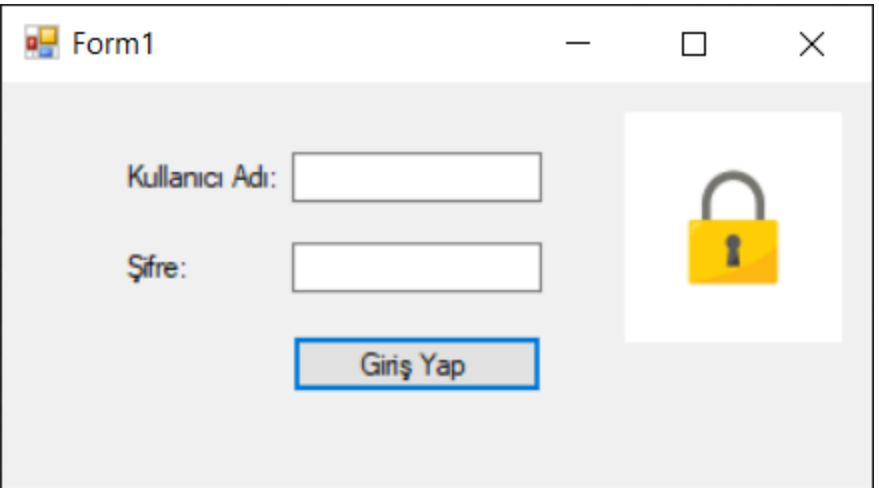
# EMM3115
# İLERİ PROGRAMLAMA – C#
# FORM UYGULAMALARI

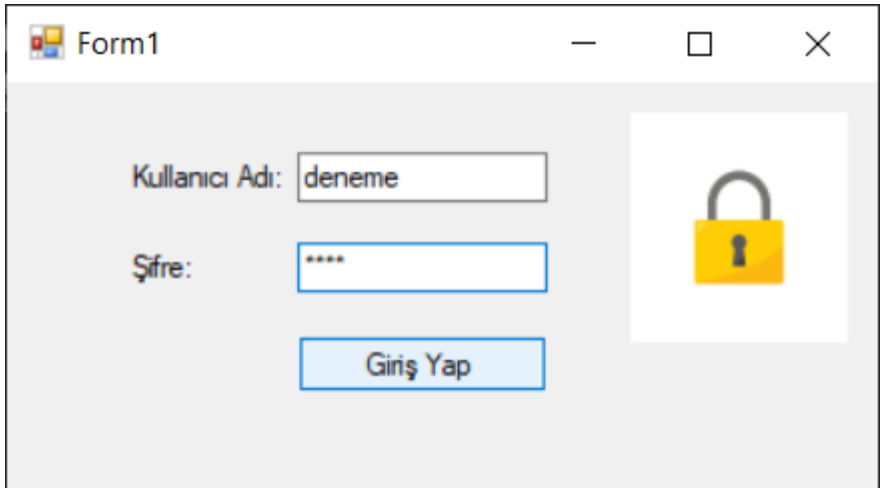
3

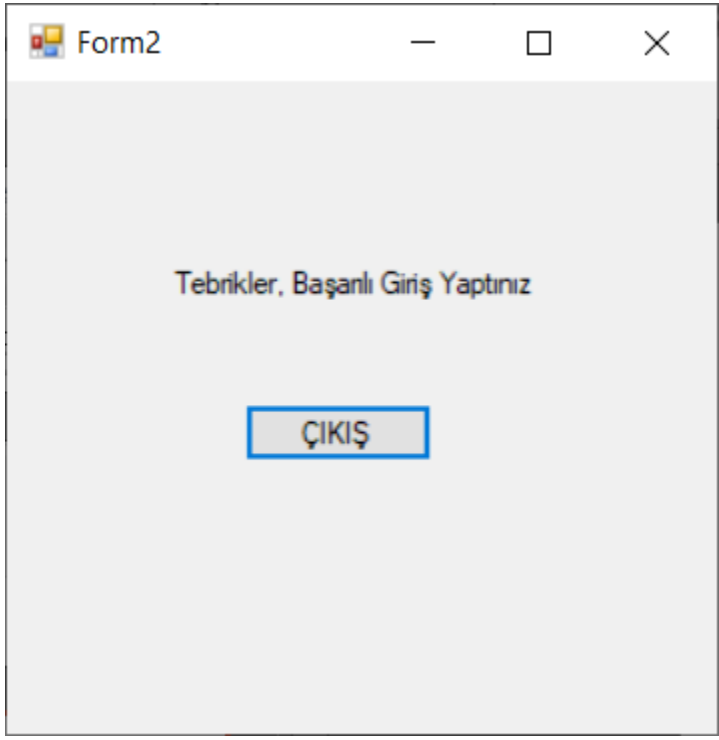
Doç. Dr. İbrahim KÜÇÜKKOÇ

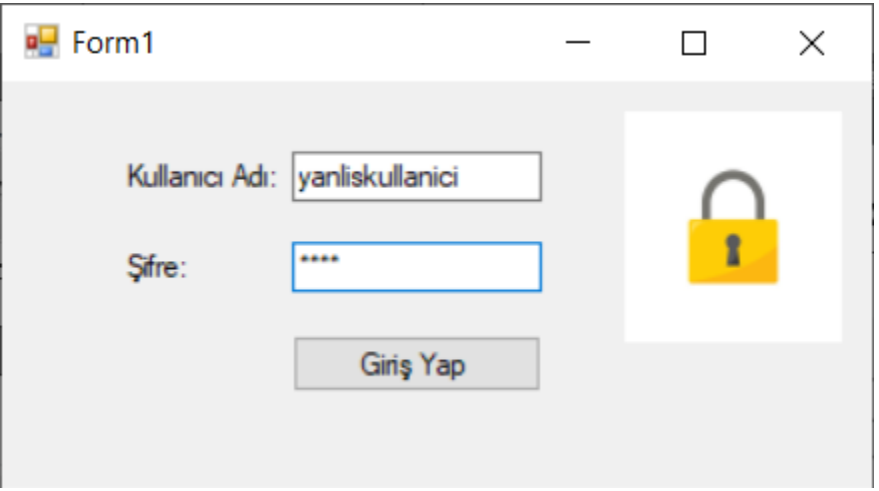
Web: http://ikucukkoc.baun.edu.tr

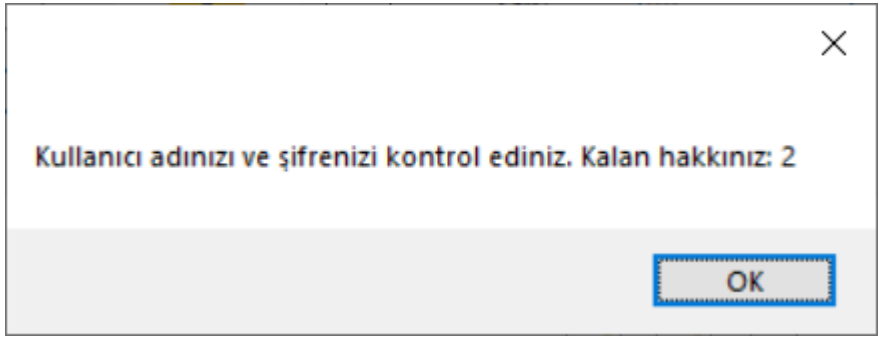
Email: ikucukkoc@balikesir.edu.tr

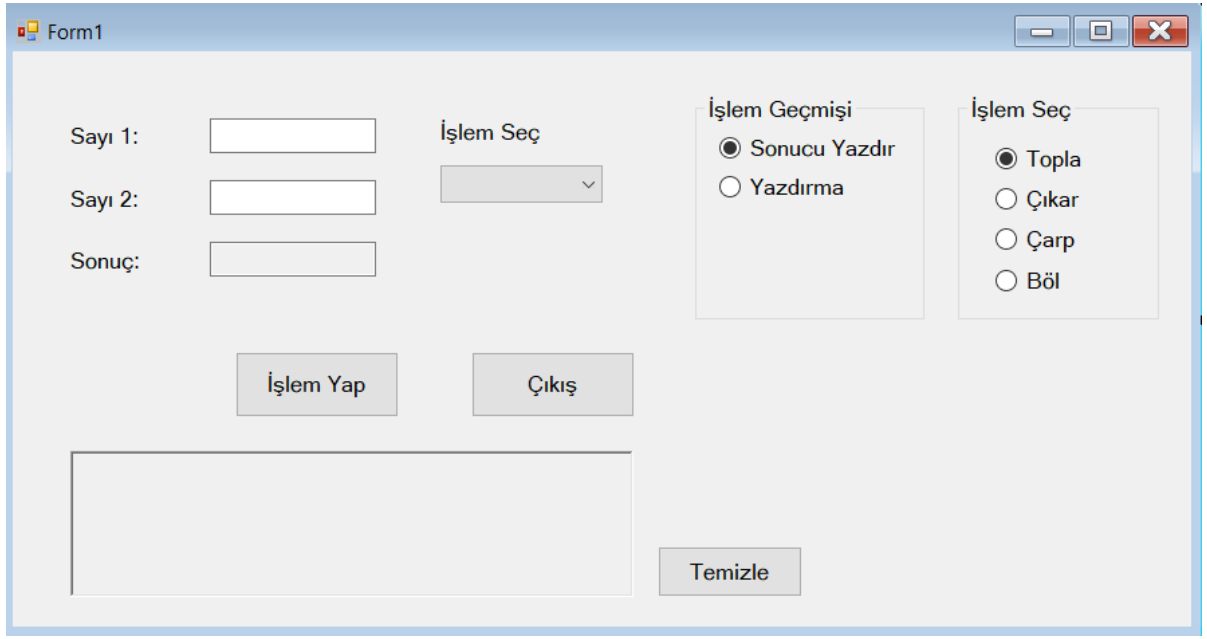

Form1
Kullanıcı Adı:
Şifre:
Giriş Yap


Form1
Kullanıcı Adı:
deneme
Şifre:
****
Giriş Yap


Form2
Tebrikler, Başanlı Giriş Yaptınız
ÇIKIŞ


Form1
Kullanıcı Adı:
yanliskullanici
Şifre:
****
Giriş Yap


Kullanıcı adınızı ve şifrenizi kontrol ediniz. Kalan hakkınız: 2
OK



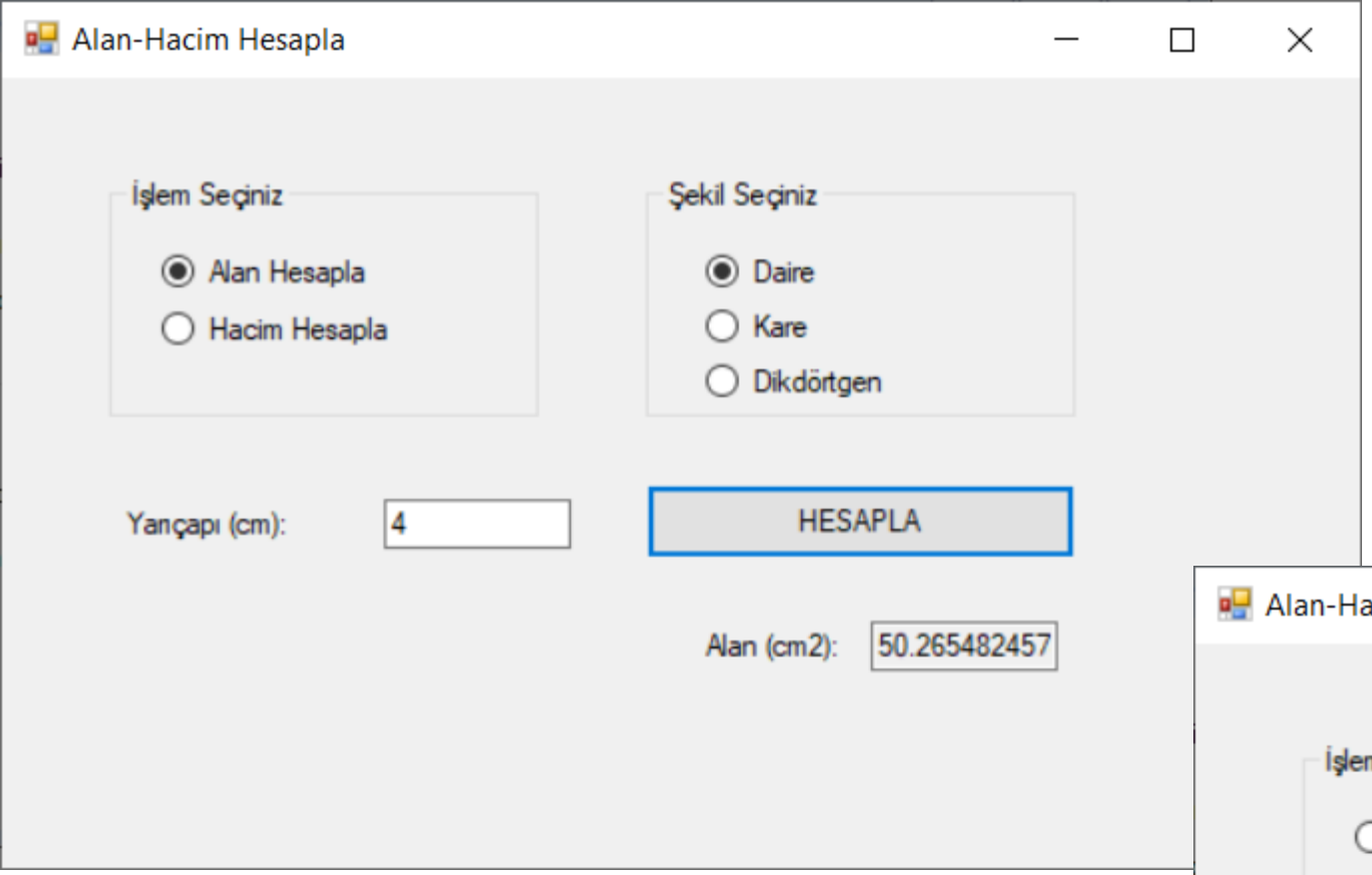

```
private void button2_Click(object sender, EventArgs e)
{//ÇIKIŞ
    //Application.Exit();

    DialogResult sonuc;
    sonuc = MessageBox.Show("Çıkmak istediğinize emin misiniz?","Dikkat",MessageBoxButtons.OKCancel,MessageBoxIcon.Warning );
    if (sonuc == DialogResult.OK)
    {
        Application.Exit();//Uygulamadan tamamen çıkar
        //Form1.ActiveForm.Close();//Aktif formu kapat
    }
}
```

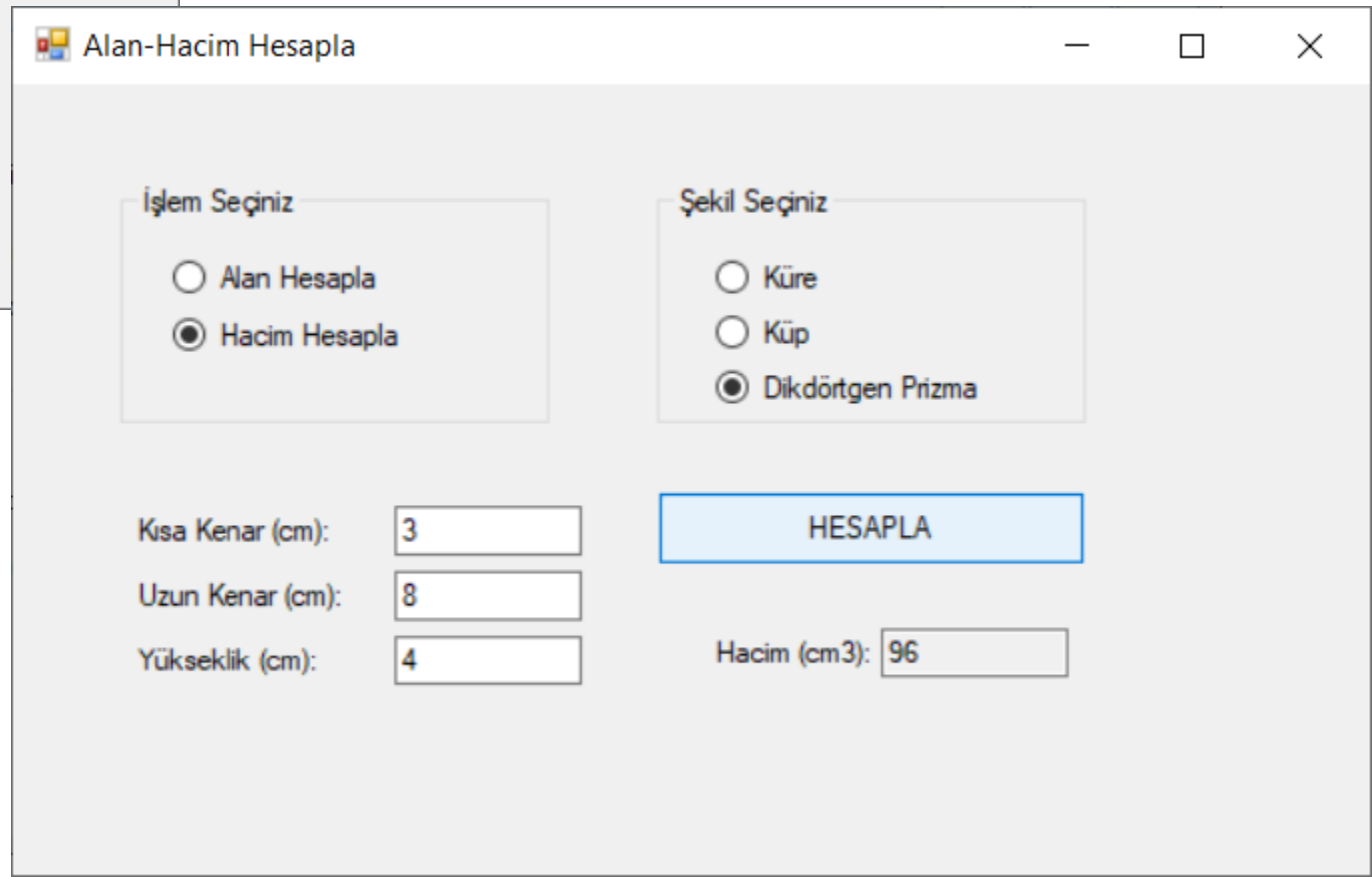
```
private void button1_Click(object sender, EventArgs e)
{//İşlem Yap
    double sayi1 = -1;
    double sayi2 = -1;

    if (textBox1.Text != "")
    {
        sayi1 = Convert.ToDouble(textBox1.Text);
    }
    if (textBox2.Text != "")
    {
        sayi2 = Convert.ToDouble(textBox2.Text);
    }

    double sonuc = -1;
    String islem = "";

    /*
    if (radioButton1.Checked == true)//topla
    {
        sonuc = sayi1 + sayi2;
        islem = "+";
    }
    else if (radioButton2.Checked == true)//çıkar
    {
        sonuc = sayi1 - sayi2;
        islem = "-";
    }
    else if (radioButton3.Checked == true)//çarp
    {
        sonuc = sayi1 * sayi2;
        islem = "x";
    }
    else if (radioButton4.Checked == true)//böl
    {
        sonuc = sayi1/sayi2;
        islem = "/";
    }*/
```

```
if (comboBox1.Text == "Topla")
{
    sonuc = sayi1 + sayi2;
    islem = "+";
} else if (comboBox1.Text == "Çıkar")
{
   sonuc = sayi1 - sayi2;
    islem = "-";
}
else if (comboBox1.Text == "Çarp")
{
    sonuc = sayi1 * sayi2;
    islem = "*";
}
else if (comboBox1.Text == "Böl")
{
    sonuc = sayi1 / sayi2;
    islem = "/";
}
else if (comboBox1.Text == "Karekök Al")
{
    sonuc = Math.Sqrt(sayi1);
    islem = "Karekök";
}
else if (comboBox1.Text == "Kuvvetini Al")
{
    sonuc = Math.Pow(sayi1,sayi2);
    islem = "^";
}

//sonuc = Math.Round(sonuc,4);
```

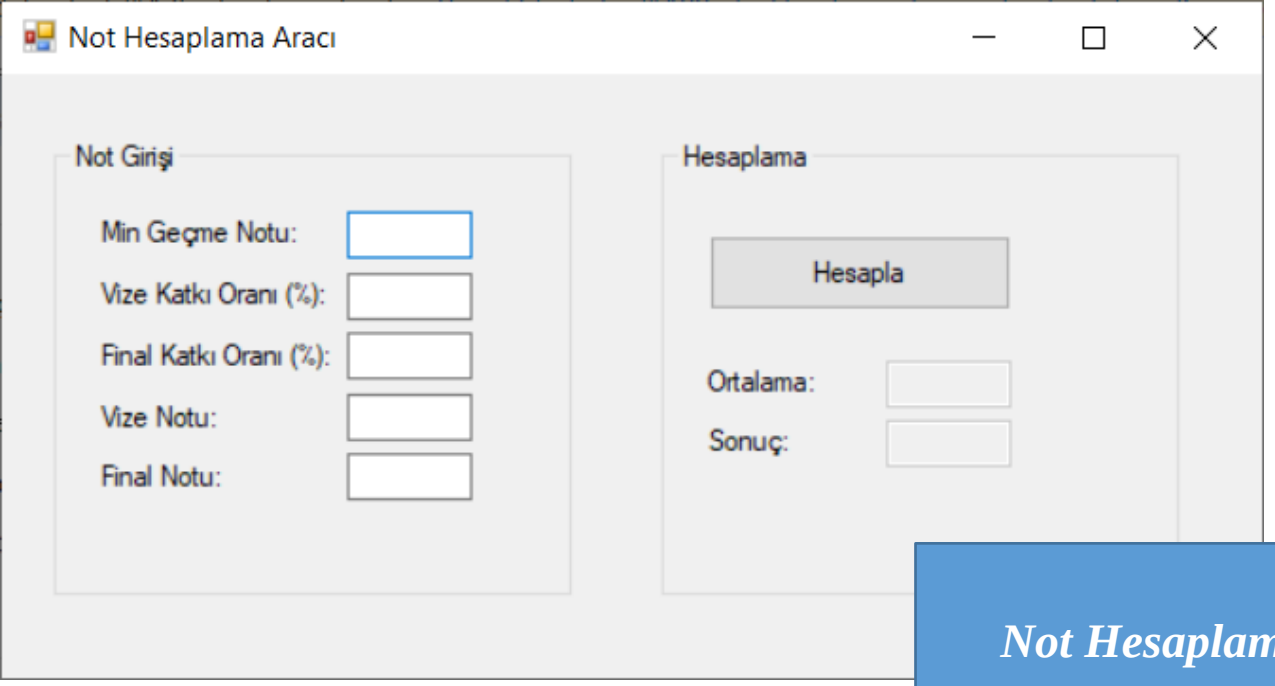
```
    //işlem geçmişi
    if (radioButton5.Checked)
    {
        if (comboBox1.Text == "Karekök Al")
        {
            richTextBox1.AppendText("Karekök("+sayi1.ToString()+")=" + sonuc + Environment.NewLine);
        }
        else
        {
            richTextBox1.AppendText(sayi1.ToString() + islem + sayi2.ToString() + "=" + sonuc + Environment.NewLine);
        }
    }
}


private void button3_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
    //richTextBox1.Text = "";
}

private void Form1_Load(object sender, EventArgs e)
{
    comboBox1.SelectedIndex = 0;
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.Text == "Karekök Al")
    {
        textBox2.Text = "";
        //textBox2.Clear();
        textBox2.ReadOnly = true;

    }
}
```

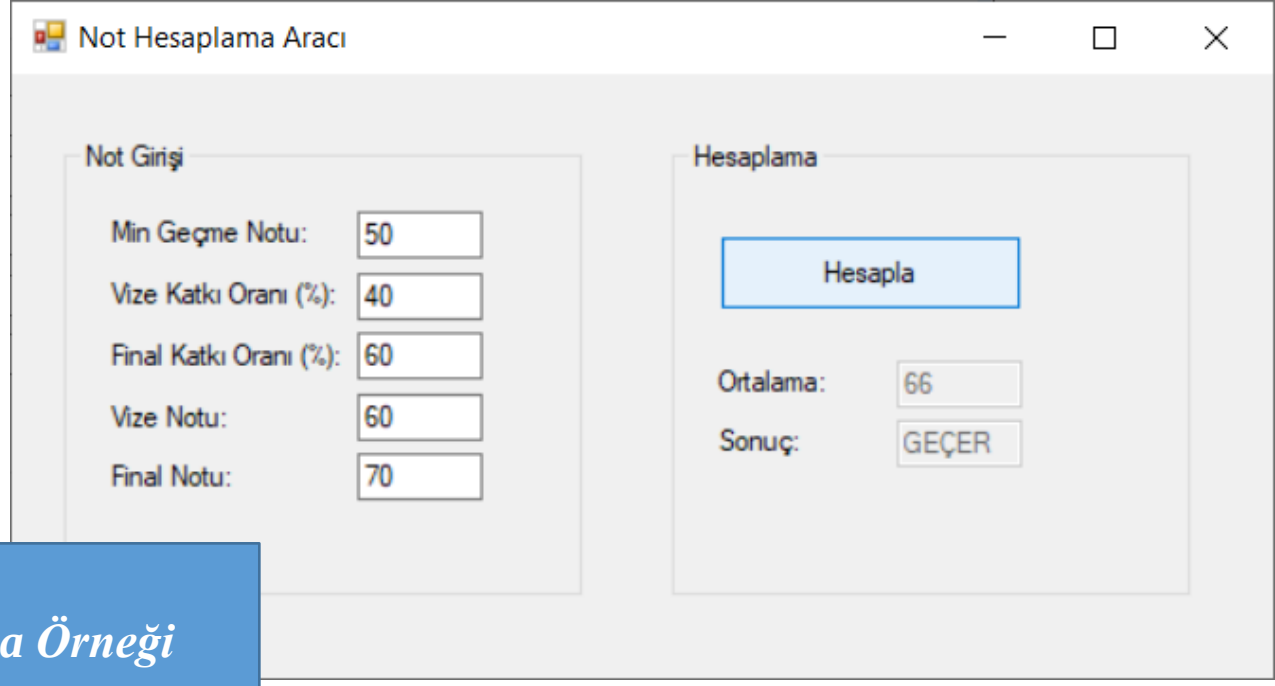
# Alan-Hacim Hesaplama Örneği

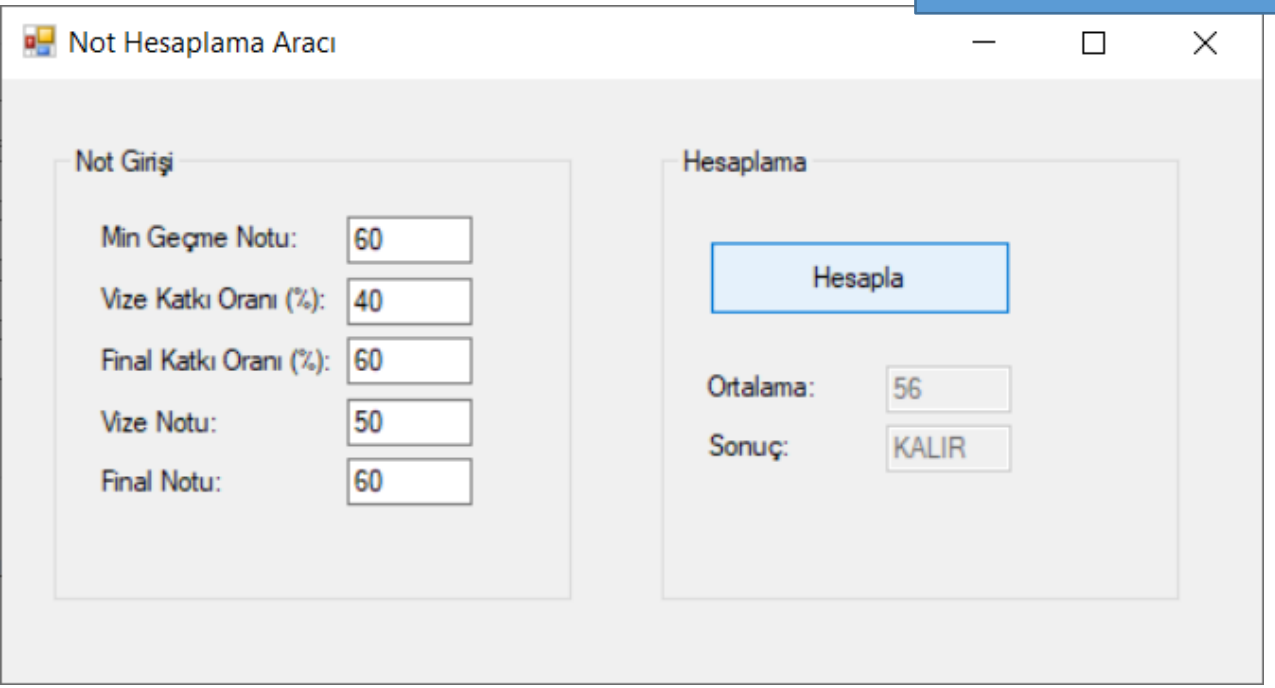


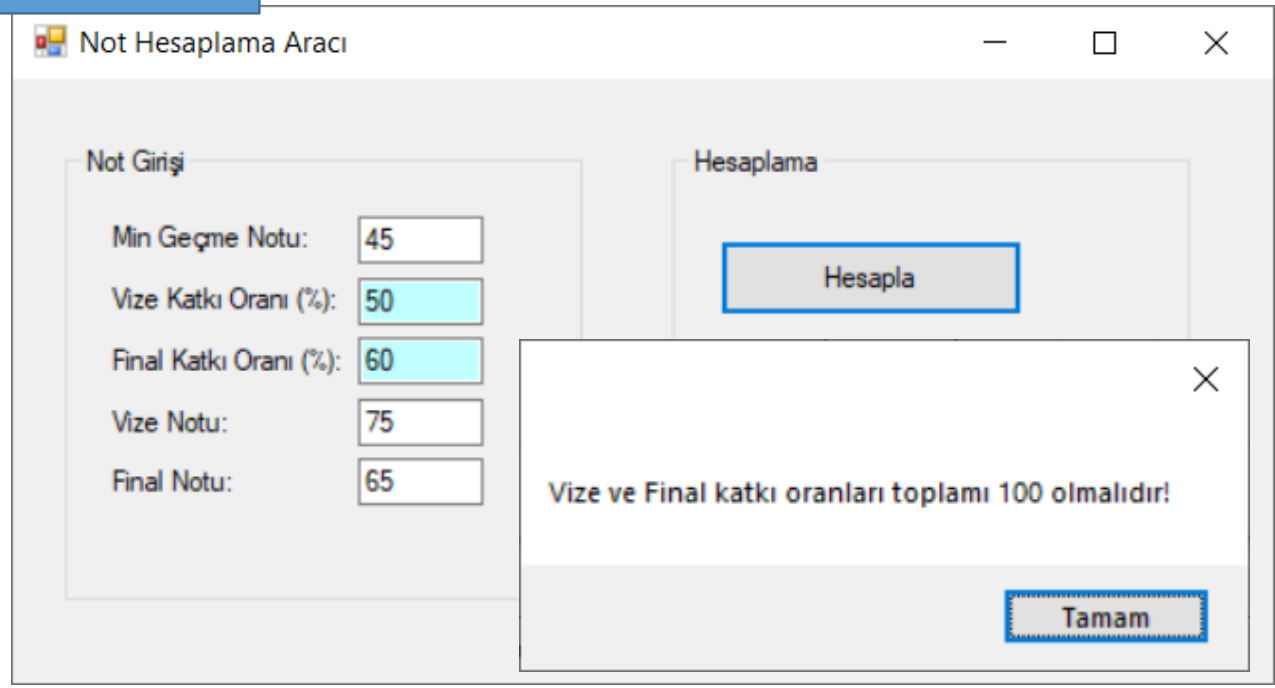



# *Not Hesaplama Örneği*

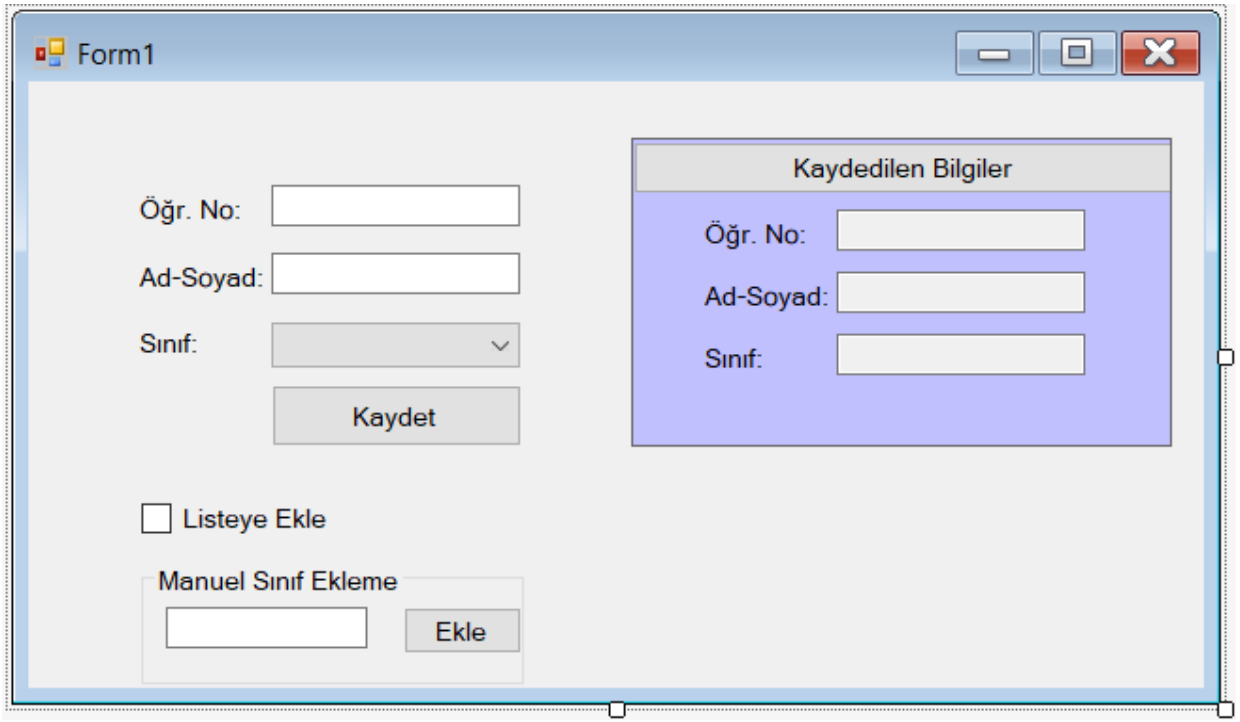


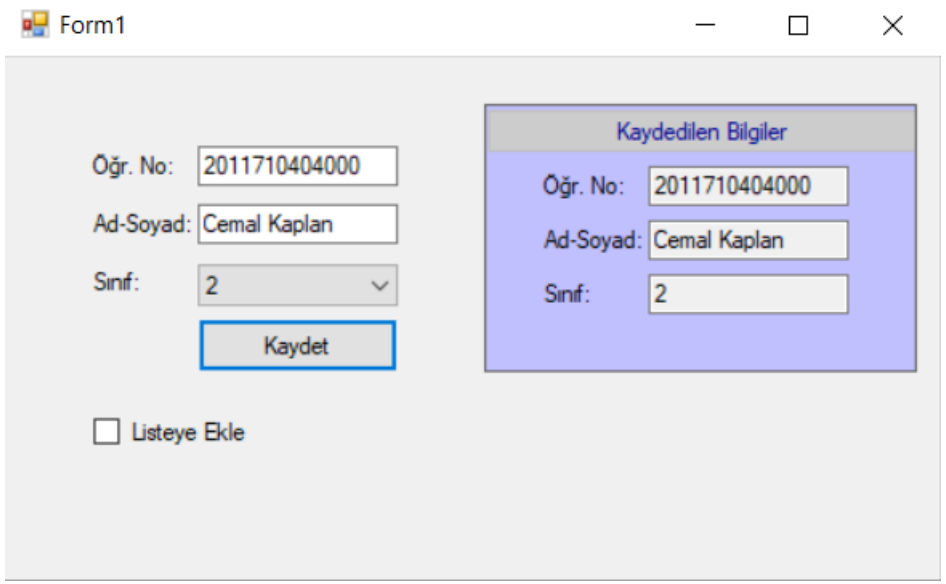



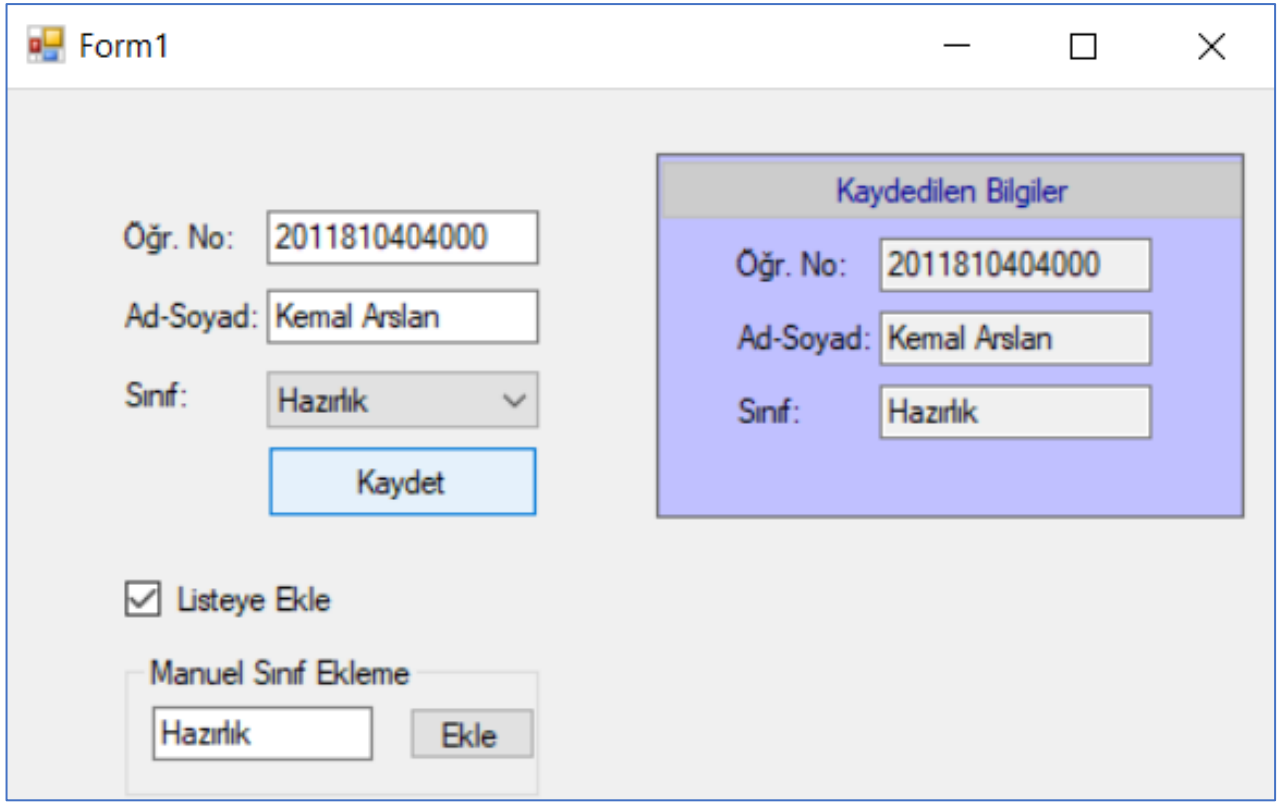



# CS3-1

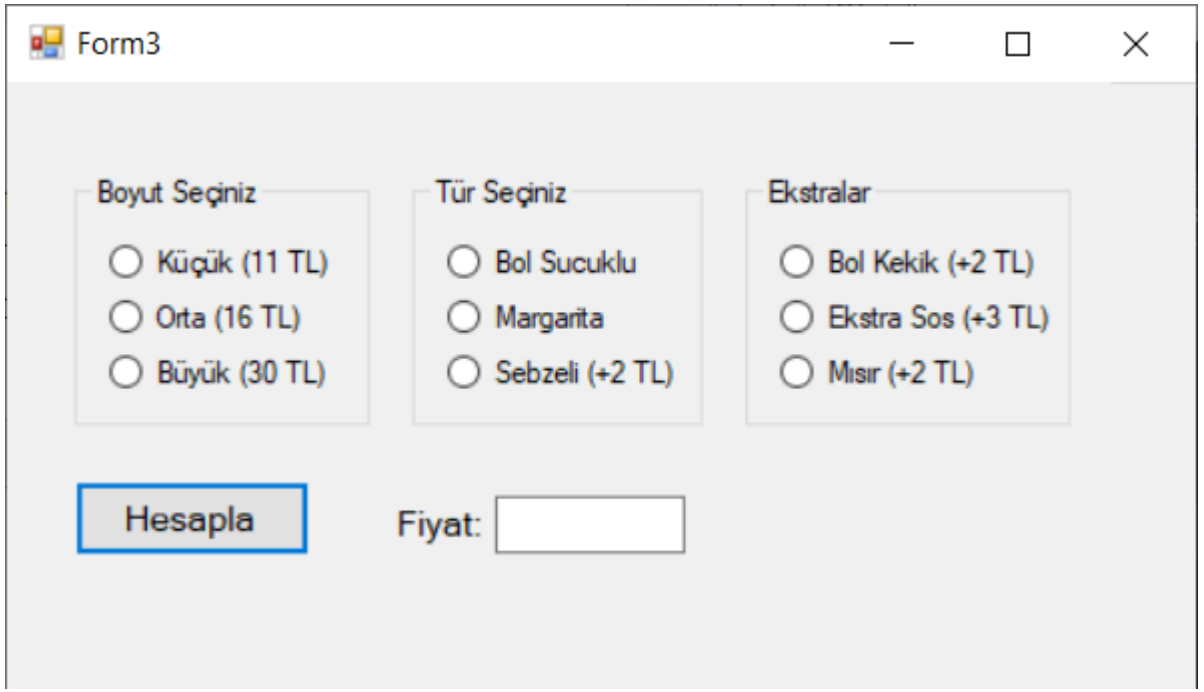

Form1
Öğr. No:
Ad-Soyad:
Sınıf:
Kaydet
Kaydedilen Bilgiler
Öğr. No:
Ad-Soyad:
Sınıf:
Listeye Ekle
Manuel Sınıf Ekleme
Ekle


