

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei

RAPORT

Lucrarea de laborator nr. 1
Medii Interactive de dezvoltare a produselor software

A efectuat:
st. gr. TI-144

Gorduz Daniel

A verificat:
lect. univ.

Irina Cojan

Chișinău 2016

Obiectivele 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 ;
3. Se elaborează un program pentru realizarea unui cronometru.
4. Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu).

Considerații teoretice:

C++Builder este un mediu de dezvoltare rapidă a aplicațiilor produs de filiala CodeGear a Embarcadero Technologies pentru scrierea programelor în limbajul de programare C++. C++Builder combină biblioteca de componente vizuale și un IDE scris în Delphi, cu un compilator C++. Ciclul de dezvoltare este în așa fel încât Delphi primește primul îmbunătățiri semnificative, urmat de C++Builder. Majoritatea componentelor dezvoltate în Delphi pot fi folosite în C++Builder fără modificări, dar nu și invers.

C++Builder include unelte care permit dezvoltarea vizuală bazată pe drag-and-drop, făcând programarea mai facilă prin implementarea unui GUI builder WYSIWYG în IDE.

Mersul lucrarii:

Laboratorul dat consta din crearea a 3 aplicatii in mediul de programare Borland c++ Builder. Pasii efectuarii aplicatiei sunt:

-Incrementor/decrementor :

- 1.Mai intai a fost creata forma si au fost aranjate butoanele, label-ul prin intermediul instrumentelor mediului de programare.
- 2.Cu ajutorul la Object Inspector au fost setate denumirile si proprietatile elementelor grafice.
- 3.A fost creata o variabila de tip intreg care inregistreaza valoarea variabilei urmata sa fie incrementata sau decrementata.
- 4.Cu ajutrul butonului increment este crea un eveniment care maresti valoarea variabilei cu 1. Analogic pentru butonul decrement.
- 5.In cimpul label este inregistrat un text specific in dependenta de butonul apasat.

Codul sursa:

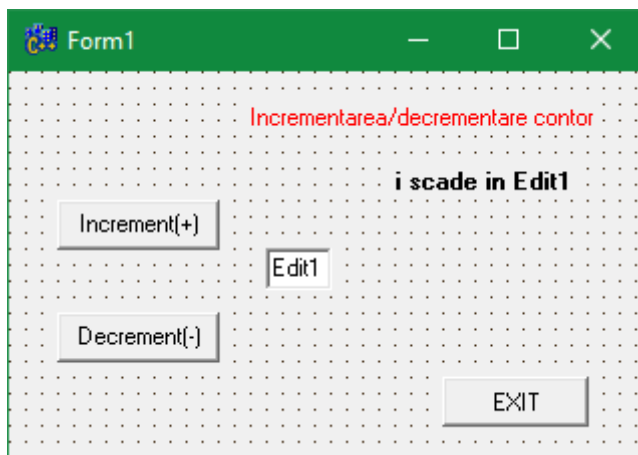
```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Lab1_MIDPS.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
int i=0;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
    Edit1->Text="0";  
    Form1->Caption="MIDPS";  
}
```

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

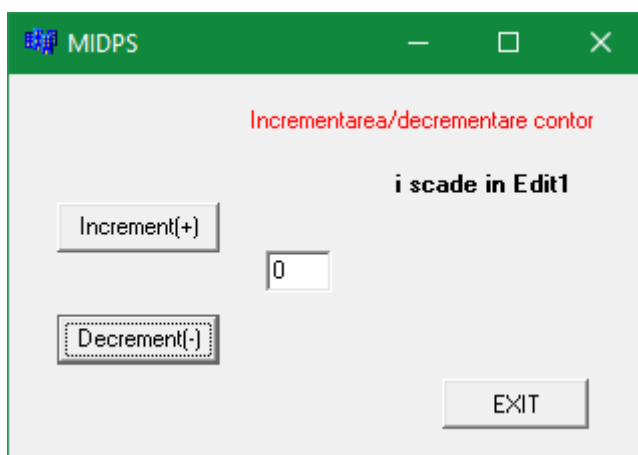
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Edit1->Text=--i;
}

