



Bilgisayar Programcılığı

Görsel Programlama-I

Öğr. Gör. Cansu AYVAZ GÜVEN

DIYALOG PENCERELERİ

- MessageBox
- File Dialog Pencereleeri
 - ColorDialog
 - FolderBrowserDialog
 - FontDialog
 - OpenFileDialog
 - SaveFileDialog

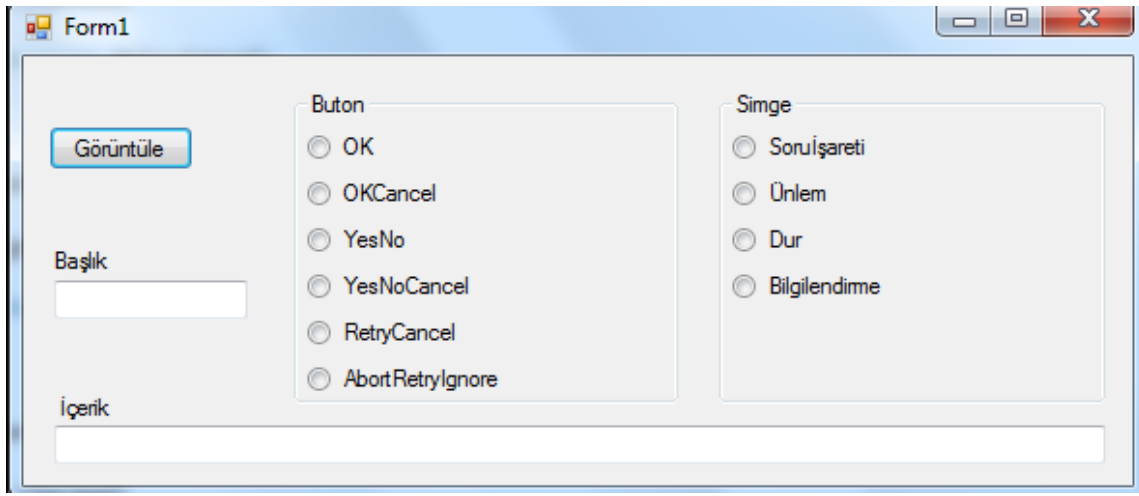
MessageBox: Kullanıcıya bilgi vermek için kullanılan mesaj kutusudur.

- **MessageBoxButtons.OK:** Mesaj kutusunda sadece **Tamam** düğmesi görüntülenir.
- **MessageBoxButtons.OKCancel:** Mesaj kutusu üzerinde **Tamam** ve **İptal** butonları görüntülenir.
- **MessageBoxButtons.YesNo:** Mesaj kutusu üzerinde **Evet** ve **Hayır** butonları görüntülenir.
- **MessageBoxButtons.YesNoCancel:** Mesaj kutusu üzerinde **Evet**, **Hayır** ve **İptal** butonları görüntülenir.
- **MessageBoxButtons.RetryCancel:** Mesaj kutusu üzerinde **Yeniden dene** ve **İptal** butonları görüntülenir.
- **MessageBoxButtons.AbortRetryIgnore:** Mesaj kutusu üzerinde **Durdur**, **Yeniden dene** ve **Yoksay** butonları görüntülenir.

Icon(Simge): MessageBox üzerindeki simgeyi ve MessageBox açıldığında oluşacak sesi belirtir.

- **MessageBoxIcon.Question:** Soru İşareti
- **MessageBoxIcon.Warning:** Ünlem İşareti
- **MessageBoxIcon.Exclamation:** Ünlem İşareti
- **MessageBoxIcon.Stop:** Dur İşareti
- **MessageBoxIcon.Error:** Dur İşareti
- **MessageBoxIcon.Hand:** Dur İşareti
- **MessageBoxIcon.Information:** Bilgilendirme İşareti
- **MessageBoxIcon.Asterisk:** Bilgilendirme İşareti

Örnek:



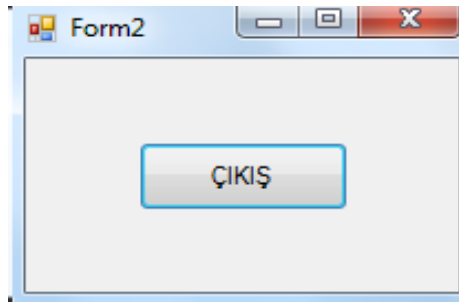
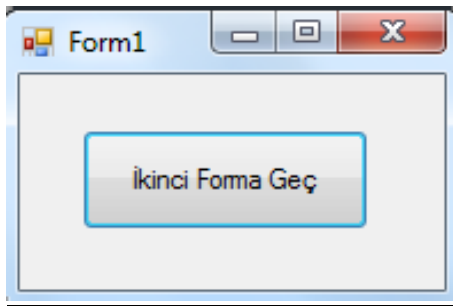
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace mBox_Dialog
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        MessageBoxButtons a = new MessageBoxButtons();
        MessageBoxIcon b = new MessageBoxIcon();
        private void rdbOK_CheckedChanged(object sender, EventArgs e)
        {
            if (rdbOK.Checked)
            {
                a = MessageBoxButtons.OK;
            }
        }
        private void rdbOKCancel_CheckedChanged(object sender, EventArgs e)
        {
            if (rdbOKCancel.Checked)
            {
                a = MessageBoxButtons.OKCancel;
            }
        }
        private void rdbYesNo_CheckedChanged(object sender, EventArgs e)
        {
            if (rdbYesNo.Checked)
            {
                a = MessageBoxButtons.YesNo;
            }
        }
    }
}
```

```

    }
    private void rdbYesNoCancel_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbYesNoCancel.Checked)
        {
            a = MessageBoxButtons.YesNoCancel;
        }
    }
    private void rdbRetryCancel_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbRetryCancel.Checked)
        {
            a = MessageBoxButtons.RetryCancel;
        }
    }
    private void rdbAbortRetryIgnore_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbAbortRetryIgnore.Checked)
        {
            a = MessageBoxButtons.AbortRetryIgnore;
        }
    }
    private void rdbSorulsareti_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbSorulsareti.Checked)
        {
            b = MessageBoxIcon.Question;
        }
    }
    private void rdbUnlem_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbUnlem.Checked)
        {
            b = MessageBoxIcon.Exclamation;
        }
    }
    private void rdbDur_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbDur.Checked)
        {
            b = MessageBoxIcon.Stop;
        }
    }
    private void rdbBilgilendirme_CheckedChanged(object sender, EventArgs e)
    {
        if (rdbBilgilendirme.Checked)
        {
            b = MessageBoxIcon.Information;
        }
    }
    private void button1_Click(object sender, EventArgs e)
    {
        MessageBox.Show(textBox2.Text, textBox1.Text, a, b);
    }
}

```

ÖRNEK-2:

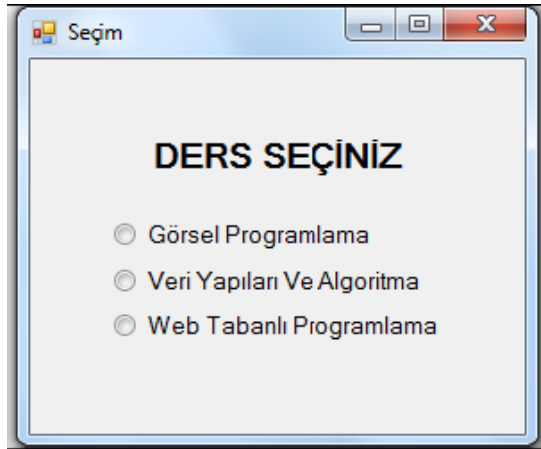


```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace mBox_Dialog2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            DialogResult dugme = new DialogResult();
            dugme = MessageBox.Show("İkinci forma geçmek istediğinizden emin misiniz?", "Devam",
            MessageBoxButtons.OKCancel, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
            if (dugme==DialogResult.OK)
            {
                this.Hide();
                Form frm = new Form2();
                frm.Show();
            }
        }
    }
}
```

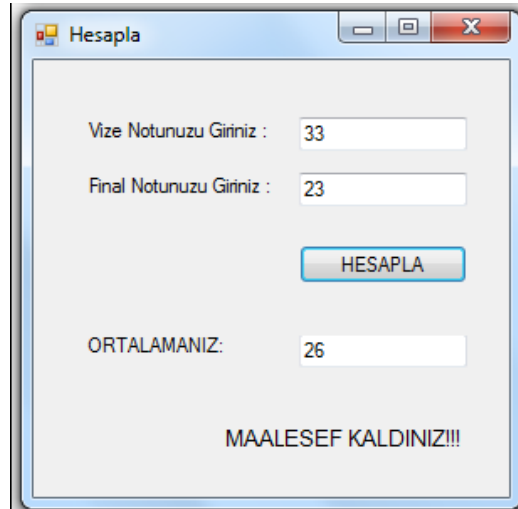
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace mBox_Dialog2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            DialogResult dugme = new DialogResult();
            dugme = MessageBox.Show("Çıkmak istediğinizden emin misini?", "Çıkış", MessageBoxButtons.OKCancel,
            MessageBoxIcon.Question);
            if (dugme==DialogResult.OK)
            {
                Application.Exit();
            }
        }
    }
}
```

