Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#2

Mediul integrat C++ Builder

Autor:
Alexandr Ialticenco

lector asistent:
Irina Cojanu
lector superior:
Svetlana Cojocaru

Obiectivele lucrarii

- Insuirea modului de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER . Realizarea unui program simplu care utilizeaza componente de tip TButton, TEdit, Tlabel, RadioButton etc.

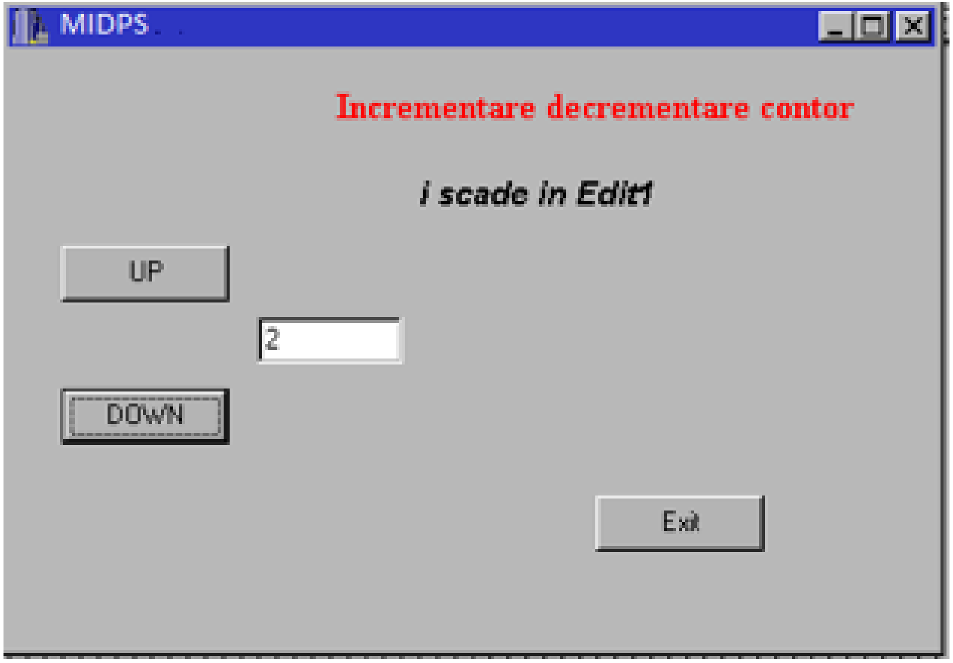
- Insuirea modului de utilizare a componentei VCL TTimer. Insuirea modului de utilizare a functiilor de lucru cu timpul sistem. Realizarea unor aplicatii de gestionare a resursei timp.

- Insuirea modului de utilizare a componentelor VCL TPaintBox si TPanel. nsuirea modului de utilizare a principalelor functii grafice ale mediului C++BUILDER . Realizarea unor elemente pentru afisarea grafica a informatiei (diagrama si bargraf).

