

Laborator – Folosirea variabilelor tip referință

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Crearea și implementarea proiect Windows Forms
- Stabilirea aspectului unui formular Windows Forms
- Scrierea de cod care să specifice comportamentul formularelor din aplicație

Condiții prealabile

Înainte de a realiza acest laborator trebuie să fiți familiarizați cu următoarele concepte:

- Crearea și folosirea claselor
- Delegați și evenimente

➤ Exercițiul 1 - Calculul sumei a doua numere

Se va crea un proiect nou Windows Forms care va avea un singur formular. În cadrul acestuia se vor găsi câte un control text pentru primul număr, pentru al doilea număr și pentru rezultat. În plus, va exista un buton care în momentul în care va fi apăsător se va calcula suma celor 2 numere.

Crearea unui proiect nou

- De la meniul **File** se alege opțiunea **New => Project**.
- În partea stângă se alege **Visual C#**, iar în zona centrală se alege **Windows Forms Application**.
- Proiectul se va numi "Calculeaza suma". Acest lucru se precizează în partea de jos a ferestrei.

Stabilirea aspectului vizual

- Apăsați dublu click pe formularul **Form1.cs**.
- Deschideți fereastra **Toolbox** de la meniul **View**.
- Adăugați 3 controale de tip **Label** și 3 de tip **TextBox**.
- Deschideți fereastra **Properties Windows** de la meniul **View**.
- Denumiți cele 3 etichete în ordine de sus în jos:
 - number1_label
 - number2_label
 - result_label
- Precizați textul care se afișează pentru fiecare:
 - Numar 1
 - Numar 2
 - Result
- Denumiți cele 3 casete text în ordine de sus în jos:
 - number1_textBox
 - number2_textBox



- `Result_textBox`
- Adăugați un buton care se va numi **sum_button** și va avea proprietatea `Text` setată la valoarea "Sum".
- Marcați proprietatea **ReadOnly** a controlului `rezultat_textBox` la valoarea **True** pentru a nu permite editarea rezultatului

Implementarea comportamentului

- Din modul **Designer** pentru **Form1.cs**, apăsați dublu clic pe butonul cu textul "Sum"
- Se va crea o metoda handler pentru evenimentul **Click**
- Preluati valorile din cele 2 casete text prin intermediul
 - `number1_textBox.Text`
 - `number2_textBox.Text`
- Interpretați valoarea de tipul **string** obținută la un număr întreg
 - `string str = number1_textBox.Text`
 - `int number1 = int.Parse(str)`
- La sfârșitul acestei metoda setați proprietatea `Text` a controlului **sum_textBox** la valoarea sumei dintre cele 2 numere

Testați aplicația

Asigurarea integrității formularului

- Există posibilitatea ca valoarea din **result_textBox** să nu coincidă mereu cu suma celor 2 numere. Acest lucru se poate întâmpla dacă se editează unul dintre numere după ce a fost calculată suma și nu se apasă butonul **sum_button**.
- Pentru a rezolva această problemă se va extinde comportamentul formularului adăugând o singură metodă handler pentru evenimentul **TextChanged** al controalelor **TextBox** corespunzătoare celor 2 numere.
- În modul designer, deschideți fereastra de proprietăți pentru **number1_textBox**.
- Vizualizați tab-ul **Events** și apoi căutați evenimentul **TextChanged**.
- În coloana corespunzătoare evenimentului scrieți o metodă cu numele "number_textBox_TextChanged".
- Folosiți aceasta metodă și pentru evenimentul cu același nume din **number2_textBox**.
- În metoda handler adăugați cod care șterge valoarea sumei din controlul **result_textBox**:
 - `result_textBox.Text = ""`
- De fiecare dată când editați valoarea pentru cele 2 numere, se va șterge ultimul rezultat.

Testați aplicația



➤ Exercițiul 2 – Tratarea situației în care se scrie un număr invalid

Se va folosi același proiect de la exercițiul 1. Se va insista asupra excepției care va fi aruncată atunci când se scrie un număr incorect în casetele text corespunzătoare celor 2 numere pentru care se calculează suma.

