**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.4-5

Disciplina: Programarea pilotată de evenimente.

Tema: Ceasul sistemului Windows. Ferestre descendent.

**Efectuat**: st.gr. TI-207 Bunescu Gabriel.

**Verificat**: asist. univ. Gaidarji Alina

Chișinău 2022

**Scopul lucrării:**

* studierea metodelor și principiilor de lucru cu timer-ul;
* studierea metodelor și principiilor de lucru cu ferestrele descendent;
* studierea principiilor de prelucrare a mesajelor parvenite de la ferestrele descendent.

**Sarcina:**

Scrieţi un program care afişează un test, utilizând radio butoane, check boxes, butoane push și pentru cronometrare de folosit timer-ul.

**Codul sursă:**

#include "framework.h"

#include "PPE.h"

#define MAX\_LOADSTRING 100

HINSTANCE hInst;

WCHAR szWindowClass[MAX\_LOADSTRING];

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

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

HWND mHwnd;

RECT clientRect;

static HWND hBtn, hNext, hPrev, hFinish, hAgain, radiob1, radiob2, radiob3, check1, check2, check3, check4, check5, check6;

static HWND hEdt1, hEdt2, text1, text2, text3, textA, textP, timer;

static HWND hNota, hTrue, hFalse;

int it, punct = 0;

char StrA[20];

WORD IDC\_CHECKBOX\_CTRL\_ID = 1, IDC\_CHECKBOX\_CTRL\_ID2 = 2, IDC\_RADIOBUT = 3;

const UINT\_PTR IDT\_TIMER1 = 1;

int a, b, Len, notaint, timeleft = 60;

wchar\_t notastr[20], timestr[20];

bool check11, check21, check22;

int StrToInt(char\* s)

{

int temp = 0;

int i = 0;

int sign = 0;

if (s[i] == '-')

{

sign = 1;

i++;

}

while (s[i] >= 0x30 && s[i] <= 0x39)

{

temp = temp + (s[i] & 0x0F);

temp = temp \* 10;

i++;

}

temp = temp / 10;

if (sign == 1)

temp = -temp;

return(temp);

}

void CalcNota() {

GetWindowText(hEdt1, (LPWSTR)&StrA[0], 5);

a = StrToInt(StrA);

GetWindowText(hEdt2, (LPWSTR)&StrA[0], 5);

b = StrToInt(StrA);

if (a == 8) punct++;

if (b == 9) punct++;

if (check11) punct++;

if (check21) punct++;

if (check22) punct++;

notaint = punct \* 2;

wsprintf(notastr, \_T("%d"), notaint);

SetWindowText(hTrue, notastr);

}

void show() {

if (it == 0) {

ShowWindow(hAgain, SW\_HIDE);

ShowWindow(hBtn, SW\_SHOWNORMAL);

ShowWindow(text1, SW\_HIDE);

ShowWindow(text2, SW\_HIDE);

ShowWindow(text3, SW\_HIDE);

ShowWindow(hNota, SW\_HIDE);

ShowWindow(hTrue, SW\_HIDE);

check11 = false;

check21 = false;

check22 = false;

a = 0;

b = 0;

punct = 0;

SendMessage(check1, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(check2, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(check3, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(check4, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(check5, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(check6, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(radiob1, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(radiob2, BM\_SETCHECK, BST\_UNCHECKED, 0);

SendMessage(radiob3, BM\_SETCHECK, BST\_UNCHECKED, 0);

SetWindowText(hEdt1, L"0");

SetWindowText(hEdt2, L"0");

SetWindowText(timer, L"Timpul: 60 secunde.");

}

if (it == 1) {

timeleft = 60;

SetTimer(mHwnd,

IDT\_TIMER1,

1000,

(TIMERPROC)NULL);

ShowWindow(hBtn, SW\_HIDE);

ShowWindow(hPrev, SW\_HIDE);

ShowWindow(text2, SW\_HIDE);

ShowWindow(check1, SW\_HIDE);

ShowWindow(check2, SW\_HIDE);

ShowWindow(check3, SW\_HIDE);

ShowWindow(check4, SW\_HIDE);

ShowWindow(check5, SW\_HIDE);

ShowWindow(check6, SW\_HIDE);

ShowWindow(text1, SW\_SHOWNORMAL);

ShowWindow(hNext, SW\_SHOWNORMAL);

ShowWindow(radiob1, SW\_SHOWNORMAL);

ShowWindow(radiob2, SW\_SHOWNORMAL);

ShowWindow(radiob3, SW\_SHOWNORMAL);

