

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

LUCRAREA DE LABORATOR#2

Realizarea unui simplu GUI Calculator

Autor:
Ernest BITCA

lector asistent:
Irina COJANU
lector superior:
Radu MELNIC

Lucrarea de laborator #2

1 Scopul lucrarii de laborator

Realizarea un simplu GUI calculator care suporta urmatoarele functii: +, -, /, *, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.

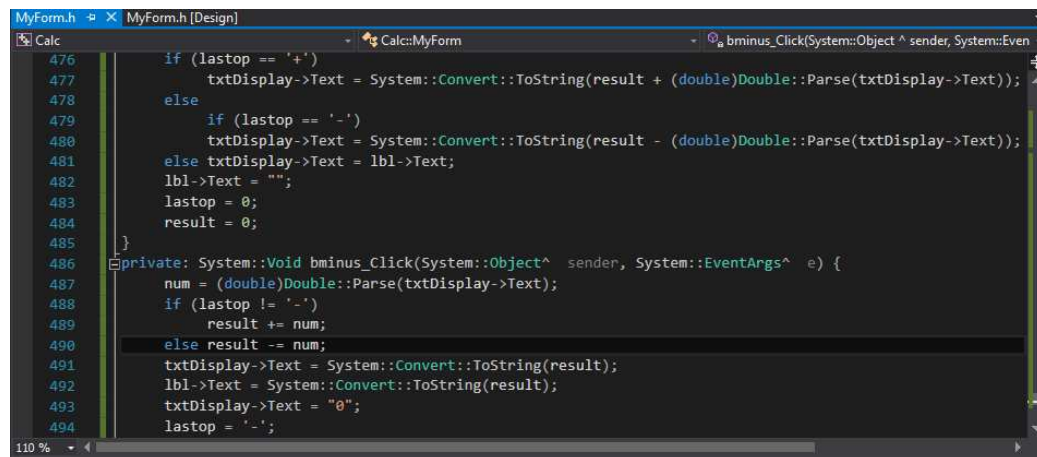
2 Mersul lucrarii de laborator

Pe parcursul lucrarii am creat 2 calculatoare. **Primul** calculator permite efectuarea operatiilor ca +, -, /, *, putere, radical, InversareSemn(+/-). Operatiile date se pot efectua in lant (nu este necesar mereu sa apasam pe semnul egal).

Al doilea calculator permite calcularea expresiilor matematice mai grele. Aici am divizat functionalitatea in 2 parti (cea grafica si insusi algoritmul de prelucrarea a expresiei pe care l-am facut in fisiere aparte).

Secvente de cod

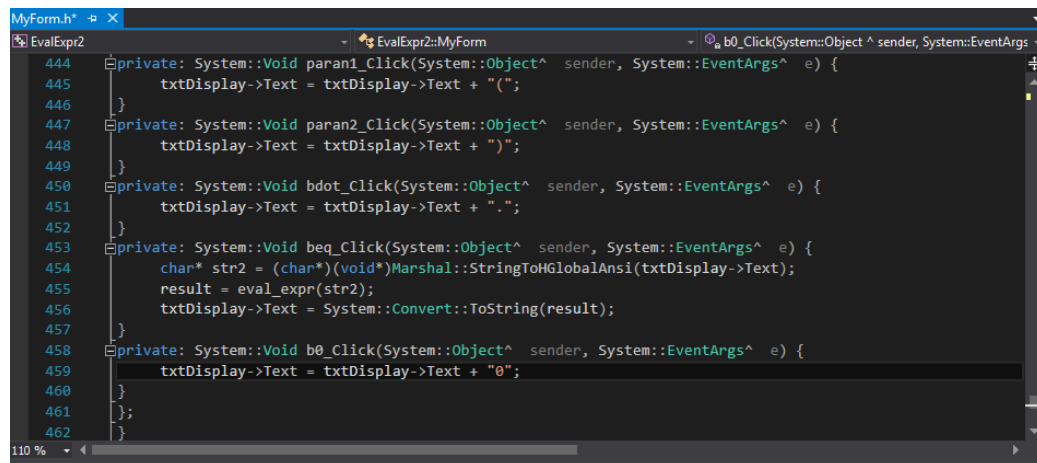
Calculator #1



```
476         if (lastop == '+')
477             txtDisplay->Text = System.Convert.ToString(result + (double)Double.Parse(txtDisplay->Text));
478         else
479             if (lastop == '-')
480                 txtDisplay->Text = System.Convert.ToString(result - (double)Double.Parse(txtDisplay->Text));
481             else txtDisplay->Text = lbl->Text;
482             lbl->Text = "";
483             lastop = 0;
484             result = 0;
485     }
486     private: System::Void bminus_Click(System::Object^ sender, System::EventArgs^ e) {
487         num = (double)Double.Parse(txtDisplay->Text);
488         if (lastop != '-')
489             result += num;
490         else result -= num;
491         txtDisplay->Text = System.Convert.ToString(result);
492         lbl->Text = System.Convert.ToString(result);
493         txtDisplay->Text = "0";
494         lastop = '-';
    }
```

