Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

**RAPORT**

Lucrare de laborator Nr.3

*Tema:*  Realizeaza un simplu GUI Calculator

A efectuat: st. gr. TI-141

Bulat Alexandru

A verificat: Cojanu Irina

Chişinău 2016

*Tema:*  Realizeaza un simplu GUI Calculator  
 **Scopul lucrării:** de realizat un calculator cu interfata  
 **Sarcina lucrării:**

* Realizeaza un simplu GUI Calculator
* Operatiile simple: +,-,\*,/,putere,radical,InversareSemn(+/-),operatii cu numere zecimale.
* Divizare proiectului in doua module - Interfata grafica(Modul GUI) si Modulul de baza(Core Module).

**Codul sursa:  
  
#include <windows.h>  
#include <strstream>  
LRESULT CALLBACK WindowProcedure(HWND, UINT, WPARAM, LPARAM);  
inline int SetDefaultFont(int identifier, HWND hwnd);  
HINSTANCE hInst;  
HWND btn\_hwnd[10];  
HWND btn\_hwnd\_dot,  
 btn\_hwnd\_plus,  
 btn\_hwnd\_minus,  
 btn\_hwnd\_times,  
 btn\_hwnd\_over,  
 btn\_hwnd\_equals;  
HWND btn\_hwnd\_ce;  
HWND lbl\_hwnd;  
const char ClassName[ ] = "Win32 Calculator";  
const int btn\_id\_0 = 0,  
 btn\_id\_1 = 1,  
 btn\_id\_2 = 2,  
 btn\_id\_3 = 3,  
 btn\_id\_4 = 4,  
 btn\_id\_5 = 5,  
 btn\_id\_6 = 6,  
 btn\_id\_7 = 7,  
 btn\_id\_8 = 8,  
 btn\_id\_9 = 9;  
const int btn\_id\_dot = 10,  
 btn\_id\_plus = 11,  
 btn\_id\_minus = 12,  
 btn\_id\_times = 13,  
 btn\_id\_over = 14,  
 btn\_id\_equals = 15;**

**const int btn\_id\_ce = 16;  
const int lbl\_id = 17;  
int APIENTRY WinMain(HINSTANCE hInstance,  
 HINSTANCE ,  
 LPSTR ,  
 int nCmdShow)  
{  
 HWND hwnd;  
 MSG messages;  
 WNDCLASSEX wincl;  
 hInst = hInstance;  
 wincl.hInstance = hInst;  
 wincl.lpszClassName = ClassName;  
 wincl.lpfnWndProc = WindowProcedure;  
 wincl.style = CS\_VREDRAW | CS\_HREDRAW;  
 wincl.cbSize = sizeof(WNDCLASSEX);  
 wincl.hIcon = LoadIcon(NULL, IDI\_APPLICATION);  
 wincl.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);  
 wincl.hCursor = LoadCursor(NULL, IDC\_ARROW);  
 wincl.lpszMenuName = NULL;  
 wincl.cbClsExtra = 0;  
 wincl.cbWndExtra = 0;**

**wincl.hbrBackground = (HBRUSH) COLOR\_BACKGROUND;  
 if(!RegisterClassEx(&wincl))  
 {  
 MessageBox(NULL,"Nu este ingregistrata clasa ferestrei","ERROR",MB\_OK | MB\_ICONERROR);  
 return 0;  
 }  
 hwnd = CreateWindowEx(0, ClassName,  
"Win32 Calculator",  
 WS\_OVERLAPPED | WS\_CAPTION | WS\_SYSMENU,  
 CW\_USEDEFAULT,  
 CW\_USEDEFAULT,  
 180,  
 180,  
 HWND\_DESKTOP,  
 NULL,  
 hInst,  
 NULL);  
 ShowWindow(hwnd,nCmdShow);  
 while(GetMessage(&messages,NULL,0,0))  
 {  
 TranslateMessage(&messages);  
 DispatchMessage(&messages);  
 }**

