

FORM EKRANI-3

FORM EKRANI

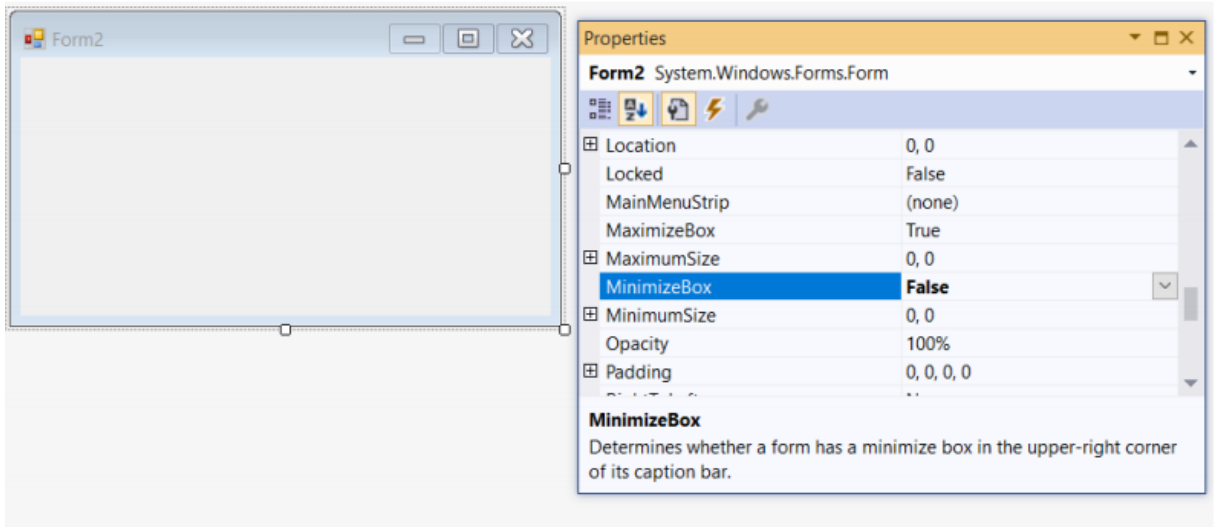
- ✓ *Form Kontrolleri*

KAZANIMLAR

- ✓ *Form kontrol geiři*
- ✓ *Veri aktarma*



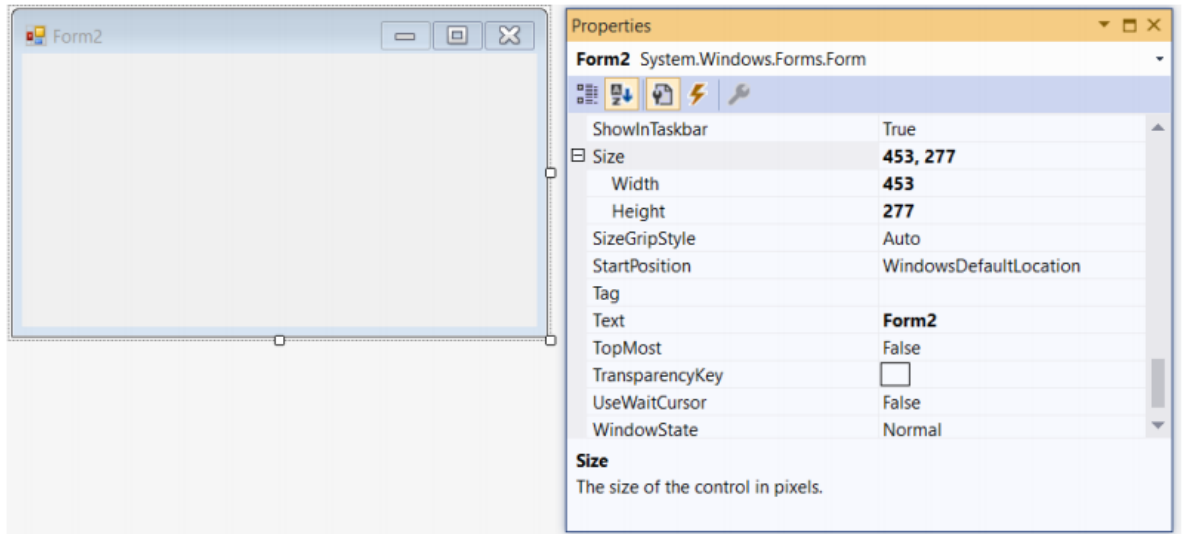
Form Özellikleri – MinimizeBox, MaximizeBox