Modificarea metodei `suma_button_Click`

- Se va include tot corpul metodei într-un bloc **try**.
- Deoarece există mai multe tipuri de excepții care pot fi aruncate de metoda **int.Parse**, se va folosi un bloc **catch** în care se prinde orice fel de eroare.
- În blocul **catch** se deschide o fereastră nouă în care utilizatorul este informat de faptul că valorile introduse pentru cele 2 numere nu pot fi citite:
 - `MessageBox.Show("Valorile celor 2 numere nu pot fi citite")`
- În momentul în care veți scrie litere sau nu veți scrie nimic, se va afișa o fereastră nouă în care va fi precizat mesajul anterior.

Testați aplicația

➤ Exercițiul 3 – Ferestre multiple

Se va crea un proiect Windows Forms nou în care se va crea un chestionar cu mai multe tipuri de întrebări. În funcție de întrebare există posibilitatea de a răspunde cu Da/Nu sau a se specifica răspunsul de la tastatură.

Crearea proiectului și a formularelor

- De la meniul File => New => Project creați un proiect Windows Form care se va numi "Chestionar".
- Editați numele formularului creat la denumirea **MainForm.cs** folosind clic dreapta **Rename** (tasta F2)
- În fereastra Solution Explorer dați clic dreapta pe proiect apoi Add => New Item pentru a crea următoarele formulare:
 - SimpleChoice
 - SpecifyAnswer

Aspectul formularelor de răspuns

- Ambele formulare vor avea un **Label** în care va fi afișat textul întrebării.
- Sub acest text, formularul **SimpleChoice** va avea 2 butoane pe care va scrie "Yes", respectiv "No", iar formularul **SpecifyAnswer** va avea o casetă text și un buton pe care va scrie "Verify"

Aspectul formularului MainForm.cs

- Adăugați un **Label** cu numele **description_Label**.
- Implicit un astfel de control se redimensionează automat în funcție de text. Pentru a dezactiva acest comportament editați proprietatea **AutoSize** la valoarea **False**.
- Plasați acest control în partea de sus și la distanță standard de marginea de sus, stânga și dreapta ale formularului.
- Adăugați 2 butoane cu numele **simpleAnswer_button** și **specifyAnswer_button** care vor avea textul "Simple answer", respectiv "Specify answer".



Implementarea comportamentului pentru MainForm.cs

- Adăugați o metodă handler pentru evenimentul **Click** al butonului **simpleAnswer_button** care va deschide formularul SimpleAnswer:
 - `SimpleAnswer form = new SimpleAnswer();`
 - `form.Show();`
- În cazul de mai sus, se permite deschiderea oricător ferestre de întrebări. În cazul în care nu se dorește acest lucru se poate folosi metoda:
 - `form.ShowDialog();`
- În mod asemănător, adăugați o metodă handler pentru evenimentul **Click** al butonului **speciftAnswer_button** care va deschide formularul SpecifyAnswer.
- Din **Designer** specificați o scurtă descriere pentru cele 2 tipuri de întrebări din fereastra de proprietăți.

Testați aplicația**Implementarea comportamentului pentru SimpleAnswer.cs**

- Adăugați o metodă handler pentru evenimentul **Load** al formularului.
- În această metodă scrieți întrebarea în eticheta creată.
- Adăugați câte o metodă handler pentru cele 2 butoane de răspuns, pe evenimentul **Click**.
- În fiecare dintre metode afișați o fereastră **TextBox** în care specificați dacă răspunsul este corect.

Implementarea comportamentului pentru SpecifyAnswer.cs

- Adăugați o metodă handler pentru evenimentul **Load** al formularului.
- În această metodă scrieți întrebarea în eticheta creată.
- Adăugați o metodă handler pentru butonul care verifică răspunsul, pe evenimentul **Click**.
- Afișați o fereastră **TextBox** cu un mesaj în care precizați utilizatorului dacă a raspuns corect.

Testați aplicația