void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    Close();
}
```

Forma:



Rezultat:



-Cronometru

- 1) A fost creata forma cu ajutorul instrumentelor Borland C++
- 2) A fost setata denumirile si parametrii elementelor grafice.
- 3) Pentru afisarea timpului curent au fost folosite 2 functii gettimeofday() si getdate()
- 4) Pentru crearea cronometrului au fost create 3 variabile: secunde, milisecunde, minute. Ele au fost conectate cu timer-ul si au fost stabilit un interval de 100 milisecunde. Odata ce variabilele ating valoarea necesara ele sunt resetate inapoi la 0, iar valoarea urmatoare sunt incrementate.
- 5) Cu ajutorul butonului reset valorile variabilelor sunt resetate la 0. La apasarea butonului START este dezactivat butonul STOP si vice-versa.

Codul sursa:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <time.h>  
#include <stdio.h>  
#include <dos.h>  
  
#include "Lab1_MIDPS_Cronometru.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
  
TForm1 *Form1;  
struct date d;  
struct time t;  
int min=0;  
int sec=0;  
int zec=0;  
//-----
```

```

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Label1->Caption="Realizarea unui Cronometru in C++ Builder";
    Form1->Caption="MIDPS";
    Label2->Caption="C++ Builder CRONOMETRU";
    Label2->Font->Color=clGreen;
    Edit2->Text="00 min 00 sec 00 zec";
    CronTimer->Enabled=false;
    Stop->Enabled=false;
    Zero->Enabled=false;
}

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

//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    Timer1->Interval=1000;
    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::StartClick(TObject *Sender)
{
    Start->Enabled=false;
    Stop->Enabled=true;
}

```

```

        CronTimer->Enabled=true;
        Zero->Enabled=false;
    }
    //-----
    void __fastcall TForm1::Label1Click(TObject *Sender)
    {
        Label1->Caption="Realizarea Unui Cronometru in C++Builder";
    }
    //-----
    void __fastcall TForm1::Label2Click(TObject *Sender)
    {
        Label2->Caption="C++Builder CRONOMETRU";
    }
    //-----
    void __fastcall TForm1::CronTimerTimer(TObject *Sender)
    {
        char buf[20];
        zec++;
        if(zec==10){
            zec=0;
            sec++;
            if(sec==59){
                zec=0;
                sec=0;
                min++;
            }
        }
        sprintf(buf,"%02d min  %02d sec  %02d zec",min,sec,zec);
        Edit2->Text=(AnsiString)buf;
    }
    //-----

```

```

void __fastcall TForm1::StopClick(TObject *Sender)
{
    Start->Enabled=true;
    Stop->Enabled=false;
    Zero->Enabled=true;
    CronTimer->Enabled=false;

}