MinimizeBox Özelliği: Simge Durumuna Küçültme

MaximizeBox Özelliği: Ekranı Kaplama MinimizeBox =false,
MaximizeBox=true

Formun boyutları: Width (yatay boyutu) , Height (yüksekliği)



Formun Ekrandaki Konumu (Location):

```
private void Form1_Load(object sender, System.EventArgs e)
{
    this.Location=new Point (150, 250);
}
```

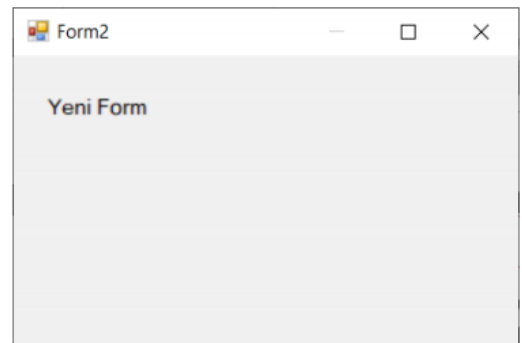
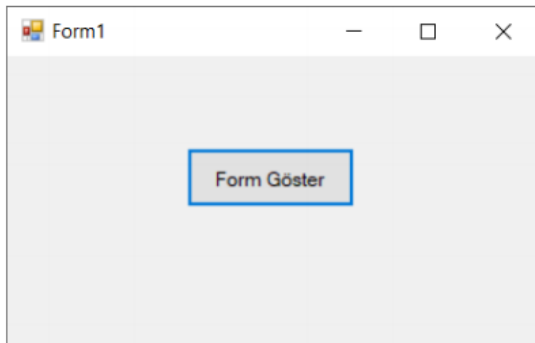
Form Özellikleri

- ✓ **BackgroundImage:** Form arka planına resim ekleme
- ✓ **Opacity:** Formun şeffaflığını ayarlama

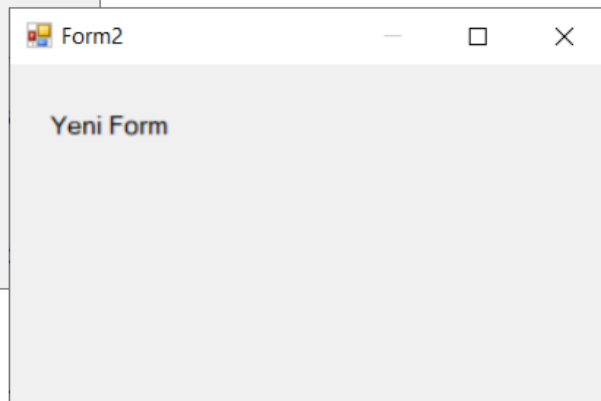
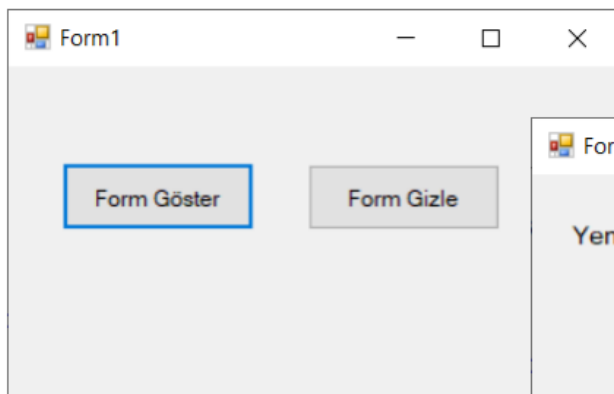


