

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Automatica și Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.1

la Medii interactive de dezvoltare a produselor soft

A efectuat:

st. gr. TI-143

Racu Ion

A verificat:

Cojocaru Svetlana

Chisinau 2016

Scopul Lucrării:

- a) Însușirea modului de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER . Realizarea unui program simplu care utilizează componente de tip *TButton*, *TEdit*, *Tlabel*, *RadioButton* etc.
- b) Însușirea modului de utilizare a componentei VCL **TTimer**. Însușirea modului de utilizare a funcțiilor de lucru cu timpul sistem. Realizarea unor aplicații de gestionare a resursei timp.
- c) Însușirea modului de utilizare a componentelor VCL **TPaintBox** și **TPanel**. Însușirea modului de utilizare a principalelor funcții grafice ale mediului C++BUILDER . Realizarea unor elemente pentru afișarea grafică a informației (diagramă și bargraf).

Sarcina Lucrării:

- 1) Vor fi examinate toate componentele prezentate în indicații teoretice;
- 2) Se modifică programul din *Project1.cpp* astfel încât să se obțină forma cu obiecte din figura 4.1 ;

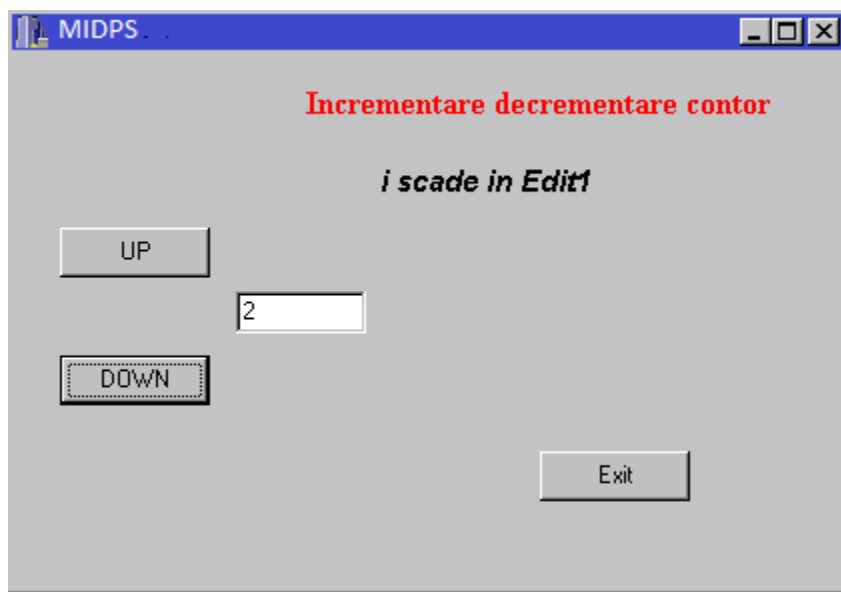


Fig. 4.1 – Realizarea 1

Se vor utiliza următoarele obiecte (în afara formei):

- două butoane (Button 1 și 2) pentru incrementarea (UP) respectiv decrementarea (DOWN) a unei variabile întregi **i** ;
- un buton (Button 3) pentru ieșirea din program (Exit);
- o casetă de editare (Edit1) unde se va afișa valoarea variabilei **i**;
- două etichete (Label1 și 2) pentru afișarea textului „**Incrementare decrementare contor.**” Respectiv a **sensului de variație a variabilei i din caseta Edit1**;
- în caption-ul formei se va afișa textul „ **MIDPS 1- A**”;
- fiecare obiect va avea hint-ul activ completat corespunzător .

3) Se elaborează un program pentru realizarea unui cronometru.

