: Satış İşlemi ve Hesaplamalar

"Bilet Kes" butonuna (`btnSatinAl`) çift tıkla ve şu kodları yaz. Burada **Hata Ayıklama**, **Mantıksal Operatörler** ve **Veri Doğrulama** kullanacağız.



```

private void btnSatinAl_Click(object sender, EventArgs e)
{
    try // Hata Ayıklama Başlangıcı
    {
        // 1. Veri Doğrulama: Koltuk seçildi mi?
        if (secilenSatir == -1 || secilenSutun == -1)

```

```

{
    MessageBox.Show("Lütfen bir koltuk seçiniz!", "Uyarı");
    return;
}

// 2. Fiyat Hesaplama
string secilenFilmAdi = cmbFilmler.SelectedItem.ToString();
// List koleksiyonundan filmi bul (LINQ yerine döngü ile bulalım ki mantığı otursun)
Film secilenFilm = null;
foreach (Film f in filmListesi)
{
    if (f.Ad == secilenFilmAdi) secilenFilm = f;
}

double biletFiyati = secilenFilm.Fiyat;

// Karar İfadeleri: İndirim kontrolü
if (rb0grenci.Checked)
{
    biletFiyati = biletFiyati * 0.80; // %20 indirim
}

// Büfe Hesabı (Mantıksal Operatörler ve Aritmetik)
double ekstra = 0;
if (chkMisir.Checked) ekstra += 50;
if (chkKola.Checked) ekstra += 30;

double toplamTutar = biletFiyati + ekstra;

// 3. Satışı Tamamla ve Listeye Ekle
lstSepet.Items.Add($"Film: {secilenFilm.Ad} | Koltuk: {lblSecilenKoltuk.Text} | Tutar: {toplamTutar} TL");

// Çok Boyutlu Diziyi Güncelle (Koltuk artık dolu)
koltukDurumu[secilenSatir, secilenSutun] = 1;

// Buton Rengini Kırmızı Yap (Dolu)
foreach (Control ctrl in pnlKoltuklar.Controls)
{
    if (ctrl is Button btn && btn.Tag.ToString() == $"{secilenSatir},{secilenSutun}")
    {
        btn.BackColor = Color.Red;
        btn.Enabled = false; // Tıklanamaz yap
    }
}

```

```

        // Toplam Ciroyu Güncelle
        toplamCiro += toplamTutar;
        lblTutar.Text = toplamTutar.ToString() + " TL";

        MessageBox.Show("İyi Seyirler! Bilet oluşturuldu.");

        // Seçimi sıfırla
        secilenSatır = -1;
        secilenSutun = -1;
        lblSecilenKoltuk.Text = "...";
    }
    catch (Exception hata)
    {
        MessageBox.Show("Bir hata oluştu: " + hata.Message);
    }
}

```



## Özet: Hangi Konuyu Nerede Kullandın?

Bu projeyi tamamladığında, aşağıdaki kavramları sadece ezberlememiş, gerçek hayatta kullanmış olacaksın:

Konu	Projedeki Karşılığı
Sınıflar ve Nesneler	<code>Film</code> sınıfı oluşturup, <code>new Film(...)</code> ile filmleri nesneye dönüştürdü.
List Koleksiyonu	Sinemadaki tüm filmleri <code>List&lt;Film&gt;</code> içinde tuttu.
Çok Boyutlu Diziler	Koltuk düzenini ( $5 \times 5$ ) <code>koltukDurumu[5,5]</code> matrisi ile yönetti.
Döngüler (Nested Loops)	25 tane butonu tek tek elle koymak yerine <code>for</code> döngüsüyle kodun kendisine çizdirdi.
Karar İfadeleri (If-Else)	Öğrenci mi Tam mı? Koltuk dolu mu boş mu? sorularına yanıt verdi.
Olaylar (Events)	Dinamik oluşturulan butonlara <code>Click</code> olayı bağladı.
Hata Ayıklama (Try-Catch)	Olası hatalara karşı <code>try-catch</code> bloğu ile programın çökmesini engelledi.
Veri Bağlama	Kod arkasındaki verileri <code>ComboBox</code> ve <code>ListBox</code> araçlarına bağladı.

