Temă laborator 1 – Sisteme de Operare

#undef UNICODE

#include <windows.h>

#include <stdio.h> // adaugarea librariei pentru functia de string, adica sprintf\_s

#include <random> // pentru generare de numere

#include <ctime> // pentru generare de numere

#include "resource.h"

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

BOOL CALLBACK DlgProc(HWND hDlg, UINT iMsg, WPARAM wParam, LPARAM lParam);

BOOL dlgActive = FALSE;

BOOL isSelectedRadioButton = FALSE; // daca s-a selectat vreun radioButton

HWND hwndMain;

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,

PSTR szCmdLine, int iCmdShow)

{

static char szAppName[] = "Dialog";

HWND hwnd;

MSG msg;

WNDCLASSEX wndclass;

wndclass.cbSize = sizeof(wndclass);

wndclass.style = CS\_HREDRAW | CS\_VREDRAW;

wndclass.lpfnWndProc = WndProc;

wndclass.cbClsExtra = 0;

wndclass.cbWndExtra = 0;

wndclass.hInstance = hInstance;

wndclass.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wndclass.hCursor = LoadCursor(NULL, IDC\_ARROW);

wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);

wndclass.lpszMenuName = NULL;

wndclass.lpszClassName = szAppName;

wndclass.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);

RegisterClassEx(&wndclass);

hwnd = CreateWindow(szAppName,

"Convertor valutar",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

NULL,

NULL,

hInstance,

NULL);

SetWindowPos(hwnd, HWND\_BOTTOM, 0, 0, 0, 0, SWP\_NOACTIVATE);

ShowWindow(hwnd, SW\_HIDE);

UpdateWindow(hwnd);

hwndMain = hwnd;

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)

{

static HINSTANCE hInstance;

switch (iMsg)

{

case WM\_CREATE:

hInstance = ((LPCREATESTRUCT)lParam)->hInstance;

if (!dlgActive) {

DialogBox(hInstance, MAKEINTRESOURCE(IDD\_DIALOG), hwnd, DlgProc);

PostMessage(hwnd, WM\_CLOSE, 0, 0);

}

return 0;

case WM\_DESTROY:

PostQuitMessage(0);

return 0;

}

return DefWindowProc(hwnd, iMsg, wParam, lParam);

}

BOOL CALLBACK DlgProc(HWND hDlg, UINT iMsg, WPARAM wParam, LPARAM lParam) {

char eur\_cursText[11], usd\_cursText[11], s[11], str1[11], str2[11], str3[11]; // bufferele pentru fiecare curs, cat si pentru fiecare rezultat

float eur\_curs, usd\_curs, low, high, ron, eur, usd;

switch (iMsg) {

case WM\_INITDIALOG:

SetDlgItemText(hDlg, IDC\_SUMA, "0"); // initializam campul sumei cu 0

SetDlgItemText(hDlg, IDC\_FINAL\_RON, "0"); // initializam campul ron(final) cu 0

SetDlgItemText(hDlg, IDC\_FINAL\_EUR, "0"); // initializam campul eur(final) cu 0

SetDlgItemText(hDlg, IDC\_FINAL\_USD, "0"); // initializam campul usd(final) cu 0

low = 3;

high = 5.50;

// intervalul de randomizare

srand(time(NULL)); // o randomizare mai corecta, imprumutata din documentatie (cpp docs)

eur\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // generam primul curs euro-ron

usd\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // generam primul curs usd-ron

while (eur\_curs < usd\_curs) { // daca euro < usd atunci se randomizeaza iar

eur\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // se regenereaza alt curs euro-ron

usd\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // se regenereaza alt curs usd-ron

}

sprintf\_s(eur\_cursText, "%.2f", eur\_curs); // buffer

SetDlgItemText(hDlg, IDC\_CURS\_EUR, eur\_cursText); // afisam cursul euro-ron

sprintf\_s(usd\_cursText, "%.2f", usd\_curs); // buffer

SetDlgItemText(hDlg, IDC\_CURS\_USD, usd\_cursText); // afisam cursul usd-ron

return TRUE;

case WM\_CLOSE:

dlgActive = FALSE;

EndDialog(hDlg, 0);

return TRUE;

case WM\_COMMAND:

switch (LOWORD(wParam)) {

case ID\_CURS:

low = 3;

high = 5.50;