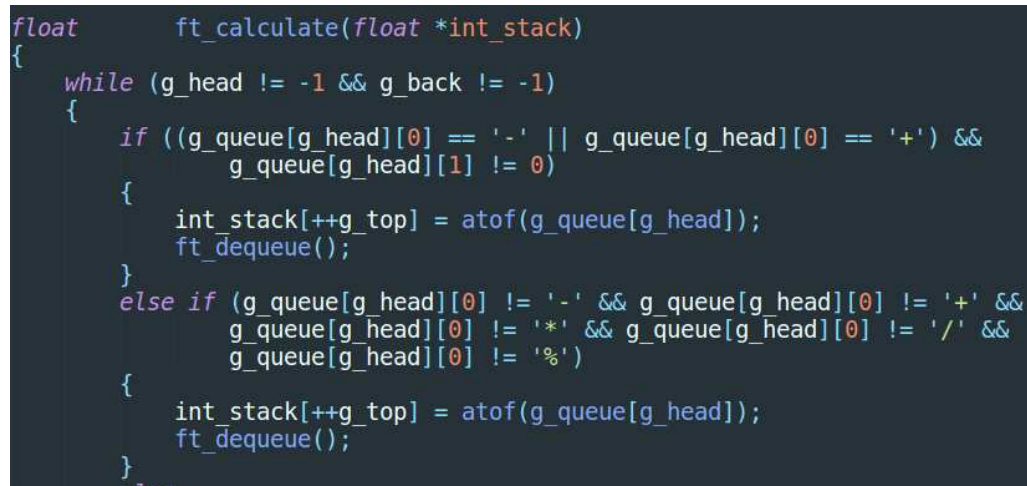
Calculator #2

Partea grafica



```
444 private: System::Void paran1_Click(System::Object^ sender, System::EventArgs^ e) {
445     txtDisplay->Text = txtDisplay->Text + "(";
446 }
447 private: System::Void paran2_Click(System::Object^ sender, System::EventArgs^ e) {
448     txtDisplay->Text = txtDisplay->Text + ")";
449 }
450 private: System::Void bdot_Click(System::Object^ sender, System::EventArgs^ e) {
451     txtDisplay->Text = txtDisplay->Text + ".";
452 }
453 private: System::Void beq_Click(System::Object^ sender, System::EventArgs^ e) {
454     char* str2 = (char*)(void*)Marshal::StringToGlobalAnsi(txtDisplay->Text);
455     result = eval_expr(str2);
456     txtDisplay->Text = System::Convert::ToString(result);
457 }
458 private: System::Void b0_Click(System::Object^ sender, System::EventArgs^ e) {
459     txtDisplay->Text = txtDisplay->Text + "0";
460 }
461 };
462 }
```