Se vor utiliza următoarele obiecte:

- o formă (*Form1*) pe care sunt dispuse celelalte obiecte și în *Caption*-ul căreia se va afișa textul „**MIDPS**”;

- patru butoane (*Button 1, 2, 3, 4*) cu următoarele funcții:
 - *Button1* – pornirea cronometrului(*Caption Start*);
 - *Button2* – oprirea cronometrului(*Caption Stop*);
 - *Button3* – inițializarea cronometrului(*Caption Zero*);
 - *Button4* – ieșirea din program (*Caption Exit*).
- două timere (*Timer1* și *Timer2*) cu următoarele funcții
 - *Timer1* (*Interval=1000 ms*) utilizat la afișarea timpului curent;
 - *Timer2* (*Interval=100 ms*) utilizat pentru cronometru;
- două casete de editare (*Edit1* și *Edit2*) utilizate pentru :
 - *Edit1* - afișarea datei și orei curente;
 - *Edit2* - afișarea timpului cronometrat;
- două etichete (*Label1* și *Label2*) cu *Caption*-ul conform figurii 2.4

Observații:

- din primele trei butoane, la un un moment dat va fi activ unul singur;
- fiecare obiect va avea *hint*-ul activ completat corespunzător;

4) Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu) pe care sunt dispuse următoarele obiecte:

- o formă (*Form1*) în *Caption*-ul căreia se va afișa textul „**MIDPS**”;
- trei butoane (*Button 1, 2, 3*) cu următoarele funcții:
 - *Buton1* – activarea afișării în diagramă și în bargraf (*Caption Start*);
 - *Buton2* – oprirea afișării în diagramă și în bargraf (*Caption Stop*);
 - *Buton3* – ieșirea din program (*Caption Exit*).
- două timere (*Timer1* și *Timer2*) cu următoarele funcții
 - *Timer1* (*Interval=1000 ms*) utilizat la afișarea timpului curent;
 - *Timer2* (*Interval=500 ms*) pentru intervalul de afișare în diagramă și în bargraf;
- o casetă de editare (*Edit1*) utilizată pentru afișarea datei și orei curente;
- două etichete (*Label1* și *Label2*) cu *Caption*

Observații:

- din primele două butoane, la un un moment dat va fi activ unul singur;
- fiecare obiect va avea *hint*-ul activ completat corespunzător;

- valoarea numerică ce se va afișa în cele două elemente grafice se obține cu funcția *random* după care numărul generat se va converti în pixeli ținându-se cont de înălțimea comună a graficului și bargrafului
- pentru realizarea bargrafului se vor utiliza două obiecte de tip TPanel de culori diferite care se vor suprapune;
- pentru desenarea graficului se vor utiliza funcțiile *MoveTo*, *LineTo* iar pentru avansul acestuia funcția *CopyRect*.

