# Contents

CAPITOLUL 2	2
Aplicații de tip Windows Forms Application în C#	2
1. Introducere	2
1.2. Controale și proprietăți	3
1.3. Adaugarea mai multor form-uri	6
1.4. Scrierea și citirea din fișiere	7
2. Crearea unei aplicații de tip "Windows Forms Application" în C#	9
2.1 Crearea unui nou proiect de tip Windows Forms Application	9
2.2 Compilarea și rularea unei aplicații de tip "Windows Forms Application" în C#1	.1
3. Exerciții1	3
Tema T21	.3

## **CAPITOLUL 2**

## Aplicații de tip Windows Forms Application în C#

În acest capitol vor fi prezentate clase specializate pentru crearea ferestrelor sau a formularelor utilizând System.Windows.Forms.Form, controale cum ar fi butoane (System.Windows.Forms.Button), casete de text (System.Windows.Forms.TextBox), Label, RadioButton, ComboBox, ListBox, CheckedListBox și multe altele.

## 1. Introducere

Codul complex care realizează aplicații de tip Windows Form, necesită utilizarea de clase din două assembly-uri standard:

- System.Windows.Forms.dll: conține clasele corespunzătoare elementelor grafice utilizate în cadrul aplicației (formulare, butoane, etc) și mecanismul de execuție pentru acestea;
- System.Drawing.dll: conține structurile utilizate pentru reprezentarea geometriei (puncte, dreptunghiuri, dimensiuni, etc) și funcțiile necesare pentru desenarea pe ecran.

Clasele derivate din *Form* moștenesc proprietăți care determină atributele vizuale, metode care implementează anumite comportamente precum *Show*, *Hide*, *etc* și o serie de metode specifice (handlere) precum *Load*, *Click* etc.

Cele mai utilizate proprietați ale form-urilor sunt:

- StartPosition poziția ferestrei atunci când aceasta apare prima dată, poate fi setată și manual;
- Location(X,Y) coordonatele coltului din stanga sus al form-ului;
- ControlBox precizează dacă fereastra conține sau nu un icon;
- Opacity procentul de opacitate;
- > TransparencyKey- identifică o culoare care va deveni transparentă pe formă.

Definirea funcțiilor de tratare a unui *eveniment* asociat controlului se realizează prin selectarea grupului *Events* din fereastra *Proprieties*.

*Observatie!:* La efectuarea unui dublu click pe un *control* din Designer se va genera automat o funcție pentru evenimentul implicit.

Cele mai utilizate evenimente sunt:

Load – apare atunci când formularul este încărcat în memorie pentru prima data;

- FormClosed apare atunci când formularul este închis;
- Activated apare atunci când formularul este activ;
- > **Deactivate** apare atunci când utilizatorul va da click pe un alt formular al aplicației.

### 1.2. Controale și proprietăți

Cuvântul *control* este utilizat pentru a face referire la obiectele de interfață cum ar fi butoanele, câmpuri de editare (labels), bare de defilare (scroll bars), etc. Un **Control** este instanța unei clase derivate din *System.Windows.Forms*, fiind responsabil cu desenarea unei părți din formular. În Visual Studio .NET găsim o serie de controale standard, disponibile în Toolbox.

Fiecare control ocupă o zonă rectangulară în cadrul ferestrei, iar punctul de origine al sistemului de coordonate este coltul din stânga sus al ferestrei și unitatea de masură este pixelul.

Cele mai utilizate controale din cadrul bibliotectii Windows Forms sunt:

- Formă (Form) clasa de bază pentru o fereastră;
- Eticheta (Label) permite afişarea unui text static;
- Buton (Button) execută o secventă de instrucțiuni;
- Caseta de text (TextBox) permite afişarea şi citirea textului;
- Calendar (MonthCalendar) permite afişarea unui calendar al lunii curente;
- Caseta de validare (CheckBox) permite opţunile de a include/exclude sau da/nu;
- Caseta de listă (*ListBox*) permite afișarea unei liste din care putem alege;
- Imagine (PictureBox) utilizat pentru adăugarea imaginilor;
- Buton radio (RadioButton) utilizat pentru a selecta un singur element dintr-un grup de selecții.