Form1
Öğr. No:
2011710404000
Ad-Soyad:
Cemal Kaplan
Sınıf:
2
Kaydet
Listeye Ekle
Kaydedilen Bilgiler
Öğr. No:
2011710404000
Ad-Soyad:
Cemal Kaplan
Sınıf:
2

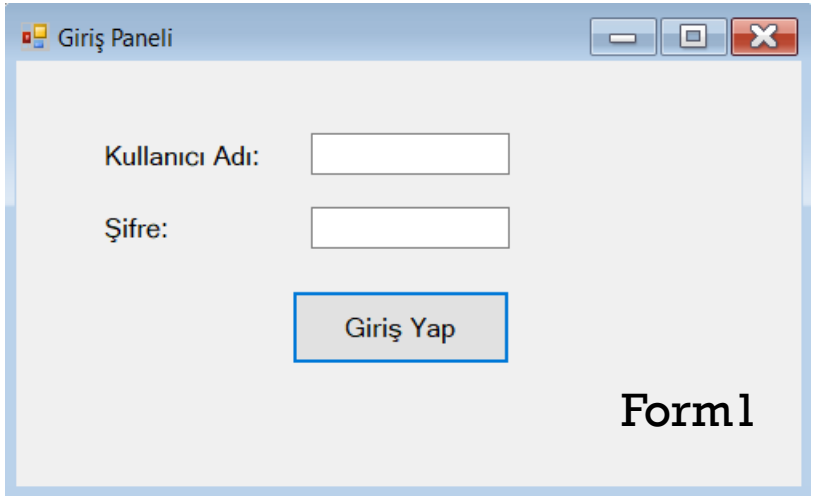

Form1
Öğr. No:
2011810404000
Ad-Soyad:
Kemal Arslan
Sınıf:
Hazırlık
Kaydet
Listeye Ekle
Manuel Sınıf Ekleme
Hazırlık
Ekle
Kaydedilen Bilgiler
Öğr. No:
2011810404000
Ad-Soyad:
Kemal Arslan
Sınıf:
Hazırlık

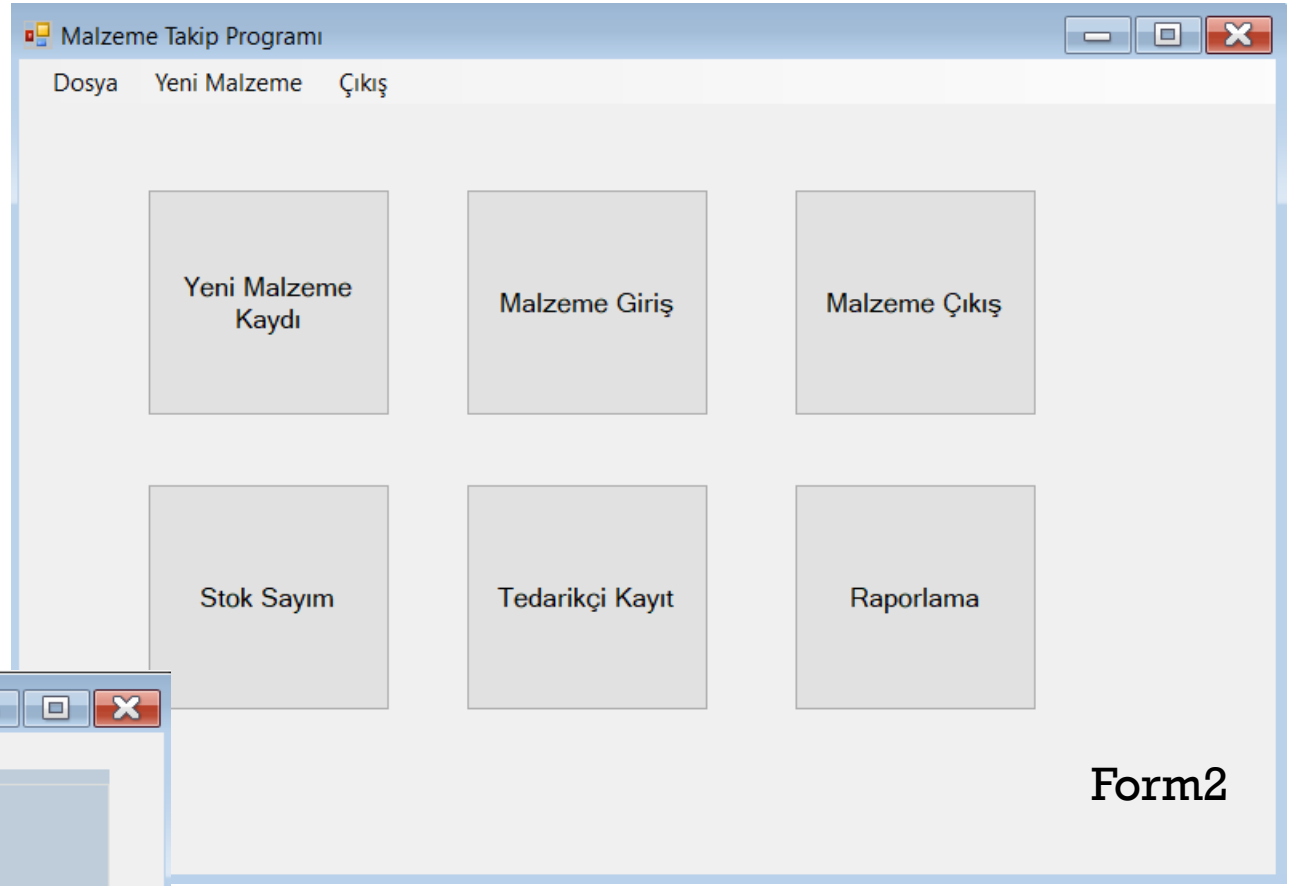
```
private void button1_Click(object sender, EventArgs e)
{
    panel1.Visible = true;
    textBox4.Text = textBox1.Text;
    textBox3.Text = textBox2.Text;
    textBox5.Text = comboBox1.Text;
}


private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked == true) {
        groupBox1.Visible = true;
    } else
    {
        groupBox1.Visible = false;
    }
}

private void button3_Click(object sender, EventArgs e)
{
    comboBox1.Items.Add(textBox6.Text);
}
```

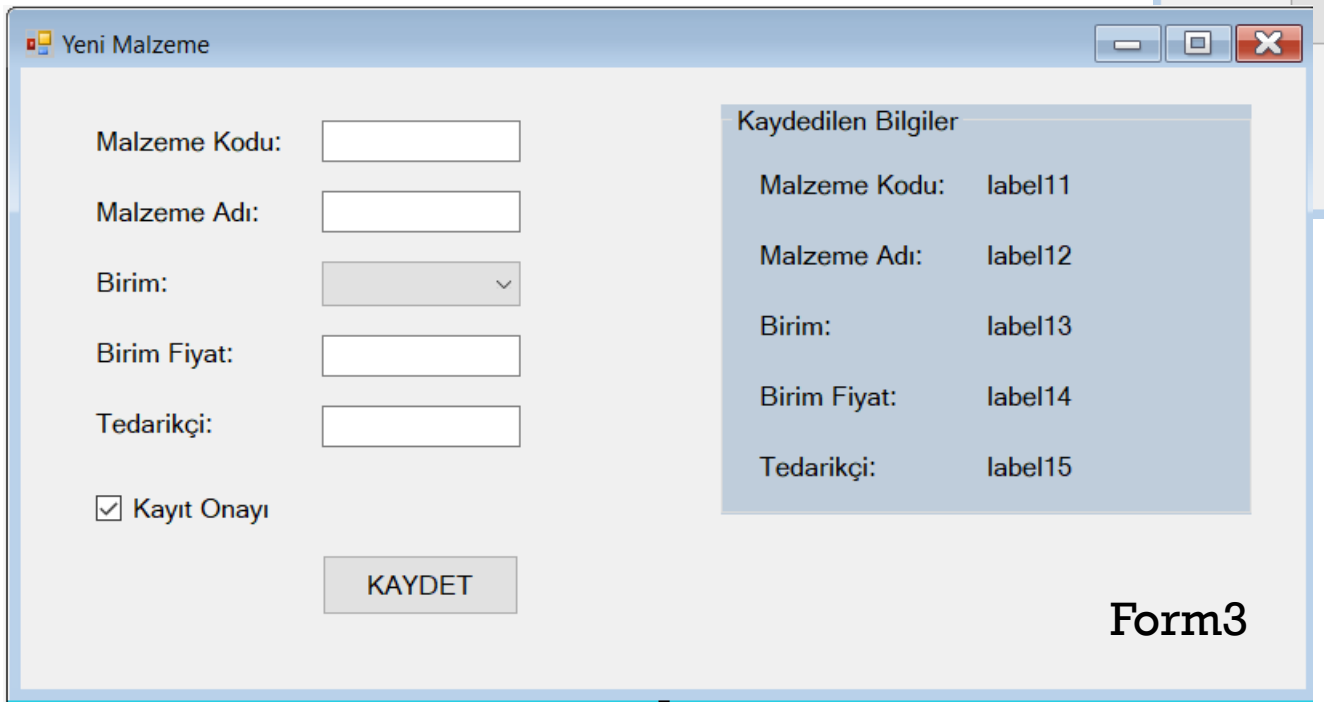
# Pizza Sipariş - ÖDEV



# YENİ FORM -Uygulama



```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    String kullaniciAdi = "admin";
    string sifre = "9876";
    int sayac = 0;

    private void button1_Click(object sender, EventArgs e)
    {
        if (textBox1.Text == kullaniciAdi && textBox2.Text == sifre)
        {
            Form2 frm2 = new Form2();
            frm2.Show();
            this.Hide();
        }
        else
        {
            sayac++;
            MessageBox.Show("Kullanıcı adınız ve/veya şifreniz yanlıştır. Kalan deneme hakkınız: " + (3-sayac), "Hata!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            if (sayac == 3)
            {
                MessageBox.Show("Başka deneme hakkınız kalmadı. Program sonlandırılıyor.");
                Application.Exit();
            }
        }
    }
}
```

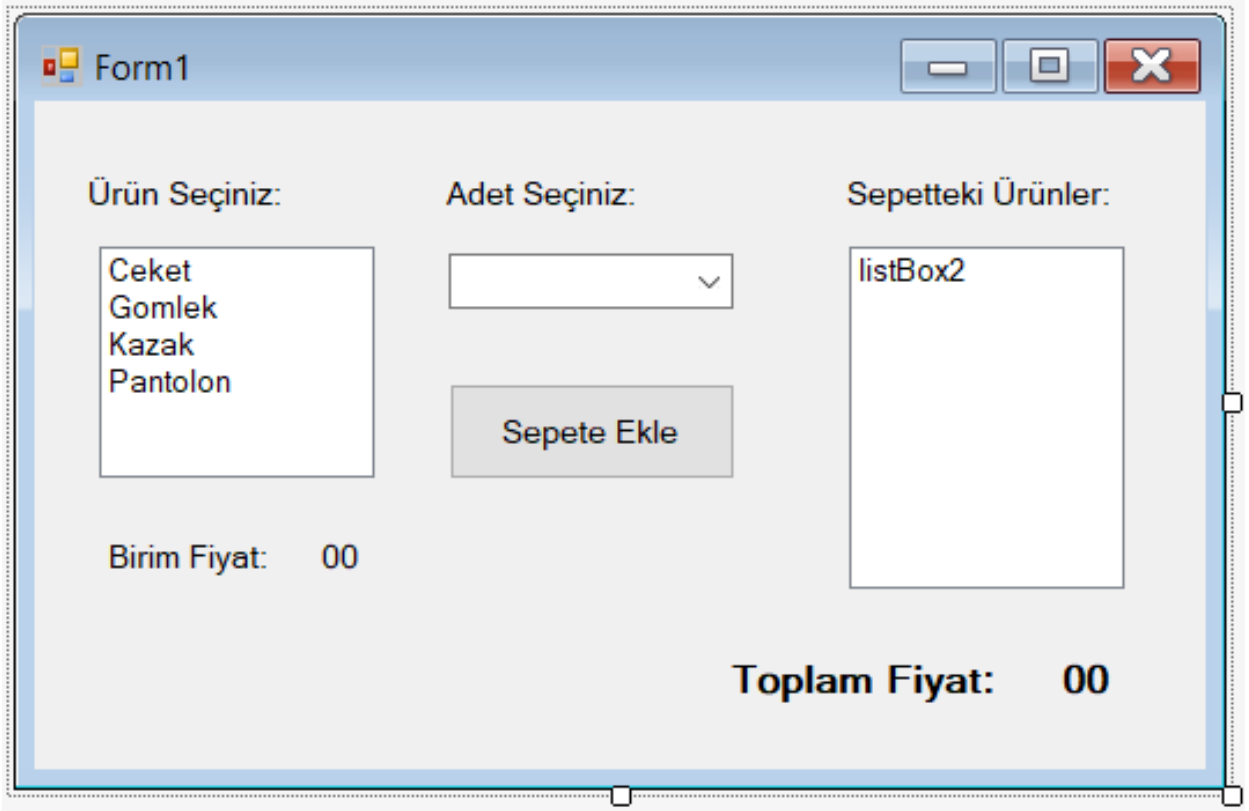
```
public partial class Form2 : Form
{
    public Form2()
    {
        InitializeComponent();
    }

    private void çıkışToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void Form2_FormClosing(object sender, FormClosingEventArgs e)
    {
        Application.Exit();
    }

    private void yeniMalzemeToolStripMenuItem_Click(object sender, EventArgs e)
    {
        button1.PerformClick();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form3 frm3 = new Form3();
        frm3.Show();
        //this.Hide();
    }
}
```

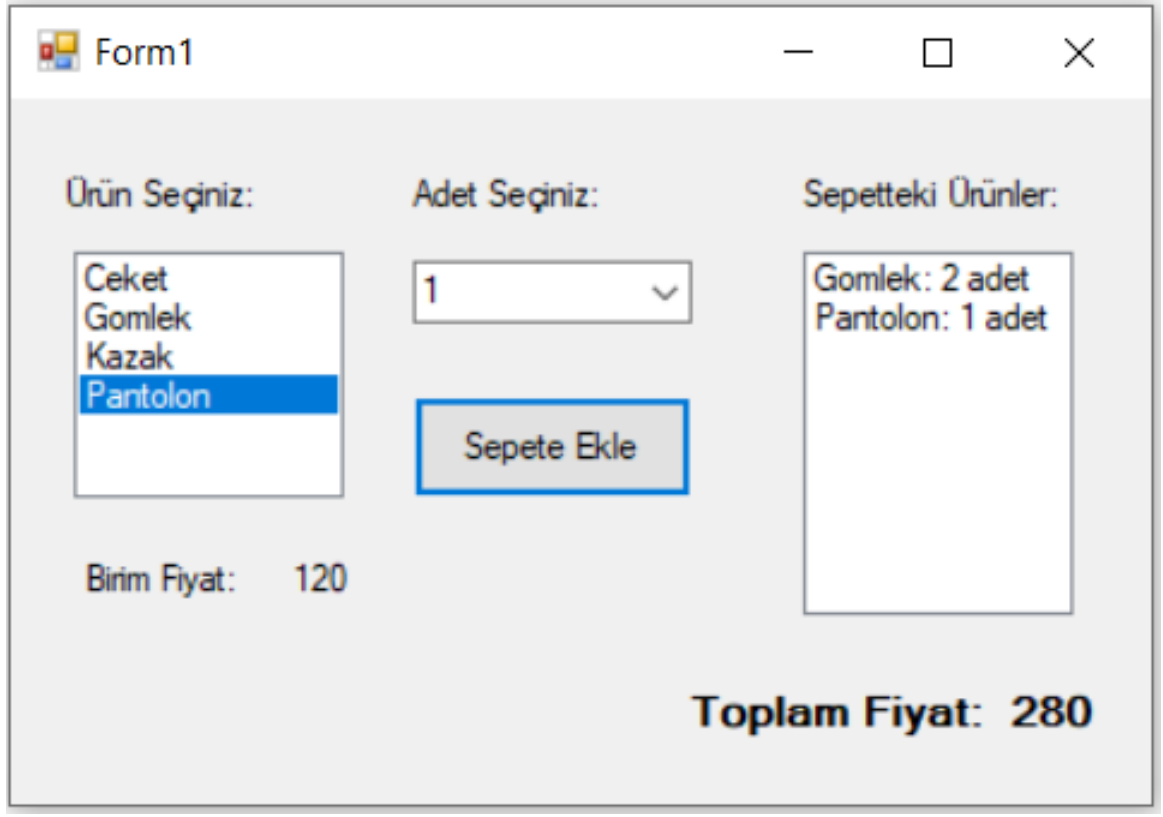
```
public partial class Form3 : Form
{
    public Form3()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        String malzemeKodu = textBox1.Text;
        String malzemeAdi = textBox2.Text;
        String birim = comboBox1.Text;
        String birimFiyat = textBox3.Text;
        String tedarikci = textBox4.Text;

        label11.Text = malzemeKodu;
        label12.Text = malzemeAdi;
        label13.Text = birim;
        label14.Text = birimFiyat;
        label15.Text = tedarikci;

        groupBox1.Visible = true;

        if (checkBox1.Checked)
        {
            MessageBox.Show("Yeni malzeme bilgisi başarılı " +
                "bir şekilde kayıt edilmiştir.");
        }
    }
```

```
    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        if (System.Text.RegularExpressions.Regex.IsMatch(textBox1.Text, "[^0-9]"))
        {
            MessageBox.Show("Lütfen sadece rakam giriniz");
            textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1);
        }

    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {
        if (System.Text.RegularExpressions.Regex.IsMatch(textBox3.Text, "[^0-9.]"))
        {
            MessageBox.Show("Lütfen sadece rakam giriniz");
            textBox3.Text = "";
            //textBox3.Text = textBox3.Text.Remove(textBox3.Text.Length - 1);
        }
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {
        if (System.Text.RegularExpressions.Regex.IsMatch(textBox2.Text, "[^A-z0-9]"))
        {
            MessageBox.Show("Lütfen sadece harf giriniz");
            textBox2.Text = "";
        }
    }
}
```

# CS3-2 – Sepete Ekle

- Bu projede Combobox ve ListBox kullanımına yönelik bir çalışma çapılacaktır. ListBox1'den seçilen ürünü, ComboBox'tan seçilen adet ile birlikte ListBox2'ye ekleyeceğiz. Bunu yaparken, butona her tıklandığında, ürünlerin birim fiyat ve seçilen adet bilgisi kullanılarak Toplam Fiyat bilgisi de güncellenecektir. Birim fiyatlar aşağıda verilmiştir.
- Ceket: 200 PB, Gömlek: 80 PB, Kazak: 100 PB, Pantolon: 120 PB.



```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox1.Text == "Ceket")
    {
        label5.Text = "200";
    } else if (listBox1.Text == "Gomlek")
    {
        label5.Text = "80";
    } else if (listBox1.Text == "Kazak")
    {
        label5.Text = "100";
    } else if (listBox1.Text == "Pantolon")
    {
        label5.Text = "120";
    }
}
```

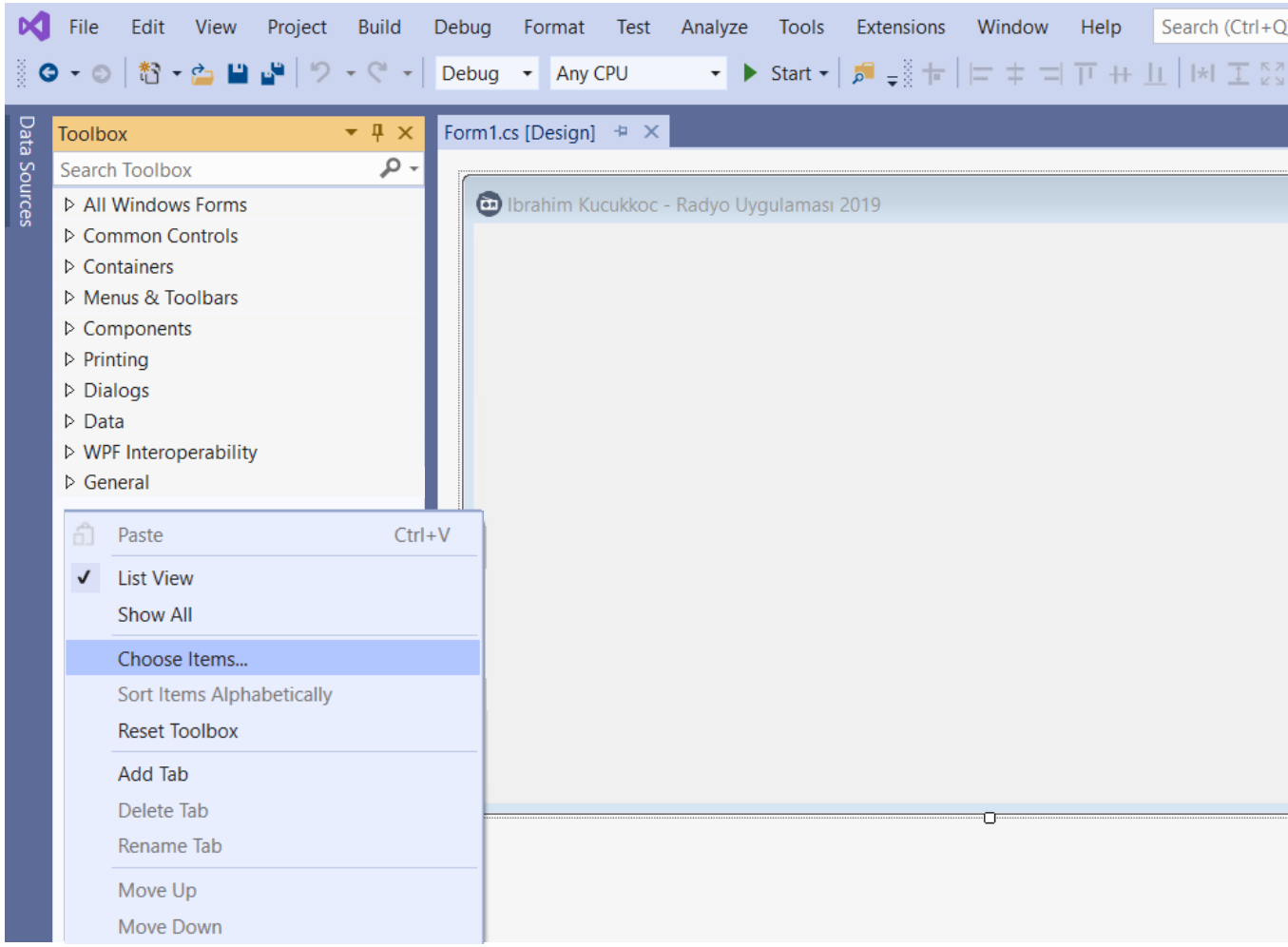
```
int toplam = 0;
private void button1_Click(object sender, EventArgs e)
{
    listBox2.Items.Add(listBox1.Text + ": " + comboBox1.Text + " adet");
    int birimfiyat = 0;
    if (listBox1.Text == "Ceket")
    {
        birimfiyat = 200;
    }
    else if (listBox1.Text == "Gomlek")
    {
        birimfiyat = 80;
    }
    else if (listBox1.Text == "Kazak")
    {
        birimfiyat = 100;
    }
    else if (listBox1.Text == "Pantolon")
    {
        birimfiyat = 120;
    }
    int adet = Convert.ToInt32(comboBox1.Text);
    toplam = toplam + birimfiyat * adet;
    label6.Text = Convert.ToString(toplam);
}
```

# Radyo Programı

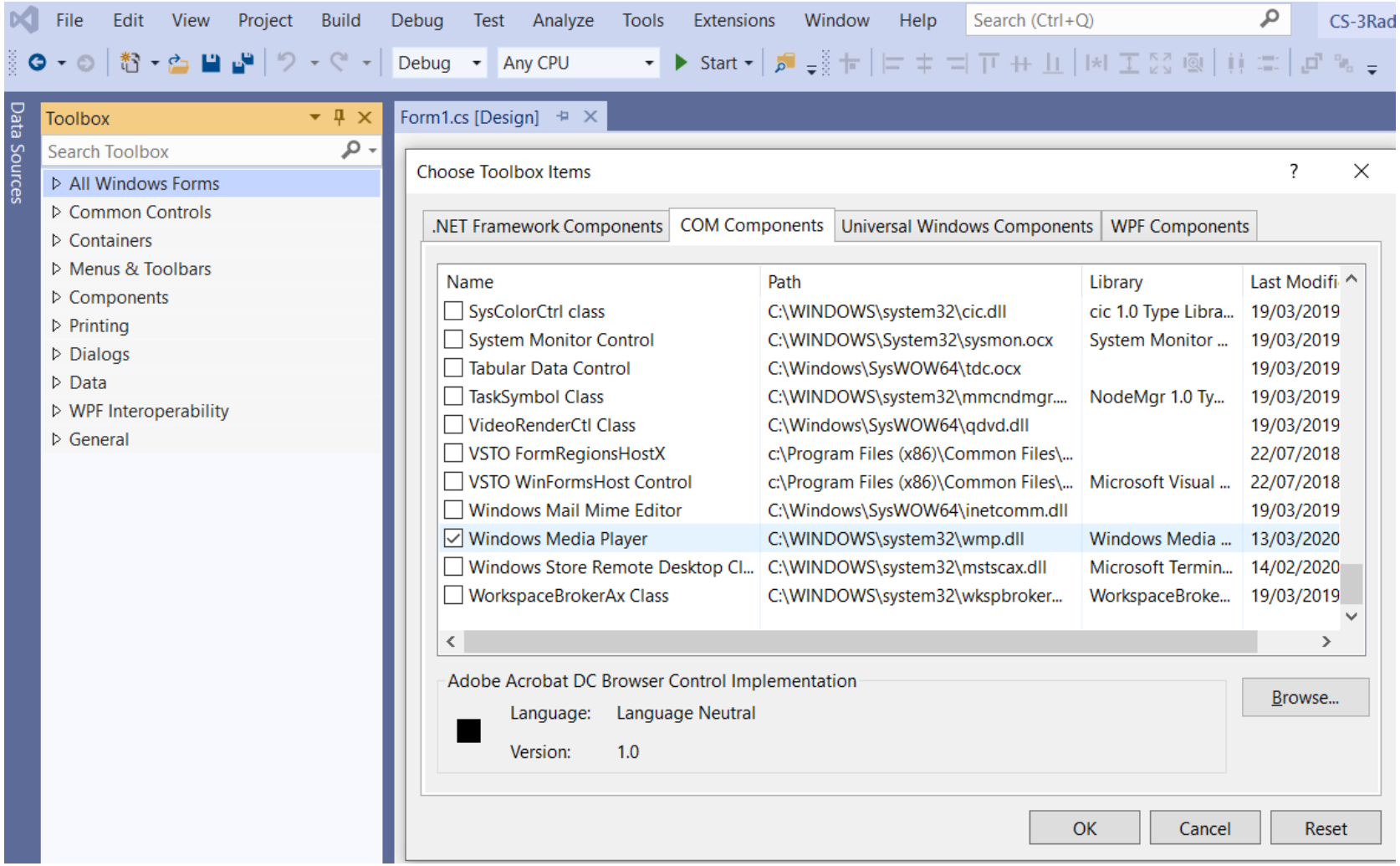
- Windows’un kendi müzik çaları olan Windows Media Player’ı C# içerisinde kullanarak mp3, wav, wmv, avi gibi dosyaları yürütebileceğiniz basit bir müzik çalar yapalım.

- Windows’un kendi müzik çaları olan Windows Media Player’ı C# içerisinde kullanarak mp3, wav, wmv, avi gibi dosyaları yürütebileceğiniz basit bir müzik çalar yapalım.

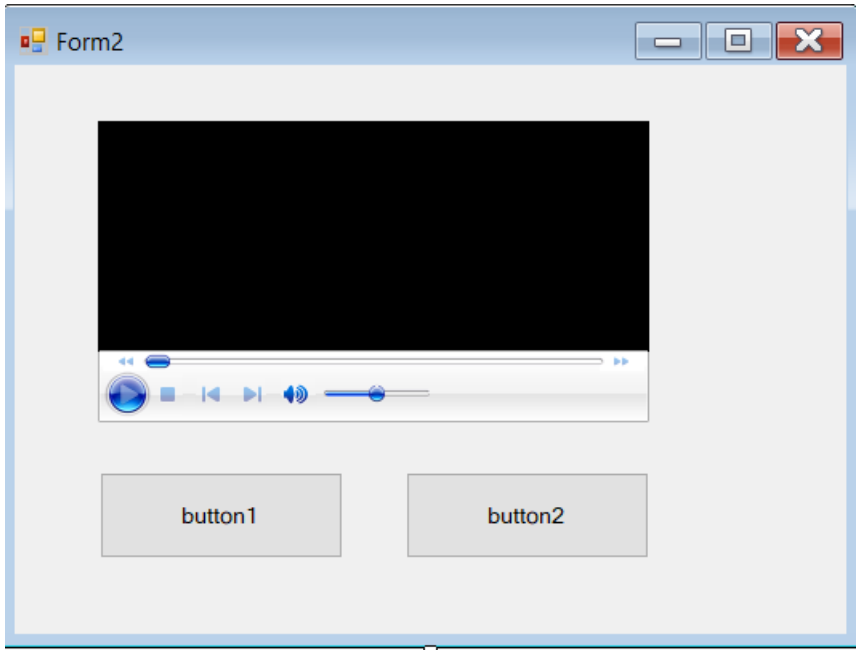
- Toolbox’ta standart olarak Windows Media Player öğesi gelmediği için bunun sonradan eklenmesi gerekir. **Toolbox** üzerinde boş bir alanda mouse sağ tık yaparak “**Choose Items**” seçeneğine tıklayın.



- Ardından açılan pencerede **Com Components** sekmesine gelip **Windows Media Player**'ı seçin ve ardından **Tamam** deyin.