```

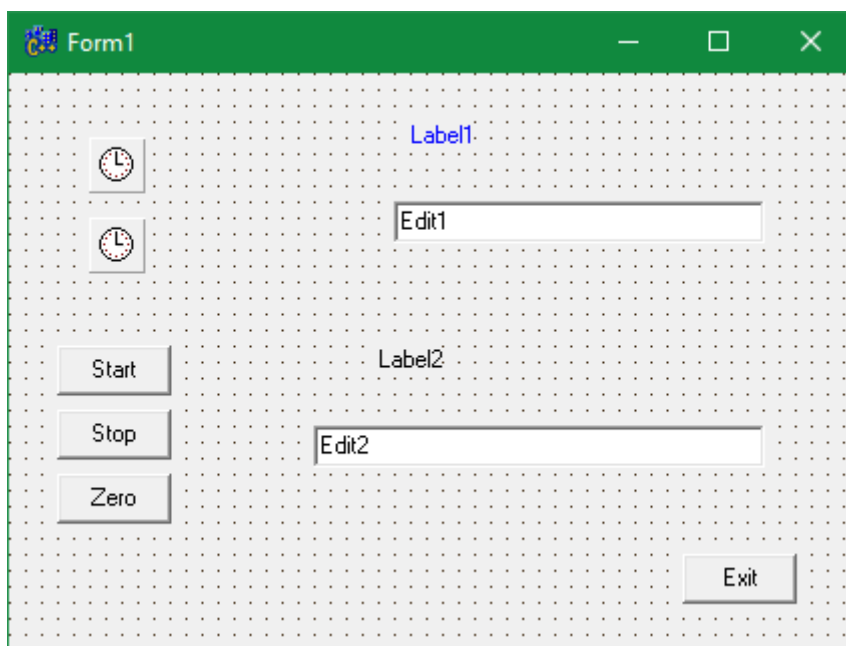
//-----

```

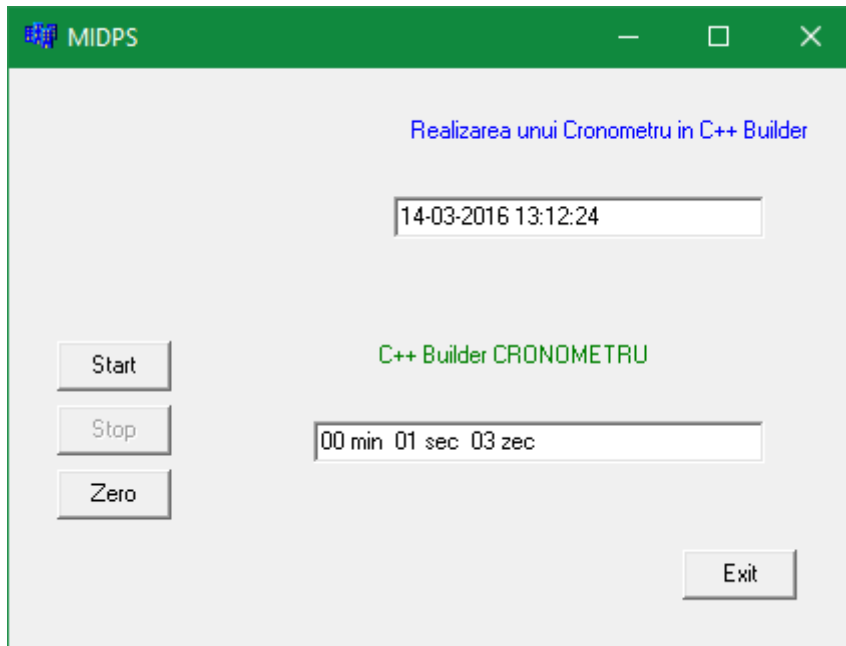
void __fastcall TForm1::ZeroClick(TObject *Sender)
{
    Stop->Enabled=false;
    char buf[20];
    min=0;
    sec=0;
    zec=0;
    sprintf(buf,"%02d min  %02d sec  %02d zec",min,sec,zec);
    Edit2->Text=(AnsiString)buf;
}

```

Forma:



Rezultat:



-Bargraf-ul si graficul

Ca si in primele 2 programe a fost creata forma si au fost setate parametrii. A fost creata zona de desen cu ajutorul instrumentului PaintBox.

- 1) Pentru desenarea Bargraf-ului au fost create 2 paneele suprapuse una deasupra alteia.
- 2) Graficul a fost creat cu ajutorul a 2 functii: MoveToEx() si pentru deplasarea la pozitia necesara, si functia LineTo() pentru desenarea liniilor.
- 3) Coordonatele liniilor au fost stabilite cu ajutorul functiei Rand() care inregistreaza valori aleatoare pentru coordonatele X si Y.
- 4) Functia LineTo() este inregistrata cu ajutorul unui timer cu interval de 500 milisecunde. Timer-ul este pornit cu ajutorul butonului START si respectiv oprit cu ajutorul butonului STOP.