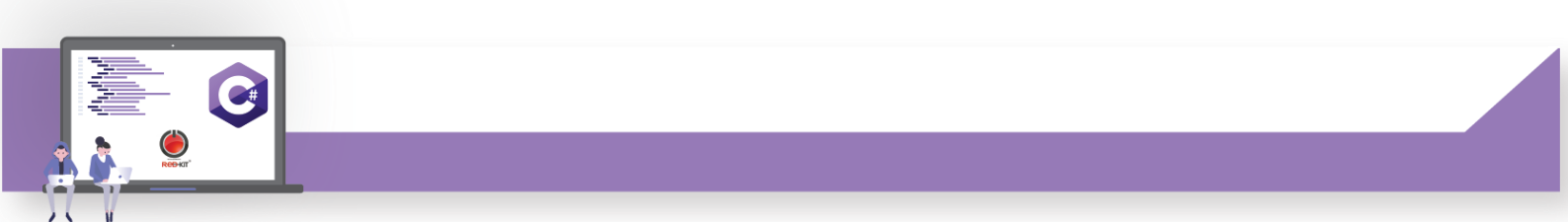
- ✓ *Show metodu birden fazla formu aynı anda ekrana getirmek için, ShowDialog ise sadece çalışmak istediğimiz formu ekrana getirmek için kullanılır. Bu form dışındakilere erişilemez.*

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 yeniForm = new Form2();
    this.Hide();
    yeniForm.Show();
}
```



