

44. ALGORİTMA GELİŞTİRME UYGULAMALARI

Aşağıdaki gibi Chat GPT nin yazdığı Programları Çalışır hale getirin ve her satırın yanına ne iş yaptığını yazın (GPT ye sorarsanız o herkesin aynı olur ve kopya alır, kendiniz yazın). Programların üzerine fazladan kendiniz birşeyler daha ekleyerek geliştirin. Çalışır ekran görüntülerini ve geliştirdiğiniz kodları ekleyin. Programı yapay Zeka yazdığı için mantıksız yerler varsa düzeltin.

Örnek 1:

```
private void ConvertButton_Click(object sender, EventArgs e)
{
    string metin = MetinTextBox.Text;
    string sessizMetin = SessizHarfleYaz(metin);
    SessizMetinLabel.Text = "Sesiz Harfle Yazılmış Metin: " + sessizMetin;
}

private string SessizHarfleYaz(string metin)
{
    char[] sesliHarfler = { 'a', 'e', 'i', 'o', 'ö', 'u', 'ü', 'A', 'E', 'I', 'O', 'Ö', 'U', 'Ü' };
    string sessizMetin = "";

    foreach (char karakter in metin)
    {
        if (Array.IndexOf(sesliHarfler, karakter) == -1)
        {
            sessizMetin += karakter;
        }
    }

    return sessizMetin;
}
```

Örnek 2:

```
private void CheckButton_Click(object sender, EventArgs e)
{
    double kenar1, kenar2, kenar3;
    if (double.TryParse(Kenar1TextBox.Text, out kenar1) && double.TryParse(Kenar2TextBox.Text, out kenar2) && double.TryParse(Kenar3TextBox.Text, out kenar3))
    {
        string ucgenTipi = BelirleUcgenTipi(kenar1, kenar2, kenar3);
        UcgenTipiLabel.Text = "Üçgen Türü: " + ucgenTipi;
    }
    else
    {
        MessageBox.Show("Geçerli kenar uzunlukları girin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private string BelirleUcgenTipi(double kenar1, double kenar2, double kenar3)
{
    if (kenar1 == kenar2 && kenar2 == kenar3)
    {
        return "Eşkenar Üçgen";
    }
    else if (kenar1 == kenar2 || kenar1 == kenar3 || kenar2 == kenar3)
    {
        return "İkizkenar Üçgen";
    }
    else
    {
        return "Çeşitkenar Üçgen";
    }
}
```

Örnek 2:

Örnek 3

```
private void HesaplaButton_Click(object sender, EventArgs e)
{
    if (DateTime.TryParse(GirisSaatTextBox.Text, out DateTime girisSaat) &&
        DateTime.TryParse(CikisSaatTextBox.Text, out DateTime cikisSaat))
    {
        double odeme = HesaplaOdeme(girisSaat, cikisSaat);
        OdemeLabel.Text = "Ödeme Tutarı: " + odeme.ToString("C"); // Para birimi formatı
    }
    else
    {
        MessageBox.Show("Geçerli saat formatı kullanın (örn. 12:30)", "Hata", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}

private double HesaplaOdeme(DateTime girisSaat, DateTime cikisSaat)
```

26

```
{
    TimeSpan kalisSure = cikisSaat - girisSaat;
    double saatlikUcret = 2.0;
    double ekUcret = 1.0;

    if (kalisSure.TotalHours <= 1)
    {
        return saatlikUcret;
    }
    else
    {
        double ekUcretTutari = Math.Ceiling(kalisSure.TotalHours - 1) * ekUcret;
        return saatlikUcret + ekUcretTutari;
    }
}
```

Örnek 1:

```
private void ConvertButton_Click(object sender, EventArgs e)
{
    string metin = MetinTextBox.Text; girilen metnin değışkene atılması
    string sessizMetin = SessizHarfleYaz(metin); metnin fonksiyona
gönderilmesi
    SessizMetinLabel.Text = "Sesiz Harfle Yazılmış Metin: " +
        sessizMetin;
}
private string SessizHarfleYaz(string metin)
{
    char[] sesliHarfler = { 'a', 'e', 'ı', 'i', 'o', 'ö', 'u', 'ü', 'A',
'E', 'I', 'İ', 'O', 'Ö', 'U', 'Ü' };
    string sessizMetin = "";
    foreach (char karakter in metin)
    {
```