Codul sursa:

```
#include <vcl.h>
#pragma hdrstop
#include <time.h>
#include <dos.h>
#include <stdio.h>
#include <math.h>
```

```

#include "Unit1.h"

//-----

#pragma package(smart_init)
#pragma resource "*.dfm"

TForm1 *Form1;

struct date d;

struct time t;

int x = 0;

int y = 100;

int pas = 0;

//-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Form1->Caption="MIDPS";
    Edit1->Clear();
    Timer1->Enabled=true;
    Label2->Caption="Resurse grafice ale mdeiului C++Builder";
    Label1->Caption="Data si ora curenta";
    StopBtn->Enabled=false;

    GreenPanel->Height = 0;
    BlackPanel->Color=RGB(0,0,0);
    GreenPanel->Color=RGB(29,120,77);

    GraphTimer->Enabled=false;
    GraphTimer->Interval=500;
}

//-----

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

```

```

//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    Timer1->Interval=1000;
    char buff[20];
    getdate(&d);
    gettime(&t);
    sprintf(buff,"%02d-%02d-%4d\t%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::StartBtnClick(TObject *Sender)
{
    StartBtn->Enabled = false;
    StopBtn->Enabled = true;
    GraphTimer->Enabled = true;
    GraphZone -> Canvas -> Pen -> Color = clBlack;
    GraphZone -> Canvas -> Brush -> Color = clSilver;
    GraphZone -> Canvas -> Brush -> Style = bsCross;
    GraphZone -> Canvas -> Rectangle(0,0,233,201);
    GraphZone -> Canvas -> Pen -> Color = clRed;
}
//-----

void __fastcall TForm1::StopBtnClick(TObject *Sender)
{
    StartBtn->Enabled = true;
    StopBtn->Enabled = false;
    GraphTimer->Enabled = false;
}
//-----