}

if (it == 2) {

ShowWindow(hPrev, SW\_SHOWNORMAL);

ShowWindow(hNext, SW\_SHOWNORMAL);

ShowWindow(text1, SW\_HIDE);

ShowWindow(hFinish, SW\_HIDE);

ShowWindow(radiob1, SW\_HIDE);

ShowWindow(radiob2, SW\_HIDE);

ShowWindow(radiob3, SW\_HIDE);

ShowWindow(text3, SW\_HIDE);

ShowWindow(hEdt1, SW\_HIDE);

ShowWindow(hEdt2, SW\_HIDE);

ShowWindow(textA, SW\_HIDE);

ShowWindow(textP, SW\_HIDE);

ShowWindow(text2, SW\_SHOWNORMAL);

ShowWindow(check1, SW\_SHOWNORMAL);

ShowWindow(check2, SW\_SHOWNORMAL);

ShowWindow(check3, SW\_SHOWNORMAL);

ShowWindow(check4, SW\_SHOWNORMAL);

ShowWindow(check5, SW\_SHOWNORMAL);

ShowWindow(check6, SW\_SHOWNORMAL);

}

if (it == 3) {

ShowWindow(hNext, SW\_HIDE);

ShowWindow(hFinish, SW\_SHOWNORMAL);

ShowWindow(text2, SW\_HIDE);

ShowWindow(check1, SW\_HIDE);

ShowWindow(check2, SW\_HIDE);

ShowWindow(check3, SW\_HIDE);

ShowWindow(check4, SW\_HIDE);

ShowWindow(check5, SW\_HIDE);

ShowWindow(check6, SW\_HIDE);

ShowWindow(text3, SW\_SHOWNORMAL);

ShowWindow(hEdt1, SW\_SHOWNORMAL);

ShowWindow(hEdt2, SW\_SHOWNORMAL);

ShowWindow(textA, SW\_SHOWNORMAL);

ShowWindow(textP, SW\_SHOWNORMAL);

}

if (it == 4) {

ShowWindow(text2, SW\_HIDE);

ShowWindow(check1, SW\_HIDE);

ShowWindow(check2, SW\_HIDE);

ShowWindow(check3, SW\_HIDE);

ShowWindow(check4, SW\_HIDE);

ShowWindow(check5, SW\_HIDE);

ShowWindow(check6, SW\_HIDE);

ShowWindow(text1, SW\_HIDE);

ShowWindow(hFinish, SW\_HIDE);

ShowWindow(radiob1, SW\_HIDE);

ShowWindow(radiob2, SW\_HIDE);

ShowWindow(radiob3, SW\_HIDE);

CalcNota();

ShowWindow(hFinish, SW\_HIDE);

ShowWindow(hNext, SW\_HIDE);

ShowWindow(hPrev, SW\_HIDE);

ShowWindow(hAgain, SW\_SHOWNORMAL);

ShowWindow(hNota, SW\_SHOWNORMAL);

ShowWindow(hTrue, SW\_SHOWNORMAL);

KillTimer(mHwnd, IDT\_TIMER1);

ShowWindow(textA, SW\_HIDE);

ShowWindow(textP, SW\_HIDE);

ShowWindow(text3, SW\_HIDE);

ShowWindow(hEdt1, SW\_HIDE);

ShowWindow(hEdt2, SW\_HIDE);

}

}

int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstance,

\_In\_ LPWSTR lpCmdLine,

\_In\_ int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

LoadStringW(hInstance, IDC\_PPE, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

if (!InitInstance(hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_PPE));

MSG msg;

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

{

if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int)msg.wParam;

}

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEXW wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

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

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDC\_PPE));

wcex.hCursor = LoadCursor(nullptr, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);

wcex.lpszMenuName = 0;

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_SMALL));

return RegisterClassExW(&wcex);

}

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

hInst = hInstance;

HWND hWnd = CreateWindowW(szWindowClass, L" Lab 4-5 Bunescu Gabriel TI-207", WS\_MINIMIZEBOX | WS\_SYSMENU,

CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

SetWindowPos(hWnd, NULL, NULL, NULL, 544, 408, NULL);

if (!hWnd)

{

return FALSE;

}

mHwnd = hWnd;

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

SetTimer(mHwnd, 1000, 1000 / 60, NULL);

return TRUE;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

switch (message)