// intervalul de randomizare

srand(time(NULL)); // o randomizare mai corecta, imprumutata din documentatie (cpp docs)

eur\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // generam primul curs euro-ron

usd\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // generam primul curs usd-ron

while (eur\_curs < usd\_curs) { // daca euro < usd atunci se randomizeaza iar

eur\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // se regenereaza alt curs euro-ron

usd\_curs = low + static\_cast<float>(rand()) \* static\_cast<float>(high - low) / RAND\_MAX; // se regenereaza alt curs usd-ron

}

sprintf\_s(eur\_cursText, "%.2f", eur\_curs); // buffer

SetDlgItemText(hDlg, IDC\_CURS\_EUR, eur\_cursText); // afisam cursul euro-ron

sprintf\_s(usd\_cursText, "%.2f", usd\_curs); // buffer

SetDlgItemText(hDlg, IDC\_CURS\_USD, usd\_cursText); // afisam cursul usd-ron

return TRUE;

case ID\_CONVERT:

if ((!GetDlgItemText(hDlg, IDC\_CURS\_EUR, eur\_cursText, 11))) { // daca este vreo valoare in campul cursului euro-ron(se stie ca GetDlgItemText returneaza o valoare TRUE daca nu este gol si 0 daca este gol)

MessageBox(hDlg, "Nu ati generat un curs valutar!", "Eroare", MB\_OK | MB\_ICONSTOP);

return FALSE;

}

int selectedRadioButton; // ce denomiare s-a selectat(RON/EUR/USD)

selectedRadioButton = -1; // initializam cu -1

if (IsDlgButtonChecked(hDlg, IDC\_SELECT\_RON)) {

isSelectedRadioButton = TRUE;

selectedRadioButton = 1; // s-a selectat RON

}

if (IsDlgButtonChecked(hDlg, IDC\_SELECT\_EUR)) {

isSelectedRadioButton = TRUE;

selectedRadioButton = 2; // s-a selectat EUR

}

if (IsDlgButtonChecked(hDlg, IDC\_SELECT\_USD)) {

isSelectedRadioButton = TRUE;

selectedRadioButton = 3; // s-a selectat USD

}

if (!isSelectedRadioButton) { // daca nu s-a selectat niciuna, atunci dam eroare

MessageBox(hDlg, "Nu ati selectat denumirea monedei 'suma'!", "Eroare", MB\_OK | MB\_ICONSTOP);

return FALSE;

}

if (!GetDlgItemText(hDlg, IDC\_SUMA, s, 11)) { // daca nu este nicio suma pusa

MessageBox(hDlg, "Nu ati introdus suma!", "Eroare", MB\_OK | MB\_ICONSTOP);

return FALSE;

}

float suma;

suma = atof(s); // convertim din string in float/double

GetDlgItemText(hDlg, IDC\_CURS\_EUR, eur\_cursText, 11); // luam cursul euro-ron

GetDlgItemText(hDlg, IDC\_CURS\_USD, usd\_cursText, 11); // luam cursul usd-ron

eur\_curs = atof(eur\_cursText); // convertim din string in float/double

usd\_curs = atof(usd\_cursText); // convertim din string in float/double

if (selectedRadioButton == 1) { // RON selectat

ron = suma;

eur = suma / eur\_curs;

usd = suma / usd\_curs;

}

else if (selectedRadioButton == 2) { // EUR selectat

ron = suma \* eur\_curs;

eur = suma;

usd = suma \* (eur\_curs / usd\_curs);

}

else if (selectedRadioButton == 3) { // USD selectat

ron = suma \* usd\_curs;

eur = suma \* (usd\_curs / eur\_curs);

usd = suma;

}

sprintf\_s(str1, "%.2f", ron); // buffer

sprintf\_s(str2, "%.2f", eur); // buffer

sprintf\_s(str3, "%.2f", usd); // buffer

SetDlgItemText(hDlg, IDC\_FINAL\_RON, str1); // afisam RON final

SetDlgItemText(hDlg, IDC\_FINAL\_EUR, str2); // afisam EUR final

SetDlgItemText(hDlg, IDC\_FINAL\_USD, str3); // afisam USD final

return TRUE;

case ID\_CANCEL:

dlgActive = FALSE;

EndDialog(hDlg, 0);

return TRUE;

}

break;

}

return FALSE;

}

|  |  |
| --- | --- |
| **Convertire #1** | **Convertire #2** |
|  |  |
| **Convertire #3** |
|  |

|  |  |
| --- | --- |
| **Eroare #1** | **Eroare #2** |
|  |  |
| **Eroare #3** |
|  |