ÖRNEK-3:

(Form1)



(Form2)



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace ortalamaHesapla
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        Hesapla hs = new Hesapla(); //Global alanda yeni formumuzu tanımladık.
        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            this.Hide(); //1. radioButton seçili ise form 1'i gizle
            hs.Show(); // 2. Formu aç
        }
        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            this.Hide(); //2. radioButton seçili ise form 1'i gizle
            hs.Show(); // 2. Formu aç
        }
        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {
            this.Hide(); //3. radioButton seçili ise form 1'i gizle
            hs.Show(); // 2. Formu aç
        }
    }
}
```



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace ortalamaHesapla
{
    public partial class Hesapla : Form //Form 2
    {
        public Hesapla()
        {
            InitializeComponent();
        }
        private void Hesapla_Load(object sender, EventArgs e)
        {
            textBox3.Visible = false;
        }
        private void btnHesapla_Click(object sender, EventArgs e)
        {
            double ort = Convert.ToDouble(textBox1.Text) * 0.3 + Convert.ToDouble(textBox2.Text) * 0.7;
            if (ort >= 60)
            {
                label4.Text = "TEBRİKLER GEÇTİNİZ...";
            }
            else
            {
                label4.Text = "MAALESEF KALDINIZ!!!";
            }
            textBox3.Visible = true;
            label3.Text = "ORTALAMANIZ: ";
            textBox3.Text = ort.ToString();
        }
    }
}
```

File Dialog Pencereleeri

Dosya Ama Diyalog Kutusu (OpenFileDialog)

OpenFileDialog, herhangi bir windows uygulamasında File → Open veya Dosya → A komutlarında ekrana gelen Dosya Ama arayüzünü uygulamalarda kullanmamızı saėlayan bir kontroldür. Toolbox üzerinden seçtiėimiz OpenFileDialog kontrolü, form üstüne bırakılsa bile, formun altında görüntülenecektir. **Toolbox**'da **Dialogs** bölümünde yer almaktadır.

Properties (Özellikler)	
AddExtension	Kullanıcı ihmal etse bile, dosya adına bir uzantının otomatik olarak eklenip eklenmeyeceėini belirler.
CheckFileExists	Kullanıcının belirlediėi dosyanın bulunmaması durumunda, bir uyarı mesajının gösterilip gösterilmeyeceėini belirler.
CheckPathExists	Kullanıcının belirlediėi yolun bulunmaması durumunda, bir uyarı mesajının gösterilip gösterilmeyeceėini belirler.
DefaultExt	Varsayılan dosya uzantısını belirler.
FileName	Diyalog kutusunda seçilen dosya ismini bir string olarak alır (saklar).
Filter	"Files of Type" "Dosya Tipi" için seçenek belirler. Buraya yazacaėımız seçeneklerdeki dosya türleri listelenir. Örneėin doc yazılırsa, yalnızca Word dosyaları listelenecektir.
InitialDirectory	Diyalog kutusu tarafından başlangıta görüntülenecek klasörü belirler.
Title	Diyalog Kutusu Başlıėı

Kaydetme Diyalog Kutusu (SaveFileDialog)

SaveFileDialog, herhangi bir windows uygulamasında File → Save veya Dosya → Kaydet komutlarında ekrana gelen Dosya Kaydetme arayüzünü uygulamalarda kullanmamızı saėlayan bir kontroldür. Toolbox üzerinden seçtiėimiz SaveFileDialog kontrolü, form üstüne bırakılsa bile, OpenFileDialog gibi formun altında görüntülenecektir.