{

case WM\_CREATE: {

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

hEdt1 = CreateWindow(L"edit", L"0",

WS\_CHILD | WS\_BORDER | ES\_RIGHT, 215, 128, 62, 20,

hWnd, 0, hInst, NULL);

hEdt2 = CreateWindow(L"edit", L"0",

WS\_CHILD | WS\_BORDER | ES\_RIGHT, 215, 201, 62, 20,

hWnd, 0, hInst, NULL);

textA = CreateWindow(L"static", L"2+2^2=:", WS\_CHILD | WS\_BORDER, 140, 128, 70, 20, hWnd, 0, hInst, NULL);

textP = CreateWindow(L"static", L"3^2=:", WS\_CHILD | WS\_BORDER, 140, 201, 70, 20, hWnd, 0, hInst, NULL);

hBtn = CreateWindow(L"button", L"Start",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE,

142, 162, 237, 62, hWnd, 0, hInst, NULL);

hNext = CreateWindow(L"button", L"Next >",

WS\_CHILD | WS\_BORDER,

409, 330, 107, 29, hWnd, 0, hInst, NULL);

hPrev = CreateWindow(L"button", L"< Prev",

WS\_CHILD | WS\_BORDER,

10, 330, 107, 29, hWnd, 0, hInst, NULL);

hFinish = CreateWindow(L"button", L"FINISH",

WS\_CHILD | WS\_BORDER,

397, 321, 119, 36, hWnd, 0, hInst, NULL);

hAgain = CreateWindow(L"button", L"Again?",

WS\_CHILD | WS\_BORDER,

189, 267, 118, 38, hWnd, 0, hInst, NULL);

radiob1 = CreateWindow(L"button", L"FUA",

WS\_CHILD | BS\_AUTORADIOBUTTON,

40, 162, 54, 33, hWnd, 0, hInst, NULL);

radiob2 = CreateWindow(L"button", L"FEIE",

WS\_CHILD | BS\_AUTORADIOBUTTON,

223, 162, 54, 33, hWnd, 0, hInst, NULL);

radiob3 = CreateWindow(L"button", L"FCIM",

WS\_CHILD | BS\_AUTORADIOBUTTON,

422, 162, 54, 33, hWnd, (HMENU)IDC\_RADIOBUT, hInst, NULL);

check1 = CreateWindow(L"button", L"Nr. 12",

WS\_CHILD | BS\_AUTOCHECKBOX,

39, 126, 83, 33, hWnd, 0, hInst, NULL);

check2 = CreateWindow(L"button", L"Nr. 1",

WS\_CHILD | BS\_AUTOCHECKBOX,

223, 126, 111, 33, hWnd, (HMENU)IDC\_CHECKBOX\_CTRL\_ID, hInst, NULL);

check3 = CreateWindow(L"button", L"Nr. 10",

WS\_CHILD | BS\_AUTOCHECKBOX,

422, 126, 71, 33, hWnd, 0, hInst, NULL);

check4 = CreateWindow(L"button", L"Nr. 2",

WS\_CHILD | BS\_AUTOCHECKBOX,

39, 198, 82, 33, hWnd, (HMENU)IDC\_CHECKBOX\_CTRL\_ID2, hInst, NULL);

check5 = CreateWindow(L"button", L"Nr. 11",

WS\_CHILD | BS\_AUTOCHECKBOX,

223, 198, 95, 33, hWnd, 0, hInst, NULL);

check6 = CreateWindow(L"button", L"Nr. 13",

WS\_CHILD | BS\_AUTOCHECKBOX,

422, 198, 86, 33, hWnd, 0, hInst, NULL);

text1 = CreateWindow(L"static", L"Cum se numeste facultatea ", WS\_CHILD | WS\_BORDER, 151, 66, 233, 20, hWnd, 0, hInst, NULL);

text2 = CreateWindow(L"static", L"Care camine au 12 etaje?:", WS\_CHILD | WS\_BORDER, 100, 66, 200, 20, hWnd, 0, hInst, NULL);

text3 = CreateWindow(L"static", L"Calculati:", WS\_CHILD | WS\_BORDER, 151, 66, 160, 20, hWnd, 0, hInst, NULL);

hNota = CreateWindow(L"static", L"Nota: ", WS\_CHILD | WS\_BORDER, 151, 66, 70, 20, hWnd, 0, hInst, NULL);

hTrue = CreateWindow(L"static", L"", WS\_CHILD | WS\_BORDER, 250, 66, 50, 20, hWnd, 0, hInst, NULL);

timer = CreateWindow(L"static", L"Timpul: 60 secunde.", WS\_CHILD | WS\_BORDER | WS\_VISIBLE, 10, 8, 141, 22, hWnd, 0, hInst, NULL);

break;

}

case WM\_COMMAND:

if (lParam == (LPARAM)hBtn)

{

it++;

show();

break;

}

if (lParam == (LPARAM)hNext)