Proprietați ale Clasei Control care determina poziționarea controalelor:

- Location proprietate de tip Point care determină poziția față de controlul părinte;
- Size proprietate de tip Size care determină dimensiunea controlului;
- Top, Left permit citirea și modificarea distanței pe verticală și orizontală față de originea controlului părinte;
- Bottom, Right permit citirea distanței de la marginea de jos / din dreapta a controlului față de controlul părinte;
- Margin setează distanța dintre marginile controlului curent și controlul părinte;

Padding – setează dintre marginile controlului curent și controalele copil;

Alte proprietăți ale Clasei Control disponibile pentru toate controalele utilizate în cadrul bibliotecii Windows Forms:

- Text proprietate de tip string care permite modificarea textului afişat de către control;
- Name proprietate de tip string care permite asocierea unei denumiri controlului pentru uşurarea regăsirii ulterioare în cadrul arborelui de controale;
- ForeColor, BackColor proprietăți de tip System.Drawing.Color care permit schimbarea culorii utilizate pentru textul și pentru fundalul controlului;
- Font proprietate de tip System.Drawing.Font care permite modificarea stilului utilizat pentru afişarea textului în cadrul controlului;
- Enabled proprietate de tip bool care permite activarea și dezactivarea controlului.

Pentru a adauga controale, din ToolBox, selectam controlul dorit si il plasam pe suprafata formularului. Toate controalele au proprietati, dacă dăm click pe o proprietate, pe ultimul rând se vor afișa informații suplimentare a proprietații selectate (Fig. 1).

Pr	operties	<del>т</del> Д	Х						
b	btn_Click System.Windows.Forms.Button								
	1 💱 🖓 🗲 🎤								
Ξ	Data		*						
Ŧ	(ApplicationSettings)								
Ŧ	(DataBindings)								
	Tag								
Ξ	Design								
	(Name)	btn_Click							
	GenerateMember	True							
	Locked	False							
	Modifiers	Private	-						

#### (Name)

Indicates the name used in code to identify the object.

Fig. 1 Proprietăți ale controlului Button

Exemplu 1.) În figura de mai jos este prezentată o aplicație care la apăsarea butonului "Click !" din formă, va afișa o fereastră cu numele MyFirst MessageBox și cu mesajul "Acesta este un MessageBox!". Nota: Dacă omitem "MyFirst MessageBox", atunci fereastra nu va avea nici un nume (titlu).

#### Capitolul 2

#### Informatică Industrială – Îndrumator de laborator

WindowsFormsApp2 (Running) - Microsoft Visual Studie           File         Edit         View         Project         Build         Debug         Team           O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O         O = O </th <th>Tools Test RTools Analyze Window Help Any CPU -  ▶ Continue =   20 = 1 ■</th> <th>() : - : <i>골</i> - <b>는 作: '과 '과 '목</b> 및 '와 '과 - <b>약</b> Applic 중 '24' Stack Frame.</th> <th>کې کې Quick Launch کې کې ation Insights - ج</th>	Tools Test RTools Analyze Window Help Any CPU -  ▶ Continue =   20 = 1 ■	() : - : <i>골</i> - <b>는 作: '과 '과 '목</b> 및 '와 '과 - <b>약</b> Applic 중 '24' Stack Frame.	کې کې Quick Launch کې کې ation Insights - ج
Program.cs = Form2.cs [Design] = Form1.cs	s		▼ Diagnostic Tools
<pre>Ed WindowsformsApp2 9 Unitsing=System.Windows.Forms; 10 = namespace-WindowsFormsApp2 12 { {public-partial-class-Form1:: 14 }public-partial-class-Form1:: 14 }public-Form1() 16 } 17 18 } 19 20 20private-void-btn_Click_Cl 21 22 } 23 } 24 } 24 } 26 0</pre>	<pre>Form ; ; ick(object-sender, EventArgs-e) ta-este-un-MessageBox!-", "-MyFirst-MessageBox</pre>	- ", bin, Click, Click, Object sender, EventArgs e)	CPU (% of all processors)     100     Summary Events Memory Usage     Events     Summary Events Memory Usage     Events     Summary Events Memory Usage     Events     Summary Events (0 of 0)     Memory Usage     Take Snapshot     CPU Usage     Take Snapshot     CPU Usage     Take Snapshot     CPU Usage
100 % • <			
Name Value	Туре	Entre Solution     Search Error List     Search Error List     Code Description	0 0 of 1 Message Proj
Autos Locals Watch 1		Call Stack Breakpoints Exception Settings Command Window	v Immediate Window Output Error List