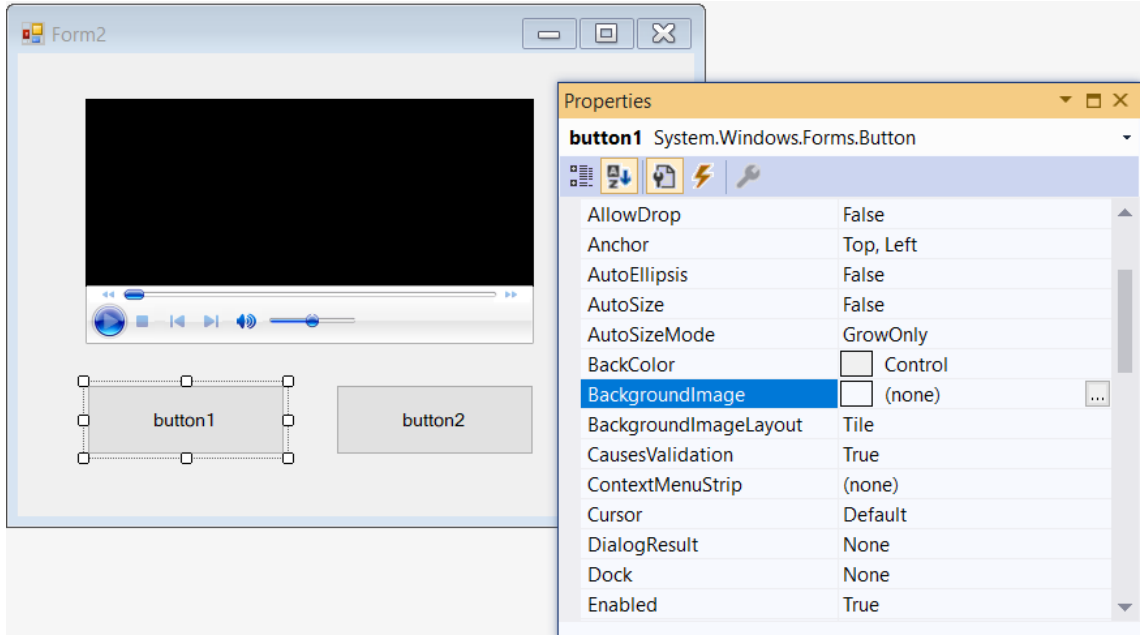


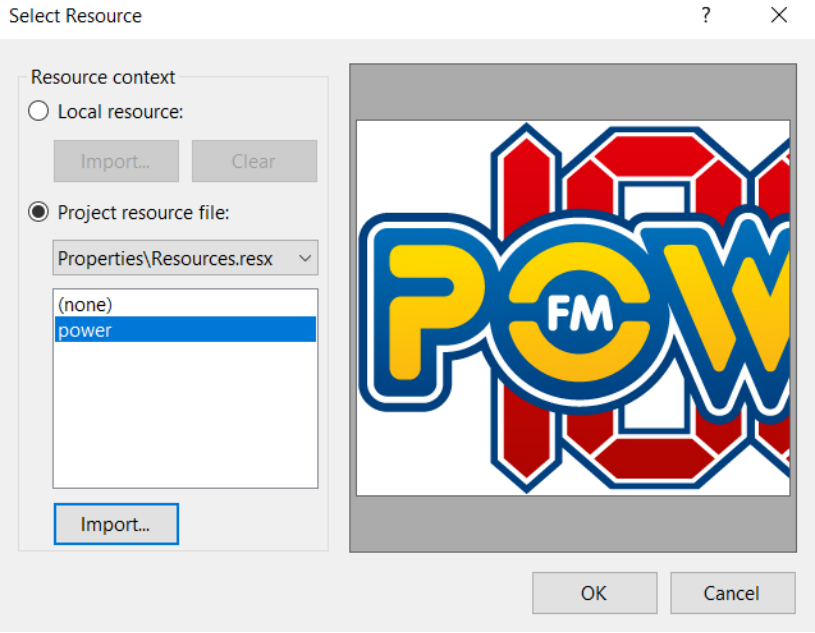
- Şimdi Toolbox içerisine **Windows Media Player** öğesi eklendi. Artık formu tasarlarken bu öğeden faydalanabilirsiniz. Formumuzu yandaki gibi tasarlayabiliriz:



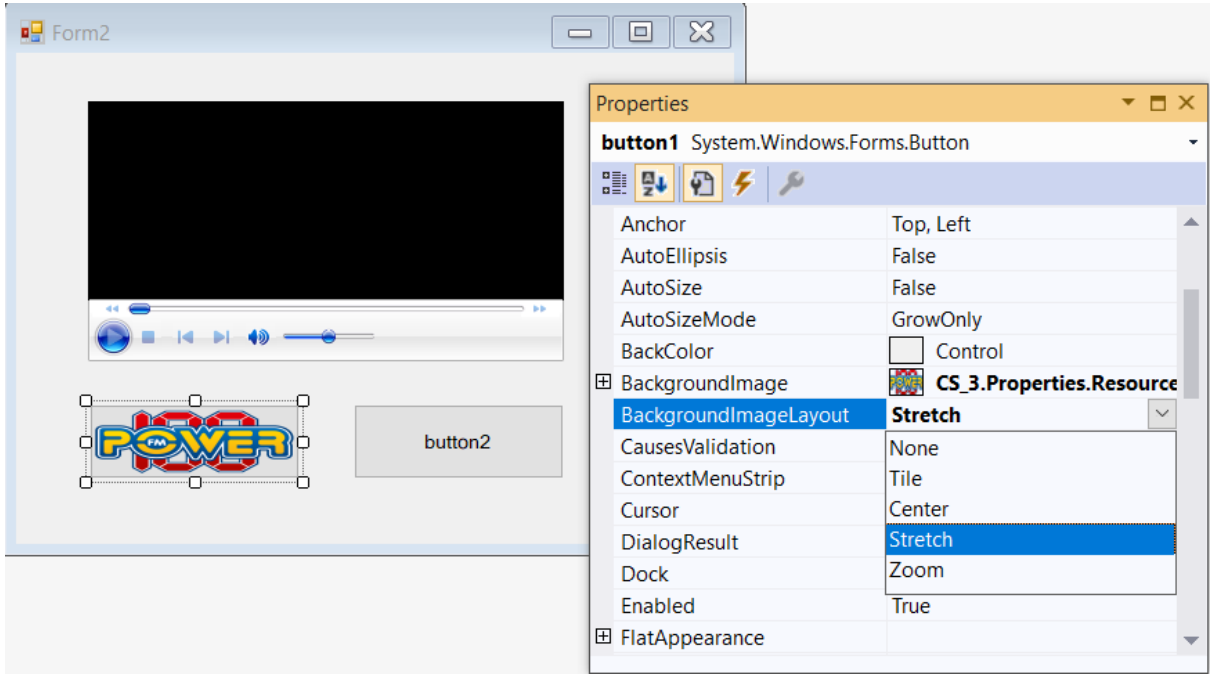

- Butonların içerisine link ekleyerek tıklandığında belirli radio frekanslarının çalmasını sağlayacağız.
- Örneğin buton1'e tıklandığında aşağıdaki linkte bulunan media oynatılsın: https://listen.powerapp.com.tr/powerfm/abr/powerfm/128/playlist.m3u8
- Bunu yapmak için buton1'in mouse click olayına aşağıdaki komutu yazmamız yeterlidir:
  *axWindowsMediaPlayer1.URL = "https://listen.powerapp.com.tr/powerfm/abr/powerfm/128/playlist.m3u8";*
- Benzer şekilde buton2 için de aşağıdaki komutu yazalım:
  *axWindowsMediaPlayer1.URL = "http://fenomen.listenfenomen.com/fenomen/128/icecast.audio";*
- Artık buton1' basıldığında PowerFM, buton2'ye basıldığında ise RadyoFenomen çalacaktır.
- Diğer radyoların çalma linkleri Google üzerinden yapılacak basit bir araştırma ile bulunabilir.

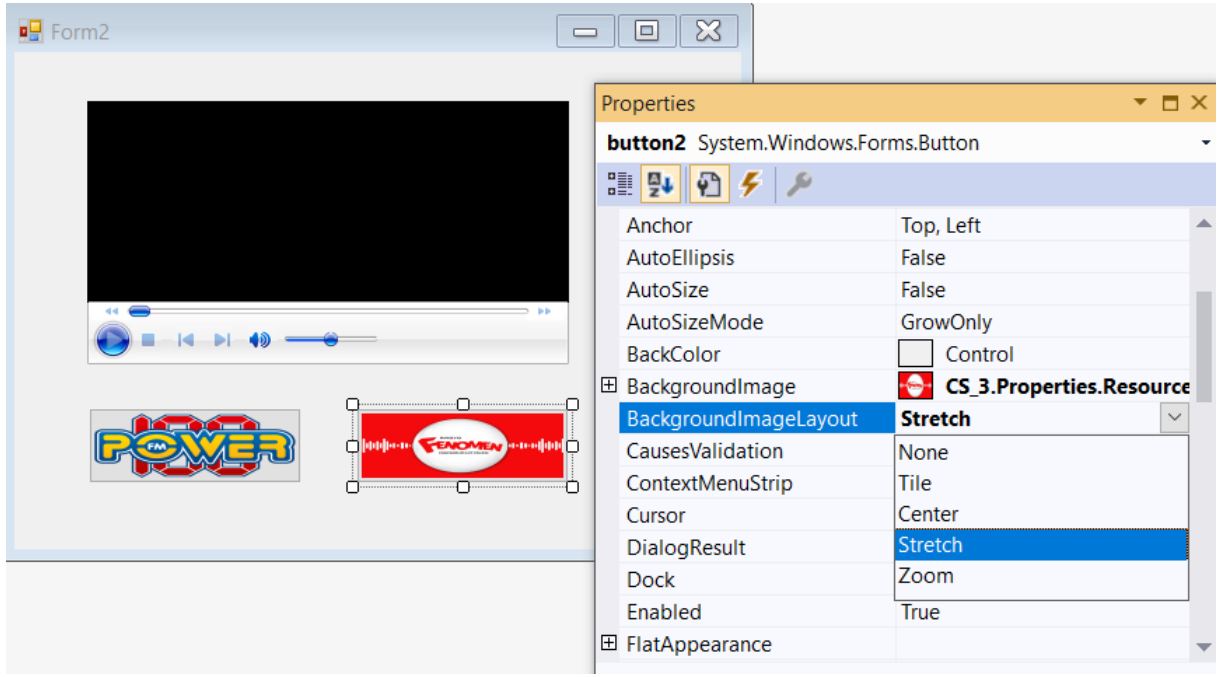
- Butonların üzerine de radyolara ait görseller aşağıdaki gibi eklenebilir. Bunun için butonun **BackgroundImage** özelliği kullanılarak **Import** butonu ile görsel seçilir.



- Her iki butonumuz için de görseller seçildikten sonra bu görsellerin buton üzerinde doğru yerleşiminin sağlanması için **BackgroundImageLayout** özelliğinden **Stretch** seçilir.



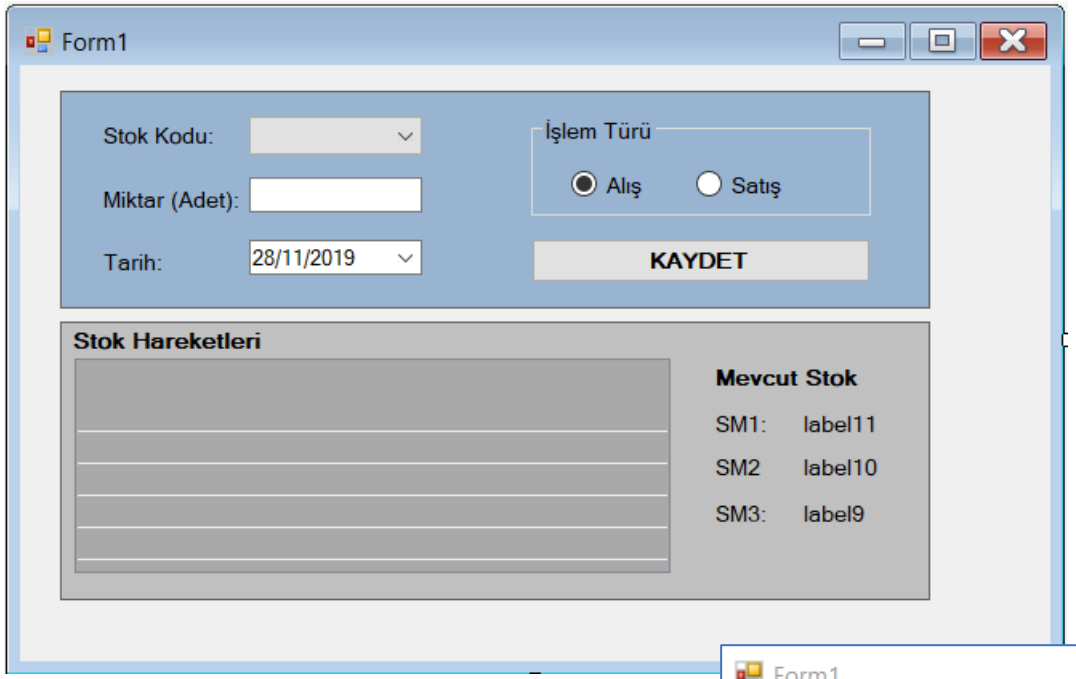



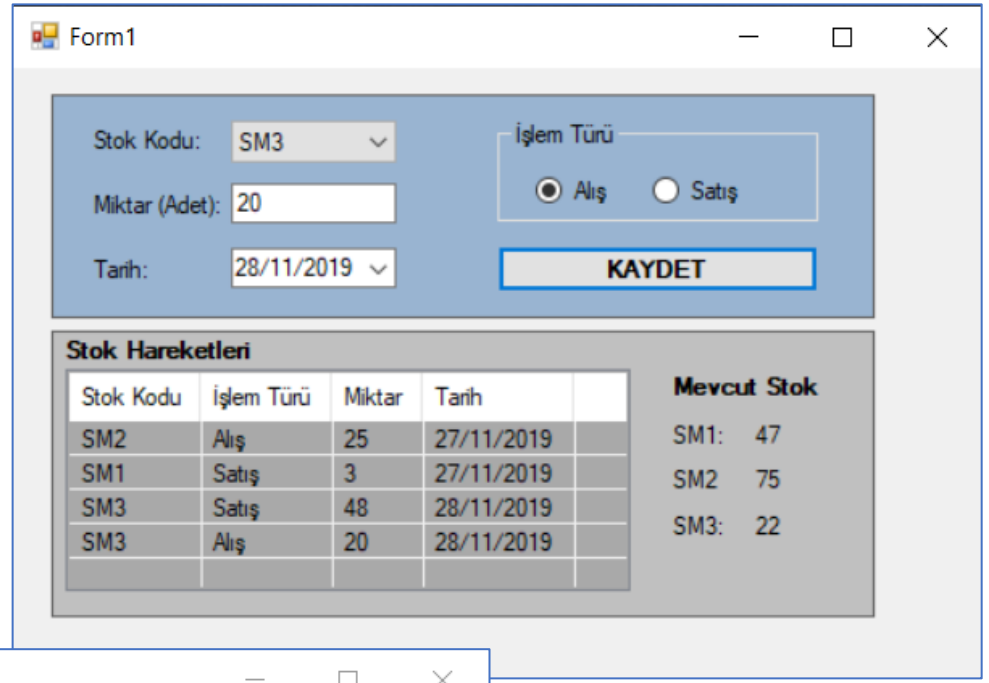
- Artık programımızı çalıştırdığımızda butonlara tıklayınca radio kanallarımız çalışır durumdadır. Daha fazla kanal ekleyerek tasarım üzerinde değişiklikler yapılarak aşağıdaki gibi daha fazla kanaldan oluşan bir form elde edilebilir:

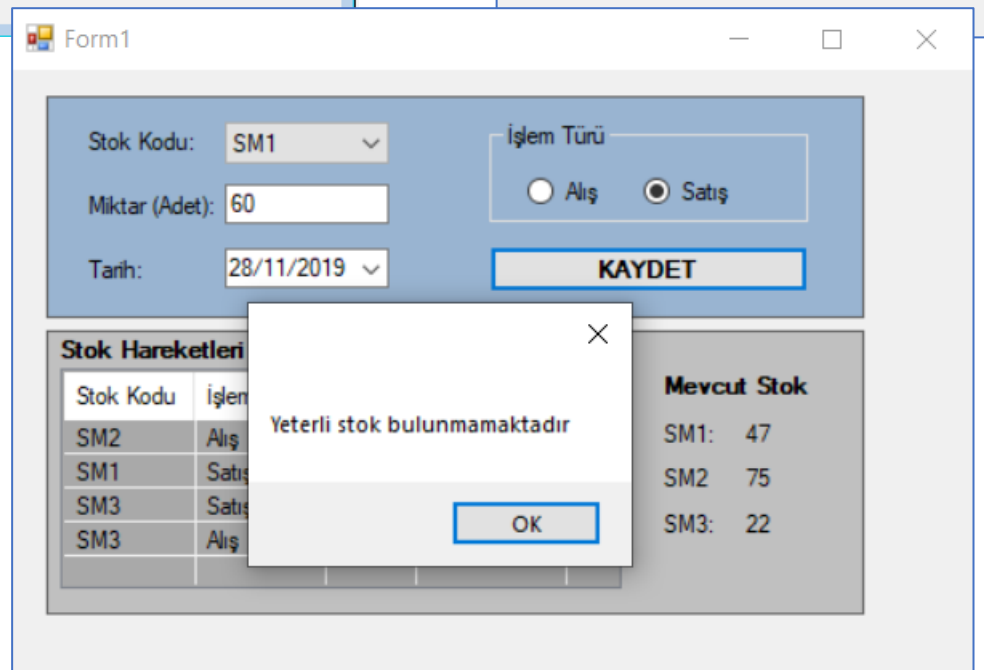
# ÖDEV-1

- Formunuza aşağıdaki gibi **TARA** ve **TARAMAYI DURDUR** butonları ekleyiniz.
- TARA butonuna bastığınızda eklemiş olduğunuz kanallar arasında tarama yapılsın, her 8 saniyede bir yeni kanala geçilsin. Son kanala ulaşıldığında ise başa geri dönülsün
  - *İpucu: Döngü kullanmanız gerekiyor :)*
- TARAMAYI DURDUR butonuna basıldığında ise o sırada çalmakta olan kanalda kalınsın.

# CS-4 Stok Hareketleri



```
private void Form1_Load(object sender, EventArgs e)
{
    listView1.Columns.Add("Stok Kodu", 65);
    listView1.Columns.Add("İşlem Türü", 65);
    listView1.Columns.Add("Miktar", 45);
    listView1.Columns.Add("Tarih", 75);
    label11.Text = stoksm1.ToString();
    label10.Text = stoksm2.ToString();
    label9.Text = stoksm3.ToString();
}
```

```
listView1.Items.Add(comboBox1.Text);//stok kodu
String islem = "";
if (radioButton1.Checked)
{
    islem = "Alış";
}
else if (radioButton2.Checked)
{
    islem = "Satış";
}
listView1.Items[listView1.Items.Count - 1].SubItems.Add(islem);//islem
listView1.Items[listView1.Items.Count - 1].SubItems.Add(textBox1.Text);//miktar
listView1.Items[listView1.Items.Count - 1].SubItems.Add(dateTimePicker1.Text);//tarih
```

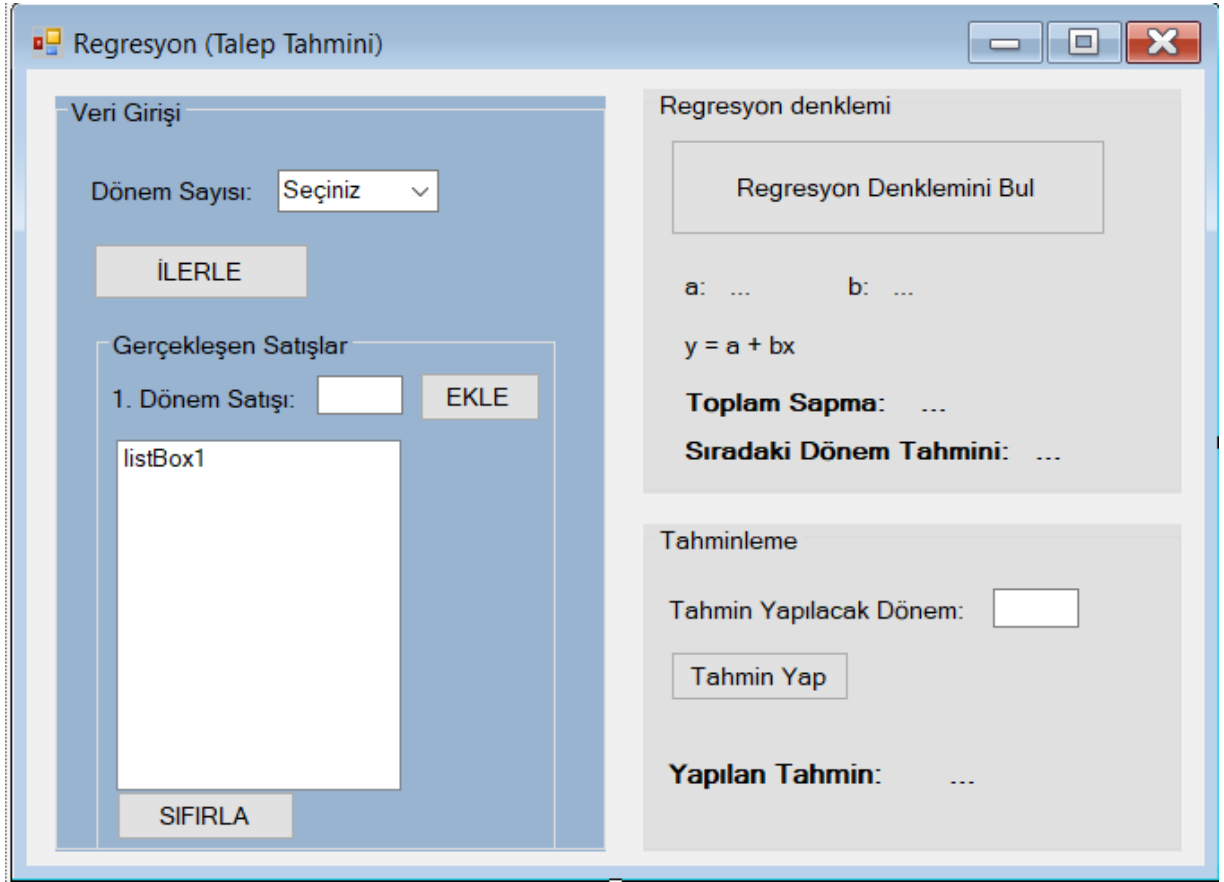
```
if (islem == "Alış")
{
    if (comboBox1.Text == "SM1")
    {
        stoksm1 += Convert.ToInt32(textBox1.Text);
        label11.Text = stoksm1.ToString();
    }
    else if (comboBox1.Text == "SM2")
    {
        stoksm2 += Convert.ToInt32(textBox1.Text);
        label10.Text = stoksm2.ToString();
    }
    else if (comboBox1.Text == "SM3")
    {
        stoksm3 += Convert.ToInt32(textBox1.Text);
        label9.Text = stoksm3.ToString();
    }
}
else if (islem == "Satış")
{
    if (comboBox1.Text == "SM1")
    {
        stoksm1 -= Convert.ToInt32(textBox1.Text);
        label11.Text = stoksm1.ToString();
```

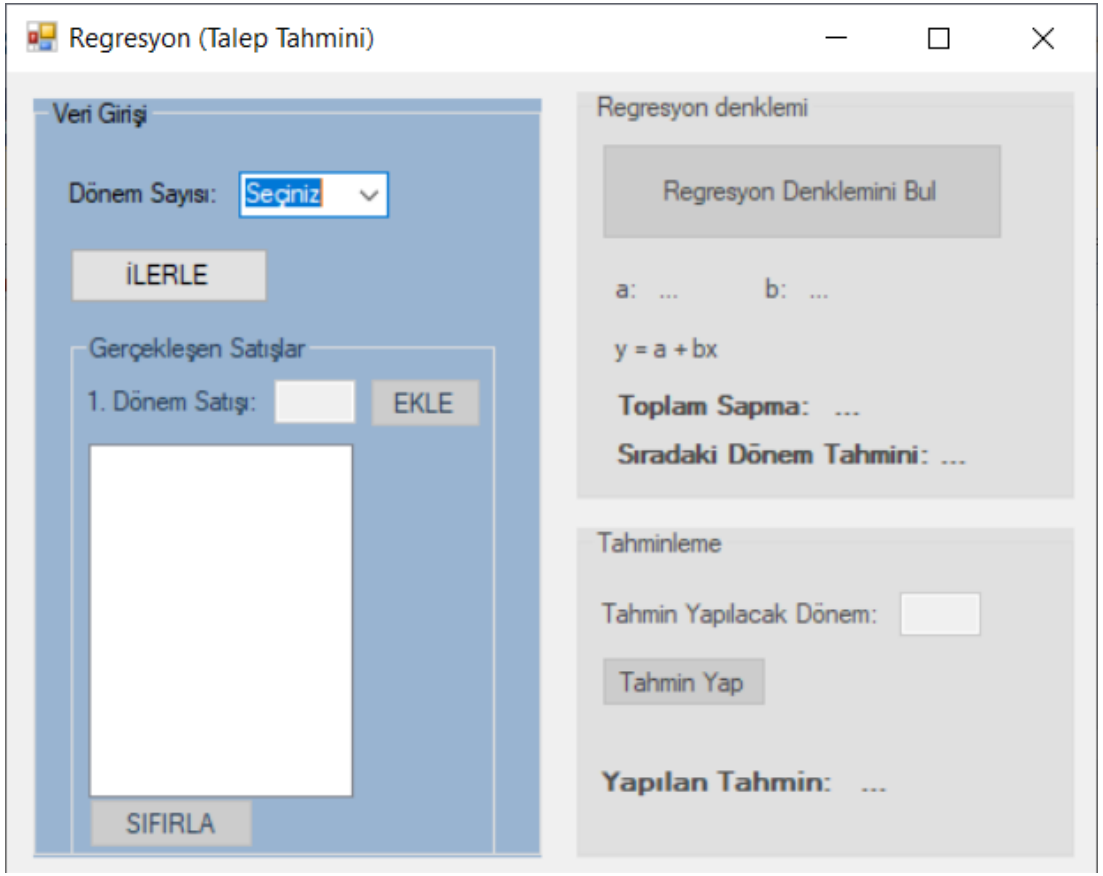
```
Boolean yeterli = true;
if (radioButton2.Checked)
{
    if(comboBox1.Text == "SM1")
    {
        if (Convert.ToInt32(textBox1.Text) > stoksm1)
        {
            MessageBox.Show("Yeterli stok bulunmamaktadır");
            yeterli = false;
        }
    }
    if (comboBox1.Text == "SM2")
    {
        if (Convert.ToInt32(textBox1.Text) > stoksm2)
        {
            MessageBox.Show("Yeterli stok bulunmamaktadır");
            yeterli = false;
        }
    }
    if (comboBox1.Text == "SM3")
    {
        if (Convert.ToInt32(textBox1.Text) > stoksm3)
        {
            MessageBox.Show("Yeterli stok bulunmamaktadır");
```

# CS-5 Regresyon Analizi (Talep Tahmini)

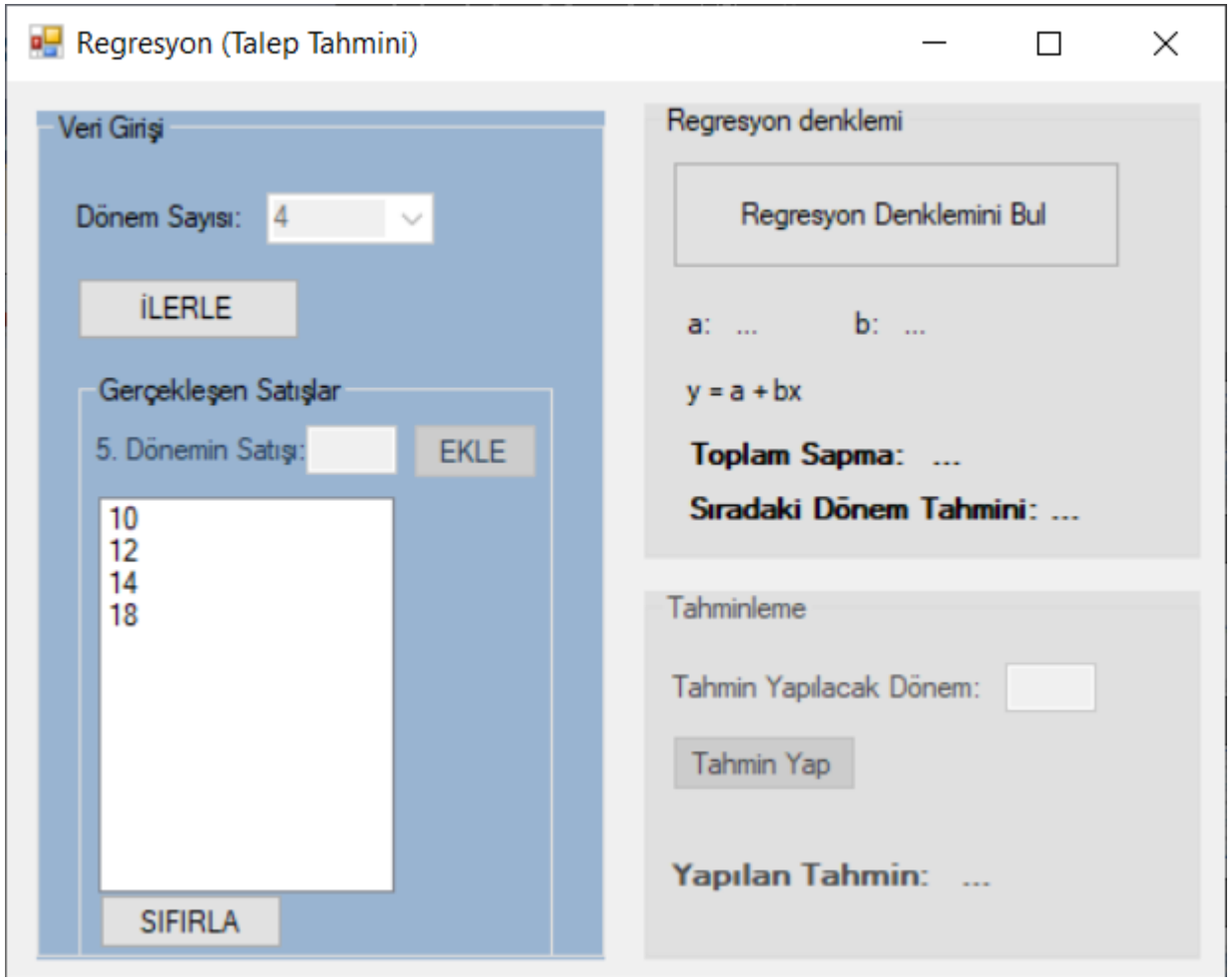
$$y = a + bx$$

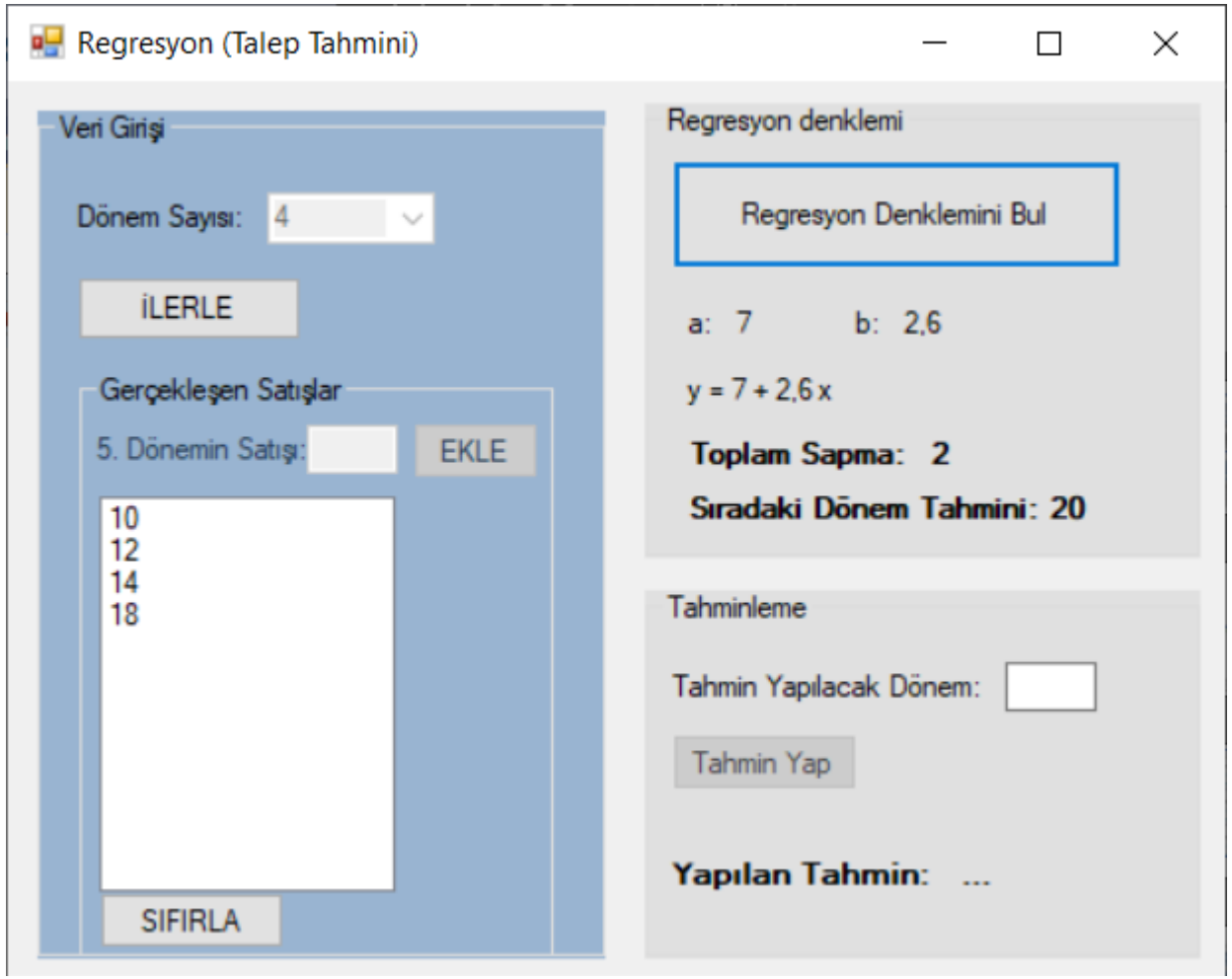
- Bu projede Regresyon Analizi ile talep tahmini yapan bir program ve arayüz geliştirilecektir. Önceki dönemlerde gerçekleşen satışlara ilişkin veri yine arayüz vasıtası ile alınacaktır. Bunun için öncelikle arayüzden dönem sayısı seçilecek ve **İLERLE** butonuna basılacaktır. Ardından, her bir döneme ilişkin satış verisi birer birer TextBox'a girildikten sonra **EKLE** butonuna basılarak ListBox'a eklenecektir.



- İstenen dönem kadar veri girildikten sonra **Regresyon Denklemini Bul** butonu aktif hale gelecektir. Bu butona basıldığında “a ve b parametreleri”, “regresyon denklemi”, “toplam sapma” ve “sıradaki dönemin talebi” bulunacaktır.