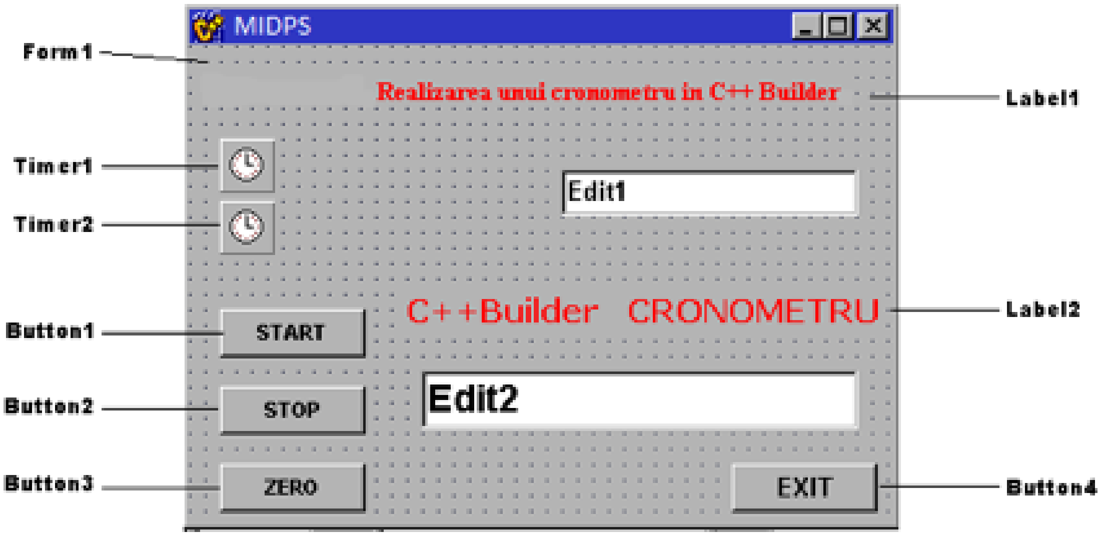
Sarcina lucrarii

- Vor fi examinate toate componentele prezentate in indicatii teoretice;

- Se modifica Project1.cpp ca sa se obtina forma de mai jos



- Se elaboreaza un program pentru realizarea unui cronometru. Se vor utiliza urmtoarele obiecte, evidentiate n figura de mai jos:

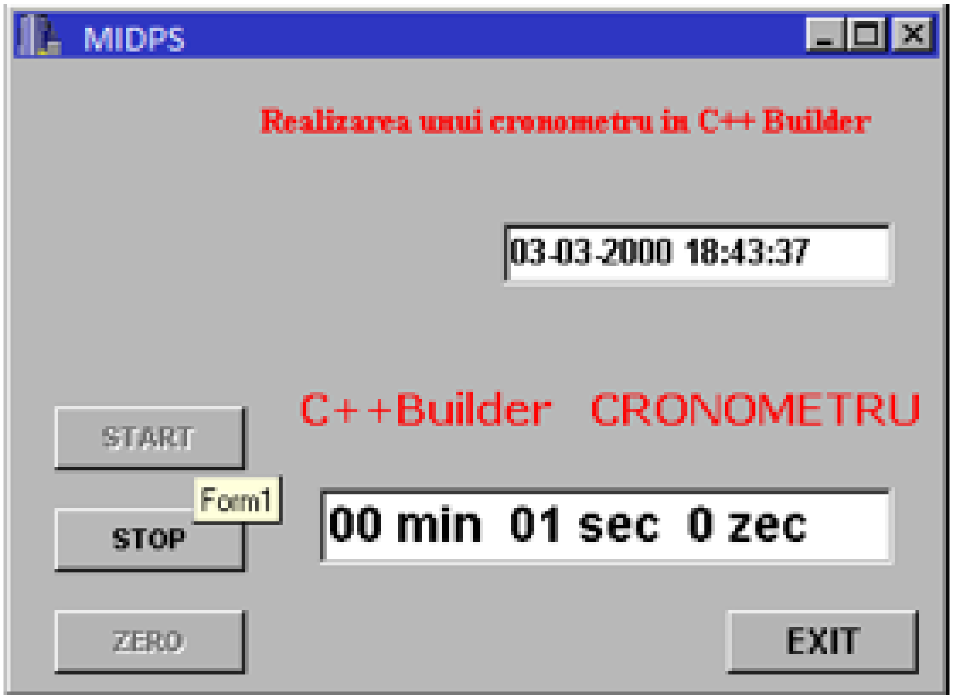


Realizarea unui cronometru in C++ Builder

Edit1

C++Builder CRONOMETRU

Edit2



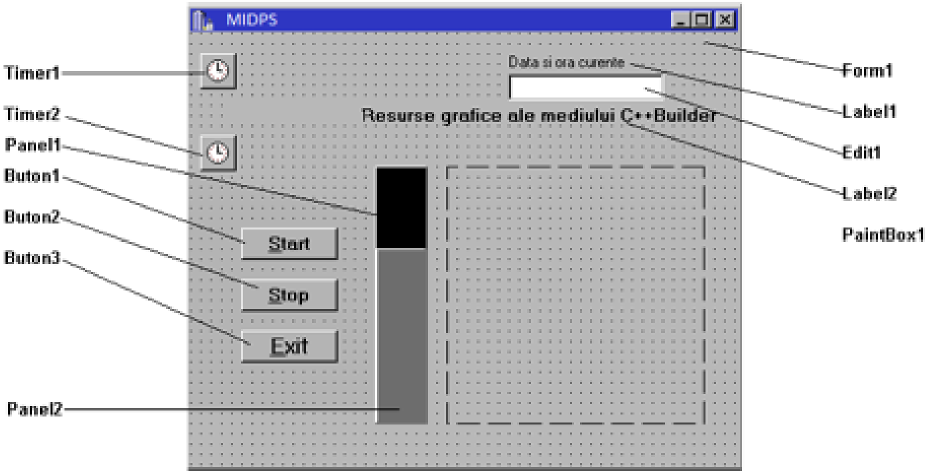
Realizarea unui cronometru in C++ Builder

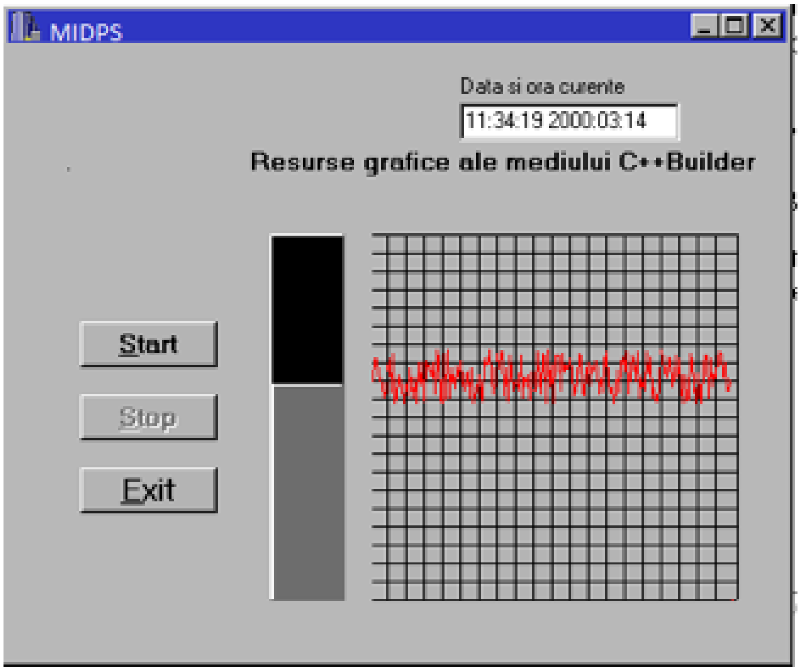
03-03-2000 18:43:37

C++Builder CRONOMETRU

00 min 01 sec 0 zec

- Se elaboreaza un program pentru realizarea a doua elemente de afisare (bargraf si diagram cu avans continuu) pentru care forma arat ca n figura de mai jos pe care sunt dispuse urmtoarele obiecte:





11:34:19 2000:03:14

1 Incrementator

1.1 Listingul

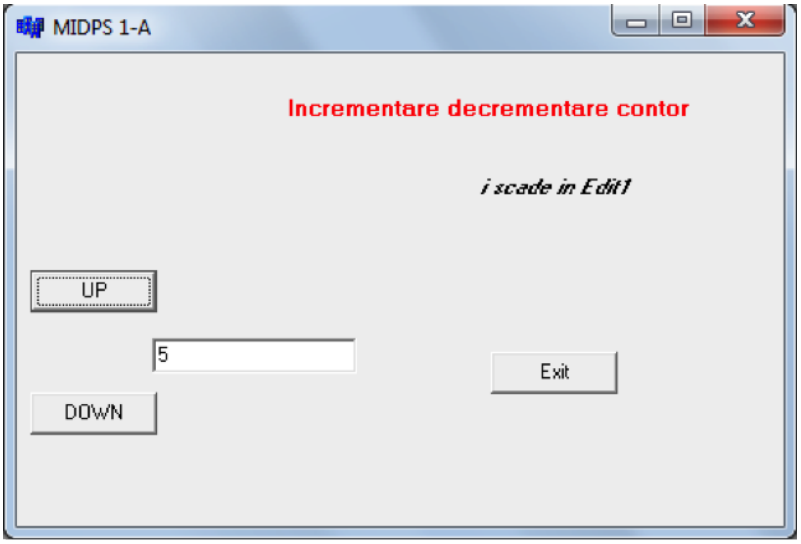
Modul GUI va fi encapsulat in clasa Form1.

```
//—————


#include <vcl.h>

#pragma hdrstop



#include ”Unit1.h”

//—————
#pragma package(smart_init)

#pragma resource ”*.dfm”

TForm1 *Form1;

int i;

//————
__fastcall TForm1::TForm1(TComponent* Owner)

        : TForm(Owner)

{

}

//————
```

```
void __fastcall TForm1::Button3Click(TObject *Sender)

{

Close();

}

//---------

void __fastcall TForm1::FormCreate(TObject *Sender)

{

i = 0;

Edit1->Text = i;

}

//---------
void __fastcall TForm1::Button1Click(TObject *Sender)

{

Edit1->Text = ++i;

}

//----------

void __fastcall TForm1::Button2Click(TObject *Sender)

{

Edit1->Text = --i;

}
```

```
//—————————
```

1.2 Screenshot-ul aplicatiei



Incrementare decrementare contor

i scade in Edit1

5

2 Cronometru

2.1 Listingul

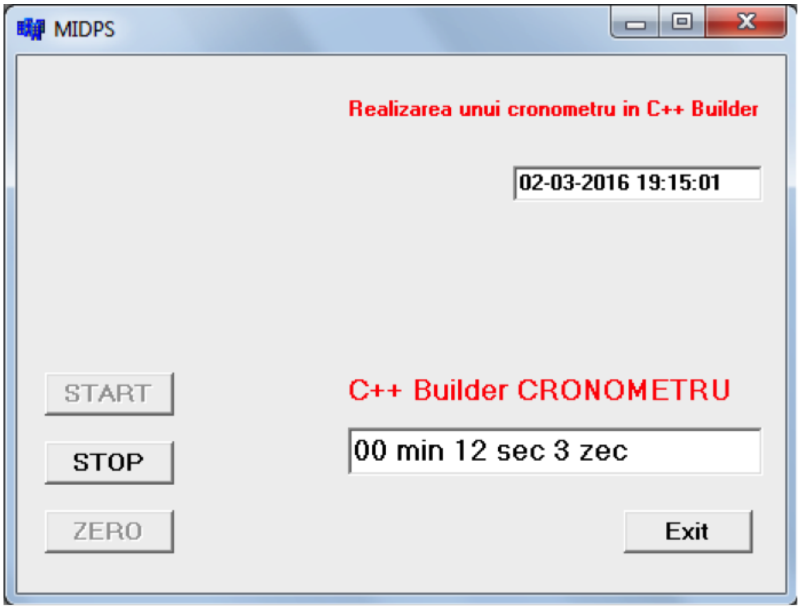
```
//—————————

#include <vcl.h>

#include <stdio.h>

#pragma hdrstop

#include "Unit1.h"
```

```
//—————

#pragma package(smart_init)

#pragma resource ”∗.dfm”

#include ”dos.h”

TForm1 ∗Form1;

struct time t;

struct date d;

int min, sec, zec;

//—————

__fastcall TForm1::TForm1(TComponent∗ Owner)

        : TForm(Owner)

{

}

//—————


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::Button1Click(TObject *Sender)

{

Timer2->Enabled = true;

Button2->Enabled = true;

Button1->Enabled = false;

Button3->Enabled = false;

}

//----------

void __fastcall TForm1::Timer2Timer(TObject *Sender)

{

zec += 1;

if (zec >= 10){

zec = 0;
sec ++;

}

if (sec>=60){

sec = 0;

min ++;

}

char buf[20];

sprintf(buf,”%02d min %02d sec %d zec”,min,sec, zec);

Edit2->Text=(AnsiString)buf;

}

//----------

void __fastcall TForm1::Button2Click(TObject *Sender)

{

Timer2->Enabled = false;

Button1->Enabled = true;

Button3->Enabled = true;

Button2->Enabled = false;

}

//----------
```

```
void __fastcall TForm1::Button3Click(TObject *Sender)
{
min = sec = zec = 0;
char buf[20];
sprintf(buf,”%02d min %02d sec %d zec”,min,sec, zec);
Edit2–>Text=(AnsiString)buf;
}
//——————
void __fastcall TForm1::Button4Click(TObject *Sender)
{
Close();
}
//——————

void __fastcall TForm1::FormCreate(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;

}
```