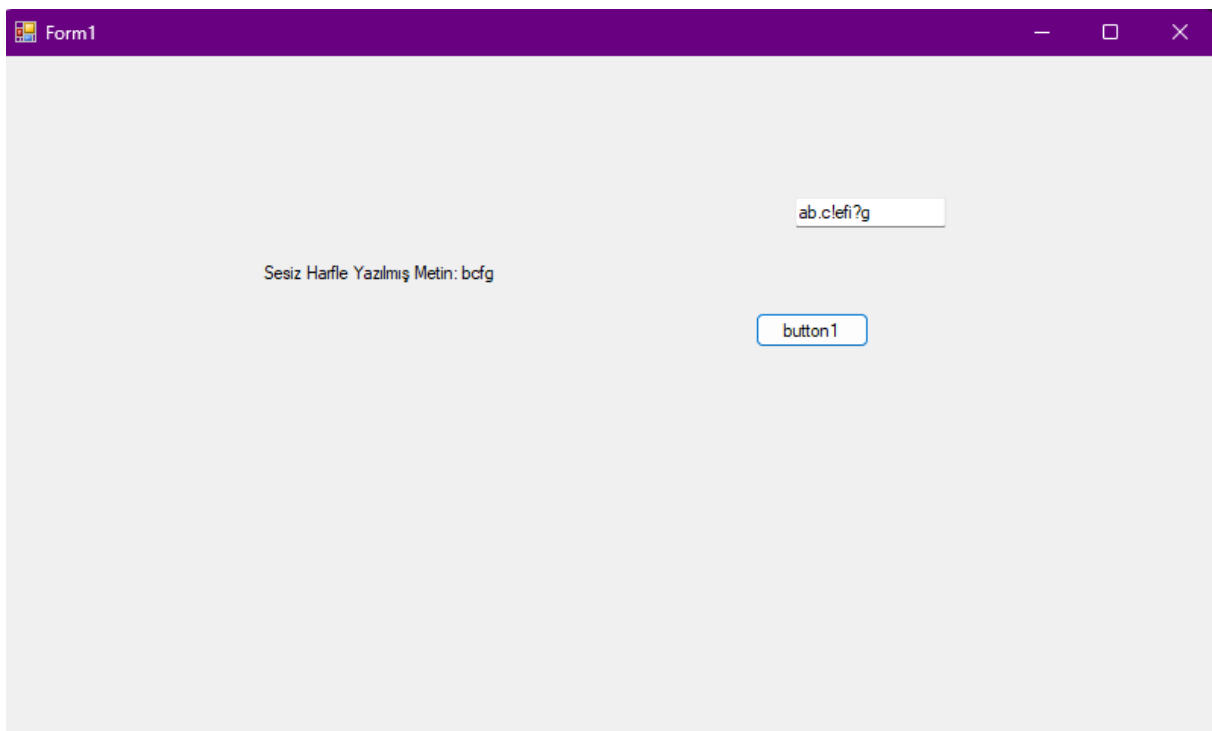
Noktalama işaretlerinin alınmaması için if'e ektradan bir koşul eklendi

```

        if (Array.IndexOf(sesliHarfler, karakter) == -1 &&
            char.IsLetterOrDigit(karakter) )
        {
            sessizMetin += karakter; incelenen char karakteri sessiz harf
ise bu sessizmetine ekleniyor
        }
        return sessizMetin;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        ConvertButton_Click( sender, e); butona tıklandığında metnin
değişmesini sağlayacak fonksiyon
    }
}

```



Örnek 2:

```

namespace berkay_2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

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