{

it++;

show();

break;

}

if (lParam == (LPARAM)hPrev)

{

it--;

show();

break;

}

if (lParam == (LPARAM)hFinish)

{

it++;

show();

break;

}

if (lParam == (LPARAM)hAgain)

{

it = 0;

show();

break;

}

if (LOWORD(wParam) == IDC\_CHECKBOX\_CTRL\_ID)

{

if (HIWORD(wParam) == BN\_CLICKED)

{

LRESULT chkState = SendMessage((HWND)lParam, BM\_GETCHECK, 0, 0);

if (chkState == BST\_CHECKED)

check21 = true;

}

}

if (LOWORD(wParam) == IDC\_CHECKBOX\_CTRL\_ID2)

{

if (HIWORD(wParam) == BN\_CLICKED)

{

LRESULT chkState = SendMessage((HWND)lParam, BM\_GETCHECK, 0, 0);

if (chkState == BST\_CHECKED)

check22 = true;

}

}

if (LOWORD(wParam) == IDC\_RADIOBUT)

{

if (HIWORD(wParam) == BN\_CLICKED)

{

LRESULT chkState = SendMessage((HWND)lParam, BM\_GETCHECK, 0, 0);

if (chkState == BST\_CHECKED)

check11 = true;

}

}

case WM\_TIMER:

{

switch (wParam)

{

case IDT\_TIMER1:

timeleft--;

wsprintf(timestr, \_T("Timpul: %d secunde."), timeleft);

SetWindowText(timer, timestr);

if (timeleft == 0) {

it = 4;

show();

KillTimer(mHwnd, IDT\_TIMER1);

MessageBox(mHwnd, L"TIMPUL A EXPIRAT!", L"LOSE", NULL);

}

}

return 0;

RedrawWindow(hWnd, NULL, NULL, RDW\_INVALIDATE);

}

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

EndPaint(hWnd, &ps);

break;

}

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

return 0;