2.2 Screenshot-ul aplicatiei



MIDPS

Realizarea unui cronometru in C++ Builder

02-03-2016 19:15:01

C++ Builder CRONOMETRU

00 min 12 sec 3 zec

3 Bargraful

3.1 Listingul

Codul sursa:

```
//–––––––––
```

```
#include <vcl.h>

#pragma hdrstop

#include <stdio.h>

#include ”Unit1.h”

#include ”dos.h”

//—————————

#pragma package(smart_init)

#pragma resource ”*.dfm”

TForm1 *Form1;

struct time t;

struct date d;

TRect rect;

int nextY, next2Y, nextP;

bool first;

//—————————

__fastcall TForm1::TForm1(TComponent* Owner)

        : TForm(Owner)

{
}

//----------------------


void __fastcall TForm1::Button3Click(TObject *Sender)

{

Close();

}

//----------------------


void __fastcall TForm1::Button1Click(TObject *Sender)

{

Button1->Enabled = false;

Button2->Enabled = true;

Timer2->Enabled = true;

first = true;

rect.Left = 0;

rect.Right = 300;

rect.top = 0;

rect.Bottom = 300;
```

```
}

//—————

void __fastcall TForm1::Button2Click(TObject *Sender)

{

Button1->Enabled = true;

Button2->Enabled = false;

Timer2->Enabled = false;

}

//—————

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;

}
```