```
Form2 yeniForm = new Form2();
private void button1_Click(object sender, EventArgs e)
{
    yeniForm.Visible = true;
}
private void button2_Click(object sender, EventArgs e)
{
    yeniForm.Visible = false;
}
}
```





✓ İptal (CancelButton) düğmesi Esc ile aynı işleve sahiptir. Forma yerleştirmiş olduğunuz düğmeye benzer işlevi vermek istiyorsanız formun özelliğine söz konusu düğmenin adını aktarmalısınız.

✓ Tamam (AcceptButton) düğmesi Enter ile tuşu aynı işleve sahiptir. Herhangi bir anda Enter tuşuna basıldığında formdaki düğmelerden birisi tıklanmış gibi işlem yapılmasını istiyorsanız o düğmenin adını formun AcceptButton özelliğine aktarmalısınız.

Form Özellikler

AcceptButton	Form üzerinde Enter tuşuna basıldığı zaman "tıklanacak" Buton kontrolü
CancelButton	Form üzerinde Esc tuşuna basıldığı zaman "tıklanacak " Button kontrolü
Opacity	Formun şeffaflık oranı(0-1arası)
MaximizeBox	Ekranı Kapla düğmesinin görünürlüğü
MinimizeBox	Simge Durumunda Küçült düğmesinin görünürlüğü
ControlBox	Close,Maximize ve Minimize düğmelerinin tümünün görünürlüğü
StartPosition	Form açıldığı zaman,ekran üzerindeki konumu
TopMost	Formun tüm pencerelerinin üzerinde gözükmesi
MinimumSize	Formun alabilceği minimum büyüklük



Form Olayları

Olay	Açıklama
Click	Form üzerine tıklandığı zaman gerçekleşir
Closing	Form kapanmadan hemen önce gerçekleşir
Closed	Form kapandıktan sonra gerçekleşir
Load	Form yüklenirken gerçekleşir
KeyDown	Form üzerindeyken bir tuşun basılması ile gerçekleşir
KeyUp	Basılan tuşun kaldırılması ile gerçekleşir.

Form Metodları

Metot	Açıklama
Hide	Formu Visible özelliğini False yaparak,gizler
Close	Formu kapatır.Eğer form başlangıç formuysa uygulama sonlanır
Show	Formu gösterir.Hide ile gizlenmişse,Visible özelliği True ile yapılır.
ShowDialog	Formu diyalog kutusu olarak gösterir.

Örnek: Formlar arası geçiş yaparak, kullanıcı tarafından girilen değerlerin listbox a aktarılması.

Form1

Ad

Şifre

Cıkıs Giriş

Form2

Kişisel Bilgiler

Ad

Soyad

Eğitim

☐ red-Kıt C

☐ mBlock

☐ wood-Kıt C

☐ wood-Kıt mBlock

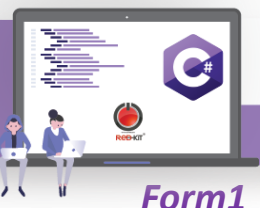
Cinsiyet

☐ Kız

☐ Erkek

listBox1

Yazdır



Form1

Form Ekranında Kullanılan Bileşenler

-2xLabel

-2xButon

-2xTextbox

-1xPictureBox

Textbox birinci kontrolümüz. **Textbox** kullanıcıdan veri almamızı ya da sisteme dışarıdan değer girmemizi sağlayan kontroldür. İkinci kontrolümüz **Label** dir. **Label** kullanıcıya form üzerinde bilgi vermek için kullanılır. Şimdi formumuza 2 adet **Textbox** ve 2 adet **Label** ve 2 adet **Button** ekleyelim.

Butonumuzun adını "**GİRİŞ**" koyduk. **Name** özelliğinede "**btn**" dedik. **Birinci label** ımızın adını "**Adı**", **ikinci label** ımızın adınıda "**Şifre**" olarak koyduk. Ve Cıkış Button umuza çift tıklayarak code behind tarafında geçiyoruz.

Programdan çıkma kod bloğumuzu yazıyoruz. Giriş Button umuza çift tıklayarak code behind tarafında geçiyoruz.

Kullanıcı adı ve şifresinin doğru yazıldığında diğer forma geçişi sağlıyoruz. Textbox lara yazılan değerleri sorguluyoruz .

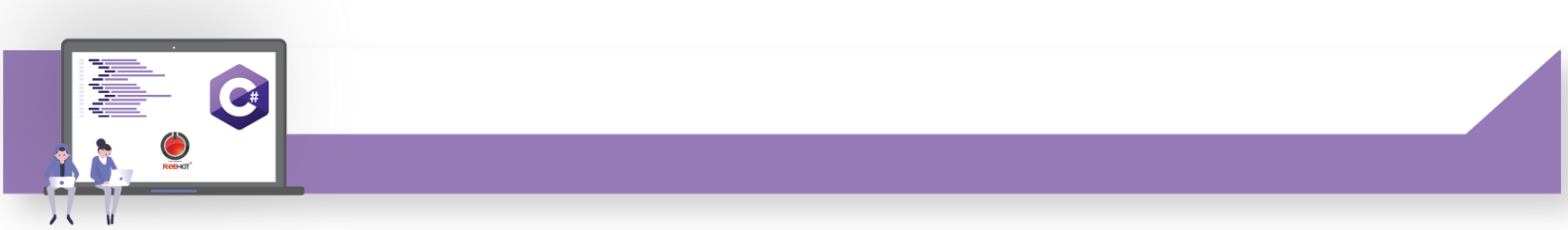
Kod Blokları:

```
private void button1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "Beyza" && textBox2.Text == "12ab")
    {
        MessageBox.Show("Hosgeldin");
        Form2 frm = new Form2();
        frm.Show();
        this.Hide();
    }
    else
        MessageBox.Show("Tekrar dene");
}
```

Form2

Form Ekranında Kullanılan Bileşenler

- ✓ 2xLabel
- ✓ 1xButon
- ✓ 2xTextbox
- ✓ 4xCheckbox
- ✓ 4xGroupBox
- ✓ 2xradioButon



Bölümlendirmek ve tasarım olarak güzel bir görünüş kazandırmak için **groupBox** kullanıyoruz. GroupBoxların text kısımlarını sırasıyla Kişisel Bilgiler, Eğitim, Cinsiyet ve Dördüncüye ise boş bırakıyoruz. İkinci kontrolümüz **Labelleri** ise; kullanıcıya form üzerinde bilgi vermek için kullanıyoruz. Sırasıyla text kısımlarını Adı ve Soyad olarak değiştiriyoruz. Butonumuzun adını "Yazdır" koyduk. Üçüncü Kontrolümüz **CheckBox** lar ise; red-Kit C, mBlock, wood-Kit C, wood-Kit mBlock olarak değiştiriyoruz.

Checkboxlar çoklu seçim işlemi için uygundur. Tek bir seçim yaptırmak istersek **radiobutton**ları kullanmalıyız. Öncelikle **string** olarak ders değişkenini tanımladık. Daha sonra if yapısı ile checkboxların seçili olduğu durumlarda ne yapması gerektiğini tanımlayarak seçim değişkenine aktarılmasını ve seçili olan seçenek ile birlikte gösterilmesini sağladık. Ekran çıktımız ise tekli ve çoklu seçim için listbox da görüntülendi. Ve son kontrolümüz **radioButonlar** ise, Kız, Erkek.

Form ekranından girilen kontrol değerlerini butona bastığımızda listbox a yazdırmasını istiyoruz. Bu yüzden bu kontrol değerlerini Yazdır Button umuza çift tıklayarak code behind tarafında geçiriyoruz.

Kod Blokları:

```
string ad = "", soyad = "";
ad = textBox1.Text;
soyad = textBox2.Text;

