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;
            }
```

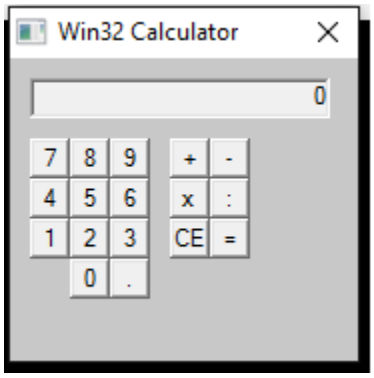
```
default:
        operand1 = operand2 = result = 0.0;
        Operator = ' ';
        dot = false;
        SetWindowText(lbl_hwnd,"Trebuie sa specificati peratorul");
        return 0;
        break;
        }
        charbuffer << result;
        charbuffer.getline(lblText,256);
        SetWindowText(lbl_hwnd,lblText);
        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