**return messages.wParam;  
}  
LRESULT CALLBACK WindowProcedure(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  
{  
 std::strstream charbuffer;  
 int i = 0;  
 char lblText[256];  
 static bool dot;  
 static double operand1;  
 static double operand2;  
 double result;  
 static char Operator;  
 switch(message)  
 {  
 case WM\_CREATE:  
 lbl\_hwnd = CreateWindowEx(WS\_EX\_CLIENTEDGE | WS\_EX\_RIGHT,"Static","0",WS\_VISIBLE | WS\_CHILD,10,10,150,20,hwnd,(HMENU) lbl\_id ,hInst,NULL);  
 btn\_hwnd[7] = CreateWindow("Button","7",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,10,40,20,20,hwnd,(HMENU) btn\_id\_7,hInst,NULL);  
 btn\_hwnd[8] = CreateWindow("Button","8",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,30,40,20,20,hwnd,(HMENU) btn\_id\_8,hInst,NULL);  
 btn\_hwnd[9] = CreateWindow("Button","9",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,50,40,20,20,hwnd,(HMENU) btn\_id\_9,hInst,NULL);  
 btn\_hwnd[4] = CreateWindow("Button","4",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,10,60,20,20,hwnd,(HMENU) btn\_id\_4,hInst,NULL);  
 btn\_hwnd[5] = CreateWindow("Button","5",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,30,60,20,20,hwnd,(HMENU) btn\_id\_5,hInst,NULL);  
 btn\_hwnd[6] = CreateWindow("Button","6",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,50,60,20,20,hwnd,(HMENU) btn\_id\_6,hInst,NULL);  
 btn\_hwnd[1] = CreateWindow("Button","1",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,10,80,20,20,hwnd,(HMENU) btn\_id\_1,hInst,NULL);  
 btn\_hwnd[2] = CreateWindow("Button","2",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,30,80,20,20,hwnd,(HMENU) btn\_id\_2,hInst,NULL);  
 btn\_hwnd[3] = CreateWindow("Button","3",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,50,80,20,20,hwnd,(HMENU) btn\_id\_3,hInst,NULL);  
 btn\_hwnd[0] = CreateWindow("Button","0",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,30,100,20,20,hwnd,(HMENU) btn\_id\_0,hInst,NULL);  
 btn\_hwnd\_dot = CreateWindow("Button",".",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,50,100,20,20,hwnd,(HMENU) btn\_id\_dot,hInst,NULL);**

**btn\_hwnd\_plus = CreateWindow("Button","+",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,80,40,20,20,hwnd,(HMENU) btn\_id\_plus,hInst,NULL); btn\_hwnd\_minus = CreateWindow("Button","-",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,100,40,20,20,hwnd,(HMENU) btn\_id\_minus,hInst,NULL);  
btn\_hwnd\_times = CreateWindow("Button","x",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,80,60,20,20,hwnd,(HMENU) btn\_id\_times,hInst,NULL);**

**btn\_hwnd\_over = CreateWindow("Button",":",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,100,60,20,20,hwnd,(HMENU) btn\_id\_over,hInst,NULL); btn\_hwnd\_equals = CreateWindow("Button","=",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,100,80,20,20,hwnd,(HMENU) btn\_id\_equals,hInst,NULL); btn\_hwnd\_ce = CreateWindow("Button","CE",WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON,80,80,20,20,hwnd,(HMENU) btn\_id\_ce,hInst,NULL);**

**for(i =0; i <= 17; i++) SetDefaultFont(i,hwnd);  
break;  
case WM\_COMMAND:  
 if(HIWORD(wParam) == BN\_CLICKED)  
{  
switch(LOWORD(wParam))  
{  
case btn\_id\_ce:  
operand1 = operand2 = result = 0.0;  
Operator = ' ';  
dot = false;  
SetWindowText(lbl\_hwnd,"0");  
break;  
case btn\_id\_plus:  
GetWindowText(lbl\_hwnd,lblText,256);  
charbuffer << lblText;  
charbuffer >> operand1;  
Operator = '+';  
dot = false;  
SetWindowText(lbl\_hwnd,"0");  
break;  
case btn\_id\_minus:  
GetWindowText(lbl\_hwnd,lblText,256);  
charbuffer << lblText;  
charbuffer >> operand1;  
Operator = '-';  
dot = false;  
SetWindowText(lbl\_hwnd,"0");  
break;  
case btn\_id\_times:  
GetWindowText(lbl\_hwnd,lblText,256);  
charbuffer << lblText;  
charbuffer >> operand1;  
Operator = '\*';  
dot = false;  
SetWindowText(lbl\_hwnd,"0");  
break;  
case btn\_id\_over:  
GetWindowText(lbl\_hwnd,lblText,256);  
charbuffer << lblText;  
charbuffer >> operand1;  
Operator = '/';  
dot = false;  
SetWindowText(lbl\_hwnd,"0");**

**break;**

**case btn\_id\_0:  
GetWindowText(lbl\_hwnd,lblText,256);  
if((lblText[0] == '0') && (!dot)) return 0;  
else strcat(lblText,"0");  
SetWindowText(lbl\_hwnd,lblText);  
break;  
case btn\_id\_1:  
GetWindowText(lbl\_hwnd,lblText,256);  
if((lblText[0] == '0') && (!dot)) strcpy(lblText,"1");  
else strcat(lblText,"1");  
SetWindowText(lbl\_hwnd,lblText);**