Fig.2 Exemplu MessageBox

Exemplu 2.) Să se creeze o aplicație care conține urmatoarele controale: două Label-uri, două TextBox-uri, și un Button (Specificații Button: Name: myBtn, Location: 90,170, Text: Get Text), care copiază textul din primul TextBox în al doilea, precum în fig. 3. În urma rulării programului vom obtine rezultatul din figura 4.

Nota: Al doilea TextBox este setat ReadOnly.

🖳 Form1	
Text	
Copied Text	
Ge	t Text

Fig.3 Exemplu MessageBox

#### Informatică Industrială – Îndrumator de laborator

IdetTextFromTextBox (flumning) - Microsoft Visual Studio         File       Edit       View       Project       Build       Debug       Text       R Tools       Analyze       Window       Help         Image: Image	♡   → ☆ ? ☆   提 = 15 項 ] 耳 왜   ■ 왜 해 제 = ♥ Application I	🝸 🜮 Quick Launch 🔑 🗕 🤧 🗙
Process: [9124] GetTextFromTextBox.exe • 📧 Lifecycle Events • Thread: •	💎 🐔 Stack Frame: 👻 🚽	
Form1.cs a 🕫 🗶 Form1.cs (Design) a		Tools → म × ≌
🖾 GetTextFromTextBox.Form1	<ul> <li></li></ul>	• 💠 🔍 Q 👔 🔨 🗸 🖥
<pre>EdeflectionTextBox</pre>	- ♥ myBtr_Clicklobject sender, EventArgs e)	CPUCE     Control Contro Control Control Control Control Control Control Control
		Take Snapshot
		CPU Usage
100 % -		Record CPU Profile
Autos	▼ # × Error List	- # ×
Name Value Type	Entire Solution - 😵 0 Errors 🔥 0 Warnings 🕕 0 o	of 1 Message Y Build + IntelliSense -
	Search Error List	- م
	Image: Code         Description	Project File
Autos Locals Watch 1	Call Stack Breakpoints Exception Settings Command Window Imr	mediate Window Output Error List

Fig.4 Exemplu MessageBox

### 1.3. Adaugarea mai multor form-uri

În general, aplicațiile conțin mai multe forme. Există o formă principală de unde putem începe să navigam prin cele secundare. Astfel, dupa crearea unui proiect nou de tip Windows Form care conține un buton în formular, la acționarea butonului se va deschide o nouă formă.

Pentru a adăuga un nou formular proiectului, selectam Project  $\rightarrow$  Add Windows Form(Fig. 5).