Listingul programului

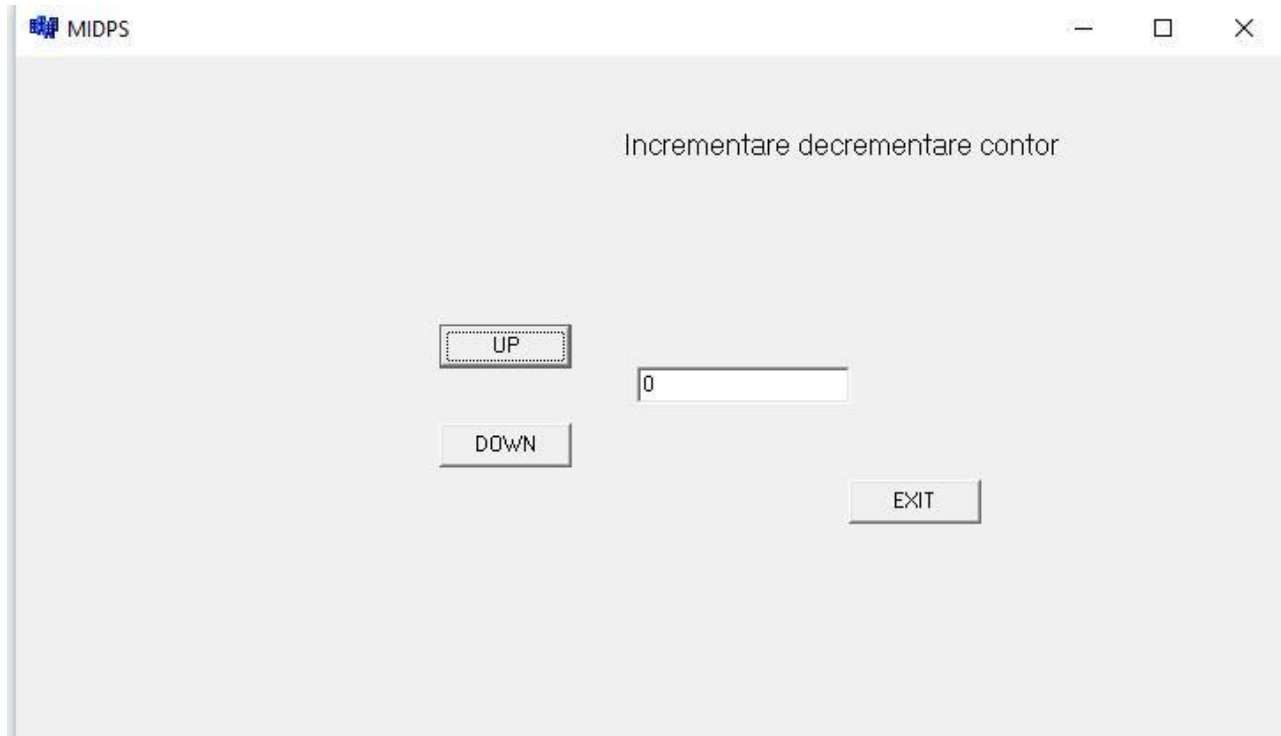
```
a) //-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int a=0;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Close();
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    ++a;
    Edit1->Text=a;
}
//-----
```

```

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    --a;
    Edit1->Text=a;
}
//-----

```



b)

```

//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "stdio.h"
#include "dos.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

```

```

struct date d;
struct time t;
int seconds=0;
int minutes=0;
int hours=0;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Timer2->Enabled = true;
    Button1->Enabled=false;
    Button2->Enabled=true;
    Button3->Enabled=false;
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Timer2->Enabled = false;
    Button2->Enabled=true;
    Button3->Enabled=true;
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    char buf[20];
    getdate(&d);
    gettime(&t);
    sprintf(buf,"%02d-%02d-%4d
%02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,t.ti_hour,t.ti_min,t.ti_sec);
    Edit1->Text=(AnsiString)buf;
}