```

```

private void CheckButton_Click(object sender, EventArgs e)
{
    double kenar1, kenar2, kenar3; kenarlar için 3 tane değişken tanımlandı

    if (double.TryParse(Kenar1TextBox.Text, out kenar1) &&
        double.TryParse(Kenar2TextBox.Text, out
            kenar2) && double.TryParse(Kenar3TextBox.Text, out kenar3)) string
değerler double a dönüştürülüyor
    {
        string ucgenTipi = BelirleUcgenTipi(kenar1, kenar2,
kenar3);değerler ilgili fonksiyona gönderiliyor ve bir tane string döndürülüyor

        UcgenTipiLabel.Text = "Üçgen Türü: " + ucgenTipi;
        Ekрана üçgenin hangi tipte olduğunu gösterecek stringin labeli
değiştiriliyor.
    }
    else
    {
        MessageBox.Show("Geçerli kenar uzunlukları girin.", "Hata",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error); string değerler double a formatında
yazılmamışsa hata mesajı gösteriliyor.
    }
}
private string BelirleUcgenTipi(double kenar1, double kenar2, double
kenar3)
{

```

Kenar uzunluğu 0 veya negatif olamayacağından bunun kontrolünü sağlayan bir koşul eklendi.

```

    if (kenar1 <= 0 || kenar2 <= 0 || kenar3 <= 0)
    {
        MessageBox.Show("Kenar uzunluğu 0 veya negatif olamaz.", "Hata",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return "Hata!";
    }
    else
    {
        Burada gelen değerlere göre üçgenin tipi belirleniyor ve
string döndürülüyor.

        if (kenar1 == kenar2 && kenar2 == kenar3)
        {
            return "Eşkenar Üçgen";
        }
        else if (kenar1 == kenar2 || kenar1 == kenar3 || kenar2 ==
kenar3)
        {
            return "İkizkenar Üçgen";
        }
        else
        {
            return "Çeşitkenar Üçgen";
        }
    }
}

```

```

    }
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
}

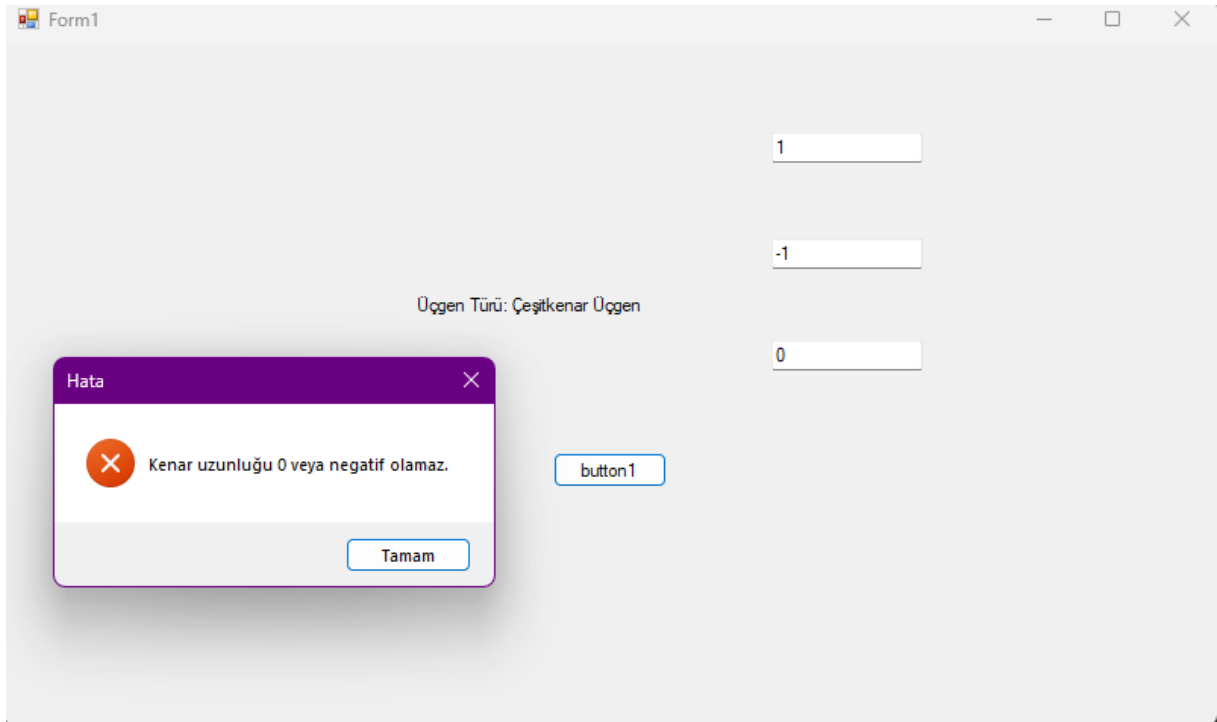
private void Kenar1TextBox_TextChanged(object sender, EventArgs e)
{
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    CheckButton_Click(sender, e); butona tıklandığında hesaplama yapan
fonksiyon
}
}