}

**Rezultatul execuției:**

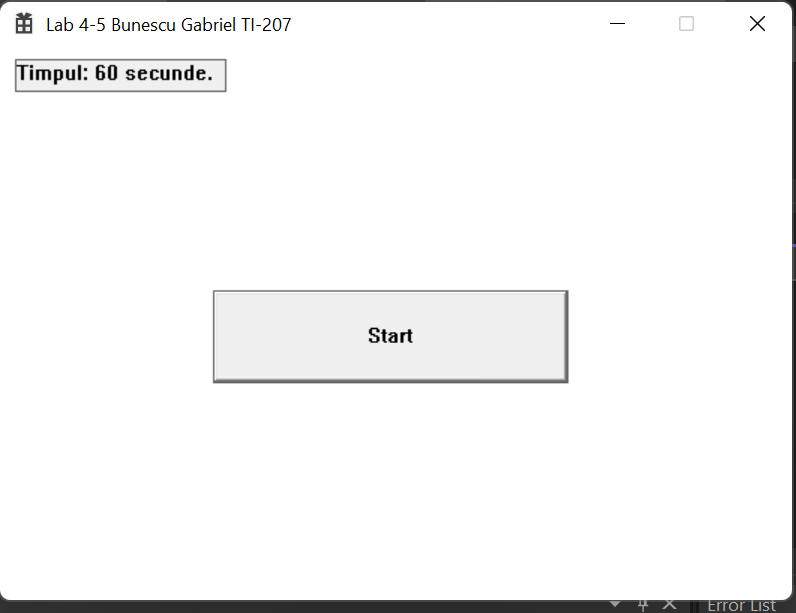


Figura 1. Rezultatul execuției, Start

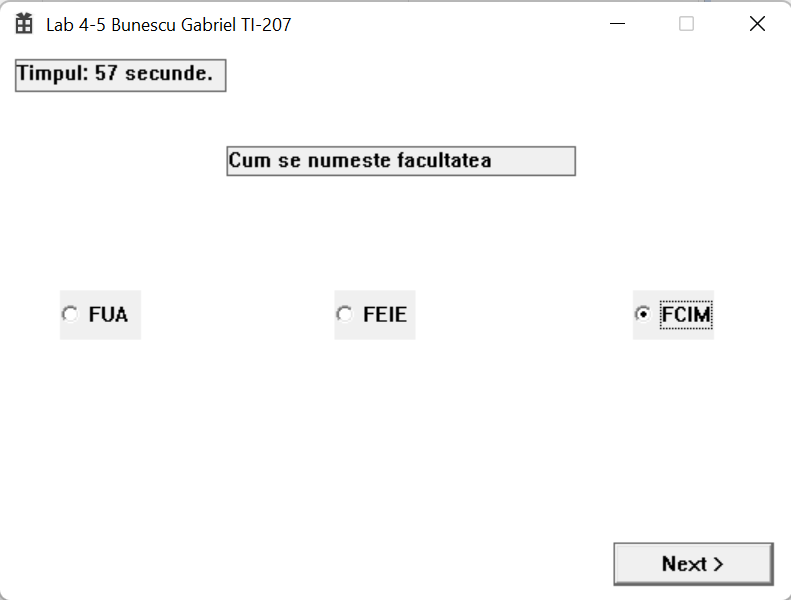


Figura 2. Prima Întrebare.

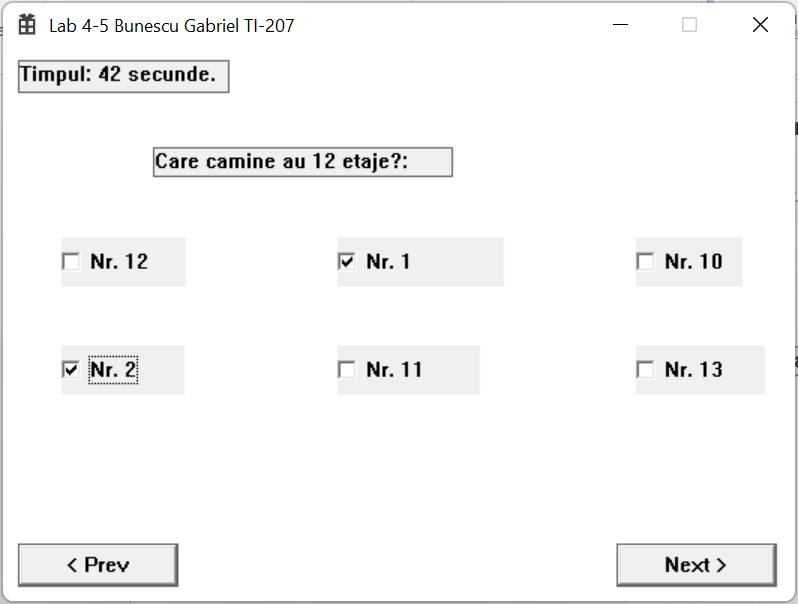


Figura 3. A doua Întrebare.

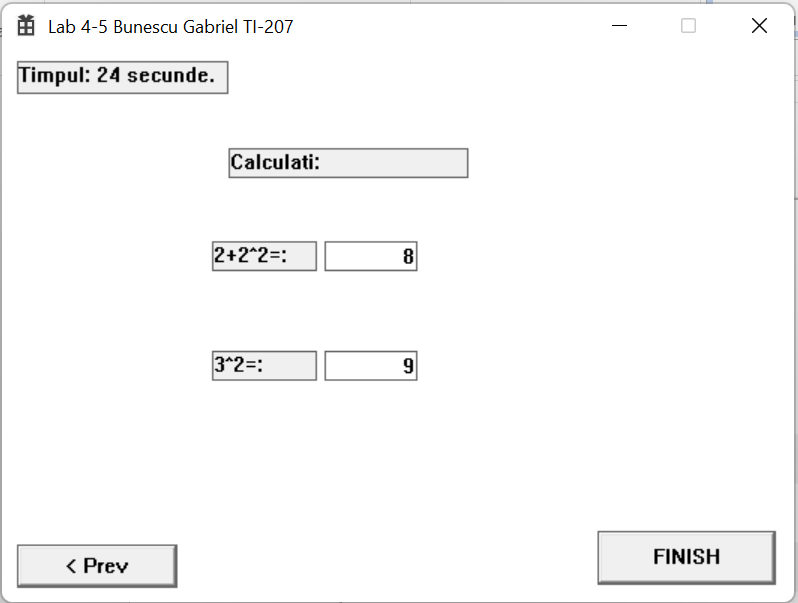


Figura 4. A treia Întrebare.



Figura 5. FINISH

**Concluzie:**

In urma efectuarii acestei lucrari de laborator am studiat bazele şi principiile de lucru cu timer-ul. Am studiat metodele si principiile de lucru cu ferestrele descendent si a mesajelor parvenite de la acestea. De asemenea , in aceasta lucrare am elaborat un program care afiseaza un test cu cateva intrebari , 2 dintre care permit raspunsul utilizand radio butoane, 2 utilizand check boxes si in cele din urma butoanele push. Pentru a cronometra timpul efectuarii testului am folosit functiile timer-ului.