**Tebrikler!** Artık sadece kod yazan biri değil, bir otomasyon geliştirebilen bir yazılımcısın.



## Proje-2: Notmatik (Basit Not Hesaplama)

**Amaç:** Öğrencinin Adını, Vize ve Final notunu girip ortalamasını hesaplamak, dersten geçip geçmediğini bulmak ve listeye eklemek.

### BÖLÜM 1: Arayüz (Form) Tasarımı

Öğrencilerinden formu şu şekilde tasarlamalarını iste. Bu tasarım daha sade ve anlaşılır.

### Form Ayarları:

- **Text:** Notmatik v1.0
- **Size:** 400, 500
- **StartPosition:** CenterScreen

### Araçlar (Toolbox):

1. **Label (3 Adet):** "Ad Soyad:", "Vize (%40):", "Final (%60):"

2. **TextBox (3 Adet):**

- `txtAdSoyad`
- `txtVize`
- `txtFinal`

3. **Button (1 Adet):**

- `btnHesapla` (Text: HESAPLA VE EKLE)

4. **Label (Sonuç için):**

- `lblSonuc` (Büyük fontlu, başlangıçta "Durum: ..." yazsın)

5. **ListBox (1 Adet):**

- `lstKayitlar` (Geçmiş hesaplamaları tutacak)

---

## BÖLÜM 2: Kodlama (Adım Adım)

Bu kodlarda **Değişkenler**, **Tür Dönüşümü**, **Matematiksel İşlemler** ve **If-Else** yapılarına odaklanıyoruz.

Kodu `btnHesapla` butonuna çift tıklayarak yazdırıyoruz.

```
private void btnHesapla_Click(object sender, EventArgs e)
{
    try
    {
        // 1. ADIM: Değişken Tanımlama ve Veri Alma
        // Kullanıcının girdiği yazıları (string) sayıya çevirmeliyiz.
        string adSoyad = txtAdSoyad.Text;

        // Veri Doğrulama: Kutular boş mu?
        if (txtVize.Text == "" || txtFinal.Text == "")
        {
            MessageBox.Show("Lütfen notları giriniz!");
            return; // İşlemi durdur
        }

        double vize = Convert.ToDouble(txtVize.Text);
        double final = Convert.ToDouble(txtFinal.Text);
```

```

// 2. ADIM: Matematiksel İşlemler (İşlem Önceliği)
// Vizenin %40'ı, Finalin %60'ı alınır.
double ortalama = (vize * 0.4) + (final * 0.6);

// 3. ADIM: Karar Yapıları (If-Else)
string durum = "";

if (ortalama >= 50)
{
    durum = "GEÇTİ";
    lblSonuc.ForeColor = Color.Green; // Yazı rengi yeşil olsun
}
else
{
    durum = "KALDI";
    lblSonuc.ForeColor = Color.Red; // Yazı rengi kırmızı olsun
}

// 4. ADIM: Sonucu Ekrana Yazdırma
lblSonuc.Text = "Ort: " + ortalama.ToString() + " - " + durum;

// 5. ADIM: Listeye Ekleme (ListBox)
// Formatlı yazdırma ($ işaretini ile)
lstKayitlar.Items.Add($"{{adSoyad}} -> Ort: {{ortalama}} ({durum}}");

// 6. ADIM: Temizlik (İsteğe bağlı)
txtAdSoyad.Clear();
txtVize.Clear();
txtFinal.Clear();
txtAdSoyad.Focus(); // İmleci tekrar isme getir
}
catch (Exception)
{
    // Hata Ayıklama: Sayı yerine harf girilirse burası çalışır
    MessageBox.Show("Lütfen not kısmına sadece sayı giriniz!", "Hata");
}
}

```



## Neden Bu Proje Daha Basit?

Önceki projeye (Sinema) kıyasılsak:

Özellik	Sinema Projesi (Zor)	Notmatik Projesi (Kolay)
<b>Veri Yapısı</b>	Class (Sınıf) ve List<Film>	Basit Değişkenler (double, string)
<b>Döngüler</b>	İç içe For döngüleri (Matris)	Döngü yok (Sadece akış var)
<b>Diziler</b>	Çok Boyutlu Dizi [,]	Dizi yok

**Görsellik**

Dinamik buton oluşturma

Sabit araçlar (Sürükle-Bırak)