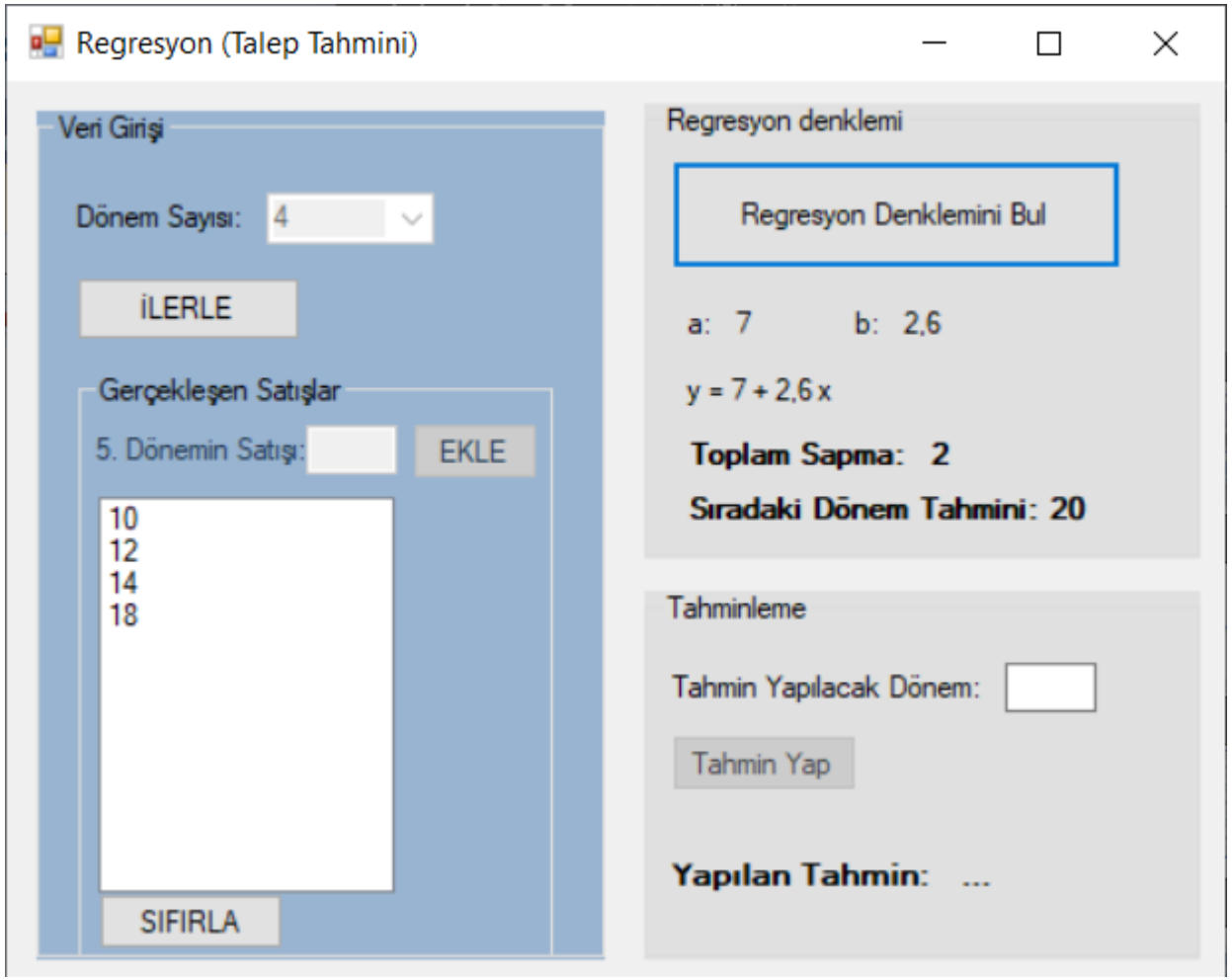


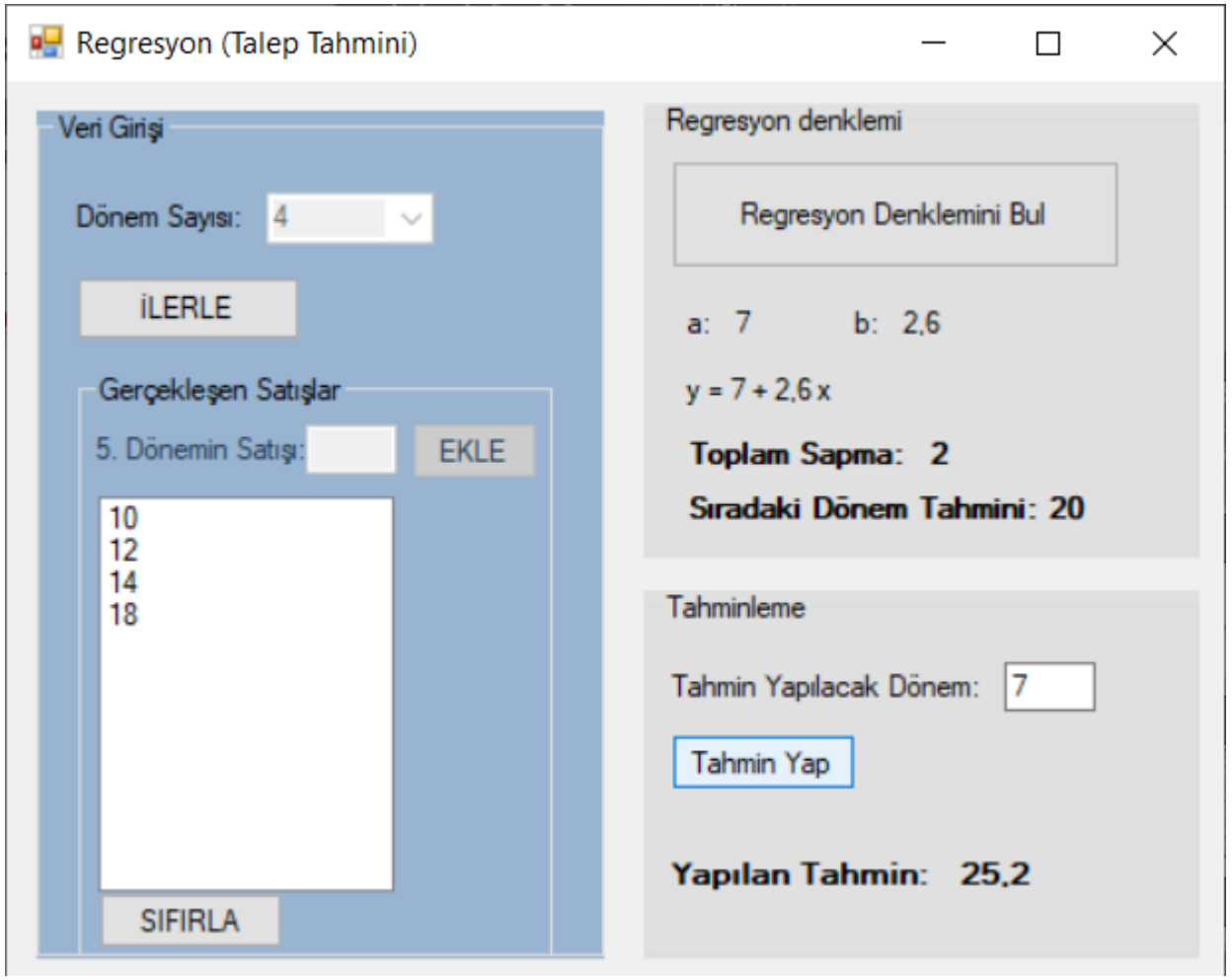



- Y = a + bX denkleminde a ile b parametrelerinin alacağı değerler aşağıdaki formüller kullanılarak ile bulunur:

$$b = \frac{(n\sum_{i=1}^{n} X_i Y_i) - (\sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i)}{(n\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i \sum_{i=1}^{n} X_i)}$$

$$a = \frac{\sum_{i=1}^{n} Y_i - b\sum_{i=1}^{n} X_i}{n}$$






- Ayrıca, elde edilen a ve b parametreleri ve dolayısıyla regresyon denklemi kullanılarak herhangi bir dönem için de talep tahmininde bulunulabilecektir.

```
public partial class Form1 : Form
{
    static int n = 10;
    int k = 0;
    double b;
    double a;

    public Form1()
    {
        InitializeComponent();
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    //İLERLE
    private void button1_Click(object sender, EventArgs e)
    {
        n = Convert.ToInt32(comboBox1.Text);
        comboBox1.Enabled = false;
        groupBox2.Enabled = true;
        button2.Enabled = false;
        textBox1.Select();
    }
```

```
//EKLE
private void button2_Click(object sender, EventArgs e)
{
    if (k < n)
    {
        listBox1.Items.Add(textBox1.Text);
        k++;
        label2.Text = (k+1).ToString() + ". Dönemin Satışı:";
        textBox1.Clear();
        button2.Enabled = false;
        textBox1.Select();
    }
    if (k == n)
    {
        label2.Enabled = false;
        button2.Enabled = false;
        textBox1.Enabled = false;
        groupBox2.Enabled = true;
        groupBox3.Enabled = true;
    }

}
```

```
//Hesapla
private void button4_Click(object sender, EventArgs e)
{
    double[] Y = new double[n];
    double[] X = new double[n];

    for (int i = 0; i < n; i++)
    {
        Y[i] = Convert.ToDouble(listBox1.Items[i]);
        X[i] = i + 1;
    }

    double XToplam = 0;
    double YToplam = 0;
    double XKareToplam = 0;
    double XYToplam = 0;

    for (int i = 0; i < n; i++)
    {
        XToplam += X[i];
        YToplam += Y[i];
        XKareToplam += X[i] * X[i];
        XYToplam += X[i] * Y[i];
    }
    b = (n * XYToplam - (XToplam * YToplam)) / (n * XKareToplam - XToplam * XToplam);
    a = (YToplam - b * XToplam) / n;

    label6.Text = Math.Round(a, 4).ToString();
    label7.Text = Math.Round(b, 4).ToString();
    label3.Text = "y = " + Math.Round(a, 4).ToString() + " + " + Math.Round(b, 4).ToString() + " x";
```

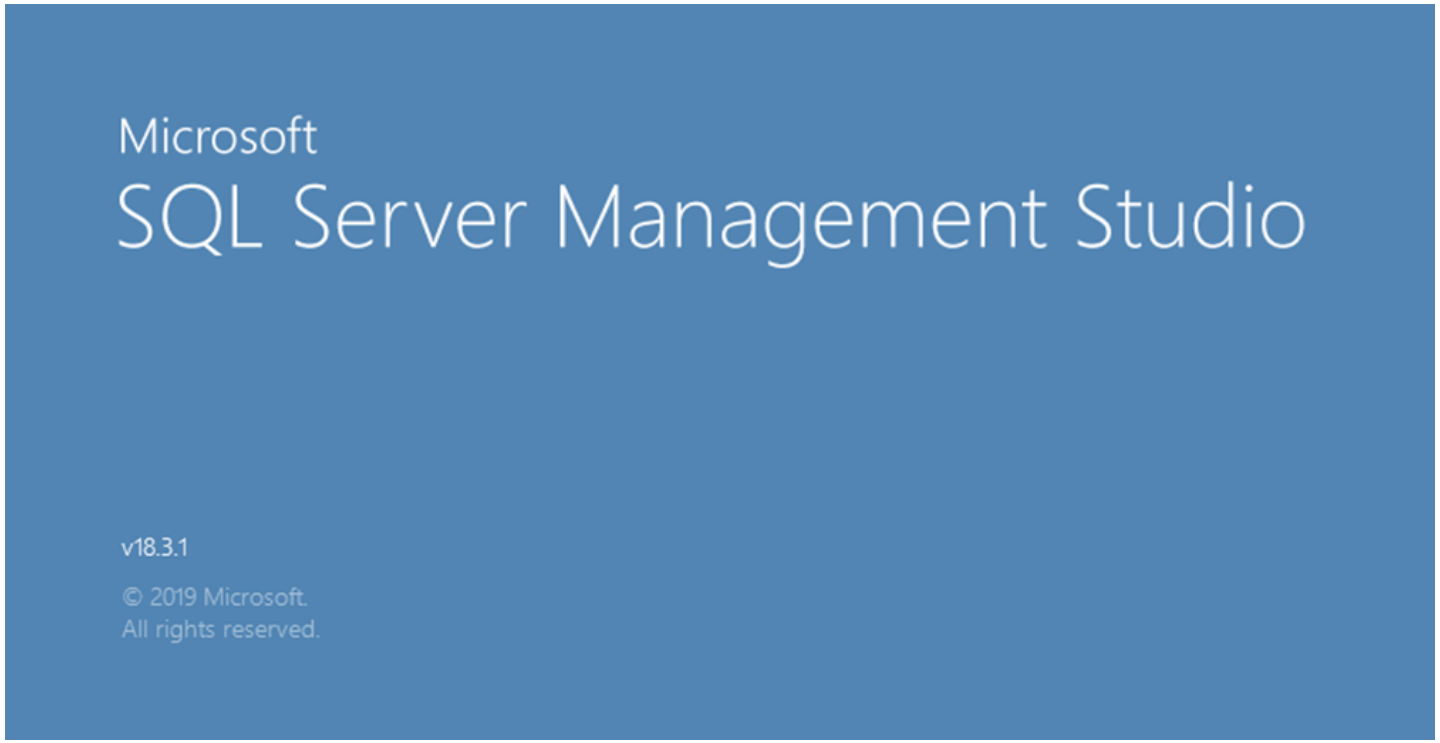
```
    //sapmayı hesapla
    double[] tahmin = new double[n];

    double y;
    for (int i = 0; i < n; i++)
    {
        y = a + b * (i + 1);
        tahmin[i] = y;
    }

    double sapma = 0;
    for (int i = 0; i < n; i++)
    {
        sapma += Math.Abs(Y[i] - tahmin[i]);
    }
    label11.Text = Math.Round(sapma, 4).ToString();
    label13.Text = Math.Round(a+b*(n+1), 4).ToString();
    groupBox4.Enabled = true;
}
```

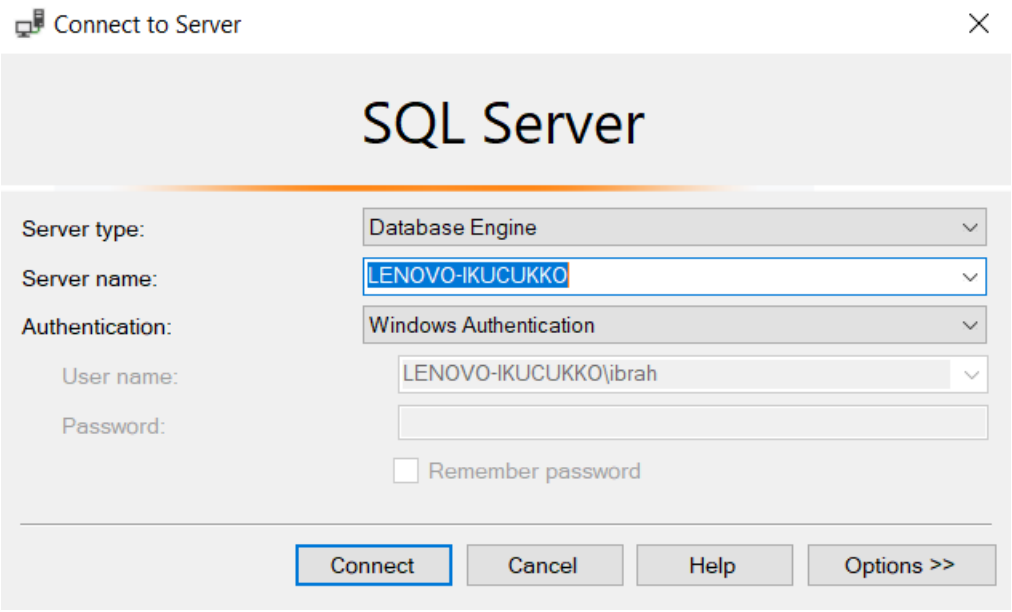
```
//TAHMİN YAP
private void button5_Click(object sender, EventArgs e)
{
    int x = Convert.ToInt32(textBox2.Text);
    double y = a + b * x;
    label10.Text = Math.Round(y, 2).ToString();
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    if (textBox2.Text != "")
    {
        button5.Enabled = true;
    }
    else
    {
        button5.Enabled = false;
    }
}
```

```
//SIFIRLA
private void button3_Click(object sender, EventArgs e)
{
    comboBox1.Enabled = true;
    comboBox1.Text = "1";
    label2.Enabled = true;
    label2.Text = "1. Dönemin Satışı:";
    button2.Enabled = false;
    textBox1.Enabled = true;
    listBox1.Items.Clear();
    k = 0;
    n = Convert.ToInt32(comboBox1.Text);
    groupBox2.Enabled = false;
    groupBox3.Enabled = false;
    groupBox4.Enabled = false;
}
```

# Veritabanı Kullanımı - Öğrenci Kayıt Projesi

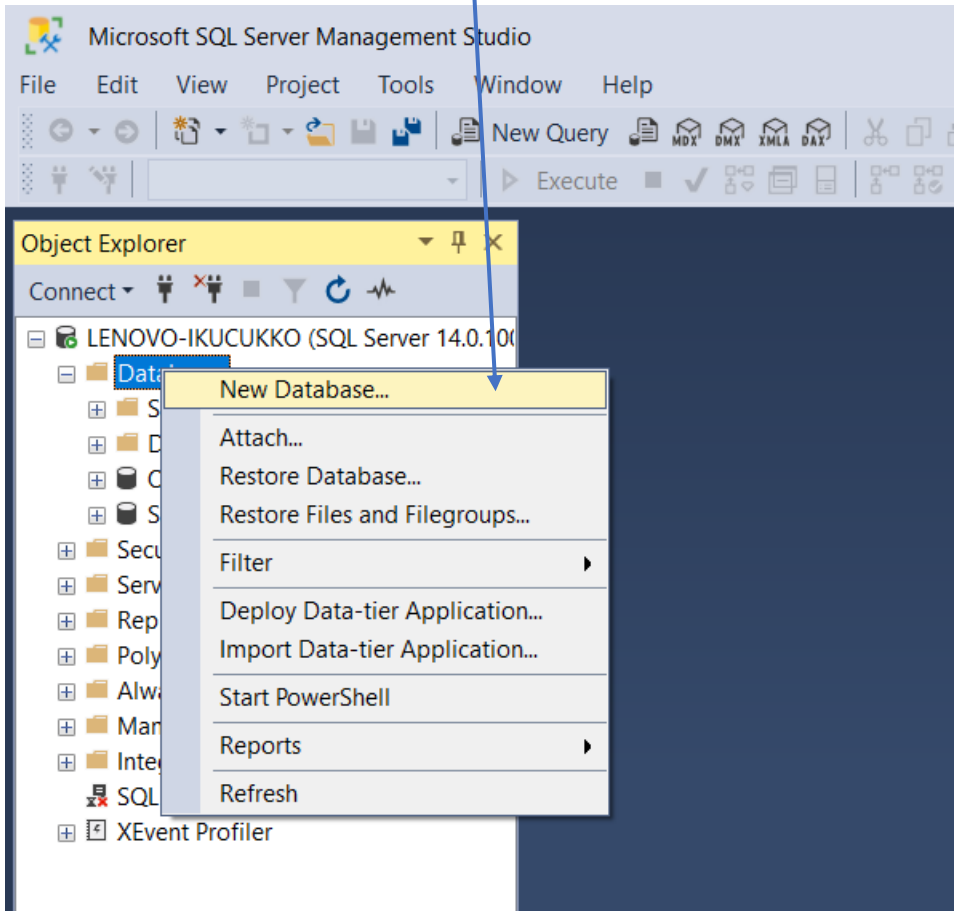
- Bu proje için öncelikle **MS SQL Server Management Studio** bilgisayara kurulmalıdır. Nasıl kurulacağına ilişkin buradan yardım alabilirsiniz: https://www.youtube.com/watch?v=HRejBl0KZp0

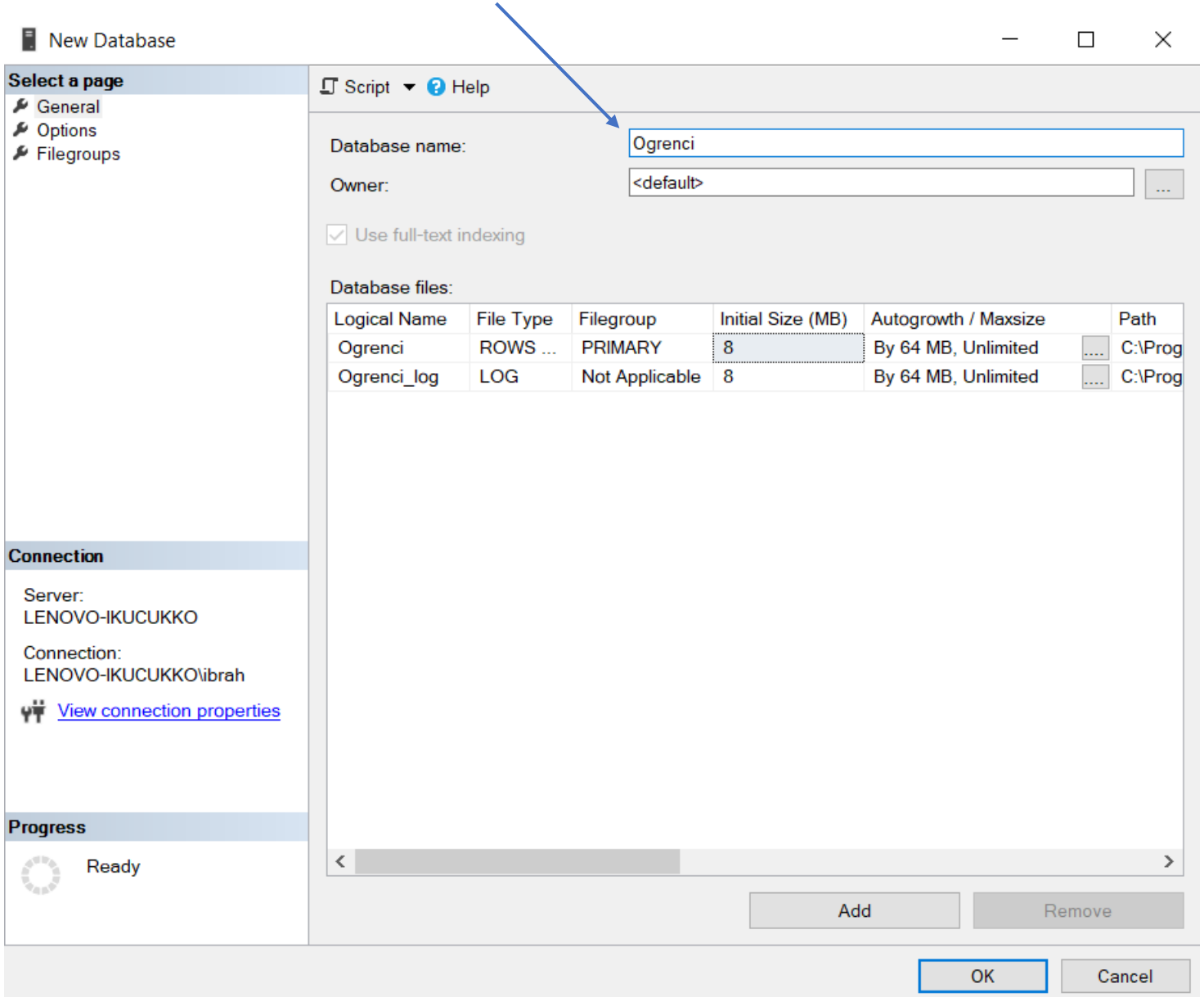


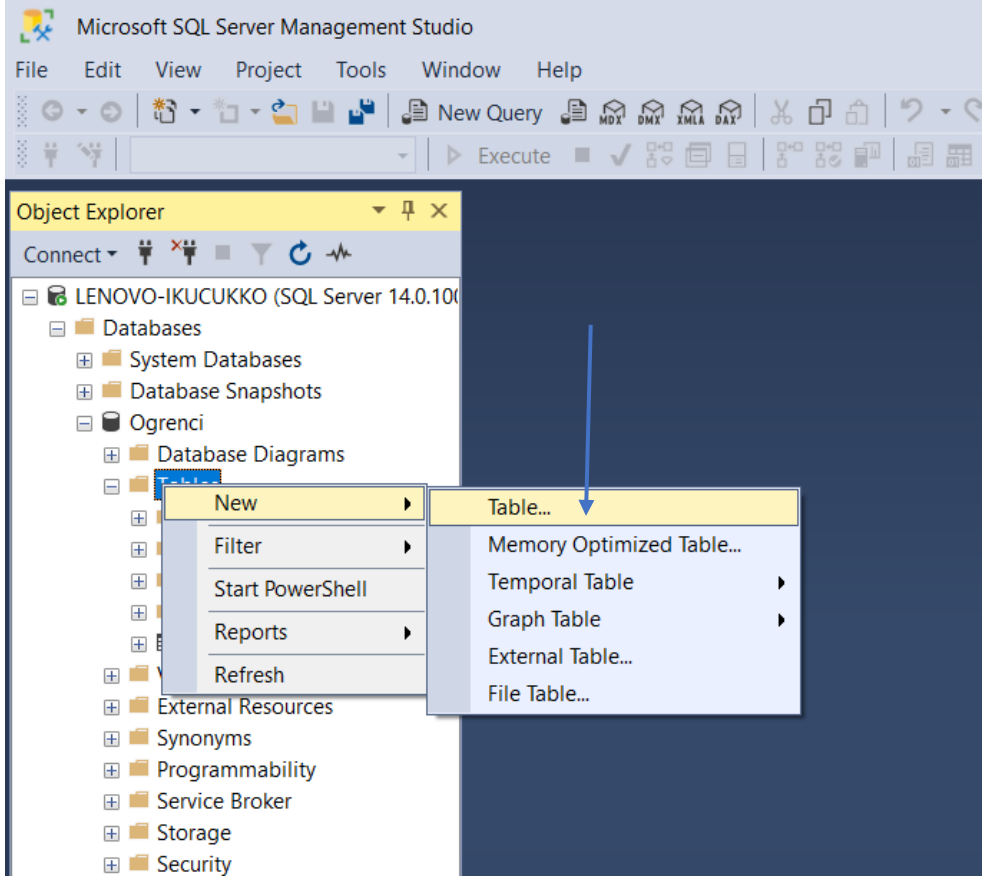
- Yardım Videosu: https://www.youtube.com/watch?v=9ai0Rc-4LKY

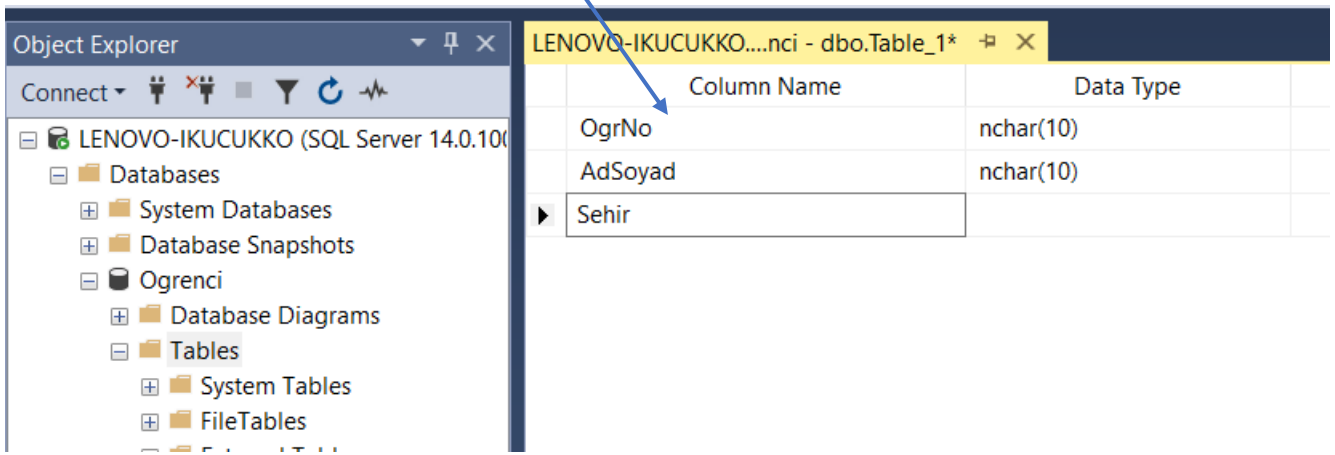
- **Database -> New Database** yolu izlenerek sadece **Database Name** kısmına isim verildikten sonra **OK**'e tıklanır.



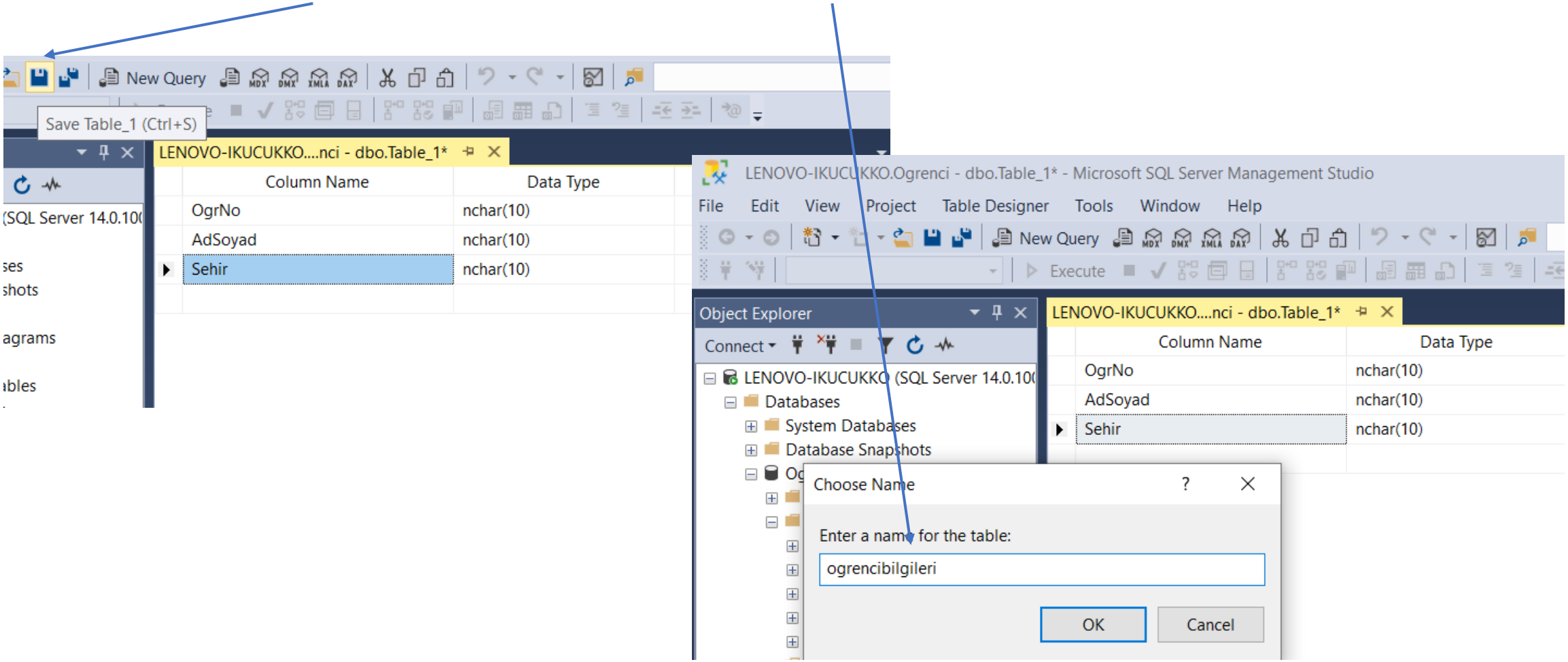



- Sonra, oluşturulan veritabanının içinde **Tables**'ın üzerine sağ tıklanarak **New -> Table** tıklanır. Oluşturulan tabloda kolon isimlerine OgrNo, AdSoyad ve Sehir yazılır. Bu aşamada tablo hala kaydedilmemiştir.

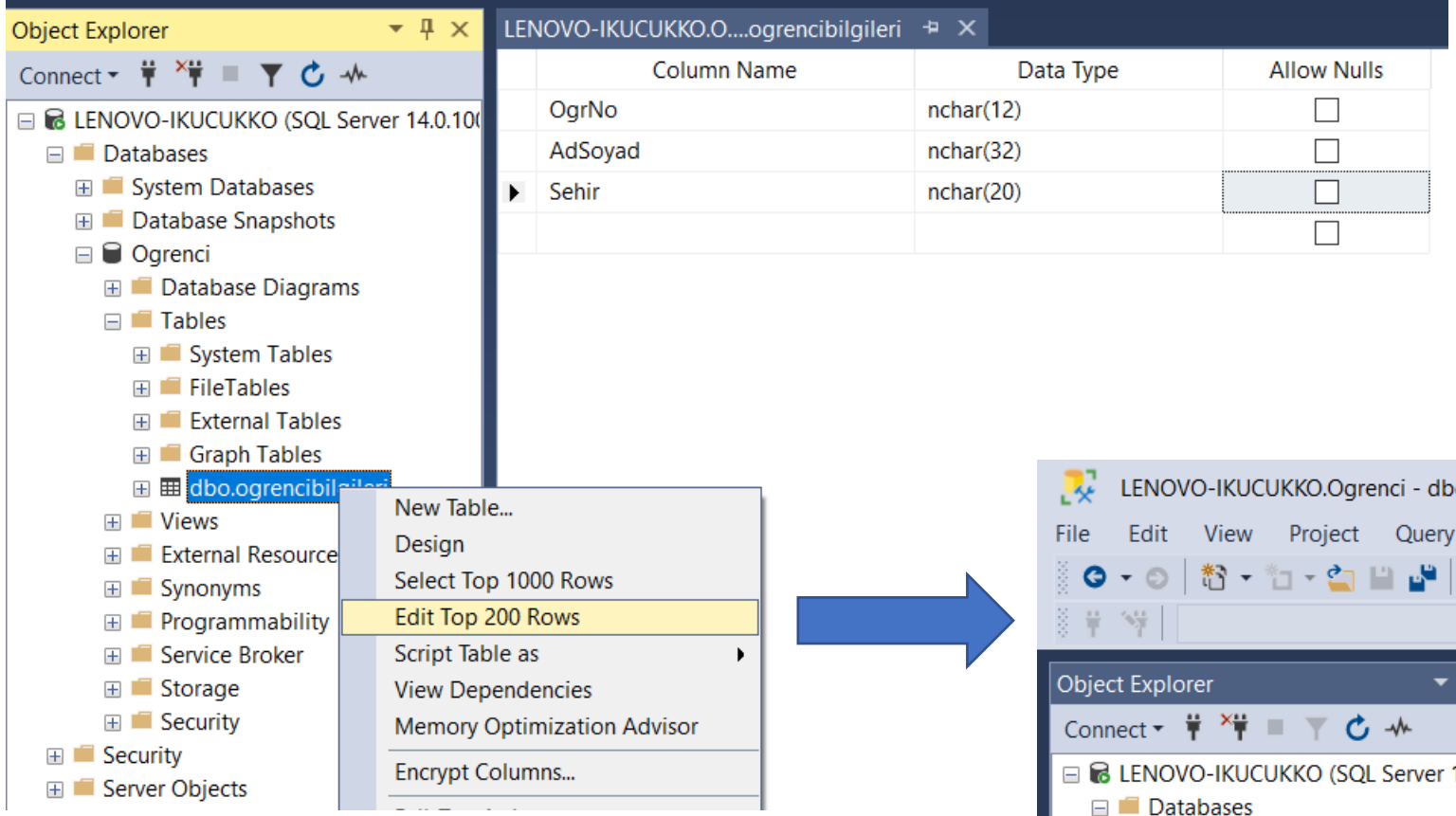


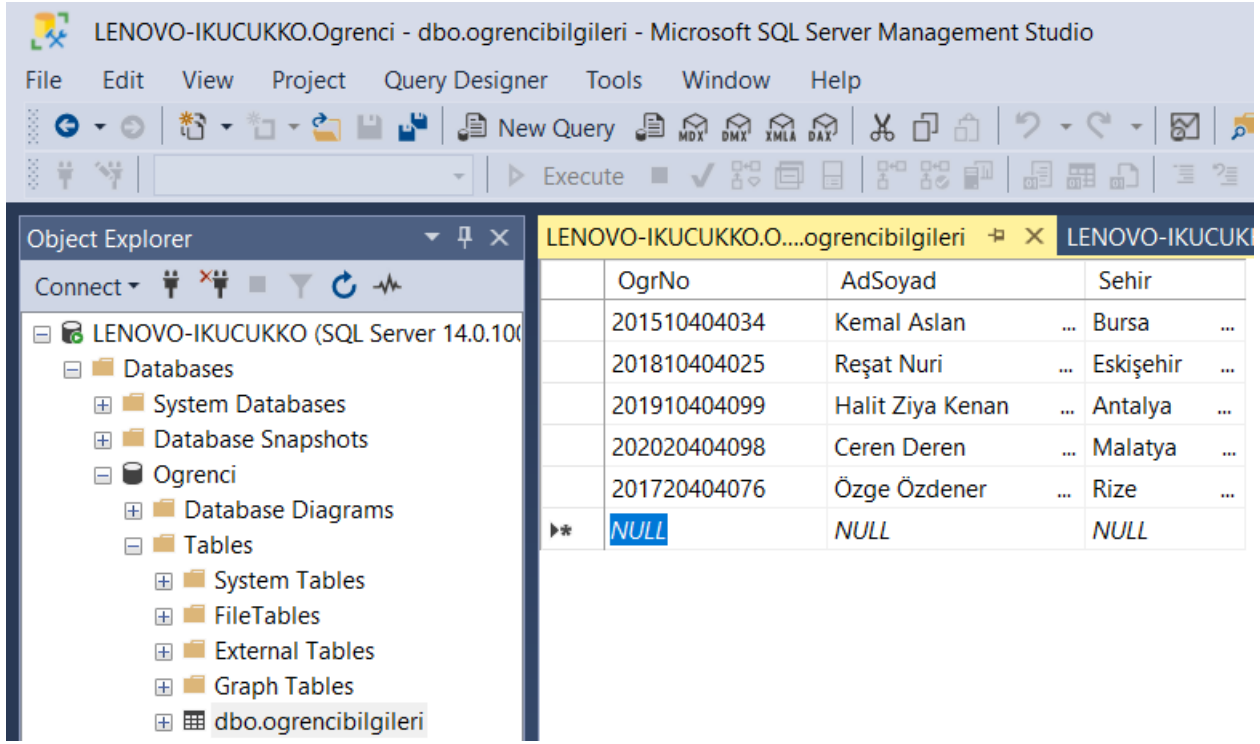



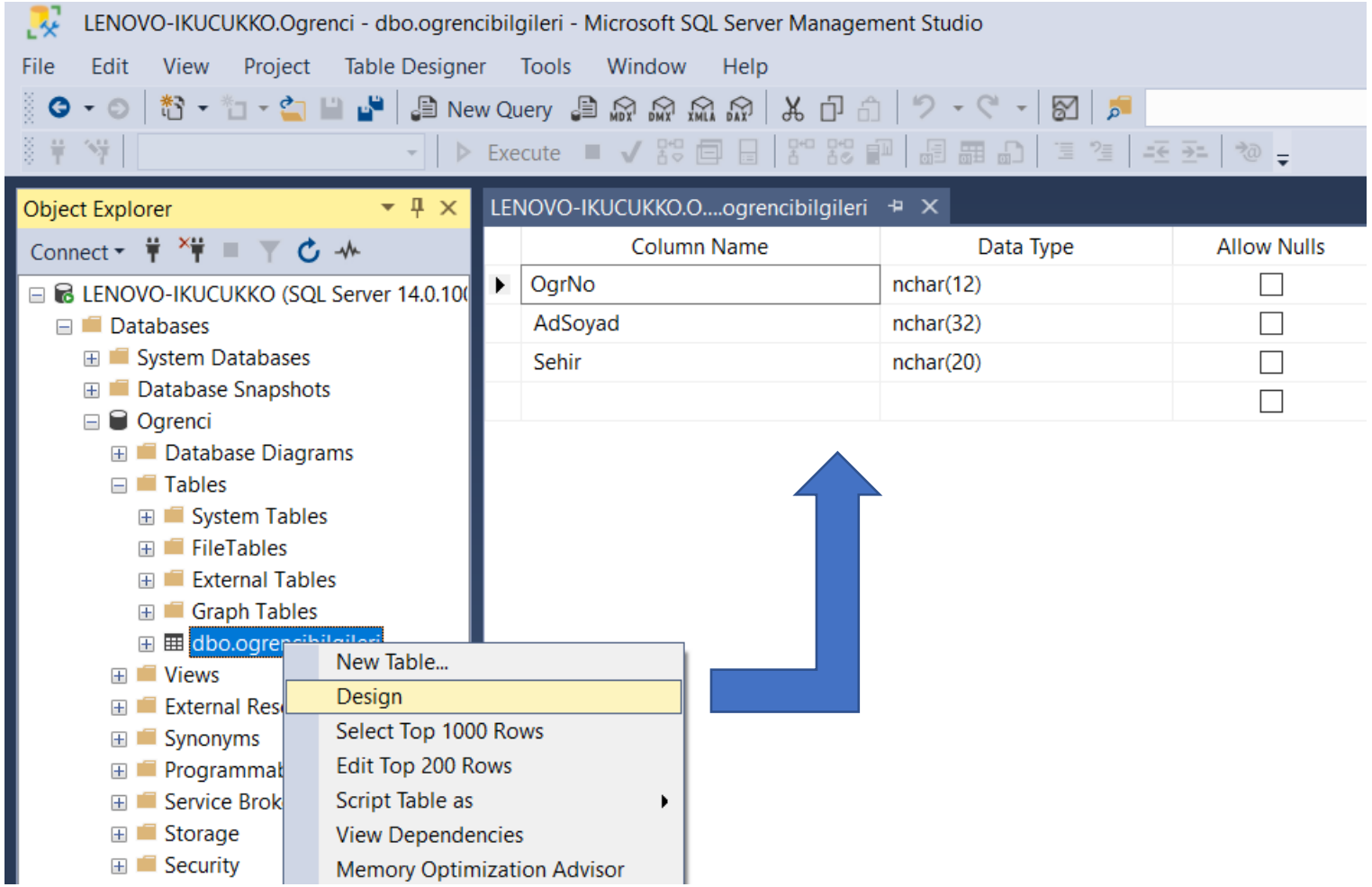
- Kaydetmek için **Kaydet** butonuna tıklanır, tabloya isim verilir.



Tabloya veri eklemek için aşağıdaki yol izlenir.