Properties (Özellikler)	
AddExtension	Kullanıcı ihmal etse bile, dosya adına bir uzantının otomatik olarak eklenip eklenmeyeceėini belirler.
CheckFileExists	Kullanıcının kaydedeceėi dosyanın bulunması durumunda, bir uyarı mesajının gösterilip gösterilmeyeceėini belirler. Özellikle dosyanın aynı isimli bir başka dosya üzerine kaydedilmesini engelleme açısından faydalıdır.
CheckPathExists	Kullanıcının belirlediėi yolun bulunmaması durumunda, bir uyarı mesajının gösterilip gösterilmeyeceėini belirler.
CreatePrompt	Kullanıcının belirlediėi dosyanın oluşturulması izni için diyalog kutusunun gösterilip gösterilmeyeceėini belirler.
DefaultExt	Varsayılan dosya uzantısını belirler.
FileName	Diyalog kutusunda belirlenen dosya ismini bir string olarak alır (saklar).
Filter	"Files of Type" "Dosya Tipi" için seçenek belirler.
InitialDirectory	Diyalog kutusu tarafından başlangıta görüntülenecek klasörü belirler.
OverwritePrompt	Mevcut bir dosya üzerine yazma durumunda uyarı diyalog kutusunun çıkıp çıkmayacaėını belirler.
Title	Diyalog Kutusu Başlıėı

Yazı Tipi Diyalog Kutusu (FontDialog)

FontDialog, geliştirdiğimiz uygulamalarda yazı tipini belirlememizi sağlayan bir kontroldür. Bu diyalog kontrolü de diğer kontroller gibi form altında görüntülenir.

Properties (Özellikler)	
Color	Seçilen yazı rengi
MaxSize	Kullanıcının seçebileceği maksimum font büyüklüğü
MinSize	Kullanıcının seçebileceği minimum font büyüklüğü
ShowColor	Diyalog kutusunda yazı rengi seçiminin gösterilip gösterilmemesini belirler.
ShowEffects	Diyalog kutusunda fonta ilgili çeşitli özelliklerin (üztü çizili, altı çizili gibi) gösterilip gösterilmemesini belirler.

Renk Diyalog Kutusu (ColorDialog)

ColorDialog, geliştirdiğimiz uygulamalarda kullanabileceğimiz yazı rengini belirlememizi sağlayan bir kontroldür. Belirlediğimiz rengi kontrol özelliklerinde (buton rengi, formun zemin rengi, vb. gibi) kullanabileceğimiz gibi, grafik uygulamalarında da kullanabiliriz. Bu diyalog kontrolü de diğer kontroller gibi form altında görüntülenir.

Properties (Özellikler)	
AllowFullOpen	Kullanıcının özel renklerini tanımlamasına (Custom Colors) izin verilip verilmemesini belirler.
AnyColor	Basic Colors (Temel Renkler) setinde tüm renklerin gösterilip gösterilmeyeceğini belirler.
Color	Seçilen renk
FullOpen	Color diyalog kutusu açıldığında, Custom Colors kontrolünün olup olmayacağını belirler. AllowFullOpen değeri False ise bu özelliğin bir önemi yoktur.
SolidColorsOnly	True olması durumunda kullanıcı yalnızca Solid renkleri seçebilmekle kısıtlanacaktır.

Klasör Gezgini (FolderBrowserDialog)

Klasör seçmek ve klasör oluşturmak için kullanılan bir kontroldür.

Properties (Özellikler)	
Description	Diyalog kutusunun üstündeki açıklama yazısını belirler.
RootFolder	Klasörleri listelerken kök klasörünün belirlenmesini sağlar.
SelectedPath	Varsayılan olarak seçili klasörü gösterir.

ÖRNEK:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace folderBrowserDialog
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnBrowser_Click(object sender, EventArgs e)
        {
            dlgBrowser.ShowDialog(); //Diyalog Kutusunu gösterir.
        }
        private void btnOpen_Click(object sender, EventArgs e)
        {
            dlgOpen.ShowDialog(); //Diyalog Kutusunu gösterir.
        }
        private void btnSave_Click(object sender, EventArgs e)
        {
            dlgSave.ShowDialog(); //Diyalog Kutusunu gösterir.
        }
        private void btnColor_Click(object sender, EventArgs e)
        {
            dlgColor.ShowDialog(); //Diyalog Kutusunu gösterir.
        }
        private void btnFont_Click(object sender, EventArgs e)
        {
            dlgFont.ShowDialog(); //Diyalog Kutusunu gösterir.
        }
    }
}
```