```

The screenshot shows a Windows application window titled "Form1". The window contains a user interface with three text input fields on the right, each containing a digit (1, 2, 3). In the center, a label displays the text "Üçgen Türü: Çeşitkenar Üçgen". At the bottom center, there is a button labeled "button1".



Örnek 3:

```
using System;
using System.Collections.Generic;

namespace berkay_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

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

        private void HesaplaButton_Click(object sender, EventArgs e)
        {
            Girilen değerler datetime objesine dönüştürülüyor.
            if (DateTime.TryParse(GirisSaatTextBox.Text, out DateTime
            girisSaat) &&
                DateTime.TryParse(CikisSaatTextBox.Text, out DateTime cikisSaat))
            {
                double odeme = HesaplaOdeme(girisSaat, cikisSaat); saat değerleri
                fonksiyona gönderilip ücret için bir double da saklanıyor.
                OdemeLabel.Text = "Ödeme Tutarı: " + odeme.ToString("C"); değer
                ekrananda gösterilmesi için label değiştiriliyor
            }
        }
    }
}
```

```

        else
        {
            Eğer yanlış formatta dönüştürme gerçekleşirse hata mesajı
            gösteriliyor.
            MessageBox.Show("Geçerli saat formatı kullanın (örn. 12:30)",
            "Hata", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
    private double HesaplaOdeme(DateTime girisSaat, DateTime cikisSaat)
    {
        Eğer giriş saati çıkış saatinden büyükse bunun
        kontrolünü sağlayan koşul eklendi.

        if(girisSaat.CompareTo(cikisSaat) > 0 )
        {
            MessageBox.Show("Giriş Saati Çıkış Saatinden Büyük
            Olamaz!", "Hata", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            return 0.0;
        }
        else {
            burada iki saatin arasındaki farka göre ücret hesaplanıyor
            TimeSpan kalisSure = cikisSaat - girisSaat;
            double saatlikUcret = 2.0;
            double ekUcret = 1.0;
            if (kalisSure.TotalHours <= 1)
            {
                return saatlikUcret;
            }
            else
            {
                double ekUcretTutari = Math.Ceiling(kalisSure.TotalHours - 1) *
                ekUcret;
                return saatlikUcret + ekUcretTutari;
            }
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        HesaplaButton_Click(sender, e); butona tıklandığında hesaplamanın
        gerçekleştirilmesini sağlayacak fonksiyon
    }

    private void GirisSaatTextBox_TextChanged(object sender, EventArgs e)
    {
    }

    private void CikisSaatTextBox_TextChanged(object sender, EventArgs e)
    {
    }

    private void label1_Click(object sender, EventArgs e)
    {
    }

    private void label2_Click(object sender, EventArgs e)

```

```
{  
}  
private void OdemeLabel_Click(object sender, EventArgs e)  
{  
}  
}  
}
```

The screenshot shows a Windows Form titled "Form1" with a light gray background. In the center, the text "Ödeme Tutarı: ₺7,00" is displayed. To the right, there are two text boxes: "Giriş Saati:" with the value "18:30" and "Çıkış Saati:" with the value "12:30". Below these, there is a button labeled "button1". In the bottom-left corner, a modal error dialog box is open. The dialog has a purple title bar with the text "Hata" and a close button. It contains an orange circular icon with a white 'X' and the text "Giriş Saati Çıkış Saatinden Büyük Olamaz!". At the bottom of the dialog is a button labeled "Tamam".

Form1

Giriş Saati: 12:30

Ödeme Tutarı: ₺7,00

Çıkış Saati: 18:30

button1