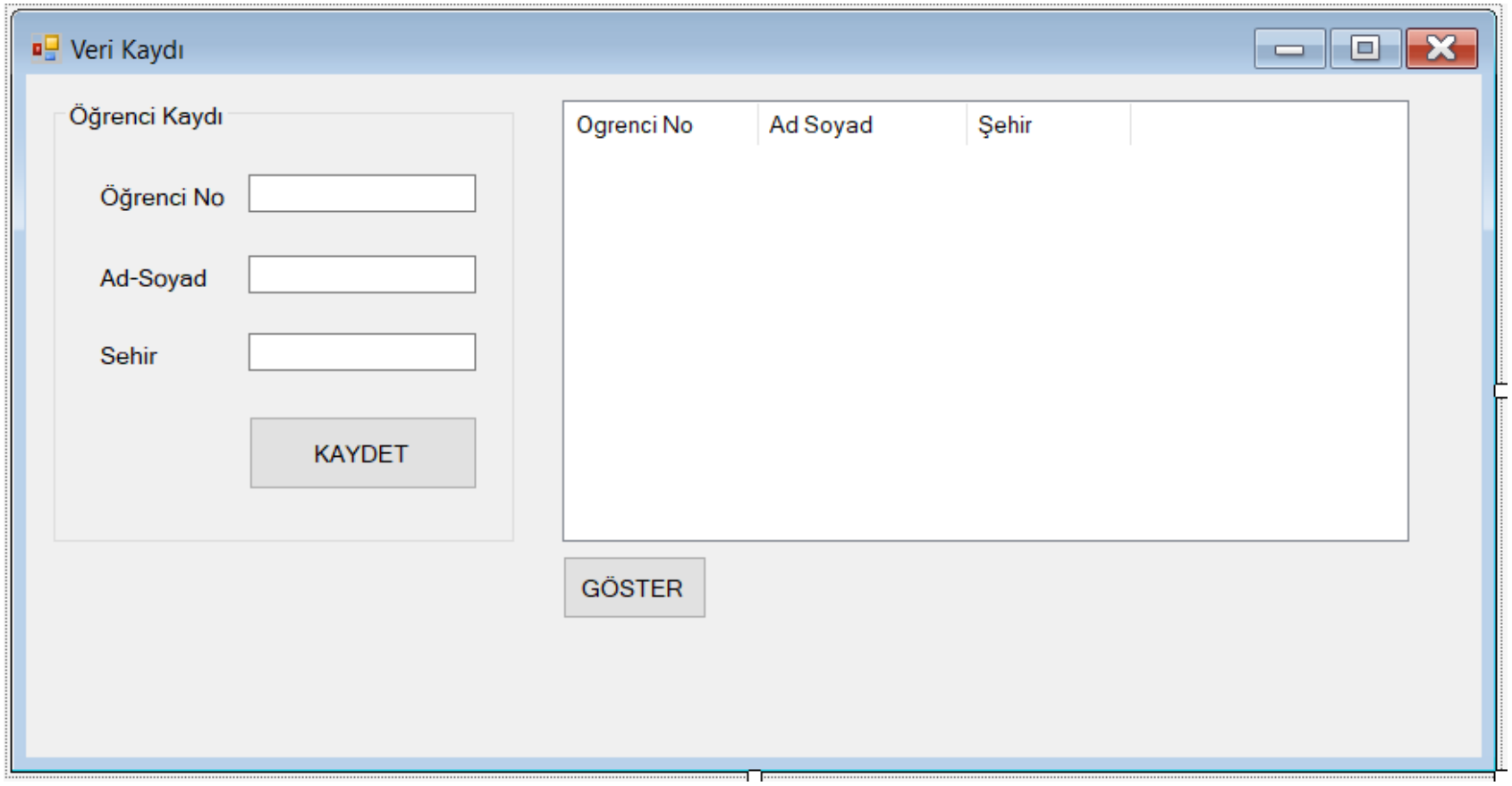


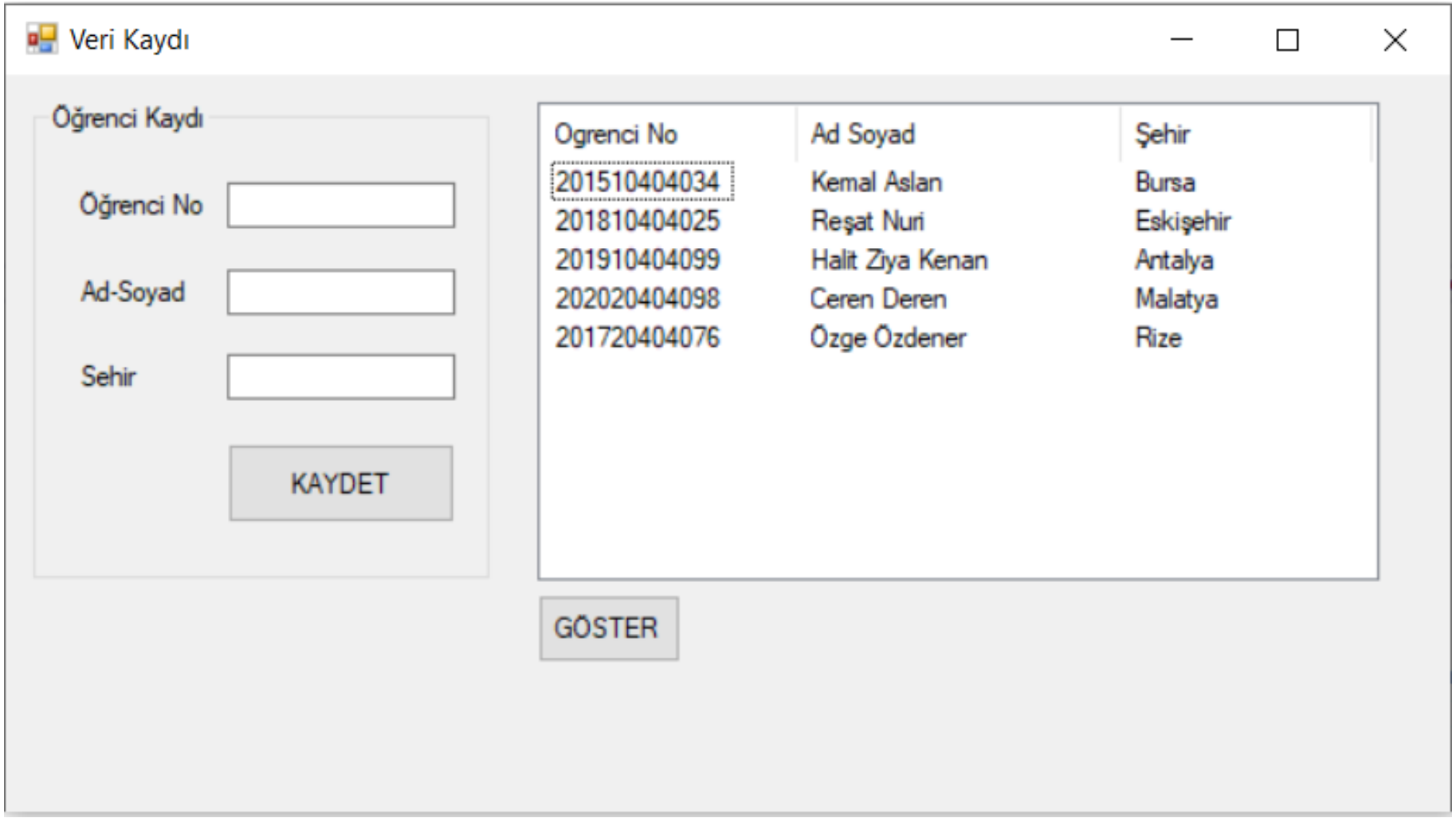



Tablo sütunlarına dönerek yeniden düzenleme yapmak, örneğin sütün eklemek için tablo isminin üzerine sağ tıklanarak **Design**'a tıklanır.

- Veritabanına veri kaydetmek ve veritabanındaki kayıtları göstermek için aşağıdaki gibi bir arayüz hazırlanmalıdır. **KAYDET** butonu veritabanına veri kaydetmek, **GÖSTER** butonu ise veritabanındaki kayıtları **ListView**'da görüntülemek için kullanılır.

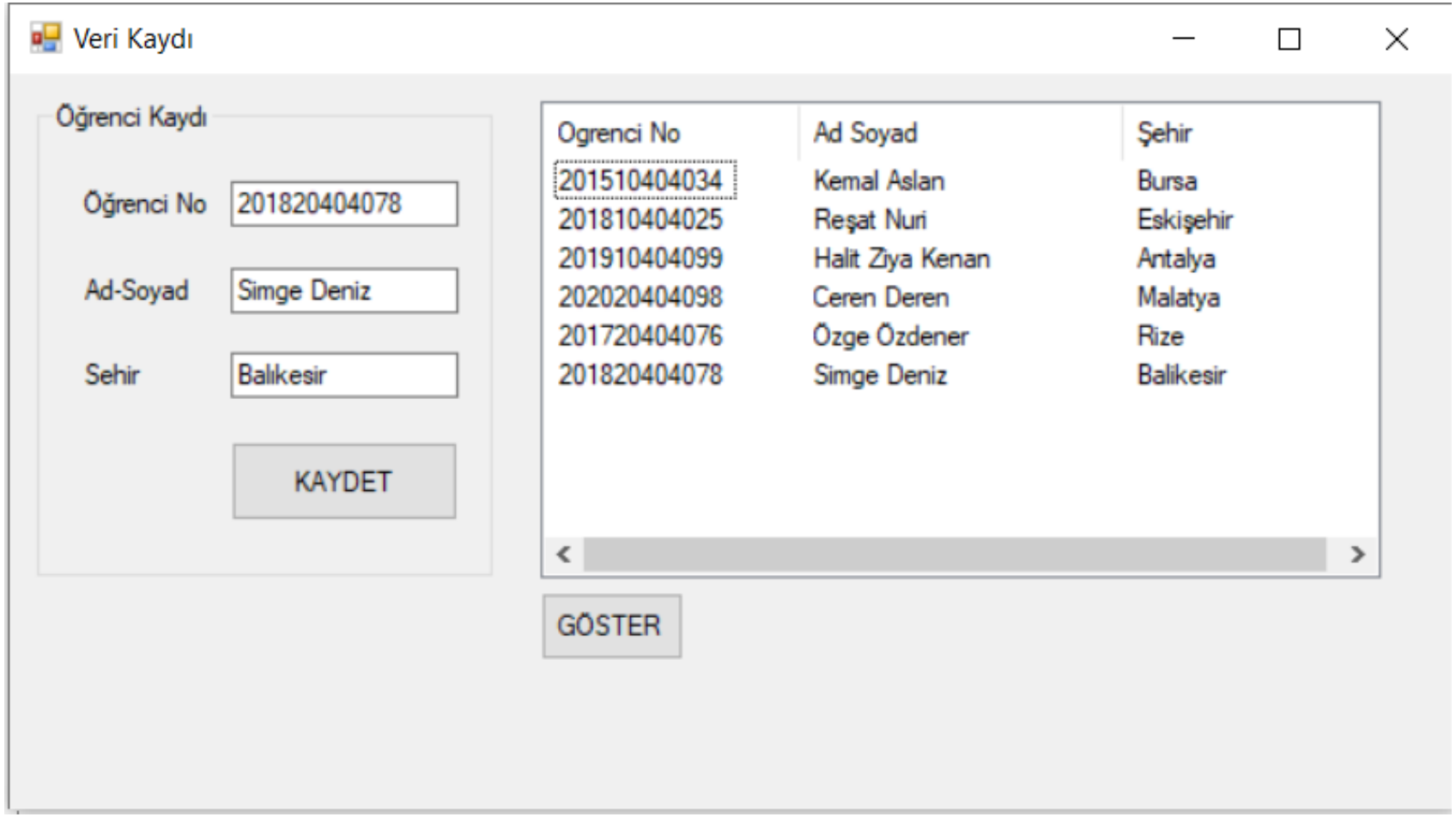


# Veritabanındaki Kayıtların Gösterilmesi



```
using System.Data.SqlClient; //SQL için gerekli kütüphane

namespace VeritabaniDersi
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        SqlConnection baglan = new SqlConnection("Data Source=LENOVO-IKUCUKKO;Initial Catalog=Ogrenci;Integrated Security=True");
        private void button1_Click(object sender, EventArgs e)
        {
            listView1.Items.Clear();
            baglan.Open();
            SqlCommand komut = new SqlCommand("Select * from ogrencibilgileri", baglan);
            SqlDataReader oku = komut.ExecuteReader();
            while (oku.Read())
            {
                listView1.Items.Add(oku["OgrNo"].ToString());
                listView1.Items[listView1.Items.Count-1].SubItems.Add(oku["AdSoyad"].ToString());
                listView1.Items[listView1.Items.Count-1].SubItems.Add(oku["Sehir"].ToString());
            }
            baglan.Close();
        }
```

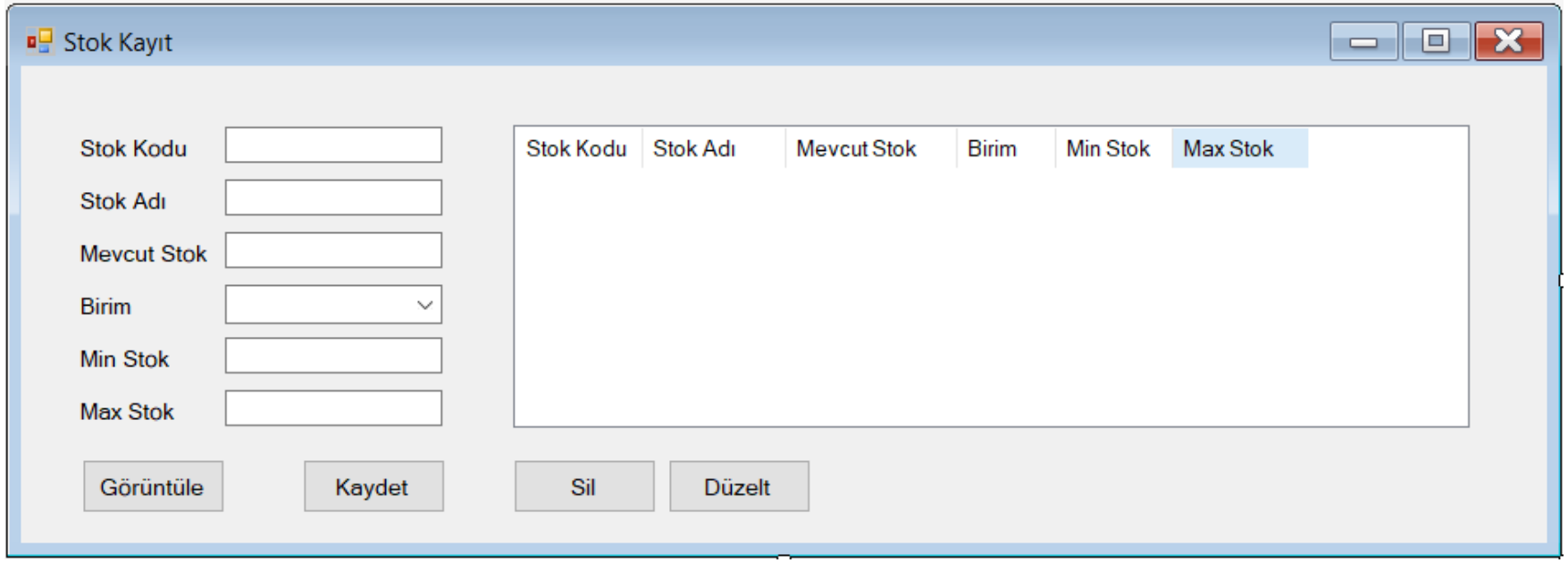
# Veritabanına Kayıt Ekleme




```
private void button3_Click(object sender, EventArgs e)
{
    baglan.Open();
    SqlCommand komut2 = new SqlCommand("Insert into ogrencibilgileri(OgrNo, AdSoyad, Sehir) values ('" + textBox1.Text+"', '"+textBox2.Text+"','"+textBox6.Text+"')", baglan);
    komut2.ExecuteNonQuery();
    baglan.Close();
    button1.PerformClick();
}
```

# Veritabanı Kullanımı - Malzeme Kayıt Projesi

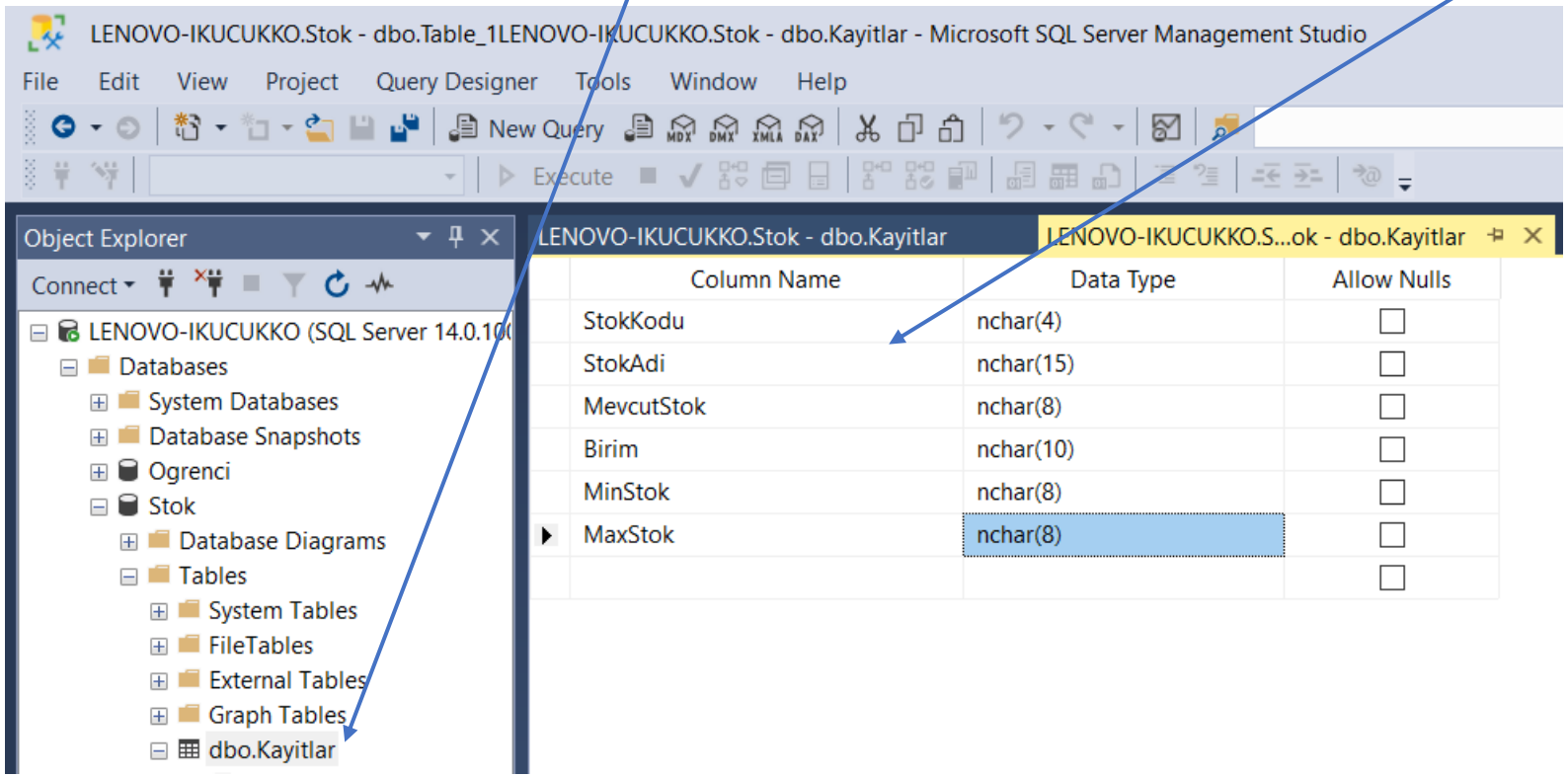
- Stok hareketlerini kaydetmek, mevcut stok miktarlarını görüntülemek, veritabanında silme ve düzeltme yapmak amacıyla verilen örnek için aşağıdaki gibi bir form tasarlanır.




- Bu projede öncekinden farklı olarak silme ve düzeltme işlemleri de yapılacaktır.

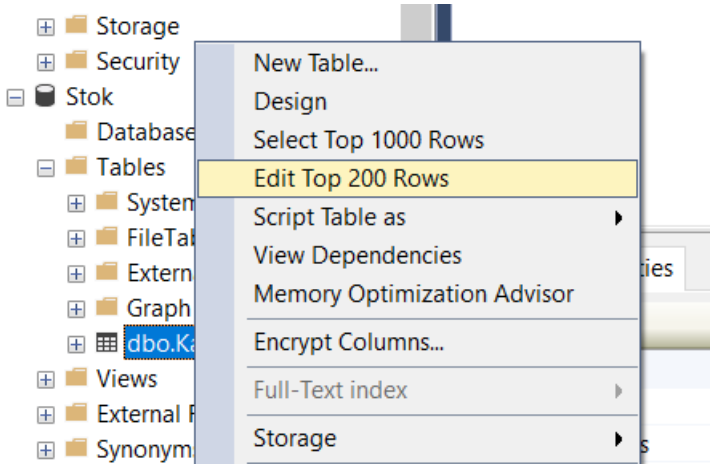
- Yardım Videosu: https://www.youtube.com/watch?v=45tftN9IeQ0&t=2581s

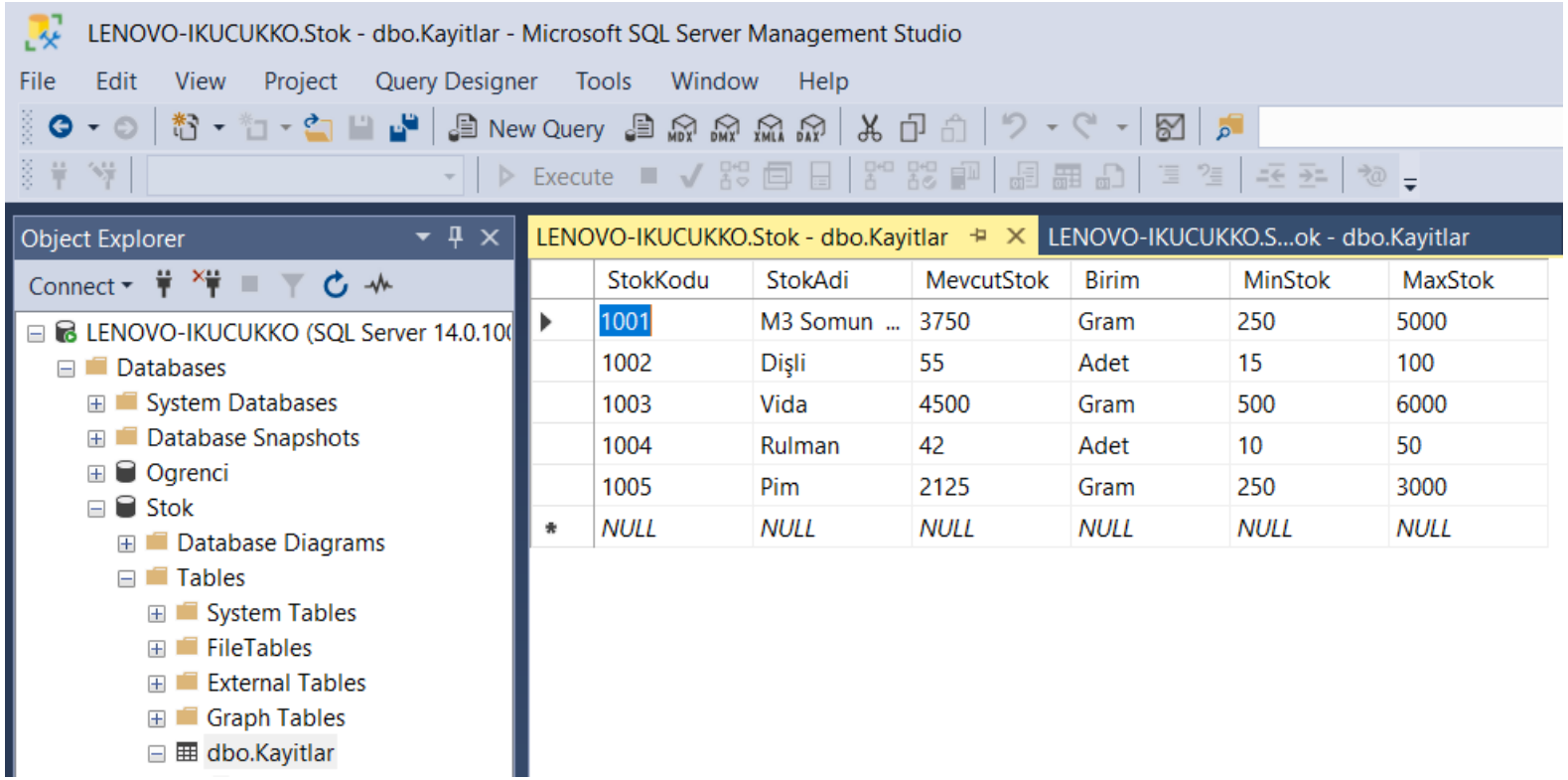
Bu proje için **Stok** isminde yeni bir veritabanı oluşturulmalıdır. Bunun için yine **Database -> New Database** yolu izlenerek **Database Name** kısmına **Stok** ismi verildikten sonra **OK**'e tıklanır.
Oluşturulan veritabanına **Kayitlar** isminde bir tablo oluşturulur ve sütunları aşağıdaki gibi isimlendirilir.




**Kayitlar** tablosunun tasarımı

- Kayitlar tablosunun üzerine sağ tıklanarak **Edit Top 200 Rows**'a tıklanıp aşağıdaki gibi stok malzemeleri ve bilgiler eklenir.

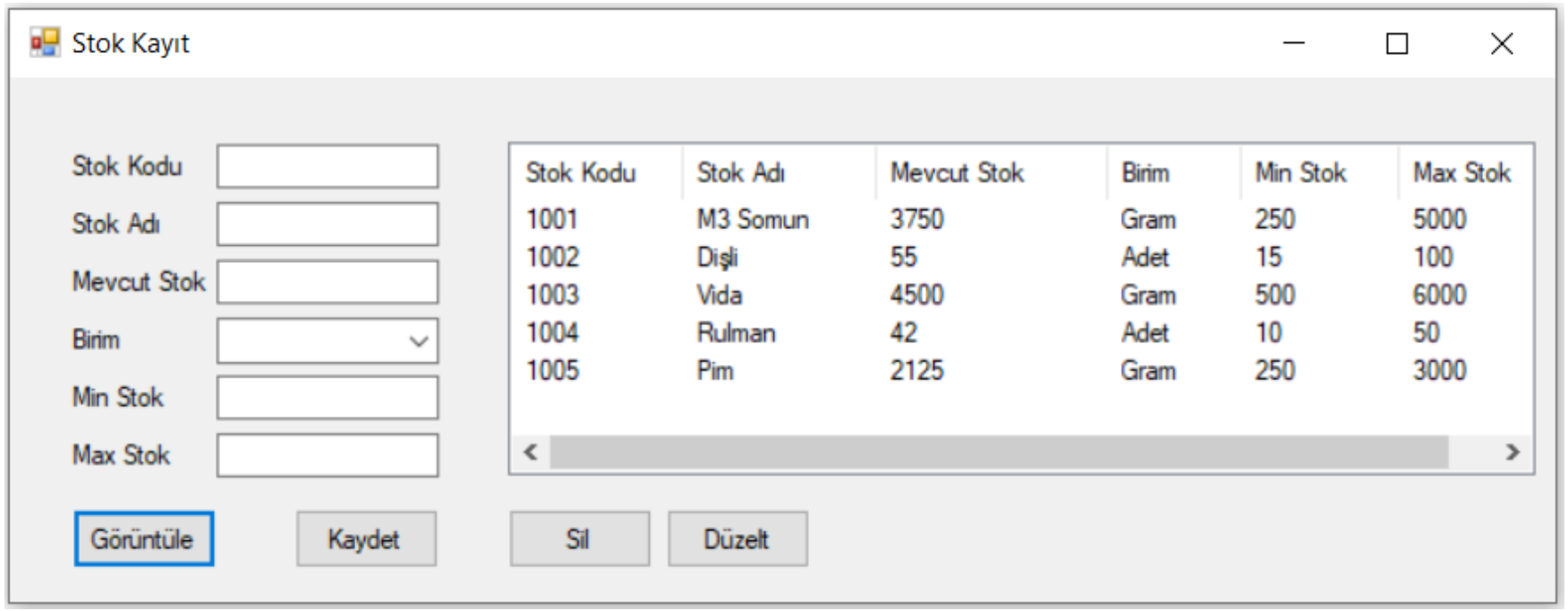



| StokKodu | StokAdi | MevcutStok | Birim | MinStok | MaxStok |
|---|---|---|---|---|---|
| 1001 | M3 Somun ... | 3750 | Gram | 250 | 5000 |
| 1002 | Dişli | 55 | Adet | 15 | 100 |
| 1003 | Vida | 4500 | Gram | 500 | 6000 |
| 1004 | Rulman | 42 | Adet | 10 | 50 |
| 1005 | Pim | 2125 | Gram | 250 | 3000 |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Kayitlar** tablosuna eklenen kayıtlar

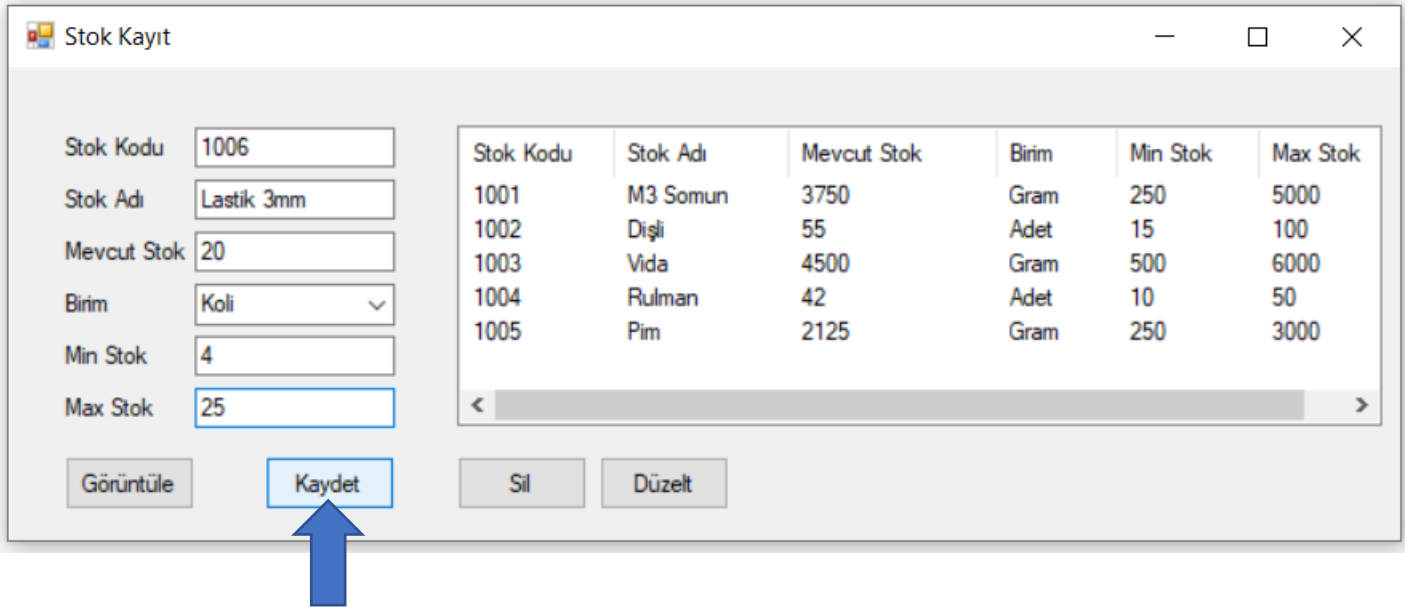
- Form tasarımı aşağıdaki gibidir. Butonların içerisine yazılacak komutlar da sonraki slaytlarda verilmiştir.