```

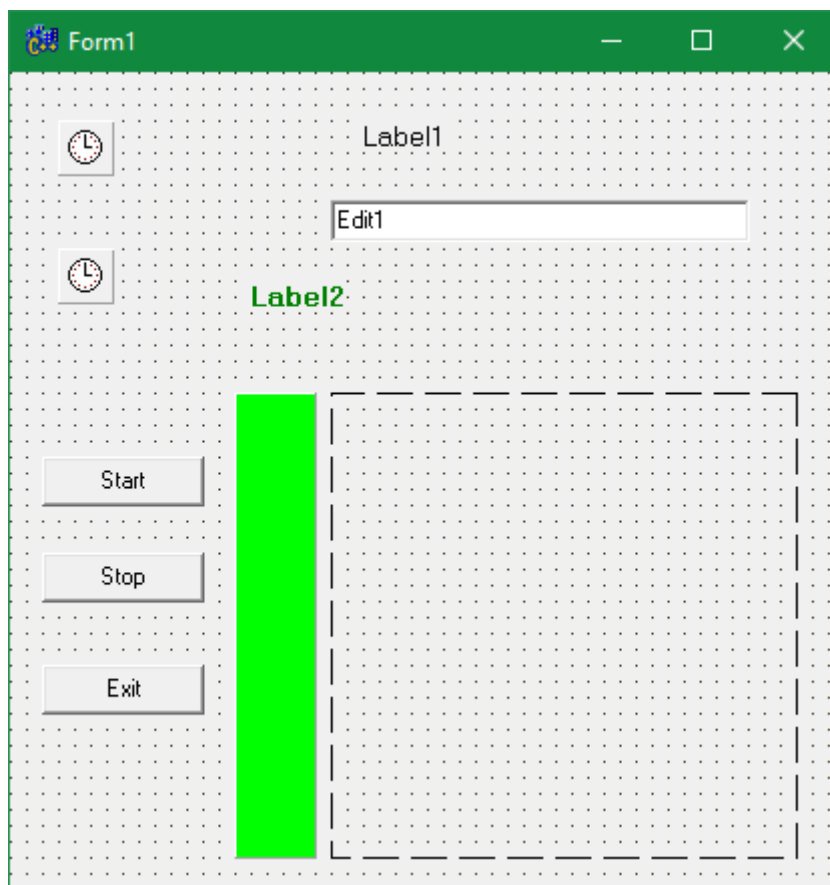
```

void __fastcall TForm1::GraphTimerTimer(TObject *Sender)
{
    GraphZone -> Canvas -> MoveTo(x,y);
    do
    {
        if (pas % 2 == 0)
        {
            y = y + rand() % 85;
            pas++;
        }
        else
        {
            y = y - rand() % 85;
            pas++;
        }
    }
    while (y < 75 || y > 125);
        x +=1 ;

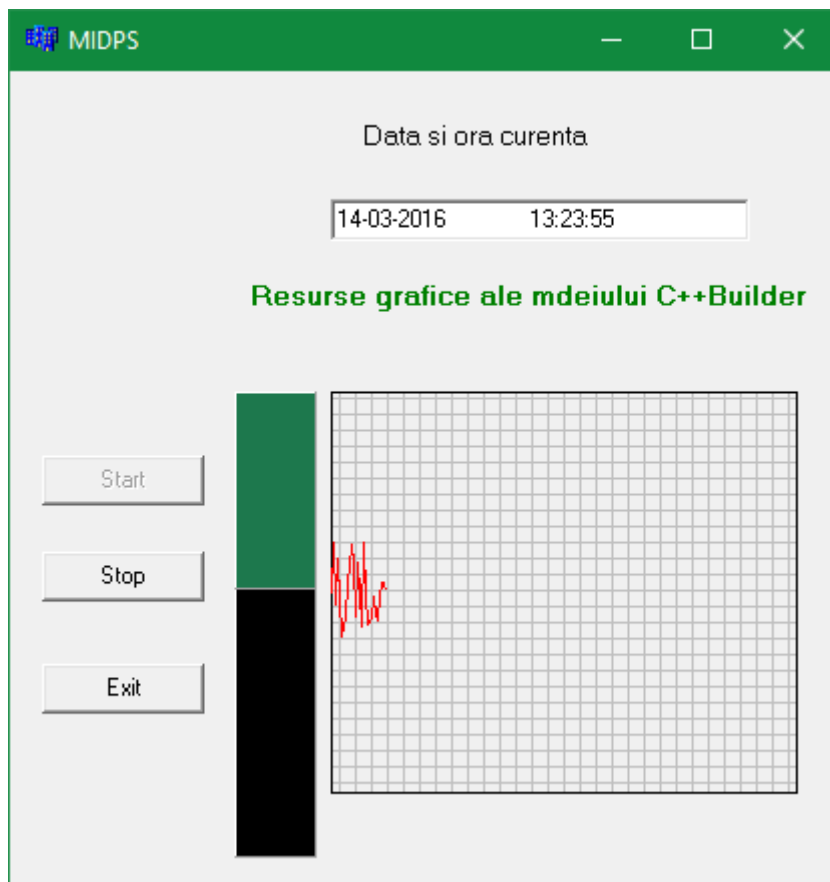
    GreenPanel -> Height = y;
    GraphZone -> Canvas -> LineTo(x,y);
    if (x >= 233)
    {
        GraphTimer -> Enabled = false;
        StopBtn->Enabled=false;
        StartBtn->Enabled=false;
    }
}

```

Forma:



Rezultat:



Concluzie

In urma efectuării acestui laborator am făcut cunostința cu posibilitățile de creare a unei aplicații cu ajutorul Mediului de dezvoltare Borland C++ Builder. Acesta este un mediu orientat pe obiect, forme și evenimente. Include unelte care permit dezvoltarea vizuală bazată pe drag-and-drop, făcând programarea mai facilă prin implementarea unui GUI builder WYSIWYG în IDE. Aceasta mi-a permis să creez o aplicație în Windows cu număr minim de cod.