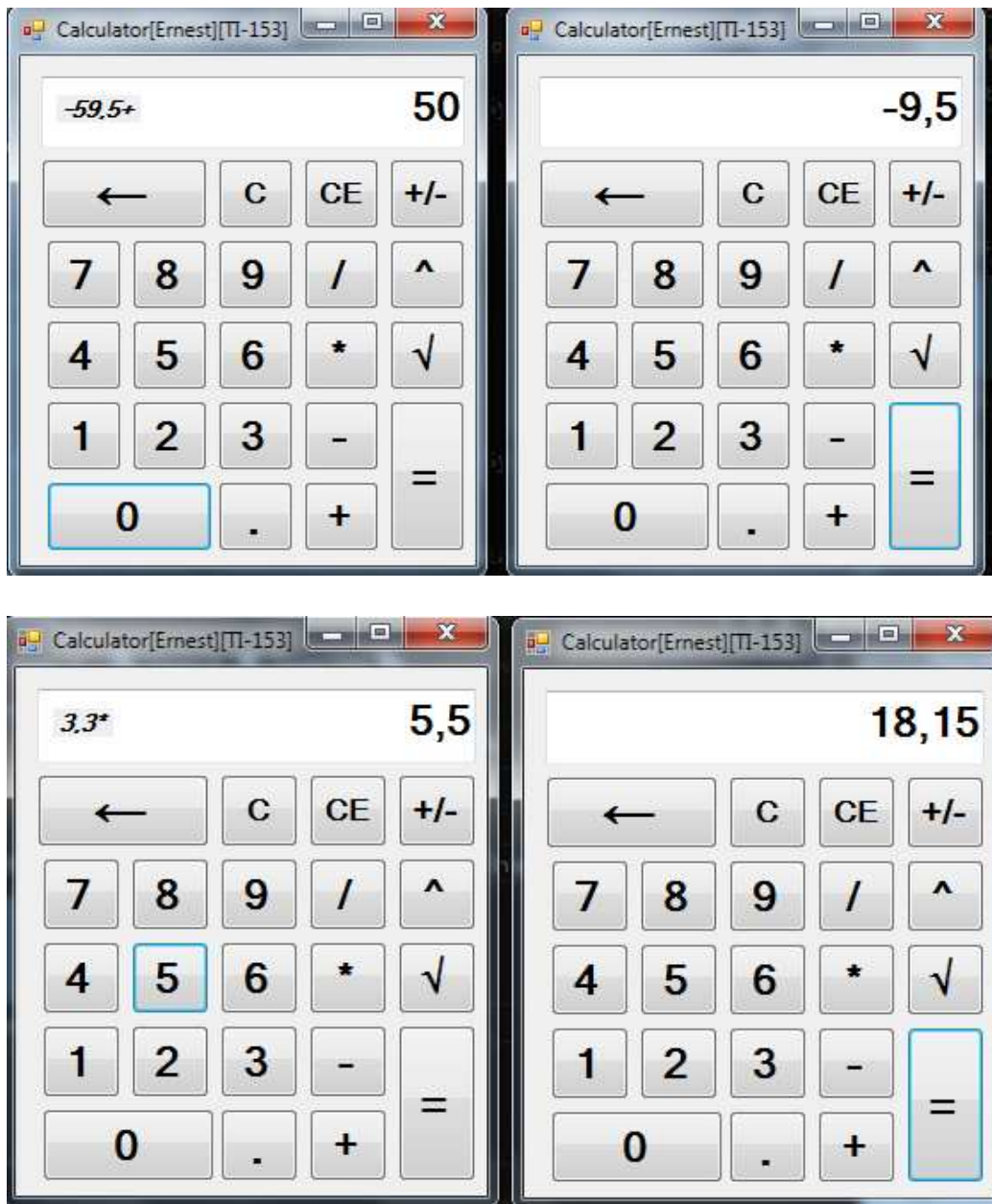
Partea functionala



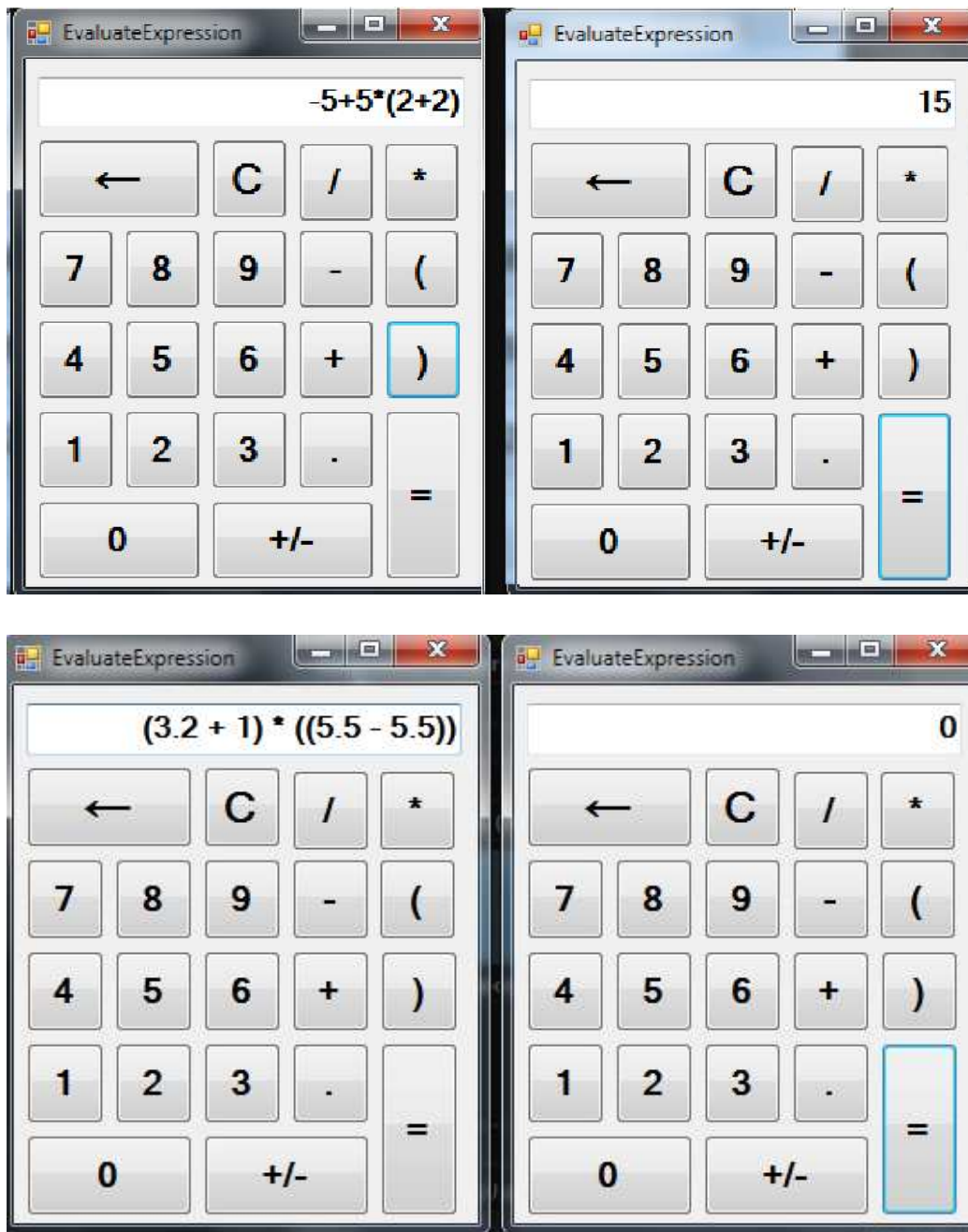
```
float ft_calculate(float *int_stack)
{
    while (g_head != -1 && g_back != -1)
    {
        if ((g_queue[g_head][0] == '-' || g_queue[g_head][0] == '+') &&
            g_queue[g_head][1] != 0)
        {
            int_stack[++g_top] = atof(g_queue[g_head]);
            ft_dequeue();
        }
        else if (g_queue[g_head][0] != '-' && g_queue[g_head][0] != '+' &&
            g_queue[g_head][0] != '*' && g_queue[g_head][0] != '/' &&
            g_queue[g_head][0] != '%')
        {
            int_stack[++g_top] = atof(g_queue[g_head]);
            ft_dequeue();
        }
    }
}
```

3 Screenshoturi

Calculator #1



Calculator #2



Concluzie

În lucrarea dată am creat 2 versiuni de calculatoare. Fiecare calculator având specificul său și construcția sa. Am învățat să ”împachetăm” programele noastre într-un mod mai prielnic unui User. Astfel putem crea aplicații User friendly. Spre deosebire de primul calculator, al doilea are separat funcționalitatea sa în alte fișiere și algoritmul de rezolvare rămâne același. În urma efectuării lucrării mi-am îmbogățit cunoștințele în ceea ce privește crearea aplicațiilor GUI. Acum putem să nu folosim mereu consola dar să comunicăm cu programul nostru printr-o formă mai plăcută și mai familiară unui simplu utilizator. Drept IDE am ales **Visual Studio** din cauza că am găsit mai multă informație despre el. Am utilizat limbajul **C++** iar partea funcțională este scrisă în **C**. Un avantaj foarte mare a unui mediu interactiv este faptul că ne permite să cream un lucru într-un timp foarte scurt și totodată să evităm multe greșeli. După părerea mea lucrarea dată este foarte importantă pentru un student TI, personal am acumulat multe cunoștințe noi în urma efectuării acesteia.