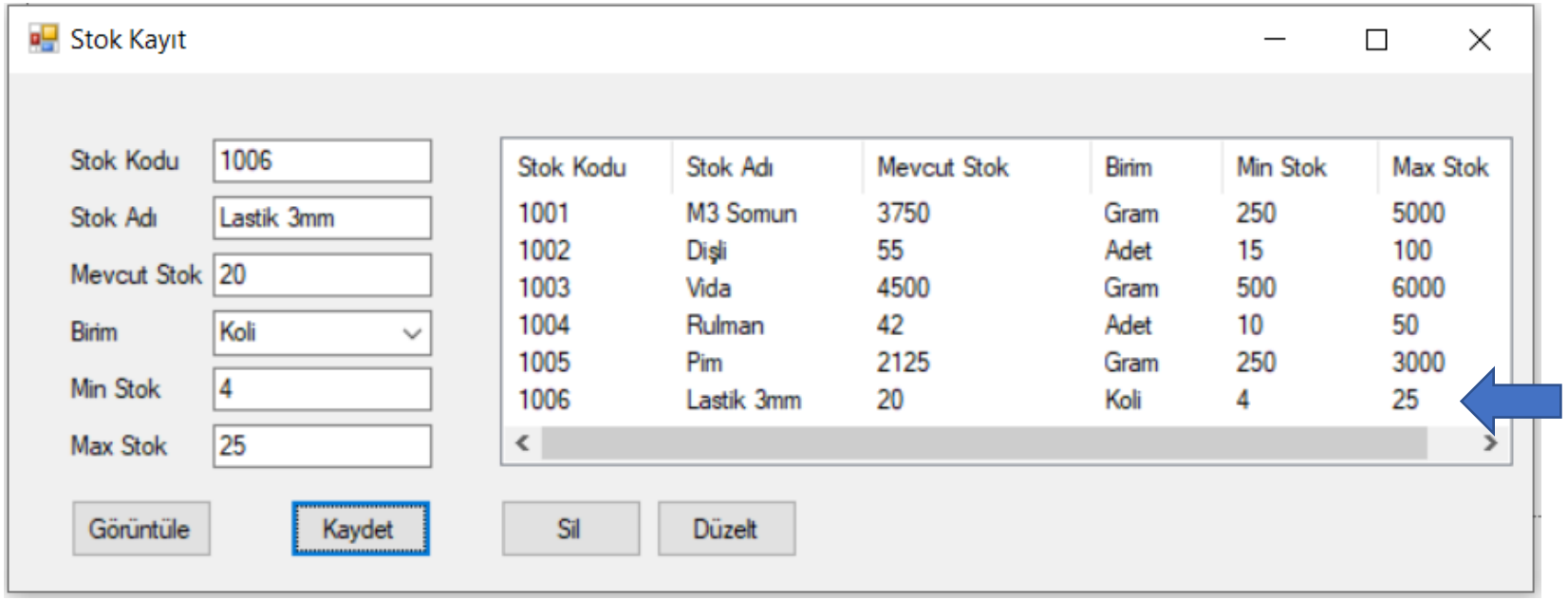

**FORM** tasarımı

```
using System.Data.SqlClient;

namespace StokKayit
{
    public partial class Form1 : Form
    {
        SqlConnection baglan = new SqlConnection("Data Source=LENOVO-IKUCUKKO;Initial catalog=Stok; Integrated Security=true");
        private void button1_Click(object sender, EventArgs e) //Görüntüle Butonu
        {
            listView1.Items.Clear();
            baglan.Open();
            SqlCommand komut = new SqlCommand("Select * from kayitlar", baglan);
            SqlDataReader oku = komut.ExecuteReader();
            while (oku.Read())
            {
                listView1.Items.Add(oku["StokKodu"].ToString());
                listView1.Items[listView1.Items.Count - 1].SubItems.Add(oku["StokAdi"].ToString());
                listView1.Items[listView1.Items.Count - 1].SubItems.Add(oku["MevcutStok"].ToString());
                listView1.Items[listView1.Items.Count - 1].SubItems.Add(oku["Birim"].ToString());
                listView1.Items[listView1.Items.Count - 1].SubItems.Add(oku["MinStok"].ToString());
                listView1.Items[listView1.Items.Count - 1].SubItems.Add(oku["MaxStok"].ToString());
            }
            baglan.Close();
        }
```

**Görüntüle** butonu komutları

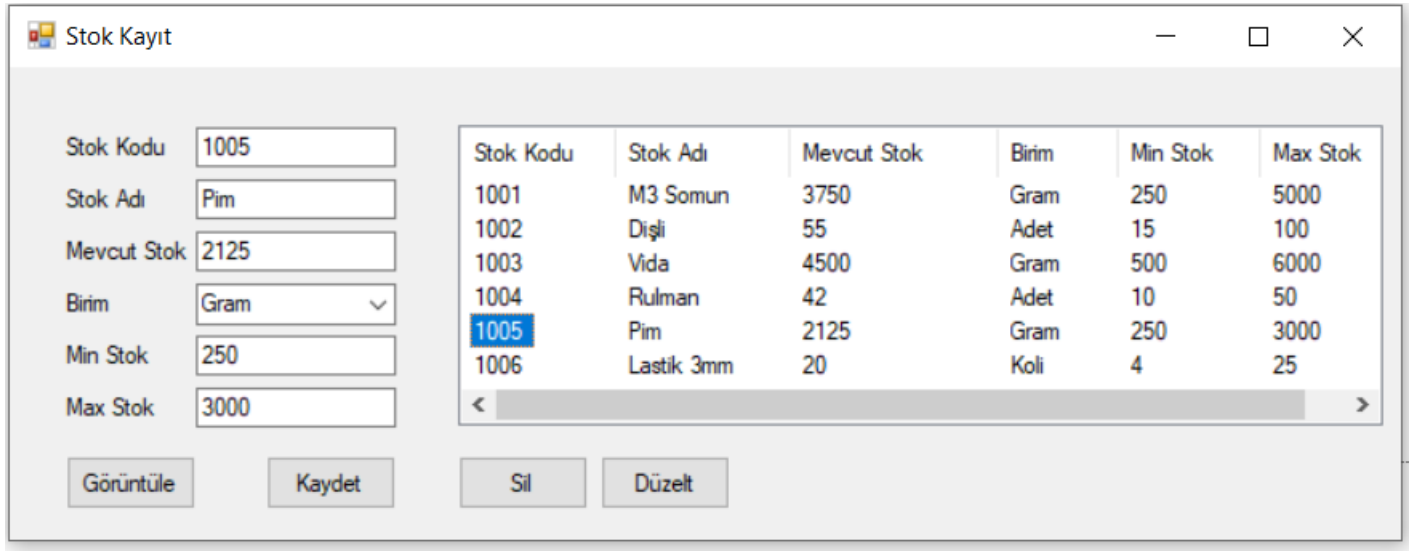

Stok Kayıt
Stok Kodu
1006
Stok Adı
Lastik 3mm
Mevcut Stok
20
Birim
Koli
Min Stok
4
Max Stok
25
Stok Kodu Stok Adı Mevcut Stok Birim Min Stok Max Stok
1001 M3 Somun 3750 Gram 250 5000
1002 Dişli 55 Adet 15 100
1003 Vida 4500 Gram 500 6000
1004 Rulman 42 Adet 10 50
1005 Pim 2125 Gram 250 3000
Görüntüle
Kaydet
Sil
Düzelt

**KAYDET**

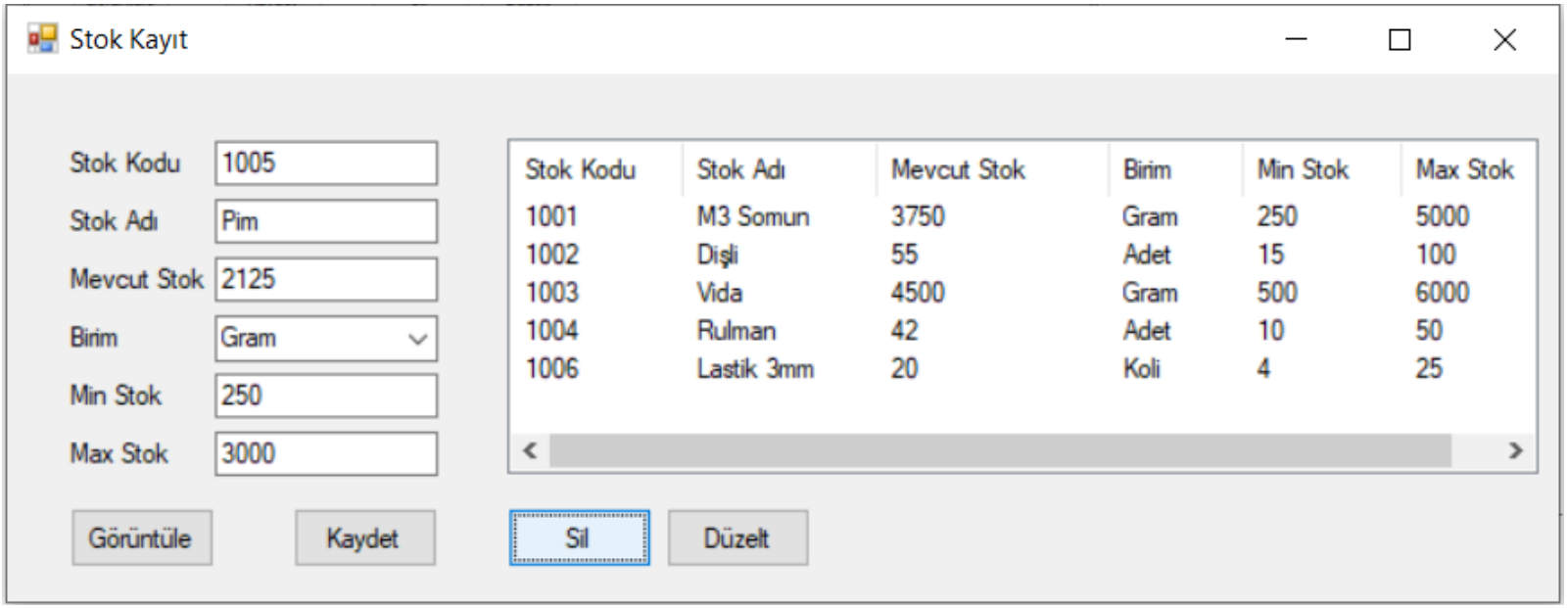

Stok Kayıt
Stok Kodu
1006
Stok Adı
Lastik 3mm
Mevcut Stok
20
Birim
Koli
Min Stok
4
Max Stok
25
Stok Kodu Stok Adı Mevcut Stok Birim Min Stok Max Stok
1001 M3 Somun 3750 Gram 250 5000
1002 Dişli 55 Adet 15 100
1003 Vida 4500 Gram 500 6000
1004 Rulman 42 Adet 10 50
1005 Pim 2125 Gram 250 3000
1006 Lastik 3mm 20 Koli 4 25
Görüntüle
Kaydet
Sil
Düzelt

```
private void button2_Click(object sender, EventArgs e) //Kaydet Butonu
{
    baglan.Open();
    SqlCommand komut2 = new SqlCommand("Insert into Kayitlar (StokKodu,StokAdi,MevcutStok,Birim,MinStok,MaxStok) " +
        "values ('"+textBox1.Text+"','"+ textBox2.Text + "','"+ textBox3.Text + "','"+comboBox1.Text+"','"+ textBox5.Text + "','"+ textBox6.Text + "')",baglan);
    komut2.ExecuteNonQuery();
    baglan.Close();
    button1.PerformClick();
}

private void Form1_Load(object sender, EventArgs e)
{
    button1.PerformClick();
}
```

**Kaydet** butonu ve **Form1_Load** komutları

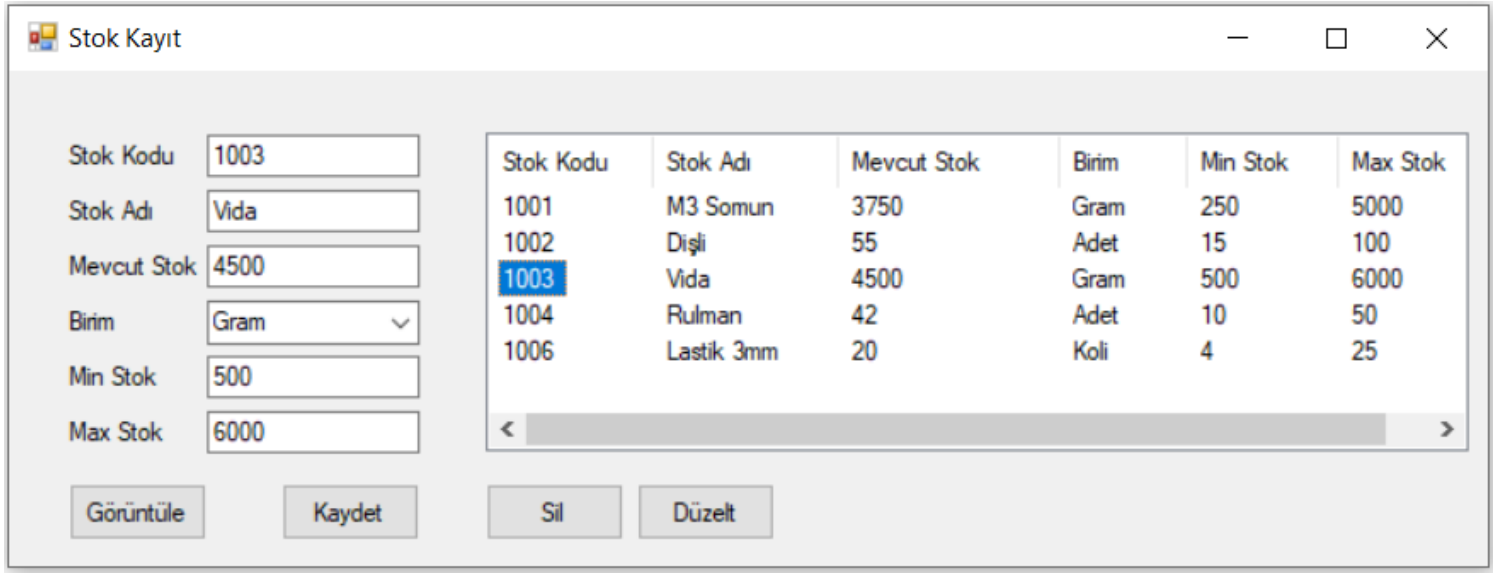

Stok Kayıt
Stok Kodu 1005
Stok Adı Pim
Mevcut Stok 2125
Birim Gram
Min Stok 250
Max Stok 3000
Stok Kodu Stok Adı Mevcut Stok Birim Min Stok Max Stok
1001 M3 Somun 3750 Gram 250 5000
1002 Dişli 55 Adet 15 100
1003 Vida 4500 Gram 500 6000
1004 Rulman 42 Adet 10 50
1005 Pim 2125 Gram 250 3000
1006 Lastik 3mm 20 Koli 4 25
Görüntüle
Kaydet
Sil
Düzelt

**SİL**


Stok Kayıt
Stok Kodu 1005
Stok Adı Pim
Mevcut Stok 2125
Birim Gram
Min Stok 250
Max Stok 3000
Stok Kodu Stok Adı Mevcut Stok Birim Min Stok Max Stok
1001 M3 Somun 3750 Gram 250 5000
1002 Dişli 55 Adet 15 100
1003 Vida 4500 Gram 500 6000
1004 Rulman 42 Adet 10 50
1006 Lastik 3mm 20 Koli 4 25
Görüntüle
Kaydet
Sil
Düzelt

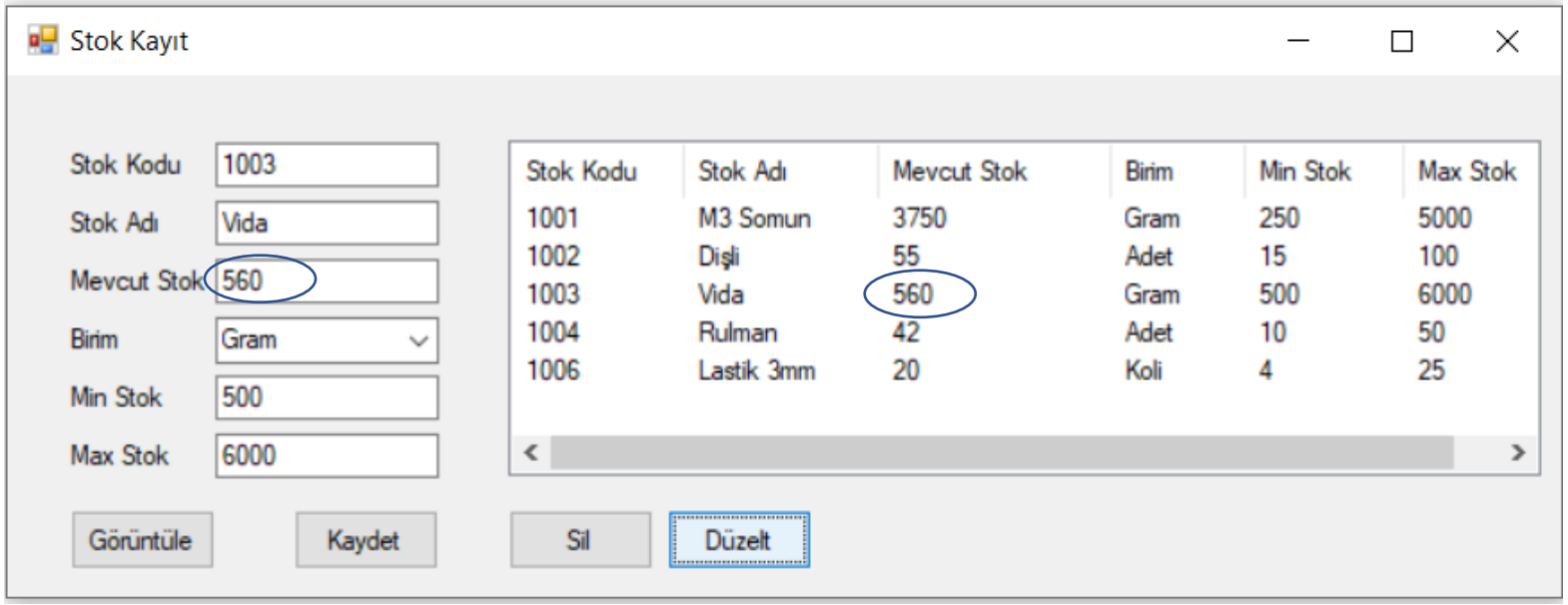
```
int kod = 0;
private void listView1_DoubleClick(object sender, EventArgs e) //listview'dan mouse ile seçim yapma
{
    kod = Convert.ToInt32(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[0].Text;//stokkodu
    textBox2.Text = listView1.SelectedItems[0].SubItems[1].Text;//stokadi
    textBox3.Text = listView1.SelectedItems[0].SubItems[2].Text;//mevcutstok
    comboBox1.Text = listView1.SelectedItems[0].SubItems[3].Text;//birim
    textBox5.Text = listView1.SelectedItems[0].SubItems[4].Text;//minstok
    textBox6.Text = listView1.SelectedItems[0].SubItems[5].Text;//maxstok
}

private void button3_Click(object sender, EventArgs e)//Sil Butonu
{
    baglan.Open();
    SqlCommand komut3 = new SqlCommand("Delete from kayitlar where StokKodu=(" + kod + ")", baglan);
    komut3.ExecuteNonQuery();
    baglan.Close();
    button1.PerformClick();
}
```

**Sil** butonu komutları




- 1003 Stok Kodlu malzeme mouse ile çift tıklanarak seçilir ve bilgilerde gerekli düzeltme yapıldıktan sonra DÜZELT butonuna tıklanır.

**DÜZELT**

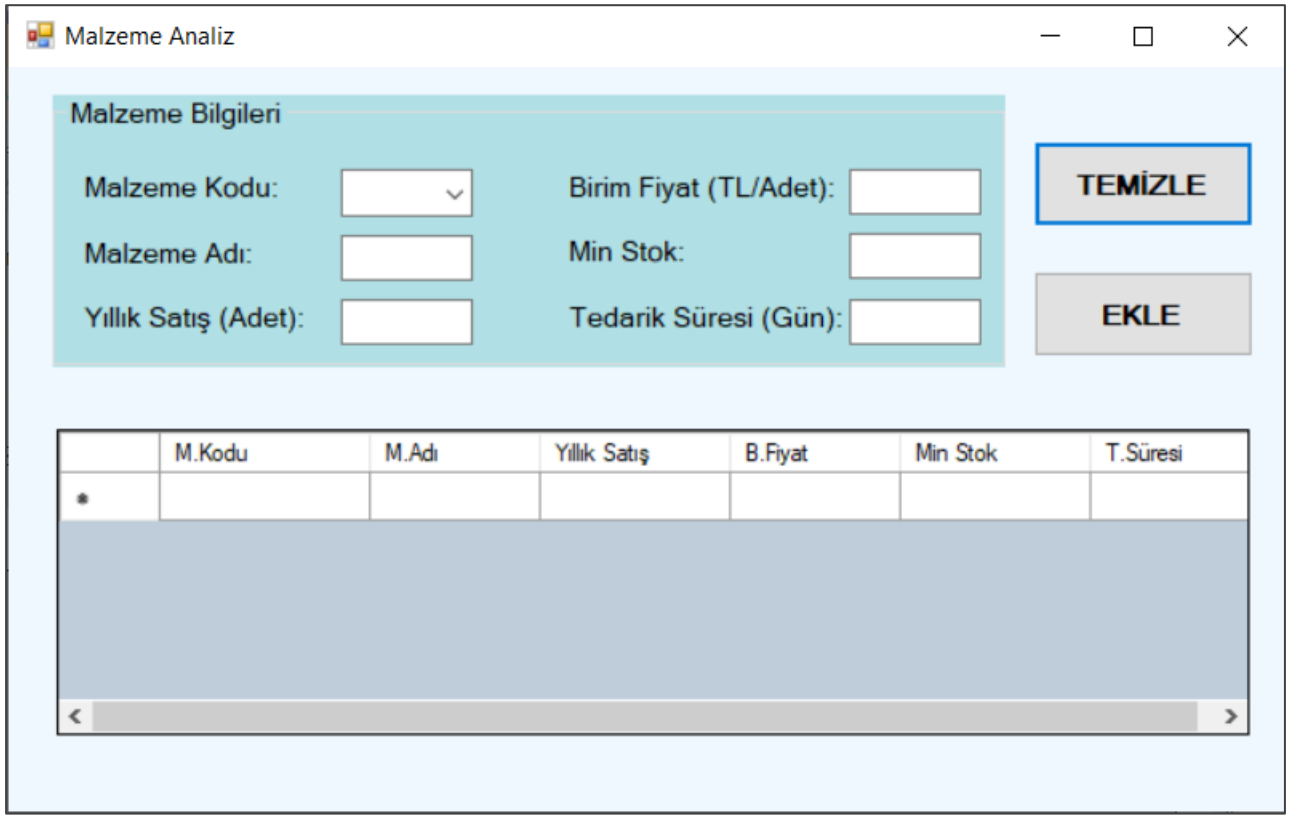


```
private void button4_Click(object sender, EventArgs e) //Düzelt Butonu
{
    baglan.Open();
    SqlCommand komut4 = new SqlCommand("Update Kayitlar set StokKodu='" + textBox1.Text.ToString() + "', StokAdi='" + textBox2.Text.ToString() + "'," +
        " MevcutStok='" + textBox3.Text.ToString()+ "', Birim='" + comboBox1.Text.ToString() + "', MinStok='" + textBox5.Text.ToString() + "'," +
        " MaxStok='" + textBox6.Text.ToString() + "' where StokKodu='" + kod+"'",baglan);
    komut4.ExecuteNonQuery();
    baglan.Close();
    button1.PerformClick();
}
```

**Düzelt** butonu komutları

# DataGridView

- Kayıtları tablolar halinde göstermek için tasarlanmış bir kontroldür. Kullanımı basittir ve kolaylıkla kişiselleştirilebilir.



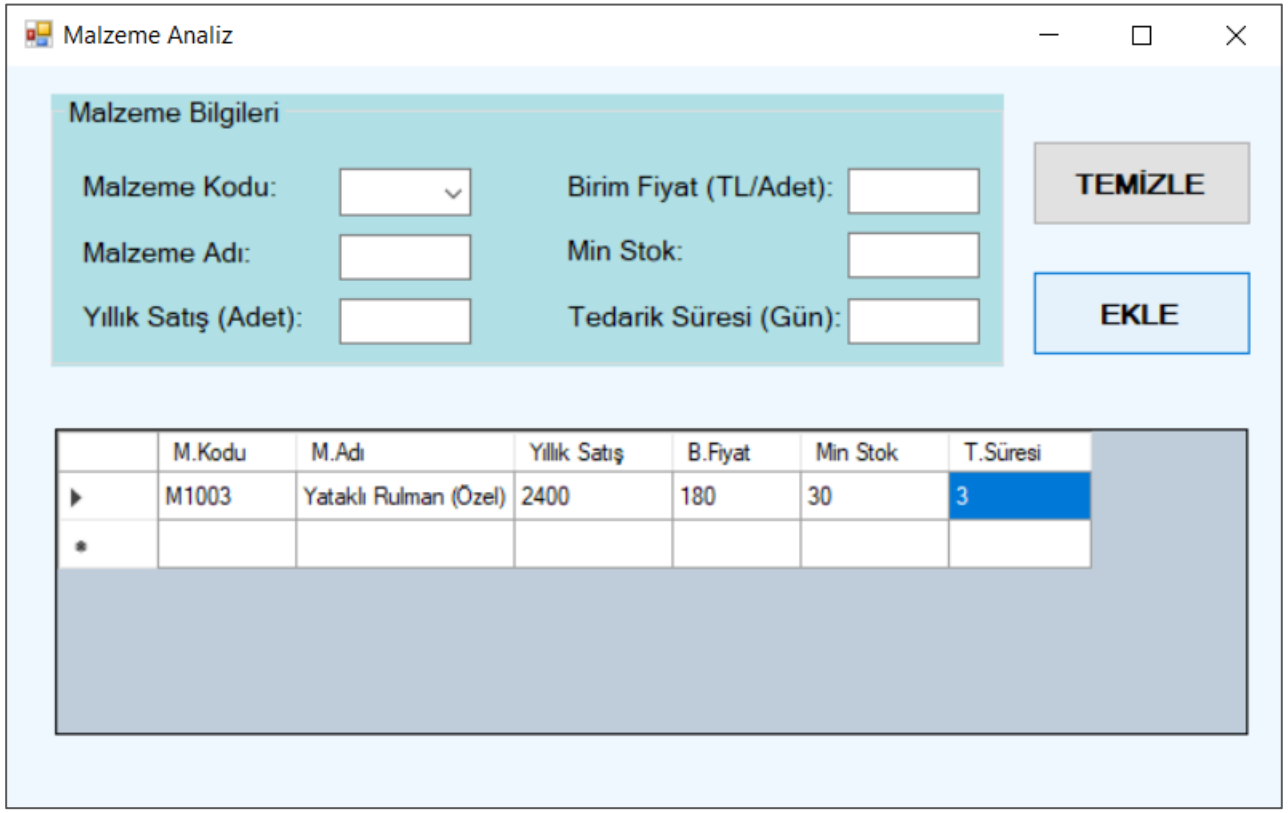

- DataGridView'da, ayarlanan sayı kadar kolon gösterilir.
- Sütun isimlerinin ayarlanması için kolon indisi verildikten sonra **.Name** özelliğiaşağıdaki gibi ayarlanır.

```
//sütun isimlerinin ayarlanması
dataGridView1.ColumnCount = 6; //kolon sayısı
dataGridView1.Columns[0].Name = "M.Kodu";
dataGridView1.Columns[1].Name = "M.Adı";
dataGridView1.Columns[2].Name = "Yıllık Satış";
dataGridView1.Columns[3].Name = "B.Fiyat";
dataGridView1.Columns[4].Name = "Min Stok";
dataGridView1.Columns[5].Name = "T.Süresi";
```

# DataGridView

- Sütun genişliklerini ayarlamak için **.Width** özelliği değiştirilebilir.

```
//Sütun başlangıç genişliklerini ayarla
dataGridView1.Columns[0].Width = 105;
dataGridView1.Columns[1].Width = 85;
dataGridView1.Columns[2].Width = 95;
dataGridView1.Columns[3].Width = 85;
dataGridView1.Columns[4].Width = 95;
dataGridView1.Columns[5].Width = 85;
```

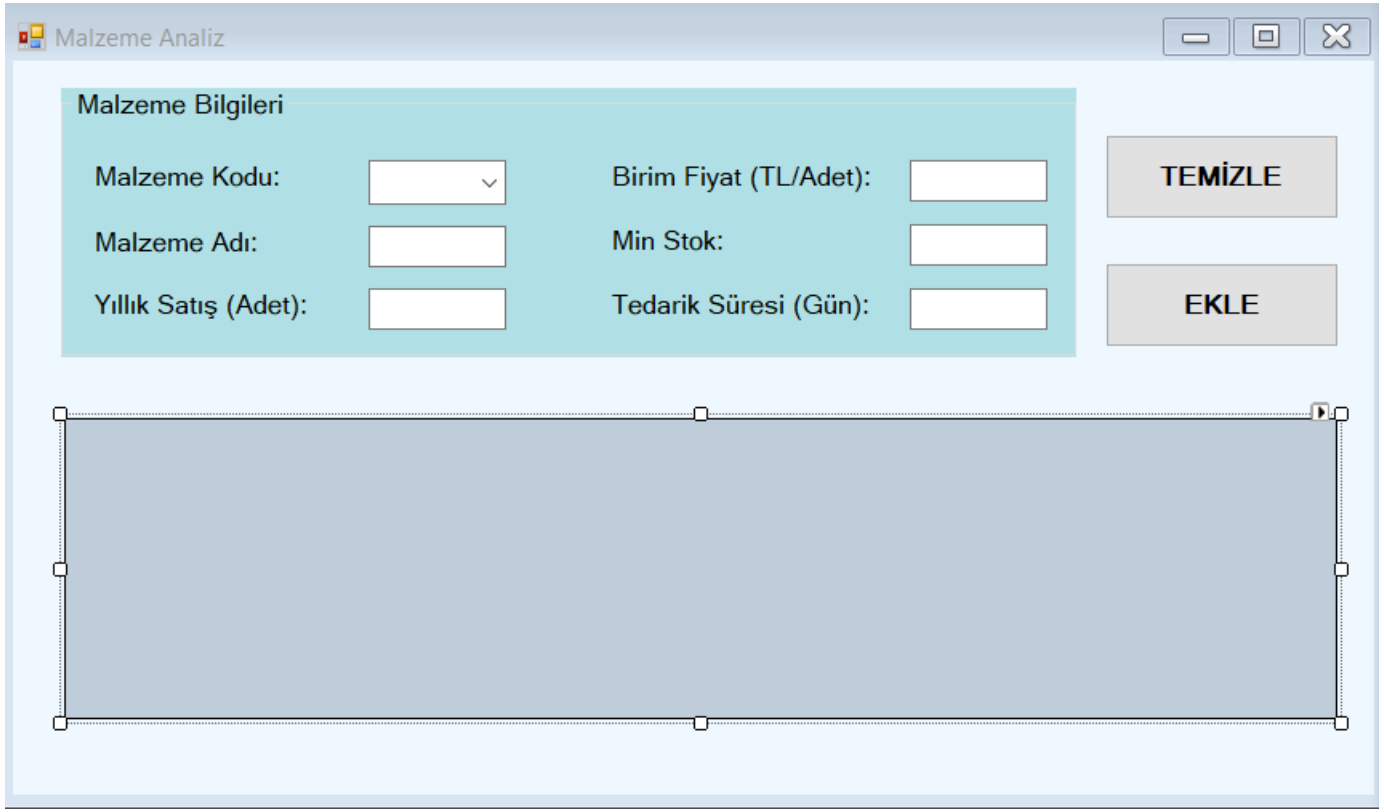


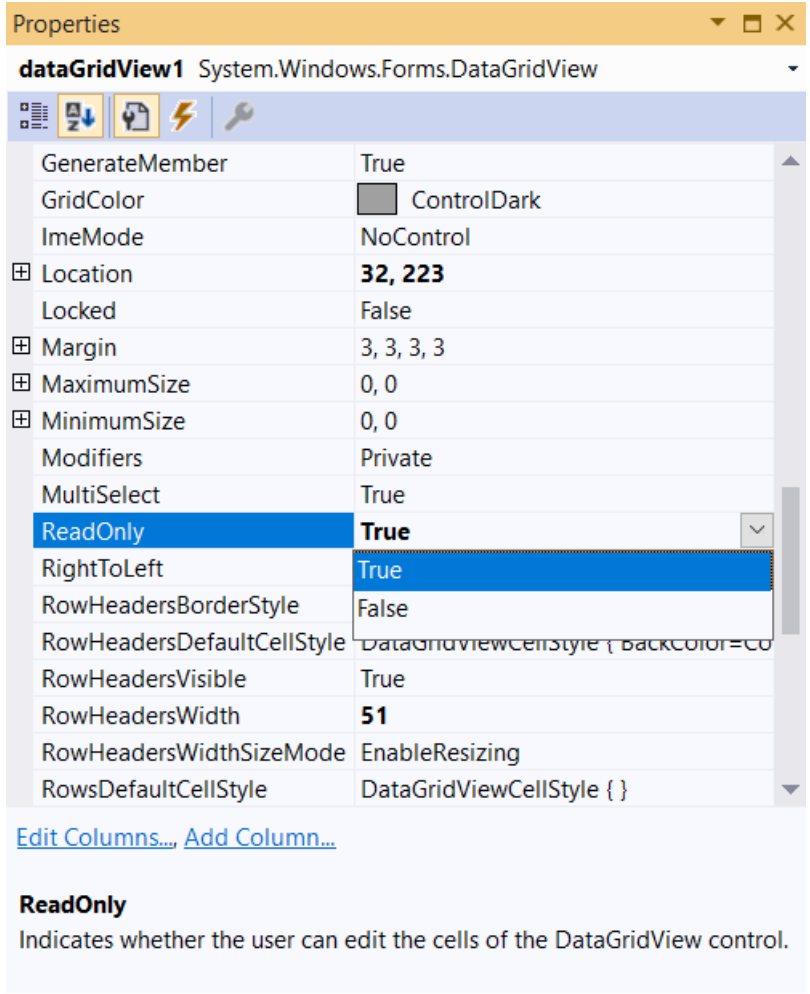

- Sütun genişliklerinin içeriğe göre otomatik olarak ayarlanması için aşağıdaki kod bloğu kullanılabilir.

```
//Sütunları İçeriğe Göre Otomatik genişlet
dataGridView1.AutoResizeColumns();
dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
```

# DataGridView

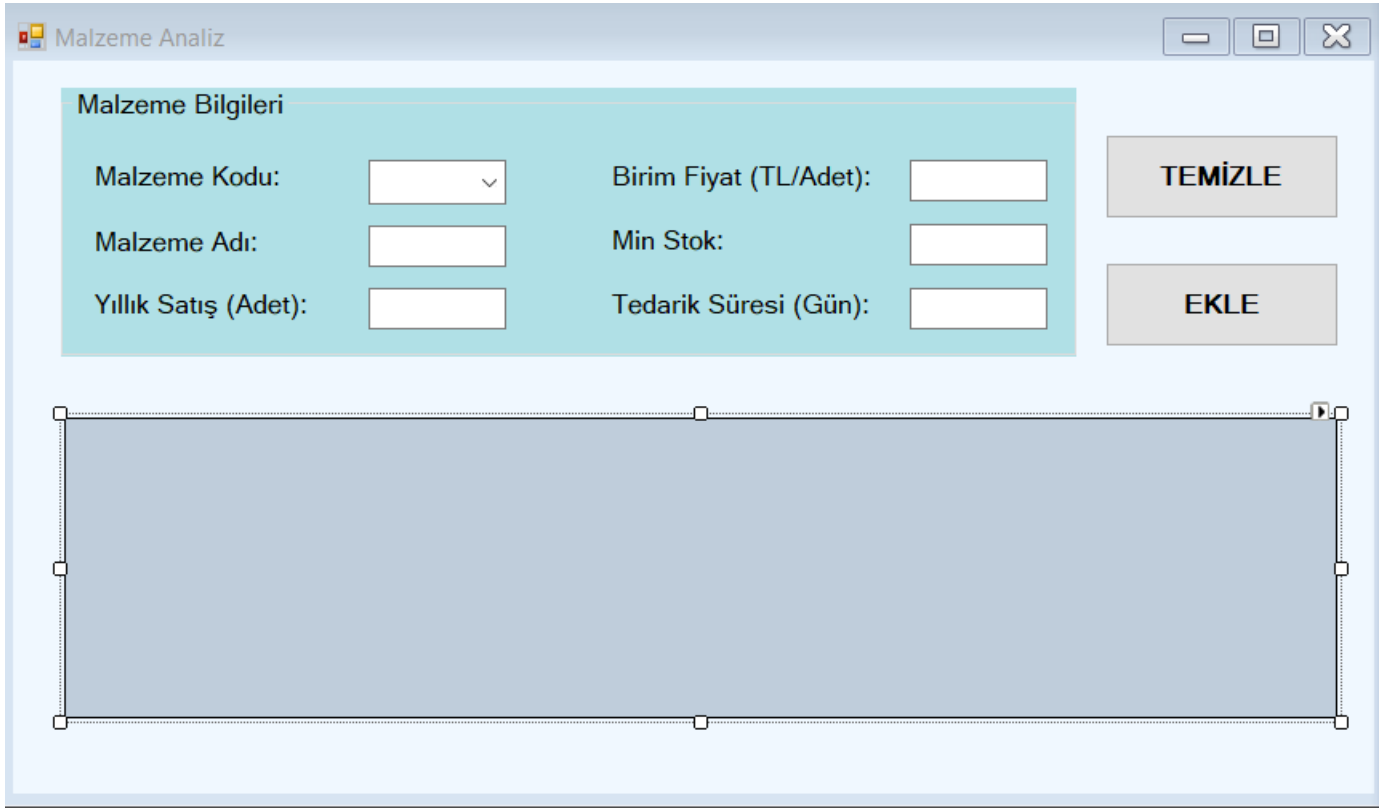
- Çalışma esnasında da DataGridView'a veri eklenmesi, değişiklik yapılması mümkündür.
- Bu özellik devre dışı bırakılmak istenirse DataGridView'ın **ReadOnly** özelliği True yapılmalıdır.

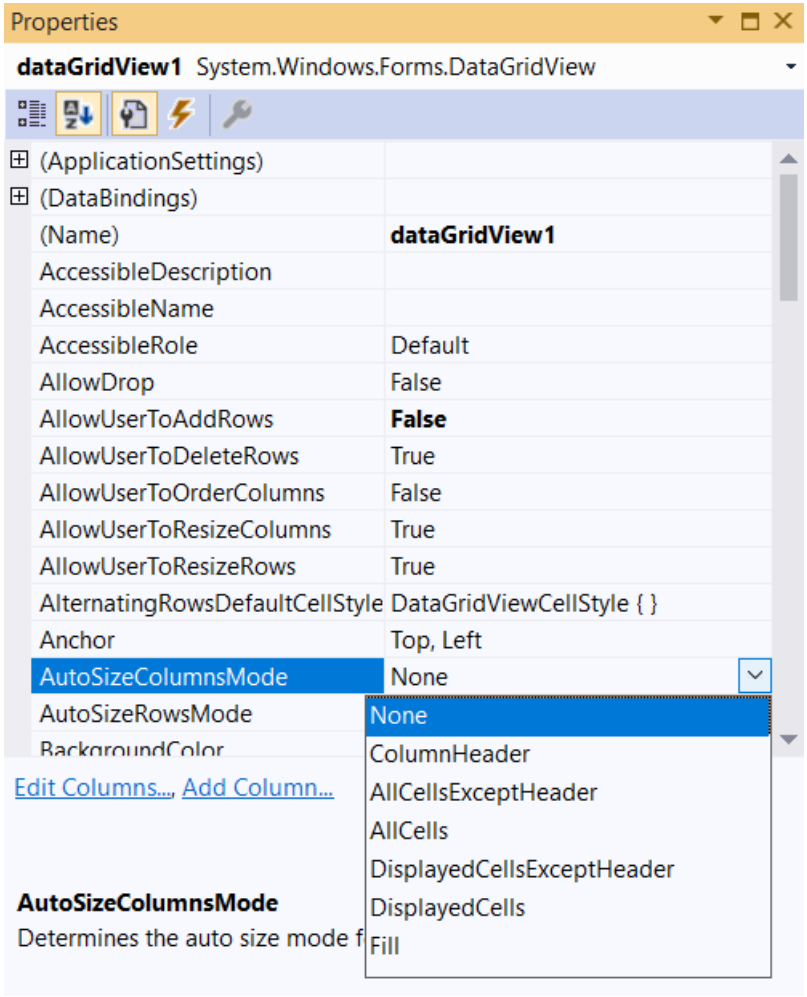


# DataGridView

- Ayrıca DataGridView aracının özelliklerinden, **sütun genişliğinin** başlığa göre veya hücre değerlerine göre otomatik olarak ayarlanması da yapılabilir.



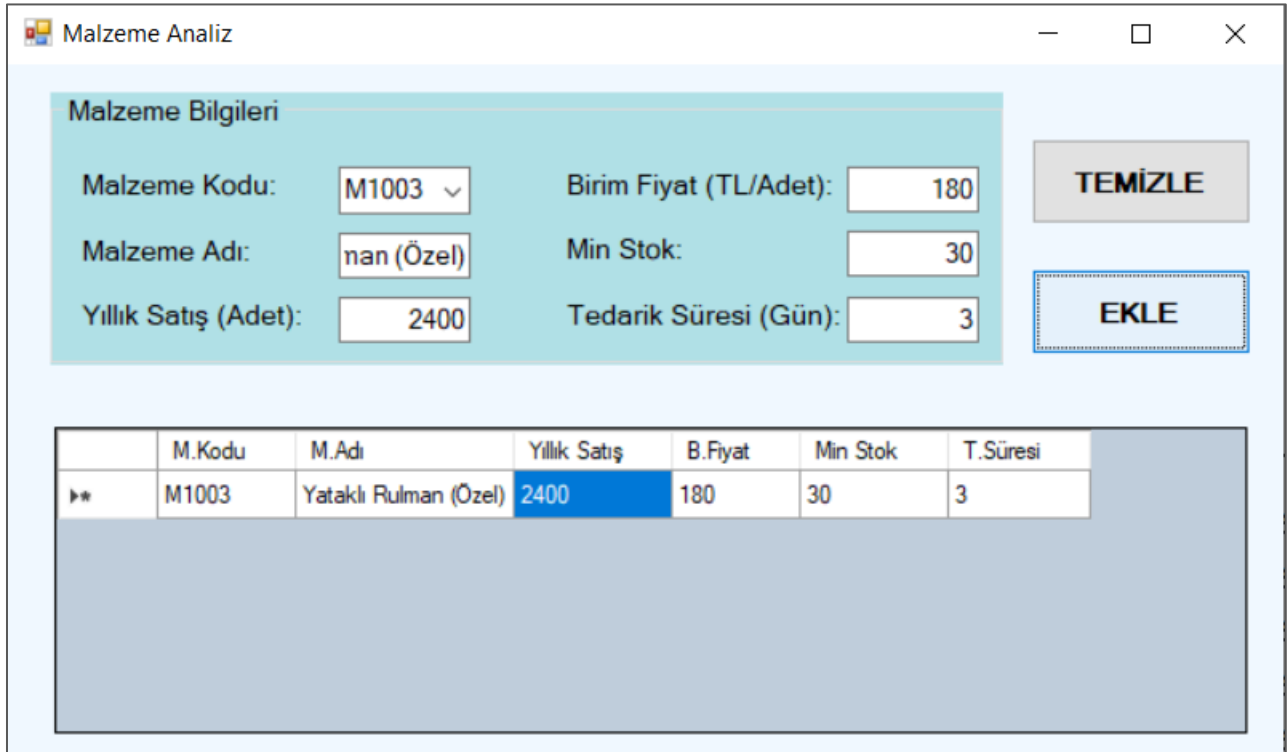



# DataGridView

- İlk satıra veri eklemek için aşağıdaki komutlar kullanılır. Bu komutlar veriyi comboBox ve textbox'lardan alarak DataGridView'a ekleyecektir.

```
dataGridView1.Rows[0].Cells[0].Value = comboBox1.Text;
dataGridView1.Rows[0].Cells[1].Value = textBox2.Text;
dataGridView1.Rows[0].Cells[2].Value = textBox3.Text;
dataGridView1.Rows[0].Cells[3].Value = textBox4.Text;
dataGridView1.Rows[0].Cells[4].Value = textBox5.Text;
dataGridView1.Rows[0].Cells[5].Value = textBox6.Text;
```

- Burada ilk satıra ait index numarasının "0" olduğuna dikkat edilmelidir.
- Satır numarası **Rows[ ]** içerisine yazıldıktan sonra sütun numaraları da yine "0." indeksten başlamak üzere **Cells[ ]** parantezleri içerisine yazılır.