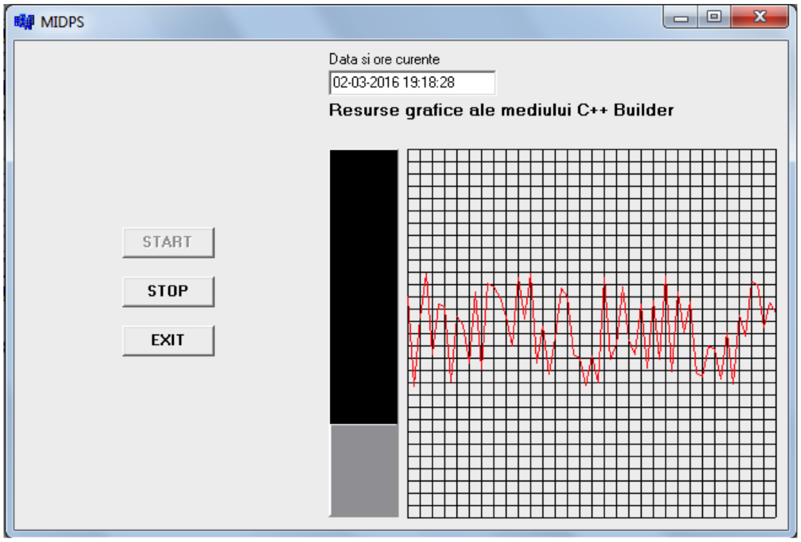
```
//----------
void cells(){


}
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
if (first){
nextP =  abs(rand()%100);
nextY = abs(rand()%100);
PaintBox1->Canvas->Pen->Color = clBlack;
for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(0,i*10);
PaintBox1->Canvas->LineTo(300,i*10);
}
for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(i*10,0);
PaintBox1->Canvas->LineTo(i*10,300);
}
first = false;
}

TRect rect2;

rect2.left = 0;

rect2.right = 300;

rect2.top = 0;

rect2.bottom = 300;

rect.left = -10;

rect.right = 290;

TRect rect3;

rect3.left = 290;
rect3.right = 300;
rect3.top = 0;
rect3.bottom = 300;

PaintBox1->Canvas->FillRect(rect3);

PaintBox1->Canvas->MoveTo(290,0);

PaintBox1->Canvas->LineTo(290,300);

for (int i =0; i<31; i++){

PaintBox1->Canvas->MoveTo(290,i*10);

PaintBox1->Canvas->LineTo(300,i*10);

}
```

```
PaintBox1->Canvas->MoveTo(290,nextY+100);

PaintBox1->Canvas->Pen->Color = clRed;

nextY = next2Y;

PaintBox1->Canvas->LineTo(295,100+nextP);



PaintBox1->Canvas->LineTo(300,100+nextY);

PaintBox1->Canvas->CopyRect(rect,PaintBox1->Canvas, rect2);

PaintBox1->Canvas->Pen->Color = clBlack;



rect3.left = 290; rect3.right = 300;
rect3.top = 0; rect3.bottom = 300;

PaintBox1->Canvas->FillRect(rect3);

PaintBox1->Canvas->MoveTo(290,0);

PaintBox1->Canvas->LineTo(290,300);



for (int i =0; i<31; i++){

PaintBox1->Canvas->MoveTo(290,i*10);

PaintBox1->Canvas->LineTo(300,i*10);

}
```

```
nextP =  abs(rand()%100);
PaintBox1->Canvas->Pen->Color = clRed;
next2Y = abs(rand()%100);
PaintBox1->Canvas->MoveTo(290,nextY+100);
PaintBox1->Canvas->LineTo(295,100+nextP);
PaintBox1->Canvas->LineTo(300,100+next2Y);
PaintBox1->Canvas->Pen->Color = clBlack;
PaintBox1->Canvas->MoveTo(300,0);
PaintBox1->Canvas->LineTo(300,300);
Panel2->Height = Panel1->Height*(nextP/100.0);
//PaintBox1->Canvas->FillRect(rect);
}
//----------------------------------------

void __fastcall TForm1::FormCreate(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;

}

//−−−−−−−−−−−
```

3.2 Screenshot-ul aplicatiei



Data si ore curente

02-03-2016 19:18:28

Resurse grafice ale mediului C++ Builder

Concluzii

In cadrul acestei lucrari de laborator am studiat principiile si modul de utilizare a celor mai importante componente ale mediului integrat C++ Builder aa ca TButton, TEdit, TTimer, TPaintBox si etc. Bazand pe cunotinele obtinute am reusit sa realizez trei aplicatii Windows cu interfata grafica, functionalul carora include nu doar interactiunea grafic dintre utilizator si aplicatia cat si gestionarea resursei de timp si afisarea grafica a informatiei (prin intermediul diagramelor si bargrafelor). Experiena si cunostinele obtinute pe parcursul indeplinirii lucrrii de laborator vor fi utile n viitor si pot fi aplicate pentru realizarea proiectelor diferite.

Bibliografie

1. http://www.functionx.com/cppbcb/ - **C++ Builder Programming**