Toolbox 🔫 👎	× Form2.cs	Form2.cs [Design]		Form1.cs [Design]							Solution E	plorer	* # ×
Search Toolbox	D → C∎ Windo	wsFormsApp3	•	MindowsFormsApp3.Form	1		▼ <sup>Q</sup> <sub>a</sub> btn_Open_C	lick(o	object sender, EventArgs e)		006	5 8 - 1	`o• <b>\$</b> ¢ ∉ @ "
General     There are no usable controls in     this group. Drag an item onto	6 7 8 9	using-System.linq; using-System.Text; using-System.Threading.Tesks; using-System.Windows.Forms;								+	Search Sol	ution Explo ion Windo	orer (Ctrl+;) P - wsFormsApp3' (1 project)
this group. Drag an term onto	x. 111 13 14 15 16 17 18 19 20 20 21 22 23 24 24 25 26	Datespace WindowsFormApp3     for public partial class form     formalic partial class form     formalic formal()     formalic formal()     formalic fo	1::Form nt(); _Click(object:sender, _new.Form2(); ;}	-(ventargs:e)				9 2 ( <del>(</del> ) <del>(</del>	Build Rebuild Clean Analyze Publish Distribute With HockeyApp Publish Stored Procedures Scope to This New Solution Explorer View		,	Propertie Reference App.conf Form1.cs Form1 Form2 Form2 Program.	ninaspos es ig I.Designer.cs I.Lesx cs
					🕄 💕 🗗 C	New Item Existing Item New Folder From Cookiecutter	Shift+Alt+A	# \$	Add Manage NuGet Packages Set as StartUp Project Debug Initialize Interactive with Project		•	cplorer Te	am Explorer
						Reference Web Reference		×	Source Control	c	• trl+X	ormsApp	B Project Properties •
	75 %	nut from Debug			కిశ	Service Reference Connected Service Analyzer		â ×	Paste Remove Rename	D	trl+V Iel	ile older	WindowsFormsApp3.cs C:\Users\Admin\source
	'Window 'Window 'Window	esFormsApp3.exe* (CLR v4.0. rsFormsApp3.exe* (CLR v4.0. rsFormsApp3.exe* (CLR v4.0. rsFormsApp3.exe* (CLR v4.0.	30319: WindowsForm 30319: WindowsForm 30319: WindowsForm 30319: WindowsForm	msApp3.exe): Loaded 'C:\WJ msApp3.exe): Loaded 'C:\WJ msApp3.exe): Loaded 'C:\WJ msApp3.exe): Loaded 'C:\WJ msApp3.exe): Loaded 'C:\WJ	<b>値</b> 11 な	Windows Form User Control Component		5	Unload Project Open Folder in File Explorer Properties	A	lt+Enter		
Server Explorer Toolbox	'Window The pro	rsFormsApp3.exe' (CLR v4.0. ogram '[12500] WindowsForms/ Output	30319: WindowsForm App3.exe" has exit	msApp3.exe): Loaded 'C:\WI ted with code 0 (0x0).	240	Class			WWT/A4*0~4*0*0*0 <sup>-0</sup> 1197507334	*	Misc	1	

Fig.5 Adaugarea unei noi Forme

Nota: Acum, avem un proiect care conține două formulare, însă acestea trebuie conectate între ele.

Formele sunt reprezentate de clase. Astfel, obiectul primei forme este creat automat atunci când programul este pornit. Pentru a doua formă, trebuie să fie creat manual, după cum urmează:

```
private void btn_Show_Click(object sender, EventArgs e)
{
     Form2 secondForm = new Form2();
     secondForm.Show(); // afiseaza a doua forma
}
```

Prin utilizarea codului de mai sus, al doilea formular va apărea de fiecare dată când faceți click pe butonul din prima formă. Pentru a face formularul să apară o singură dată, deplasam codul care creează formularul în afara metodei. Rezultatul rulării programul se poate vedea în figura de mai jos :



Fig.6 Exemplu SecondForm

Acțiune pe butonul Exit, care permite ieșirea din forma este :

```
private void btn_Exit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

### 1.4. Scrierea și citirea din fișiere

În C# fișierele text sunt gestionate prin stream-uri (fluxuri de date structurate pe octeți), și sunt în general utilizate pentru stocarea informațiilor. Pentru a putea utiliza fișierele text trebuie să adăugam spațiul de nume System.IO în program.

Cele mai utilizate clase pentru scrierea și citirea din fișiere sunt: *StreamReader* (citirea datelor din fișier) și *StreamWriter* (scrierea datelor în fișier).

Administrarea fișierelor se referă la tehnica de citire și scriere a datelor din/în fișiere. Pentru accesarea (operații de intrare-iesire) fișierului se folosesc clasele : File, StreamWriter, BinaryReader si BinaryWriter din System.IO.

Scrierea în fișier utilizând StreamWriter, se poate face astfel:

```
private void Form1_Load(object sender, EventArgs e)
{
   StreamWriter str = new StreamWriter("fisier1.txt", true);
   str.WriteLine("Acesta este un fisier text!");
   str.Close();
}
```

<u>Nota</u>: Valoare booleana **true** permite scrierea pe urmatoarea linie din fisierul text, iar daca aceasta este false se va suprascrie linia/liniile din fisierul text.

*Observație!*: După executarea codului de mai sus, se va crea automat fișier myFile.txt în același folder cu executabilul, care de regulă este **bin\Debug** (Fig.7). Fișierele se pot salva și intr-o locație specificată, adăugând în fața numelui o cale completă către fișier.



Fig.7 Locație fișier text

Pentru citirea din fișier utilizăm StreamReader:

```
private void Form1_Load(object sender, EventArgs e)
{
   StreamReader streamReader = new StreamReader(path: "fisier1.txt");
   {
     string line;
     while ((line = streamReader.ReadLine()) != null)
        MessageBox.Show(line);
   }
}
```

}

}

În figura de mai jos este prezentat un exemplu(Exemplul 1) de scriere și citire din fișier, utilizând clasele: StreamReader și StreamWriter.

WindowsFormsApp2 (Running) - Microsoft Visual Studio		💙 Quick Launch 🔑 🗕 🗗 🗙
File Edit View Project Build Debug Team Tools Test R Tools Analyze Window He	elp	PA - P
🗴 💿 - 💿 🔯 - 🏩 🔐 🦻 - 😋 - Debug - Any CPU - 🕨 Continue - 🃁 - 🕅 - 🕅 -	も と → さ ク さ # = ※ ha (第 ) 第 24 ■ 第 3 3 3 4 = ※  ♀ Applicat	ion Insights * =
Process: [13084] WindowsFormsApp2.exe - IT Lifecycle Events - Thread:	- V V Stack Frame:	
Formics B di X fiziertet Formics (Decign) Form2 er (Decign) Program er		- Diagonatis Taola
WindowsFormsApp2		
10 using-System.Windows.Forms;	our_crick(object seried), eventings c)	
11 12 © namespace - WindowsFormsApp2		Diagnostics session: 7 seconds
13 ( 14 Dimoublic-partial-class-Form1-:-Form		P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P P.
15 ·····(	💀 Form1 — 🗆 🗙	A Events
17 podzie volmi ()		d Brocers Memony (MR)
<pre>18 InitializeComponent(); 19 InitializeComponent();</pre>		16 16
20 21 Elprivate-void-btm_Click_Click(object-sender,-EventArgs-e)	Click I limbai	-
22		
<pre>24streaminiter.WriteLine("limbaj-de-programare");</pre>	de	d CPU (% of all processors)
25 streammater.cause(); 26		100 100
27 StreamReader-streamReader-=-new-StreamReader(path:"fisier.txt"); 28 ()(	programare	
29 string-line-=-string.Empty; 30 string-line-=-streamReader,ReadLine())-1=-null)		
31 (tring1) tenderave line Solit('.');		Summary Events Memory Usage CPU Usage
<pre>33 33 33</pre>		Events
<pre>34 35 35 35 35 35 35 35 35 35 35 35 35 35</pre>		Show Events (0 of 0)
36 37		Memory Usage
38 39		Take Snapshot
40		CPU Usage
42		Record CPU Profile
75 % • <		
Autos		- + ×
Name Value T	ype 🔶 Entire Solution 👻 😵 0 Errors 🔥 0 Warnings 🚺	0 of 1 Message 🍟 Build + IntelliSense 👻
	Search Error List	- م
	Code Description  Project	File Line Suppression St Y
Autos Locals Watch 1	Call Stack Breakpoints Exception Settings Command Window	Immediate Window Output Error List

Fig.8 Exemplul 1.) scriere/citire din fișier

# 2. Crearea unei aplicații de tip "Windows Forms Application" în C#

## 2.1 Crearea unui nou proiect de tip Windows Forms Application

Pentru crearea unei aplicații de tip "Windows Forms Application" se construieste un proiect nou de tip Visual C# din meniul File -> New -> Project și se alege tipul proiectului Windows Forms Application, precum în figura de mai jos.

nformatică In		Ca	pito	lul 2					
								2	