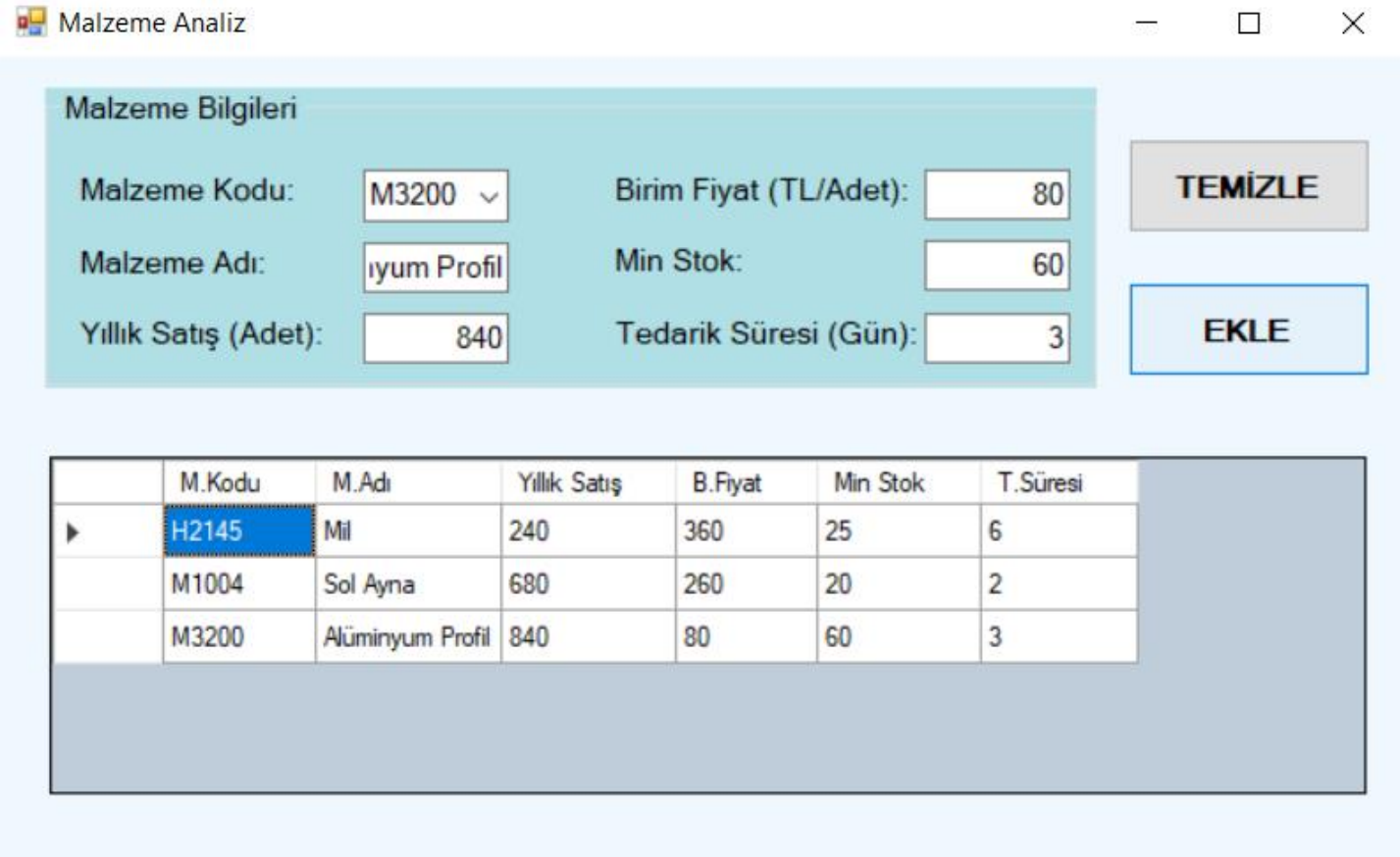


# DataGridView

- DataGridView’a başka yollarla da satır eklemesi yapılabilir. Aşağıda, bir önceki slaytta anlatılana göre daha kolay ve kullanışlı bir komut verilmektedir:

```
dataGridView1.Rows.Add(comboBox1.Text, textBox2.Text, textBox3.Text,
                       textBox4.Text, textBox5.Text, textBox6.Text);
```

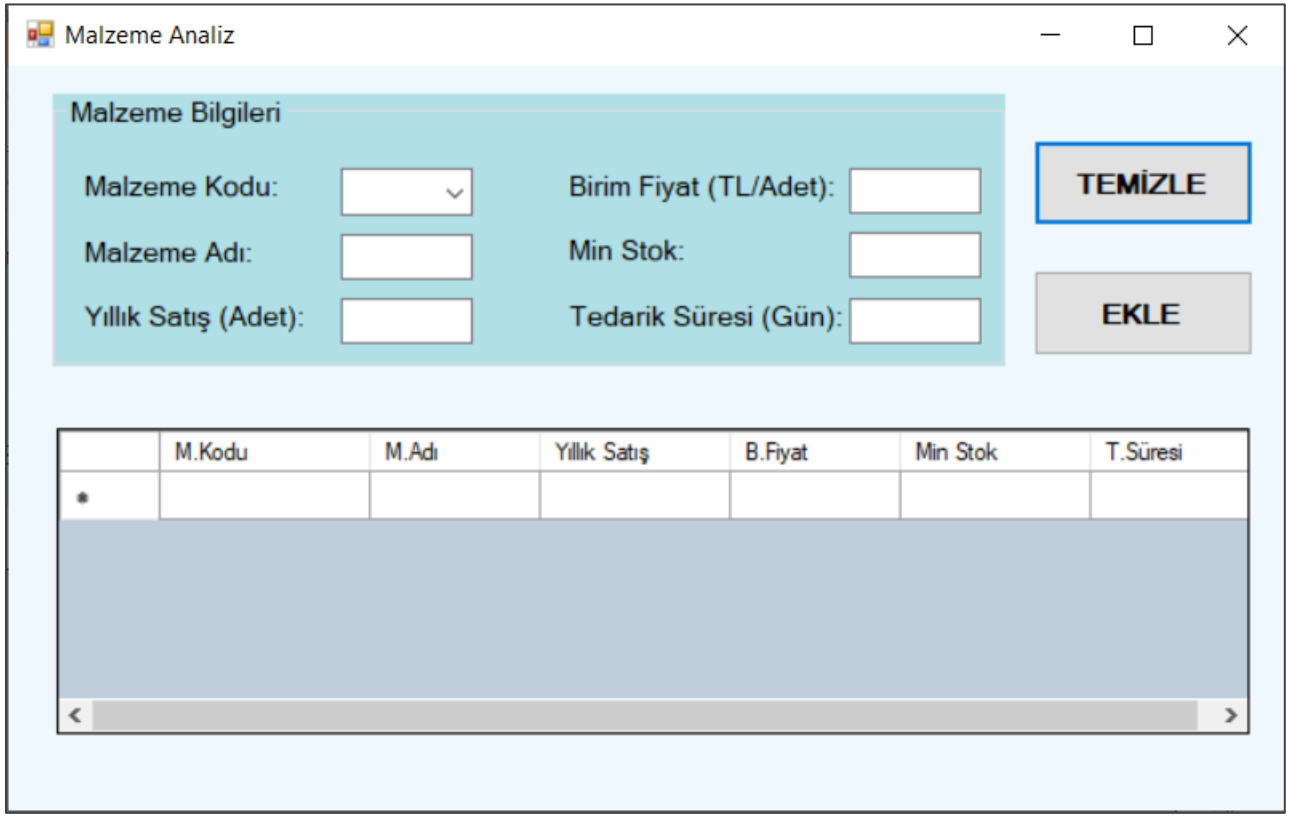
- Görüldüğü üzere alınan tüm veri tek bir komut satırı ile DataGridView’a eklenmiştir.



# DataGridView

- Aşağıdaki komutlar DataGridView'ı tamamen temizler.
- TEMİZLE butonunun içine yazılan bu kod bloğu, buton her tıklandığında DataGridView'da mevcut bütün veriyi silecektir.

```
dataGridView1.Rows.Clear();
dataGridView1.Refresh();
```



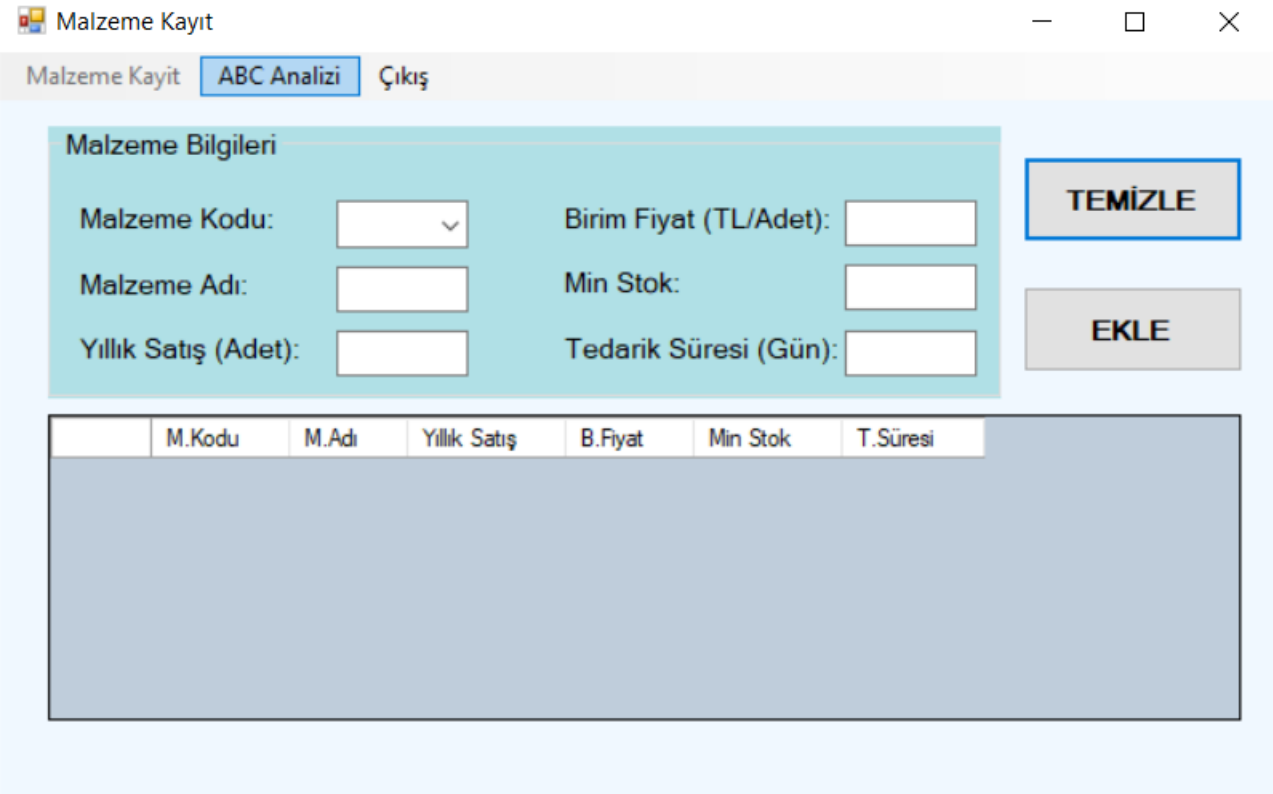
# DataGridView

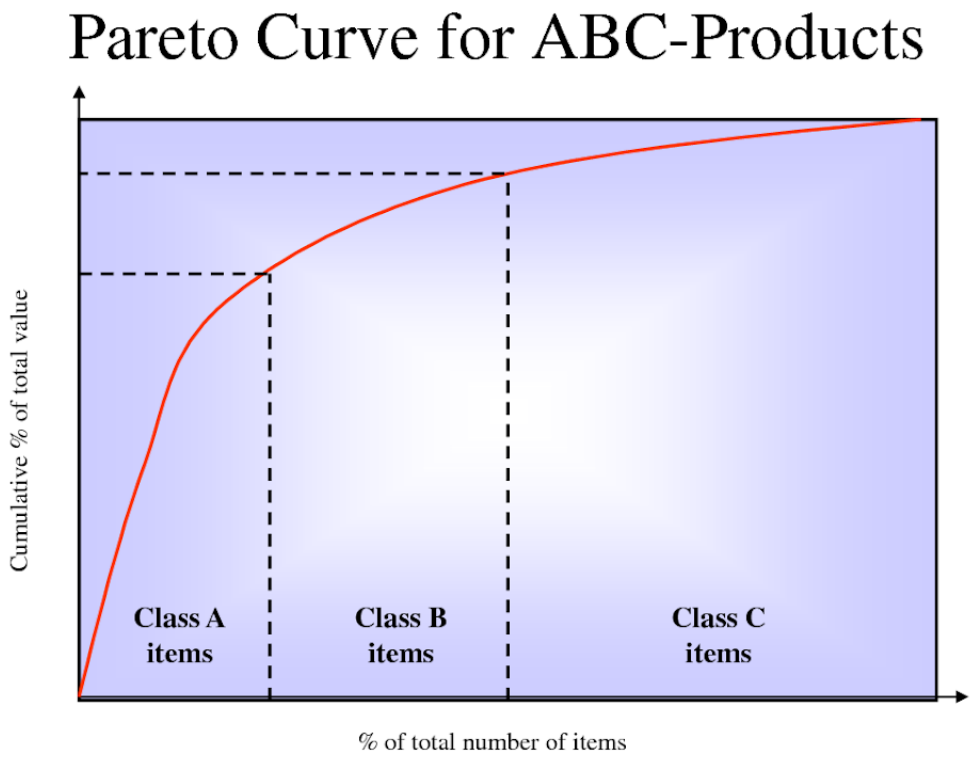
- DataGridView’da seçilen bir satırın Çalışma esnasında silinmesi mümkündür. Bu amaçla kullanılacak kodlar aşağıdaki gibidir.

```
private void button2_Click(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0)
    {
        dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);
    }
    else
    {
        MessageBox.Show("Lüffen silinecek satırı seçin.");
    }
}
```

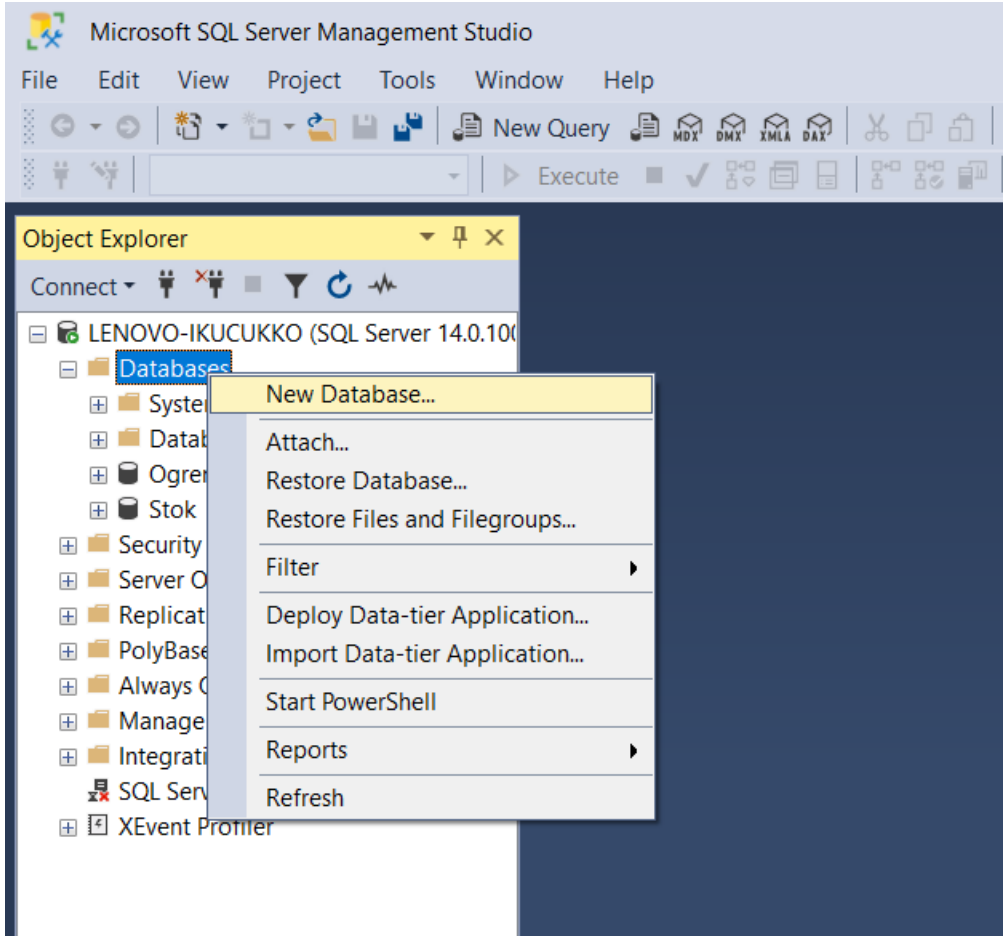
- Bunun için önce DataGridView’dan bir satır seçilmeli ve sonra ilgili butona tıklanmalıdır.
- Eğer bir satır seçilmeden butona tıklanırsa “Lütfen silinecek satırı seçin” şeklinde bir mesaj gösterilecektir.

- Projenin devamında aşağıdaki menu elemanlarında da görüldüğü gibi ABC analizinin yapılacağı bir program tasarlanacaktır.

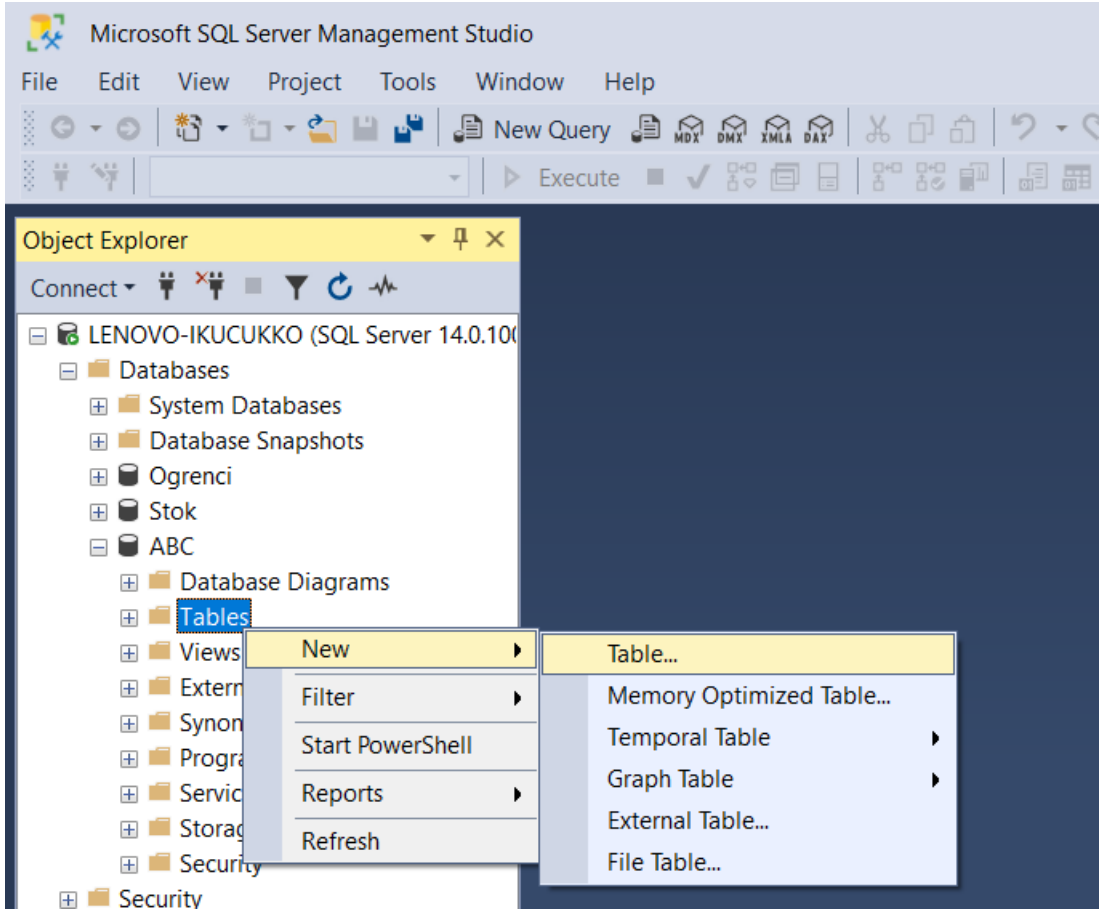


# DataGridView – Veritabanı Bağlantısı İşlemleri



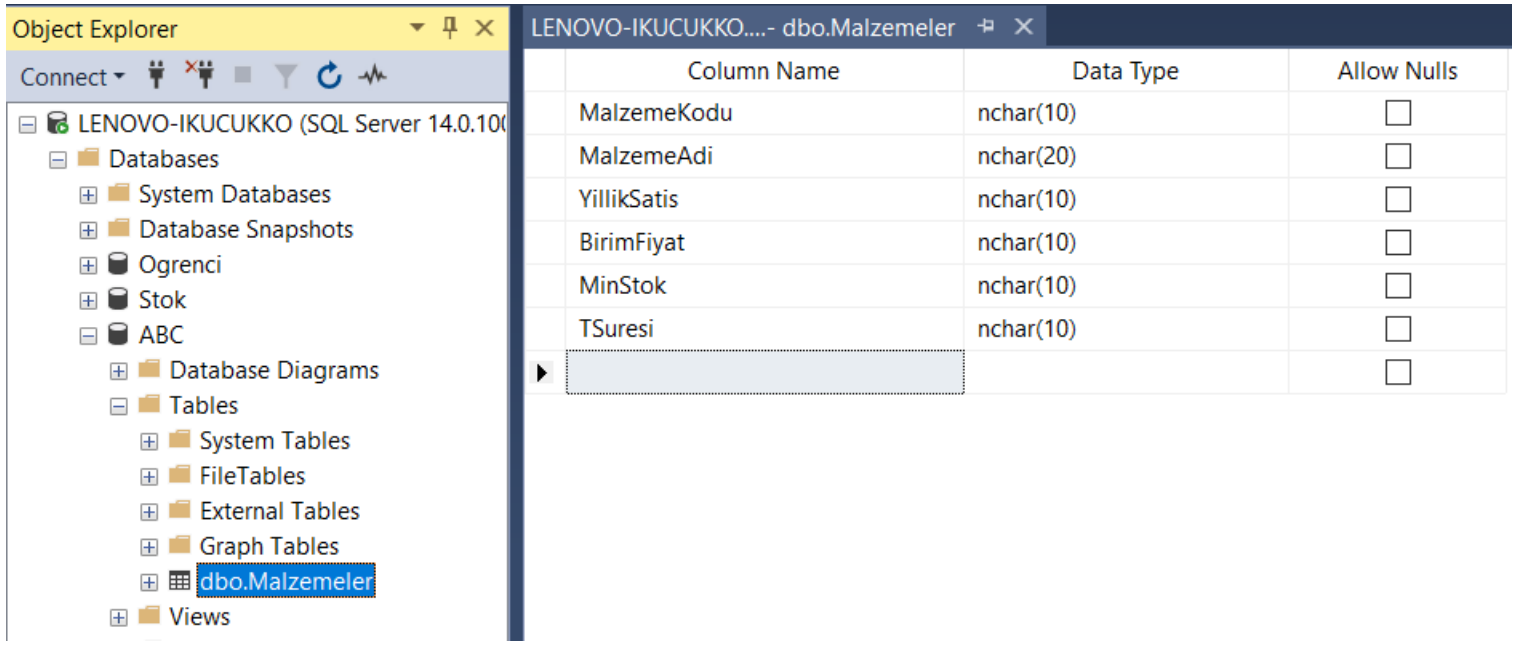

- Yeni Veritabanı Oluşturma




- Yeni Veritabanına Tablo Ekleme

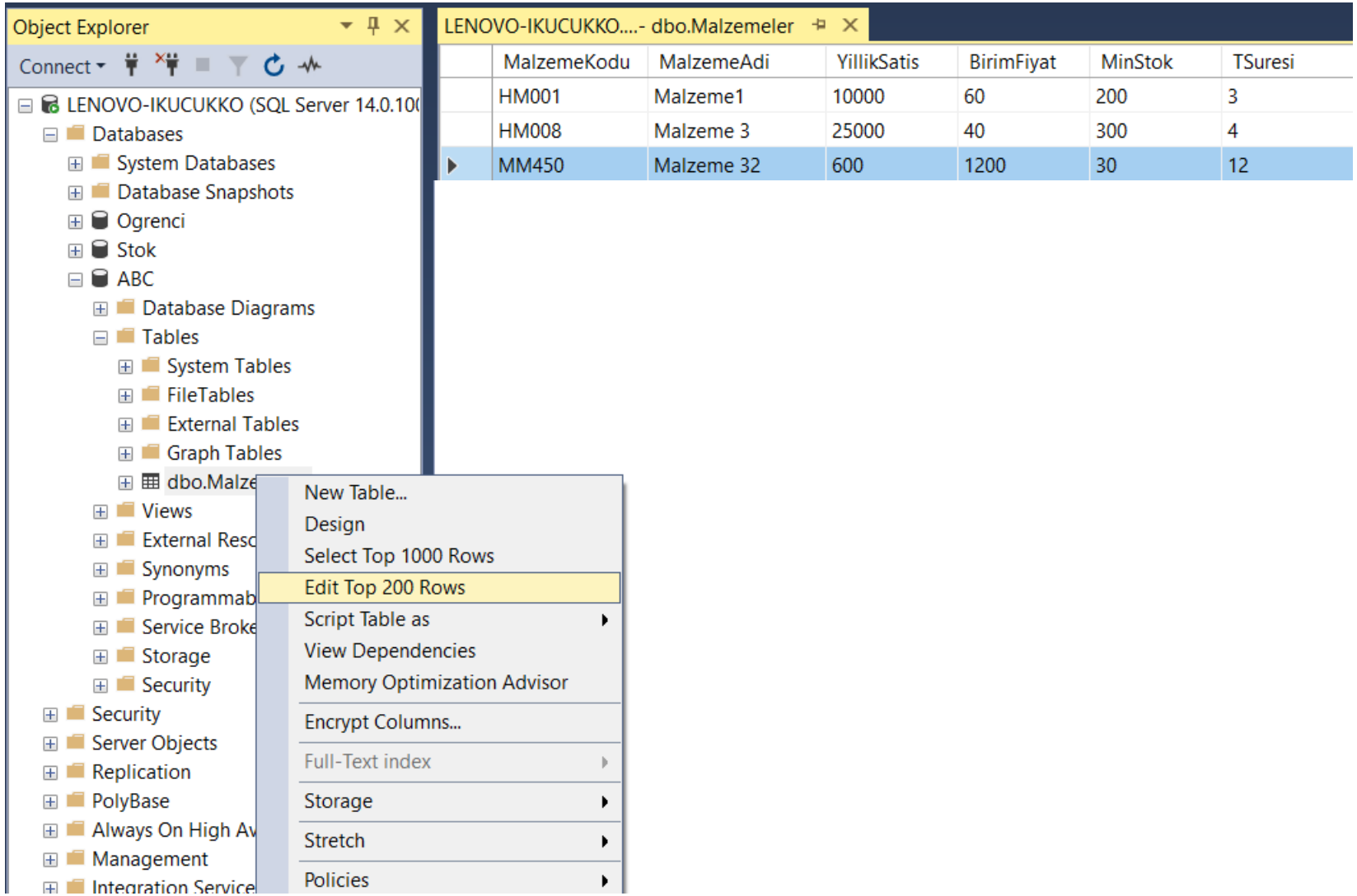
# DataGridView – Veritabanı Bağlantısı İşlemleri

- Tablo sütunlarının ayarlanması



# DataGridView – Veritabanı Bağlantısı İşlemleri

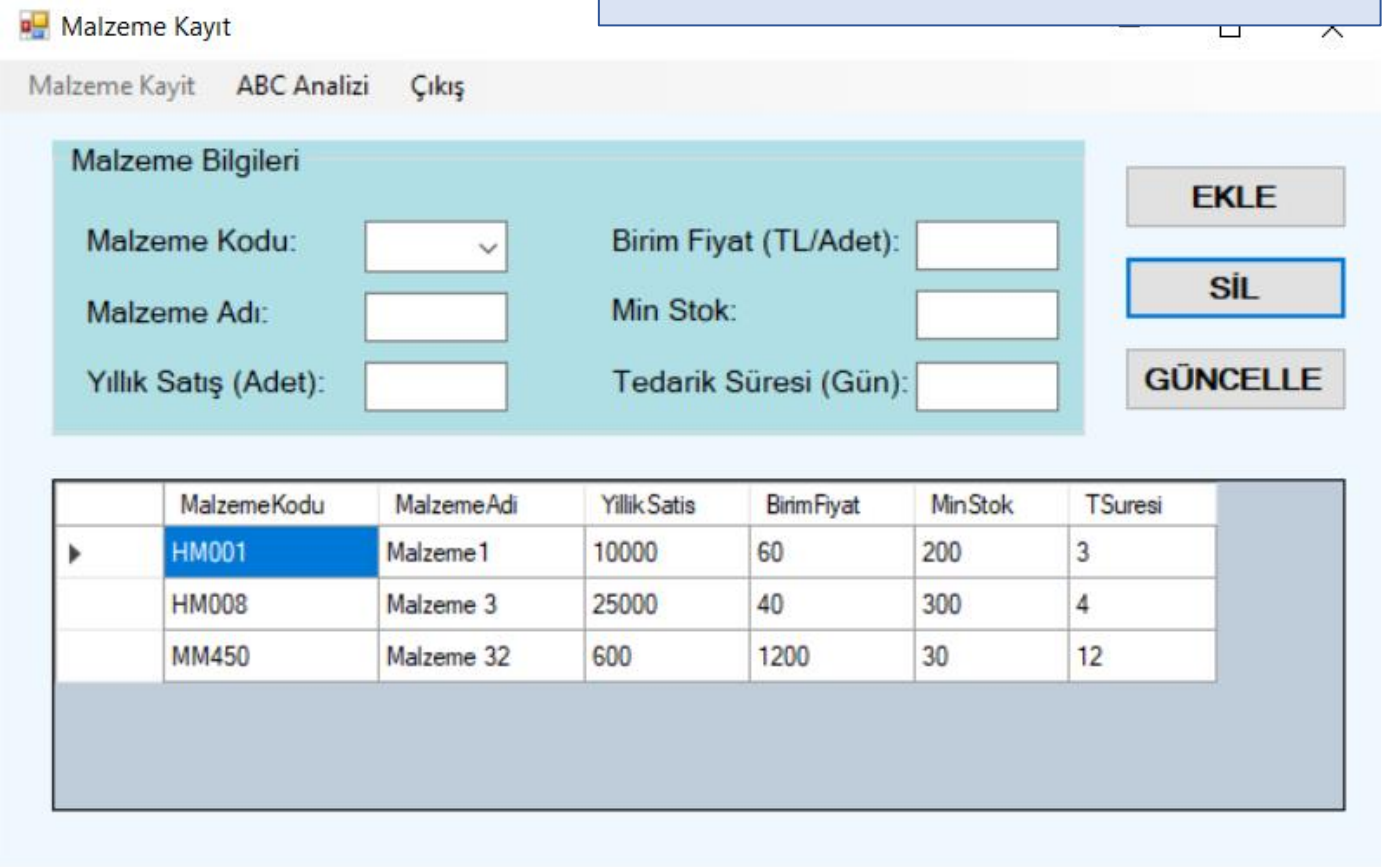
- Tabloya ilk verinin eklenmesi



# Listele()

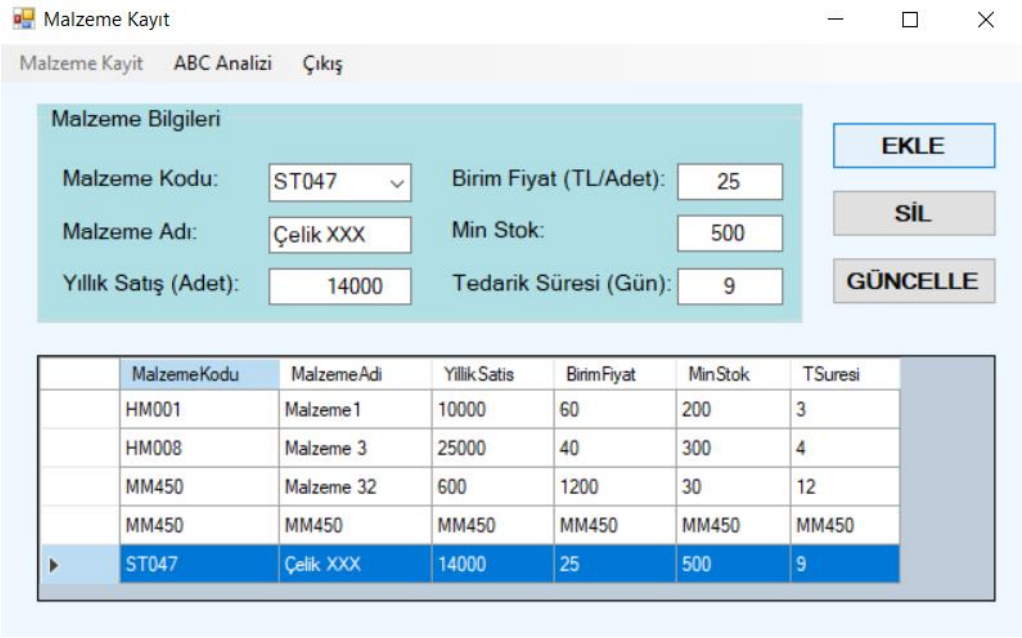
```
SqlConnection baglanti = new SqlConnection("Data Source=LENOVO-IKUCUKKO;Initial Catalog=ABC;Integrated Security=True");

private void listele()
{
    baglanti.Open();
    SqlDataAdapter da = new SqlDataAdapter("select *FROM Malzemeler", baglanti);
    DataTable tablo = new DataTable();
    da.Fill(tablo);
    dataGridView1.DataSource = tablo;
    baglanti.Close();
}

private void Form1_Load(object sender, EventArgs e)
{
    listele();
}
```

**Veritabanındaki Kayıtların DataGridView'da Listelenmesi**




- **listele ( )** metodu Form1'in Load olayında çağrılırsa form yüklendiği anda veritabanındaki kayıtlar DataGridView'da gösterilecektir.

# Veritabanına Ekleme




- **EKLE** Butonu Komutları:

```
private void button2_Click_1(object sender, EventArgs e) // EKLE Butonu
{
    String t1 = comboBox1.Text;
    String t2 = textBox2.Text;
    String t3 = textBox3.Text;
    String t4 = textBox4.Text;
    String t5 = textBox5.Text;
    String t6 = textBox6.Text;

    baglanti.Open();
    SqlCommand komut = new SqlCommand("INSERT INTO Malzemeler (MalzemeKodu, MalzemeAdi, YillikSatis, BirimFiyat, MinStok, TSuresi) " +
        "VALUES ('"+t1+"','"+t2+"','"+t3+"','"+t4+"','"+t5+"','"+t6+"')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();

    listele();
}
```

- ComboBox ve TextBox’lara girilen veriler EKLE butonuna tıklanarak veritabanına kayıt edilmek üzere gönderilir.
- Sonrasında veritabanındaki kayıtların güncel listesini DataGridView’da göstermek için **listele( )** metodu çağrılır.