listBox1.Items.Add(ad);
listBox1.Items.Add(soyad);
string Cinsiyet=null, ders="";
if (radioButton1.Checked == true)
{
    Cinsiyet = "Kız";
}
if (radioButton2.Checked == true)
{
    Cinsiyet = "Erkek";
}
listBox1.Items.Add(Cinsiyet);
if (checkBox1.Checked == true)
{
    ders = ders + " " + "red-Kit C";
}
if (checkBox2.Checked == true)
{
    ders = ders + "mBlock";
}
if (checkBox3.Checked == true)
{
    ders = ders + "wood-Kit C";
}
if (checkBox4.Checked == true)
{
    ders = ders + "wood-Kit mBlock";
}
listBox1.Items.Add(ders);
}
```



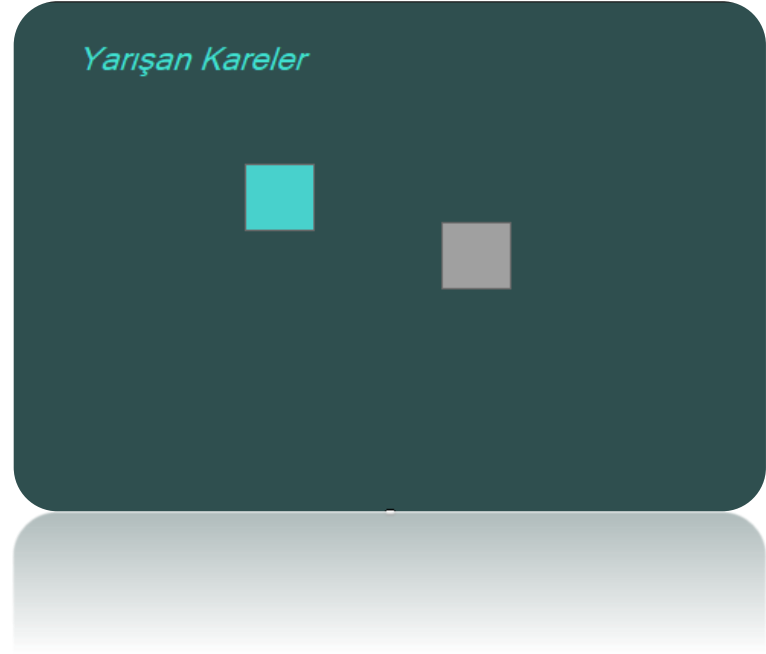
Örnek : Form ekranında oynucağımız oyundaki karelerimizin renklerini seçiyoruz ve açılan yenisayfada oyunumuzu oynuyoruz.

Oyunun Karakterlerini birlikte tasarlayalım

Karenin rengini sec

Cubuk rengini sec

Oyuna Başla



Form1

Form Ekranında Kullanılan Bileşenler

- 2xComboBox
- 5xButon
- 4xLabel
- 1xToolbar
- 2xTimer
- 2xPanel

ComboBox birinci kontrolümüz. **ComboBox** açılır menü bileşenidir. Görünüş olarak TextBox'a benzer ancak yanında küçük bir ok bulunur. Bu oka basıldığında içerisindeki elemanlar (item) seçilmek üzere listelenir. Birinci ComboBox **items** özelliğini;

Sarı

Kırmızı

Mor

Yeşil

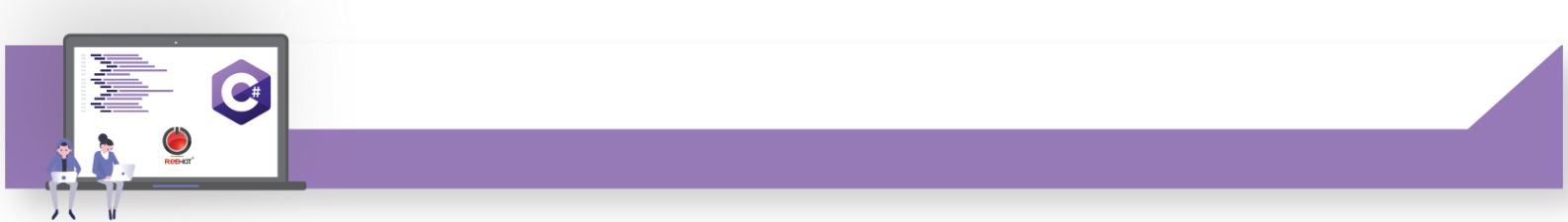
Olarak girelim. İkinci ComboBox ise;

Siyah

Beyaz

Turuncu

Yeşil



```
private void Form2_Load(object sender, EventArgs e)
{
    btn.Location = new Point(50, 60);
    btn2.Location = new Point(50, 130);
}
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (comboBox2.Text)
    {
        case "Kırmızı":
            btn2.BackColor = Color.Red;
            break;
        case "Sarı":
            btn2.BackColor = Color.Yellow;
            break;
        case "Mor":
            btn2.BackColor = Color.Purple;
            break;
        case "Yeşil":
            btn2.BackColor = Color.Green;
            break;
        default:
            MessageBox.Show("hatalı seçim");
            break;
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (btn.Left < btn2.Left)
        btn.Left++;
    if (btn.Left > btn2.Left)
        btn.Left--;
    if (btn.Top > btn2.Top)
        btn.Top--;
    if (btn.Top < btn2.Top)
        btn.Top++;
}
private void yon(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Up)
        btn2.Top -= 5;
    if (e.KeyCode == Keys.Down)
        btn2.Top += 5;
    if (e.KeyCode == Keys.Left)
        btn2.Left -= 5;
    if (e.KeyCode == Keys.Right)
        btn2.Left += 5;
}
private void button1_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
    btn.Location = new Point(40, 230);
    btn2.Location = new Point(130, 230);
}
}
```