```

```

}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Button1->Enabled=true;
    Button2->Enabled=true;
    Button3->Enabled=true;
    Edit2->Text="00:00:00";
    seconds=0;
    minutes=0;
    hours=0;
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Timer2->Enabled=false;
}
//-----

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    char timer[100];
    seconds++;
    if(seconds==60)
    {
        minutes++;
        seconds=0;
        if(minutes==60)
        {
            hours++;
            minutes=0;}
    }
    sprintf(timer,"%02d:%02d:%02d",hours,minutes,seconds);
    Edit2->Text=(AnsiString)timer;
}

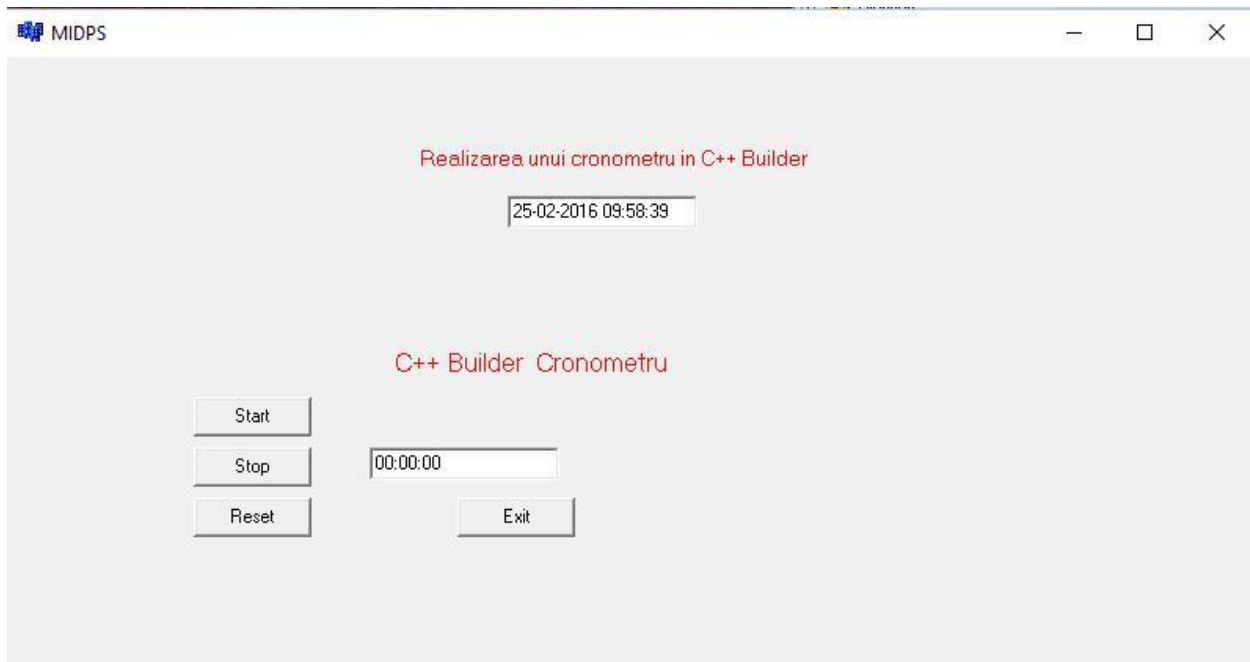
```

```

}
//-----

void __fastcall TForm1::Button4Click(TObject *Sender)
{
    Close();
}
//-----

```



c)

```

//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "dos.h"
#include <stdio.h>
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int x,y,x1,y1,i=0,k;
struct date d;
struct time t;

```



```

char buff[30];
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    gettime(&t);
    getdate(&d);
    sprintf(buff,"%02d-%02d-%4d
%02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,t.ti_hour,t.ti_min,t.ti_sec);
    Edit1->Text=AnsiString(buff);
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Close();
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Timer2->Enabled=false;
    Panel1->Color=clBlack;
    //Panel2->Height=0;
    Panel2->Color=clGray;
    Button2->Enabled=false;
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Timer2->Enabled=true;
    Button2->Enabled=true;
    Button1->Enabled=false;
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Timer2->Enabled=false;
    Button2->Enabled=false;
    Button1->Enabled=true;
}

```

```

}
//-----

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    PaintBox1->Canvas->Brush->Style=bsCross;
    PaintBox1->Canvas->Brush->Color=clBlack;
    PaintBox1->Canvas->Pen->Color=clRed;
    PaintBox1->Canvas->FloodFill(0,0,clBlack,fsBorder);
    y=rand() % 100;
    x=rand() % 7;
    if(i)
    {
        x+=x1;
    }
    if(i==0)
        PaintBox1->Canvas->MoveTo(0,10);
    else PaintBox1->Canvas->MoveTo(x1,y1);
    PaintBox1->Canvas->LineTo(x,y);
    x1=x;
    y1=y;
    i++;
    if(x==300)
    {
        PaintBox1->Repaint();
        i=0;
    }
    k=y-50;
    Panel2->Height=y;
}
//-----

```



Concluzie

In lucrarea data am obtinut abilitati in utilizarea mediului C++ Builder. Am utilizat diferite compente din C++ Builder cu ajutorul careia am realizat interfata grafica pentru program. In lucrarea data am realizat 3 miniprograme. In primul miniprogram s-a creat incrementarea si decrementarea unui contor, in al doilea un cronometru si afisarea datei curente si in al treilea miniprogram s-a realizat bargraful si diagrama.