# Veritabanından Silme

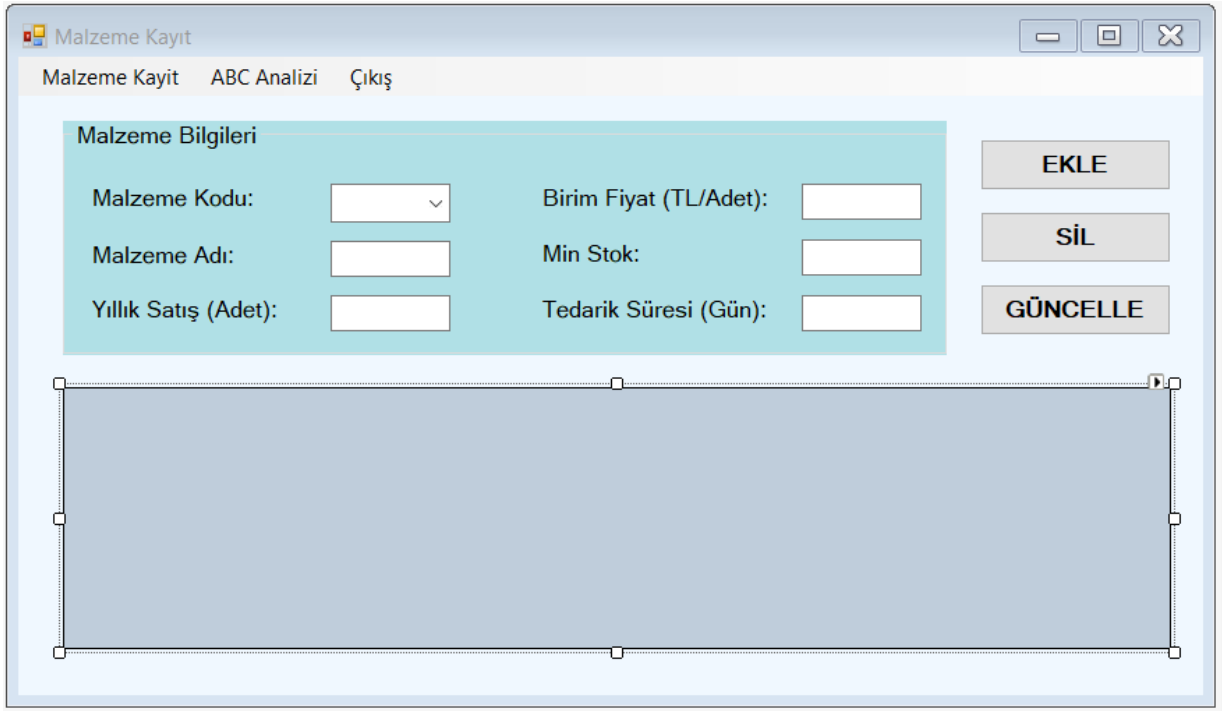
- **SİL** Butonu Komutları:

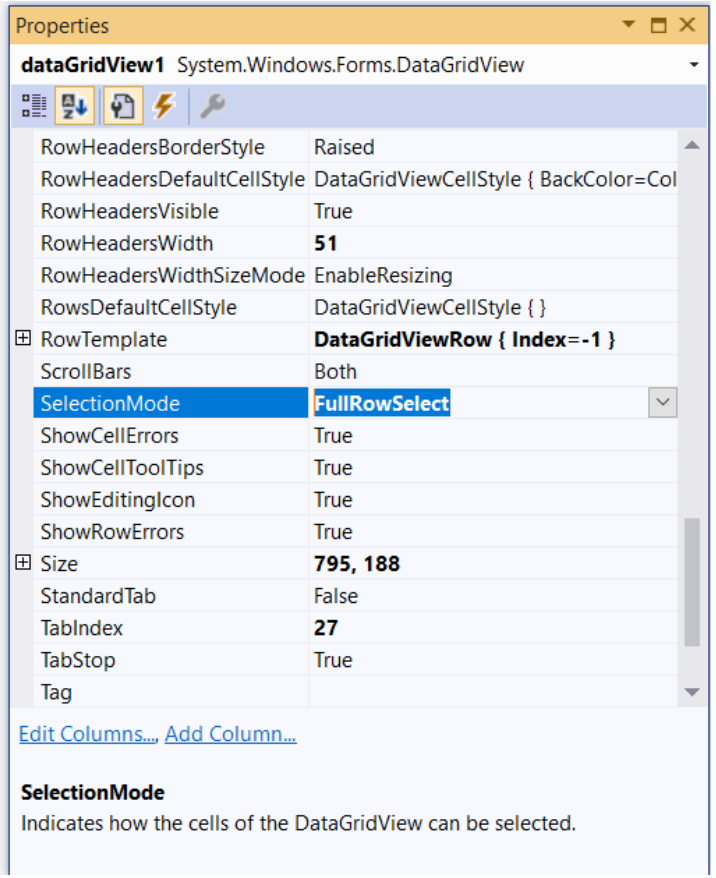
```
private void button1_Click(object sender, EventArgs e) //SİL Butonu
{
    String t1 = comboBox1.Text; //Seçilen satırın malzeme kodunu al. Bkz: dataGridView1_CellContentClick metodu

    baglanti.Open();
    SqlCommand komut = new SqlCommand("DELETE From Malzemeler WHERE MalzemeKodu=('"+t1+"')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();

    listele();
}
```

- DataGridView’a tıklandıktan sonra seçilen satır SİL butonuna tıklanarak silinir.
- ÖNEMLİ: Silme İşlemi malzeme kodu baz alınarak yapılır. Aynı malzeme koduna sahip birden fazla kayıt varsa hepsi veritabanından silinir.

# Veritabanında Kayıt Güncelleme

- Kayıt güncelleme için öncelikle güncellenecek olan kaydın seçilmesi gerekmektedir. Bu işlem DataGridView'da tıklama yoluyla yapılabilir. DataGridView'da bir hücreye tıklandığında, o satırın tamamen seçilebilmesi için **SelectionMode** özelliği **FullRowSelect** olarak ayarlanmalıdır.



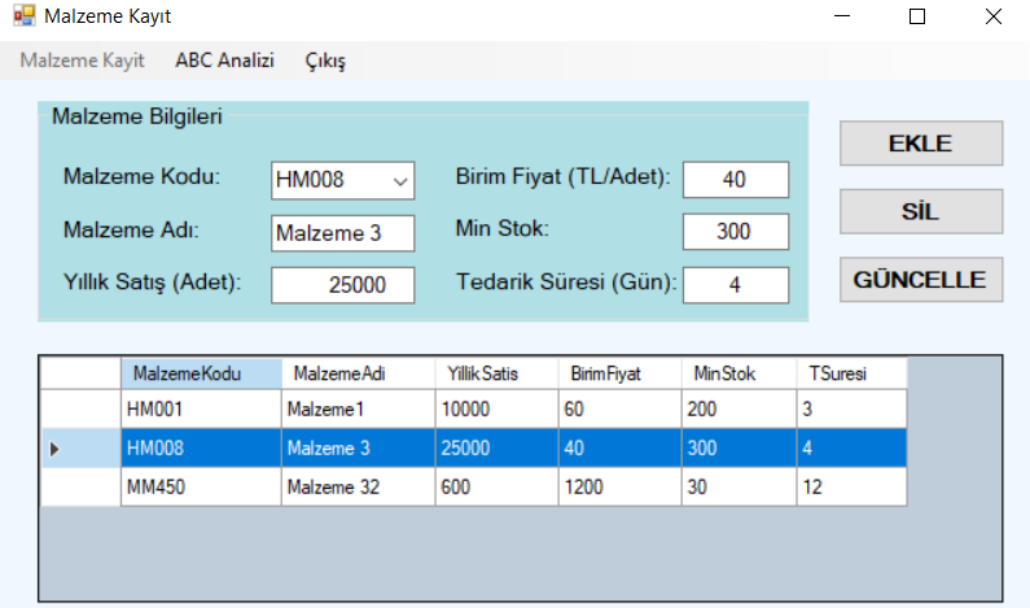




**DataGridView'da Satırın Bir Bütün Olarak Seçilmesi**

# Veritabanında Kayıt Güncelleme

- Ardından, DataGridView'da tıklama yoluyla seçilen bir satırın hücrelerindeki verinin comboBox ve textBox'lara aktarılması aşağıdaki komutlarla yapılır.

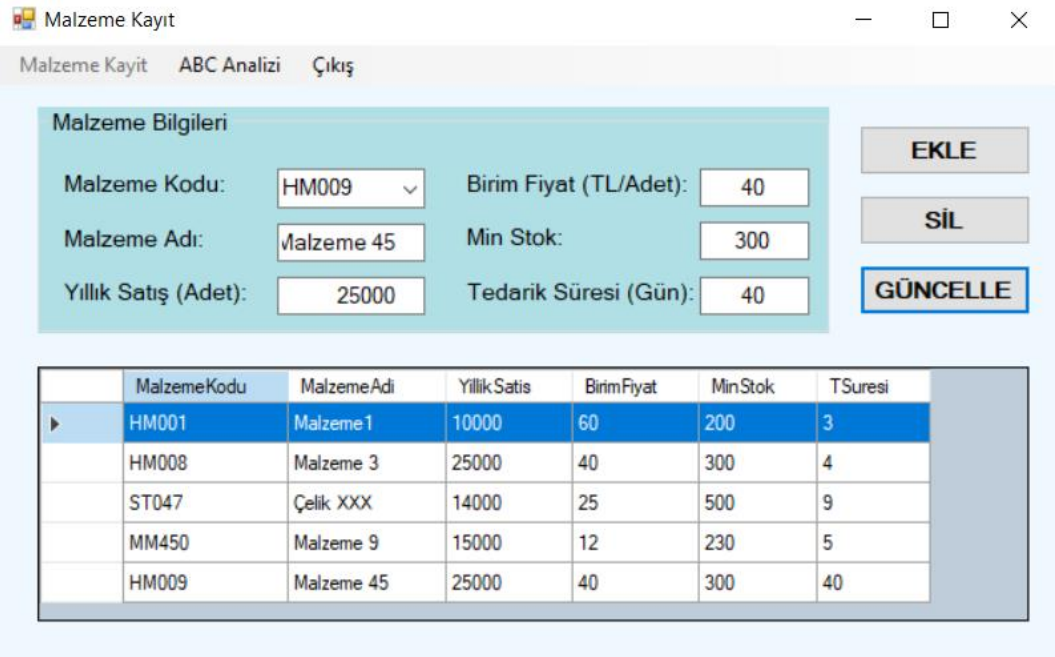
```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    comboBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox4.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    textBox5.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textBox6.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
}
```




- ÖNEMLİ: Kodlar DataGridView'ın Click olayına yazılır.

# Veritabanında Kayıt Güncelleme

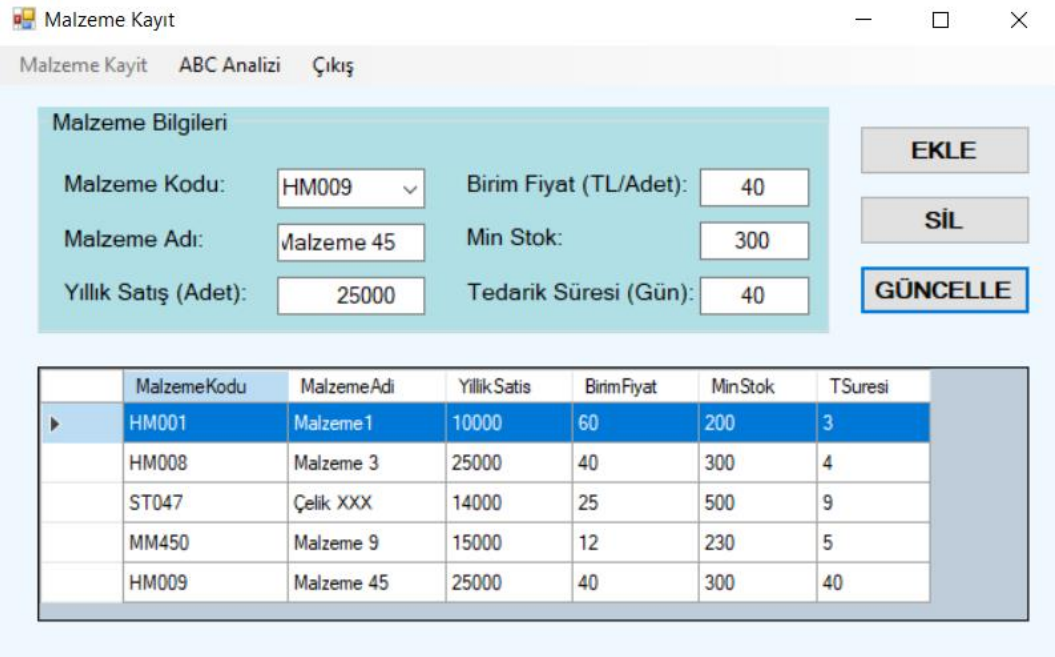
```
private void btnGuncelle_Click(object sender, EventArgs e) // GÜNCELLE Butonu
{
    String t1 = comboBox1.Text;
    String t2 = textBox2.Text;
    String t3 = textBox3.Text;
    String t4 = textBox4.Text;
    String t5 = textBox5.Text;
    String t6 = textBox6.Text;

    baglanti.Open();
    SqlCommand komut = new SqlCommand("UPDATE Malzemeler SET MalzemeKodu='" + t1 + "', MalzemeAdi='" + t2 + "', " +
        "YillikSatis='" + t3 + "', BirimFiyat='" + t4 + "', MinStok='" + t5 + "', Tsuresi='" + t6 + "' " +
        "WHERE MalzemeKodu='" + t1 + "' ", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    listele();
}
```



# Veritabanında Kayıt Güncelleme

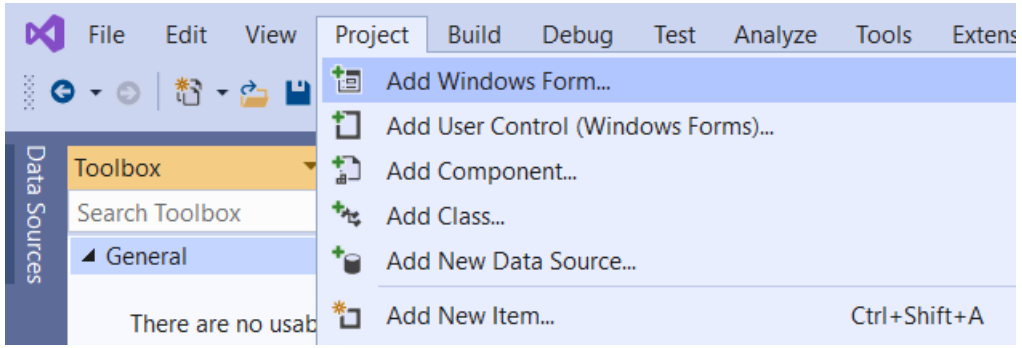
```
private void btnGuncelle_Click(object sender, EventArgs e) // GÜNCELLE Butonu
{
    String t1 = comboBox1.Text;
    String t2 = textBox2.Text;
    String t3 = textBox3.Text;
    String t4 = textBox4.Text;
    String t5 = textBox5.Text;
    String t6 = textBox6.Text;

    baglanti.Open();
    SqlCommand komut = new SqlCommand("UPDATE Malzemeler SET MalzemeKodu='" + t1 + "', MalzemeAdi='" + t2 + "', " +
        "YillikSatis='" + t3 + "', BirimFiyat='" + t4 + "', MinStok='" + t5 + "', Tsuresi='" + t6 + "' " +
        "WHERE MalzemeKodu='" + t1 + "' ", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    listele();
}
```

- Bu komut ile, seçilen bir satırdaki değerler yukarı aktarıldıktan sonra değişiklik yapılıp GÜNCELLE butonuna tıklanır. MalzemeKodu değeri baz alınarak veritabanında değişiklik yapılır.

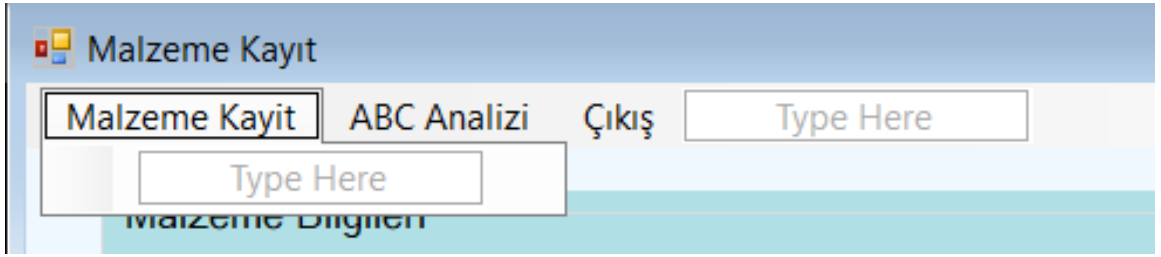


# Form2-Pareto Analizi

- Şimdi projemize ABC Analizi yapacak bir form ekleyelim. Yeni formumuzu **Project-> Add Windows Form** yolunu izleyerek ekleyebiliriz.




- Formlar arasında geçiş yapmak için de Menü yapısı kullanılabilir.  Bu amaçla **MenuStrip** kullanılarak formumuzda üst menü oluşturulabilir.




- Oluşturulan menustrip elemanlarına alt elemanlar eklenebilir ve formlar arasında geçiş yapmak için her birinin Click olayına komutlar yazılabilir.

# Formlar Arasında Geçiş

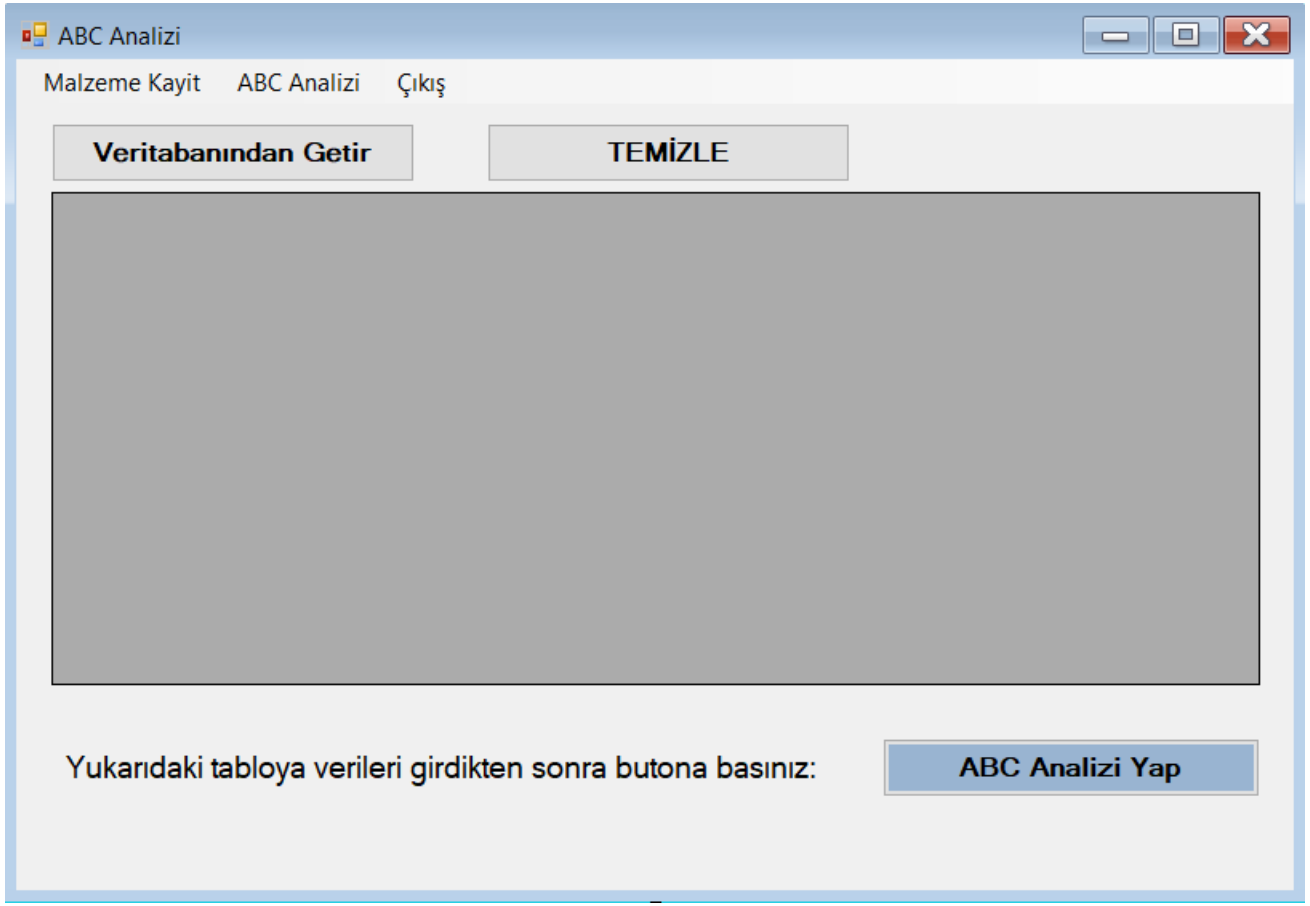
- Form1'de üst menüden ABC Analizi sekmesine tıklanınca Form2 gösterilecektir. Bu amaçla kullanılabilecek kodlar aşağıdadır.

```
private void MenuItem2_Click(object sender, EventArgs e)
{
    Form2 ABC = new Form2();
    ABC.Show();
    this.Hide();
}
```

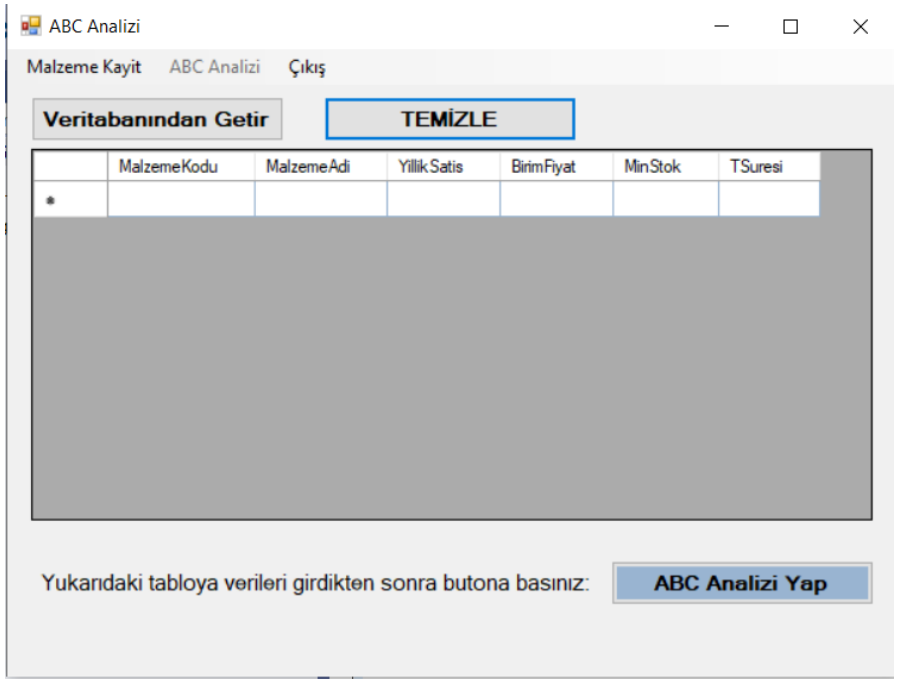
- Aynı menü yapısı Form2'de de oluşturulur. Bu defa da Malzeme Kayıt sekmesine tıklanınca aşağıdaki kodlar yardımıyla Form1 gösterilecektir.

```
private void MenuItem1_Click(object sender, EventArgs e)
{
    Form1 MalzemeKayit = new Form1();
    MalzemeKayit.Show();
    this.Hide();
}
```

İstenmeyen menü elemanlarının .Enabled özelliği False yapılabilir.




- Form2'nin tasarımı yandadır. Bu formda ABC analizi yapılmak istenen malzemeler Veritabanından Getirilebilir.
- Ayrıca manuel düzenlemeye de izin verilmektedir. Bunun için önce veritabanından veriler getirilmeli, ardından TEMİZLE butonuna basılmalıdır.

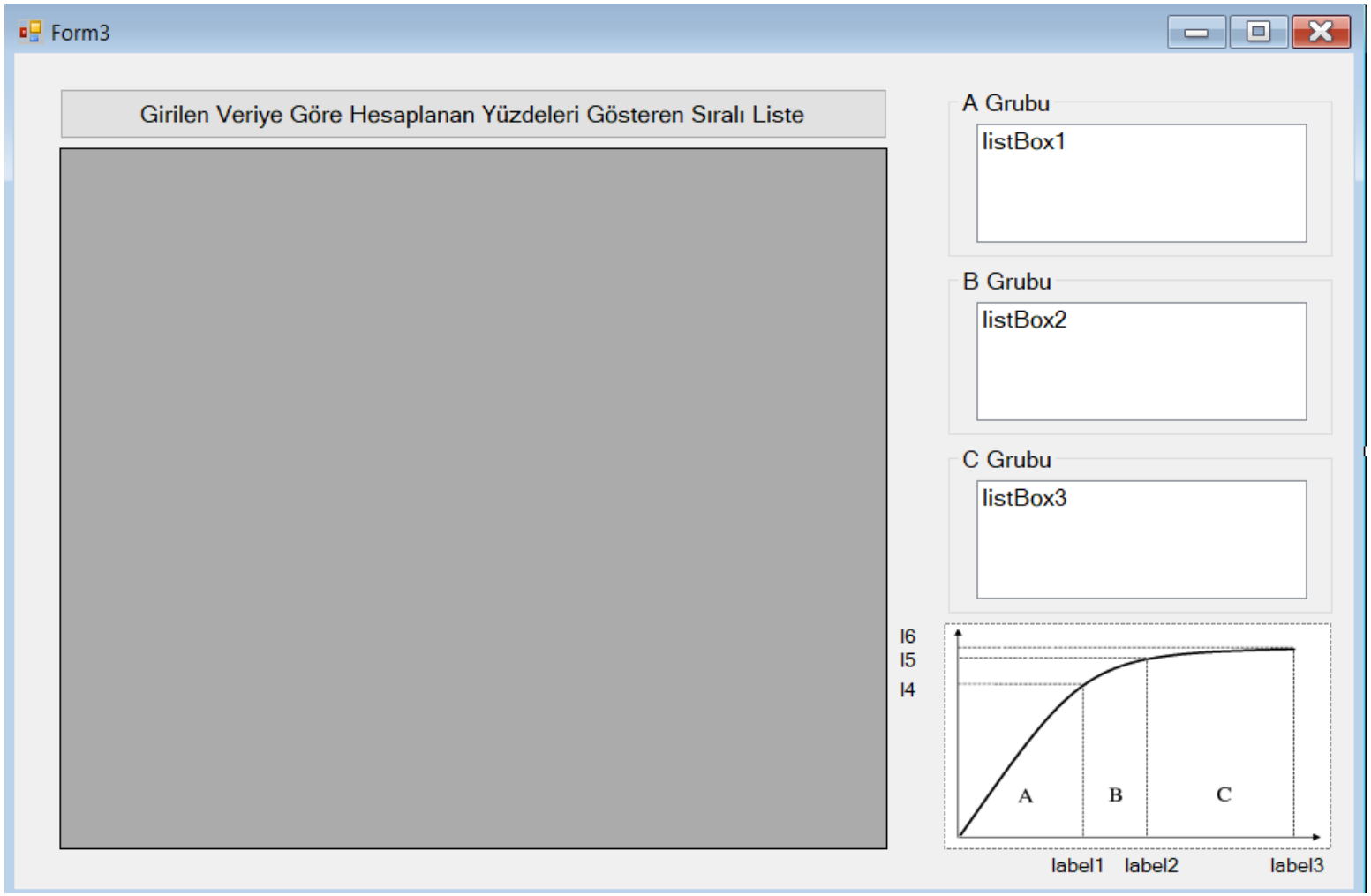


```
SqlConnection baglanti = new SqlConnection("Data Source=LENOVO-IKUCUKKO;Initial Catalog=ABC;Integrated Security=True");
SqlDataAdapter da;
DataTable tablo;
public void listele()
{
    baglanti.Open();
    da = new SqlDataAdapter("select *FROM Malzemeler", baglanti);
    tablo = new DataTable();
    da.Fill(tablo);
    dataGridView1.DataSource = tablo;
    baglanti.Close();
}


private void Form2_Load(object sender, EventArgs e)
{
    MenuItem1.Enabled = true;
    MenuItem2.Enabled = false;
}

private void MenuItem1_Click(object sender, EventArgs e)
{
    Form1 MalzemeKayit = new Form1();
    MalzemeKayit.Show();
    this.Hide();
}
```

```
private void button2_Click(object sender, EventArgs e) //ABC Analizi Yap Butonu
{
    int n = dataGridView1.Rows.Count-1;
    String[,] dizi1 = new String[n, 3];

    for (int i = 0; i < n; i++)
    {
        dizi1[i,0] = dataGridView1.Rows[i].Cells[0].Value.ToString();//MalzemeKodu
        dizi1[i,1] = dataGridView1.Rows[i].Cells[2].Value.ToString();//YillikSatis
        dizi1[i,2] = dataGridView1.Rows[i].Cells[3].Value.ToString();//BirimFiyat
    }

    Form3 ABC = new Form3(dizi1);
    ABC.Show();
    //this.Hide();
}
```

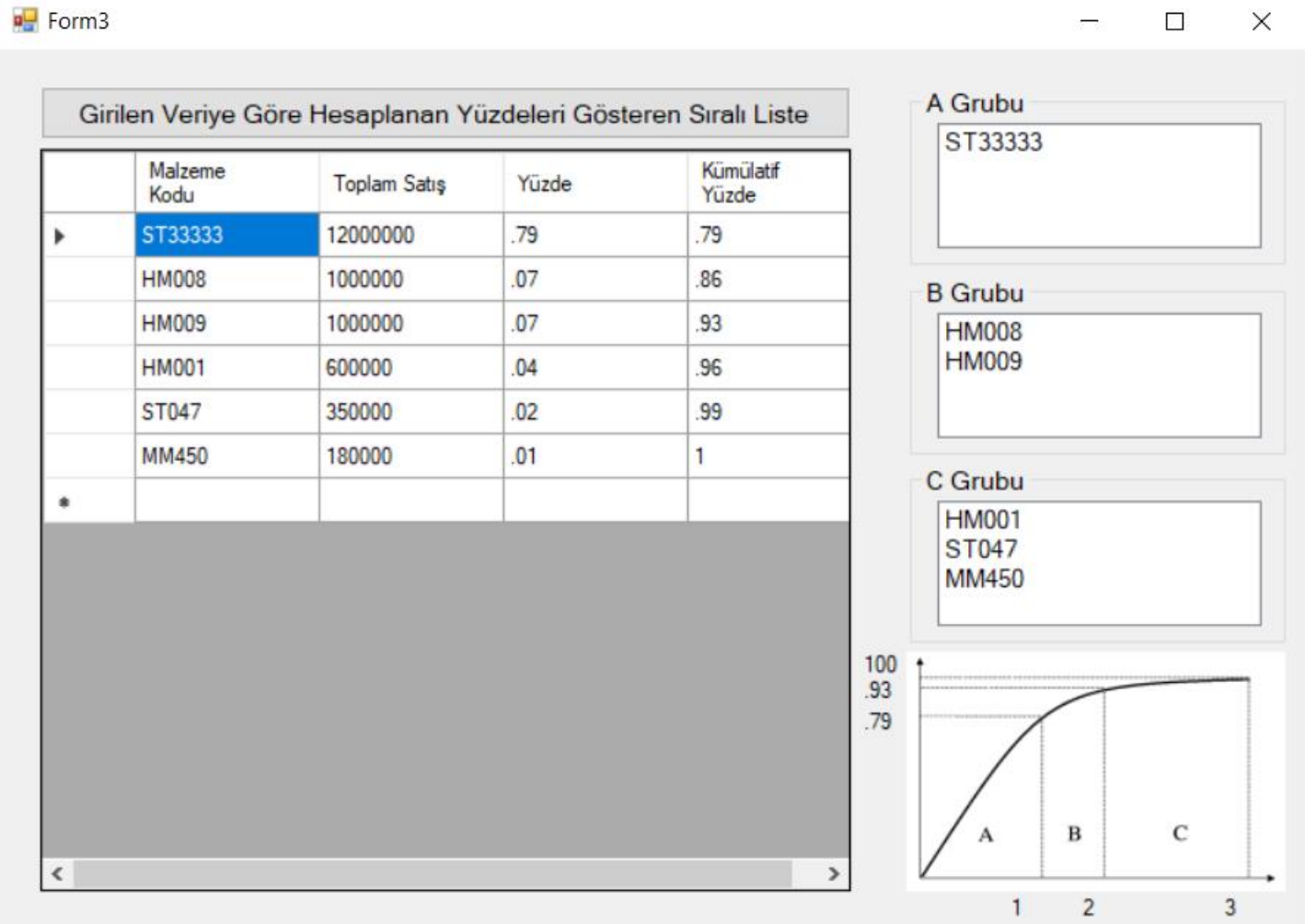
```
private void MenuItem3_Click(object sender, EventArgs e)//Çıkış Menü
{
    Application.Exit();
}

private void button1_Click(object sender, EventArgs e)//TEMİZLE Butonu
{
    tablo.Rows.Clear();
    dataGridView1.Refresh();
}

private void btnVt_Click(object sender, EventArgs e) //Veritabanından Getir
{
    listele();
}

private void Form2_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
```

# Form3 – Sonuç Ekranı




- Sonuçların gösterilmesi yeni bir form üzerinden yapılacaktır. Bir önceki formda **ABC Analizi Yap** butonuna basılması ile Form3 gösterilir.
- Ayrıca Form3'te kullanılmak üzere, ABC Analizi yapılacak malzeme bilgileri de dizi1 ismiyle Form3'e gönderilmektedir.

- ABC Analizi hesaplamaları Form3'ün Load olayına yazılmıştır.
- Form3 yüklenirken malzemelerinYillikSatis ve BirimFiyat değerleri çarpılarak Toplam Satış değerleri hesaplanır. Malzemeler Toplam Satış değerine göre azalan şekilde sıralanır. Yüzde ve Kümülatif Yüzde değerleri hesaplanarak %80 ve %95 değerlerinden A, B ve C gruplarına ayrılırlar.



```
string[,] veri;
public Form3(string[,] dizi1)
{
    veri = dizi1;
    InitializeComponent();
}

private void Form3_Load(object sender, EventArgs e)
{
    int n = veri.GetLength(0);
    string[] malzeme = new string[n];
    double[] toplamSatis = new double[n];
    double toplam = 0;

    for (int i = 0; i < n; i++)
    {
        malzeme[i] = veri[i, 0];
        toplamSatis[i] = Convert.ToDouble(veri[i, 1]) * Convert.ToDouble(veri[i, 2]);
        toplam += toplamSatis[i];
    }
```

```
//Selection Sort (Büyükten Küçüğe Sıralama)
double yedek1;
string yedek2;
int maxIndex;
//double enb = -1;
for (int i = 0; i < n-1; i++)
{
    maxIndex = i;
    for (int j = i; j < n; j++)
    {
        if (toplamSatis[j] > toplamSatis[maxIndex])
        {
            maxIndex = j;
        }
    }
    yedek1 = toplamSatis[i];
    toplamSatis[i] = toplamSatis[maxIndex];
    toplamSatis[maxIndex] = yedek1;

    yedek2 = malzeme[i];
    malzeme[i] = malzeme[maxIndex];
    malzeme[maxIndex] = yedek2;
}
```

```
//DataGridView'da sıralı listeyi göster
dataGridView1.ColumnCount = 4;
dataGridView1.Columns[0].Name = "Malzeme Kodu";
dataGridView1.Columns[1].Name = "Toplam Satış";
dataGridView1.Columns[2].Name = "Yüzde";
dataGridView1.Columns[3].Name = "Kümülatif Yüzde";

double[] yuzde = new double[n];
double[] kum = new double[n];//kümülatif yüzde

//yüzdelerin hesaplanması
double baslangic = 0;
for (int i = 0; i < n; i++)
{
    yuzde[i] = toplamSatis[i]/toplam;
    kum[i] = yuzde[i] + baslangic;
    baslangic = kum[i];
    if (kum[i] <= 0.80)
    {
        listBox1.Items.Add(malzeme[i]);
    } else if (kum[i] <= 0.95)
    {
        listBox2.Items.Add(malzeme[i]);
    } else
    {
        listBox3.Items.Add(malzeme[i]);
    }
}
```

```
    //dataGridView1'de göster
    for (int i = 0; i < n; i++)
    {
        dataGridView1.Rows.Add(malzeme[i], toplamSatis[i].ToString(), yuzde[i].ToString("#.##"),
            kum[i].ToString("#.##"));
    }

    //grafikte göster
    label1.Text = listBox1.Items.Count.ToString();
    label2.Text = listBox2.Items.Count.ToString();
    label3.Text = listBox3.Items.Count.ToString();

    double level1 = 0;
    double level2 = 0;
    for (int i = 0; i < n; i++)
    {
        if (kum[i] > level1 && kum[i]<=0.8)
        {
            level1 = kum[i];
        } else if (kum[i] > level2 && kum[i] <= 0.95)
        {
            level2 = kum[i];
        }
    }
    label4.Text = level1.ToString("#.##");
    label5.Text = level2.ToString("#.##");
    label6.Text = 100.ToString("#.##");
}
```

# Kaynakça


- C# Eğitim Kitabı 2. Baskı, Murat Yücedağ, Dikeyeksen Yayınları, 2019.
- C# Ders Notları, YMT218 Nesne Tabanlı Programlama, Fırat Üniversitesi (Erişim Tarihi 29 Ekim 2019).
- C# Programlama Ders Notları, Dr. Serkan Dişlitaş, Hitit Üniversitesi (Erişim Tarihi 29 Ekim 2019).