New Project								f	~
> Recent			.NET Fr	amework 4.6.1 - Sort by: Default	- II II		Search (Ctrl+E)		<b>م</b>
Installed			<b>N</b>	Blank App (Universal Windows)	Visual C#	Î	Type: Visual C#		
<ul> <li>Visual C#</li> <li>Windows Universal</li> </ul>				Cross-Platform App (Xamarin.Forms)	Visual C#		A project for creating an app a Windows Forms user interfa	licatio ace	n with
Windows Cli ▷ Web	assic Desktop			WPF App (.NET Framework)	Visual C#				
<ul> <li>Office/Share</li> <li>.NET Core</li> </ul>	Point		C <sup>#</sup>	Windows Forms App (.NET Framework)	Visual C#	2			
.NET Standa Android	.NET Standard Android		5``	Console App (.NET Core)	Visual C#				
Cloud Cross-Platfo	rm	۰.	<u> </u>	Console App (.NET Framework)	Visual C#				
Extensibility ▶ iOS				Class Library (.NET Standard)	Visual C#				
Test ▷ tvOS				Class Library (.NET Framework)	Visual C#				
WCF Workflow			<b>SÐ</b>	ASP.NET Core Web Application	Visual C#				
Visual Basic		-	⊕_	ASP.NET Web Application (.NET Framework)	Visual C#				
Not finding what you are looking for? Open Visual Studio Installer				Shared Project	Visual C#	-			
Name:	me: WindowsFormsApp								
Location:	C:\Users\Admin\s	soure	ce\repos		-		Browse		
Solution:	Create new solut	ion			-				
Solution name:	WindowsFormsA	pp1					<ul> <li>Create directory for solution</li> <li>Add to Source Control</li> </ul>		
							ОК	Can	icel

Fig.9 Proiect nou de tip WFApp

Se denumește proiectul WindowsFormsApp1 și se alege locația în care se va salva proiectul. Se denumeste spațiul de lucru, bifându-se și crearea unui director specific acestuia (în general, spatiul de lucru are același nume ca și numele proiectului, în cazul acesta WindowsFormsApp1, iar apoi acționăm butonul OK.

După cum se poate observa în figura de mai jos, în partea stângă se află meniul Toolbox, care conține controale (Button, ComboBox, Label, LinkLabel, ListView etc). Aceste controale se pot adăuga în formă cu "drag and drop", iar în partea dreaptă jos se găsesc proprietațile acestora.



Fig.10 Conținut proiect nou

### 2.2 Compilarea și rularea unei aplicații de tip "Windows Forms Application" în C#

Pentru compilarea aplicației WindowsFormsApplication1 se selectează **Build -> Build Solution (F6)** din meniul principal al mediului de programare. În cazul în care există erori, acestea sunt afișate în fereastra Error List (situată în partea de jos a mediului de programare). Efectuând dublu click pe fiecare eroare în parte, cursorul din program se poziționează pe linia care conține eroarea.

Rularea aplicației după compilare se poate realiza în mai multe moduri:

- rapid cu depanare (Debug -> Start Debugging sau F5);
- rulare pas cu pas (Debug -> Step Into sau F11 şi Debug -> Step Over sau F10);
- rulare rapidă până la linia marcată ca punct de întrerupere (Toggle Breakpoint sau F9 pe linia respectivă și apoi Debug -> Start Debugging sau F5). Încetarea urmăririi pas cu pas (Debug -> Stop Debugging sau Shift+F5) permite ieșirea din modul de depanare și revenirea la modul normal de lucru.

Exemplu 2.) Aplicația prezentată realizează operațiile aritmetice(+,-,/,\*) și afișează rezultatul într-un TextBox de tip Read-Only. Pentru realizarea aplicației s-au utilizat 3 label-uri, 2 textBox-uri si 4 butoane după cum se poate vedea în figura de mai jos.

🖳 Form1	
Numarul 1	+
Numarul 2	-
	1
Rezultat	Clear

Fig.11 Operații aritmetice în Win Form

Informatică Industrială – Îndrumator de laborator

