Aplicația ModalDialogExemplu

- Construiește un dialog modal care se deschide la apăsarea unui buton al formei părinte
- Datele introduse într-un control al dialogului sunt transferate unui control al formei părinte

1. Creăm un proiect nou de tipul **WindowsFormsApplication**.
2. Plasăm pe formă cu mousul un ListBox și două butoane. Setăm proprietatea Text a celor două butoane pe valorile Adauga și respectiv Iesire.
3. Adăugăm o nouă formă proiectului la *design-time*: în Solution Explorer click dreapta pe numele proiectului, alegem Add, apoi Windows Form.
4. În Solution Explorer dublu click pe numele Form2.
5. Aducem din Toolbox pe Form2 un TextBox și un buton, inscriptionat cu Ok.
6. În Solution explorer facem click pe numele Form2, alegem View Code si in clasa Form2 definim noii membri:

```
//câmp care  reține textul introdus în TextBox1
private string item;
/*proprietate care  returneaza valoarea textului; reamintim ca o proprietate este un fel
de  metoda
care permite tratatea campurilor private ca si cum ar fi publice, dar numai pentru doua
genuri de operatii: set si get.
*/
public string Item
{
   get{return item;}
}
```

Tratam evenimentul **Click** generat de apasarea butonului Ok: introducem codul de mai jos in metoda handler a evenimentului **click** al lui button1:

```
item = textBox1.Trim();
//inchidem forma Form2:
 Close();
```

7. In Form1 introducem o referinta catre Form2:

```
private Form2 f; //aceasta declaratie se scrie in clasa Form1, din fisierul Form1.cs.
```

8. Tratam evenimentul Click al butonului Adauga aflat pe Form1.
9. Pentru iesirea din aplicatie tratam evenimentul click al butonului Iesire:
   Application.Exit();
10. Compilam si rulam aplicatia.

Codul pentru Form1 este urmatorul:

```
namespace DialogModalWFA
{
   public partial class Form1 : Form
```

```
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
        private Form2 f;

        private void button1_Click(object sender, EventArgs e)
        {
            //cream forma form2
            f = new Form2();
            //afisare dialog modal
            f.ShowDialog();
            if(f.Item!="") //daca exista text introdus
            {
                //adauga itemul in listBox1
                listBox1.Items.Add(f.Item);
            }
        }

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

Codul pentru obiectele din Form2:

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

        private void Form2_Load(object sender, EventArgs e)
        {

        }
        // camp care  retine textul introdus in textBox1
        private string item;
        //proprietate get item
        public string Item
        {
            get
            {
                return item;
            }
        }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            //salvam textul introdus in control
            item = textBox1.Text.Trim();
            //inchidem forma Form2
            Close();
        }
    }
}
```

De reținut:

- Dialogurile modale se afișează cu metoda ShowDialog, iar cele nemodale cu Show.
- La închiderea cu Close un dialog nemodal se distruge (se distruge instanta clasei).
- Un dialog modal nu este distrus la închiderea cu Close. Câmpurile instanței rămân, iar forma poate fi reafișată. A se vedea accesul la câmpul f.Item după închiderea cu Close a formei Form2.
- Codul care urmează lui ShowDialog nu se execută până când forma nu se închide. După închiderea dialogului modal, putem referi în continuare clasa Form2 și membrii ei.
- Întrebare: la o a doua apăsare a butonului Form1.Button1 se creează o nouă instanță a formei Form2? Dacă da, ce se întâmplă cu cea veche? Cum mai poate fi accesată?

Fişă de lucru

1. Explicaţi în maximum două propoziţii ce se înţelege prin .NET Framework

	.NET Framework este un mediu de execuţie a aplicaţiilor pe sisteme de operare Windows, dar şi o platformă de dezvoltare a aplicaţiilor.

2. Care sunt cele două componente de bază ale Arhitecturii .NET ?

	.NET are două componente de bază: CLR (Common Language Runtime) şi BCL (Base Class Library). Prima reprezintă mediul de dezvoltare şi de execuţie a aplicaţiilor .NET, iar a doua este Biblioteca de Clase, care poate fi utilizată în programarea aplicaţiilor .NET.

3. Ce este şi ce rol îndeplineşte ***Common Language Runtime*** **(CLR)** . ?

	CLR este maşină virtuală Microsoft, o componentă a platformei .NET. Este responsabilă pentru gestionarea proceselor .NET şi pentru compilarea codului. În urma comilării, se produce un limbaj intermediar (CIL). CLR compileză prin intermediul JIT (Just In Time Compiler) acest cod, în cod maşină. Astfel se realizează portabilitatea aplicaţiilor .NET.

4. Explicaţi ce este ***Base Class Library*** şi descrieţi în cîteva fraze rolul **BCL**.

	Este o bibliotecă de clase inclusă mai ales în spaţiul de nume System. Acest spaţiu de nume defineşteclase dar şi alte spaţii de nume cu rol specific. Aplicaţiile dezvoltate pe .NET framework au utilizează această bibliotecă pentru a beneficia de avantajele codului gestionat de .NET (*managed code*). Printre aceste avantaje se numără colectarea deşeurilor (garbage collection).

5. Explicaţi modalitatea de compilare a programelor pe platforma .NET.

	CLR asigură compilarea codului sursă într-un limbaj intermediar: **CIL** (*Common Intermediate Language*). Codul CIL este memorat în fişiere numite *assembly*, apoi, în momentul execuţiei, acesta este compilat cu ajutorul compilatorului JIT (Just In Time), care produce codul maşină.

```
}
```