**break;  
case btn\_id\_2:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"2");  
 else strcat(lblText,"2");  
SetWindowText(lbl\_hwnd,lblText);  
break;  
case btn\_id\_3:  
GetWindowText(lbl\_hwnd,lblText,256);  
if((lblText[0] == '0') && (!dot)) strcpy(lblText,"3");  
else strcat(lblText,"3");  
SetWindowText(lbl\_hwnd,lblText);**

**break;  
case btn\_id\_4:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"4");  
 else strcat(lblText,"4");  
 SetWindowText(lbl\_hwnd,lblText);**

**break;  
 case btn\_id\_5:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"5");  
 else strcat(lblText,"5");  
 SetWindowText(lbl\_hwnd,lblText);**

**break;  
 case btn\_id\_6:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"6");  
 else strcat(lblText,"6");  
 SetWindowText(lbl\_hwnd,lblText);**

**break;  
 case btn\_id\_7:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"7");  
 else strcat(lblText,"7");  
 SetWindowText(lbl\_hwnd,lblText);**

**break;**

**case btn\_id\_8:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"8");  
 else strcat(lblText,"8");  
 SetWindowText(lbl\_hwnd,lblText);  
 break;  
 case btn\_id\_9:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 if((lblText[0] == '0') && (!dot)) strcpy(lblText,"9");  
 else strcat(lblText,"9");  
 SetWindowText(lbl\_hwnd,lblText);**

**break;  
case btn\_id\_equals:  
 GetWindowText(lbl\_hwnd,lblText,256);  
 charbuffer << lblText;  
 charbuffer >> operand2;  
 charbuffer.clear();  
 switch(Operator)  
 {**

**case '+':  
 result = operand1 + operand2;  
break;  
 case '-':  
 result = operand1 - operand2;  
 break;  
 case '\*':  
 result = operand1 \* operand2;**

**break;**

**case '/':  
 if(operand2 != 0) result = operand1 / operand2;**

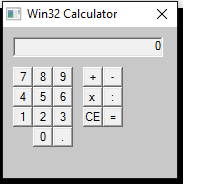
**else  
 {  
 operand1 = operand2 = result = 0.0;  
 Operator = ' ';  
 dot = false;  
 SetWindowText(lbl\_hwnd,"Nu se devide la 0");  
 return 0;  
 }  
default:  
 operand1 = operand2 = result = 0.0;  
 Operator = ' ';  
 dot = false;  
 SetWindowText(lbl\_hwnd,"Trebuie sa specificati peratorul");  
 return 0;  
 break;  
 }  
 charbuffer << result;  
 charbuffer.getline(lblText,256);  
 SetWindowText(lbl\_hwnd,lblText);  
 operand1 = operand2 = result = 0.0;  
 Operator = ' ';  
 dot = false;  
break;  
 case btn\_id\_dot:  
 if(!dot)  
 {  
 GetWindowText(lbl\_hwnd,lblText,256);  
 strcat(lblText,".");  
 SetWindowText(lbl\_hwnd,lblText);  
 dot = true;  
 }**

**break;  
 }  
 }**

**break;  
case WM\_DESTROY:  
 PostQuitMessage(0);**

**break;  
 default:  
 return DefWindowProc(hwnd,message,wParam,lParam);  
 break;  
}  
return 0;  
 }  
inline int SetDefaultFont(int identifier, HWND hwnd)  
{  
 SendDlgItemMessage(  
 hwnd,  
 identifier,  
 WM\_SETFONT,  
 (WPARAM)GetStockObject(DEFAULT\_GUI\_FONT),  
 MAKELPARAM(true, 0));  
 return 0;  
}**

**Screen capture:**



**Concluzie**: Datorita acestei lucrari am obtinut noi cunostinte si am vazut noi posibilitati de a lucre cu ajutorul programarii pirotate de avenimente saau WINAPI. Am efectuat un calculator care are functia de a face mai multe operatii matematice cum sunt adunarea,scaderea,inmultirea si impartirea. Am facut o interfta grafica cu utilizatorul care ne permite mai usor de a comunica cu codul sursa. Realizind aceasta lucrare mi-am aprofundat cunostintele in windows programing.

**Bibliografie**:  
Lectii video   
Indicatii metodice <http://moodle.ati.utm.md/mod/assign/view.php?id=987>

[Charles Petzold. Programming Windows Fifth Edition](http://elearning.utm.md/moodle/mod/resource/view.php?id=23232)