```
public partial class Form1 : Form
   {
        public Form1()
            InitializeComponent();
        }
private void btn_Addition_Click(object sender, EventArgs e)
              try
                     {
                       Rezultat.Text = (Convert.ToDouble(nr1.Text) +
                       Convert.ToDouble(nr2.Text)).ToString();
                     } catch(Exception ex)
                            {
                              MessageBox.Show("Error!" + ex);
                            }
        }
private void btn_Substraction_Click(object sender, EventArgs e)
        {
            try
                     Rezultat.Text = (Convert.ToDouble(nr1.Text) -
                     Convert.ToDouble(nr2.Text)).ToString();
                     } catch(Exception ex)
                            ł
                              MessageBox.Show("Error!" + ex);
                            }
        }
private void btn_Multiplication_Click(object sender, EventArgs e)
        ł
              try
                     Rezultat.Text = (Convert.ToDouble(nr1.Text) *
                     Convert.ToDouble(nr2.Text)).ToString();
                     } catch(Exception ex)
                            {
                              MessageBox.Show("Error!" + ex);
                            }
        }
private void btn_Divison_Click(object sender, EventArgs e)
        {
              try
                     Rezultat.Text = (Convert.ToDouble(nr1.Text) /
                     Convert.ToDouble(nr2.Text)).ToString();
                     } catch(Exception ex)
                              MessageBox.Show("Error!" + ex);
                            }
        }
private void btn_Clear_Click(object sender, EventArgs e)
            Rezultat.Text = "";
            nr1.Text = "";
            nr2.Text = "";
        }
```

## 3. Exerciții

- 1. Să se creeze un proiect de tip Windows Form, care să conțină un login (un user și o parolă citite din fișier), iar la apăsarea unui buton să se deschidă o nouă formă. A doua formă să conțină un label "Hello username" și un buton de Exit.
- 2. Să se creeze un proiect de tip Windows Form conform figurii de mai jos, primul listBox să citească obiecte din fisier, iar apoi obiectele selectate să fie copiate în cel de-al doilea listBox. Butonul Sterge permite ștergerea obiectelor din al doilea listBox.

🖳 Form2		
listBox1	Copiaza	listBox2
	Sterge	
	Exit	

Fig.12 Formă - exercitiul 2

- 3. Creați un calculator care conține trei textBox-uri (două dintre acestea sunt utilizate pentru introducerea numerelor, iar cel de-al treilea pentru afișarea rezultatul) și un meniu (MenuStrip) care conține operatorii binari.
- 4. Creați o formă care conține un TabControl cu două pagini. Prima pagină să conțină o listă cu diferite elemente (de exemplu: nume de mașini, nume de orașe, etc). Când este selectat un element, în formular va apărea o imagine specifică numelui selectat. A doua pagină are două grupuri radioButton. Selectați un radioButton din primul grup și unul din al doilea și tipăriți rezultatul într-un MessageBox.

**Tema T1:** Să se implementeze în C# o aplicație de tip WindowsForm Application care să conțină minim 5 tipuri de controale. Tema aplicației este la alegere.