

**MEDII INTERACTIVE DE DEZVOLTARE
A PRODUSELOR SOFT**

LUCRARE DE LABORATOR nr.1

MEDIUL INTEGRAT C++ BUILDER

St. gr. TI-141
Diana-Mihaela BORS

lector asistent:
Irina COJANU

lector superior:
Svetlana COJOCARU

LUCRARE DE LABORATOR #1

1. Scopul lucrării

Însușirea modului de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER.

2. 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).

3. Efectuarea lucrării de laborator

3.1. Task-uri implementate

1. Se elaborează un program pentru realizarea unui contor cu funcțiile incrementare / decrementare.
2. Se elaborează un program pentru realizarea unui cronometru.
3. Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu).

3.2. Analiza lucrării de laborator

a) Realizarea unui program simplu care utilizează componente de tip *TButton*, *TEdit*, *Tlabel*, *RadioButton* etc. (Fig. A)

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
#include "stdio.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
int i = 0;  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
  
: TForm(Owner)  
{  
Label1->Caption = "Incrementare / descrementare contor";  
}  
//-----  
void __fastcall TForm1::UpClick(TObject *Sender)
```

```

{
i++;
char buff[20];
sprintf(buff,"%d",i);
Edit1->Text = (AnsiString) buff;
Label2->Caption = "i creste in Edit1";
}
//-----

void __fastcall TForm1::DownClick(TObject *Sender)
{
i--;
char buff[20];
sprintf(buff,"%d",i);
Edit1->Text = (AnsiString) buff;
Label2->Caption = "i scade in Edit1";
}
//-----

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

```

b) Realizarea unor aplicații de gestionare a resursei timp. (Fig. B)

```

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

#pragma package(smart_init)
#pragma resource "*.dfm"
struct date d;
struct time t;
class Cronos {
private:

int minute;
int secunde;
int zecimi;
public:
Cronos(){
this->minute = 0;
this->secunde = 0;
this->zecimi = 0;
}
void incCronos(){
this->zecimi++;
if(this->zecimi == 10){
this->zecimi = 0;

```

```

this->secunde++;
}
if(this->secunde == 60){
this->secunde = 0;
this->minute++;
}
}
void zeroCronos(){
this->minute = 0;
this->secunde = 0;
this->zecimi = 0;
}
int getM(){
return this->minute;
}
int getS(){
return this->secunde;
}
int getZ(){
return this->zecimi;
}
};
Cronos cronometru;
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
Timer2->Enabled = false;
char buf[30];
sprintf(buf,"%02d min : %02d sec : %02d zec",cronometru.getM(),
cronometru.getS(),cronometru.getZ());
Edit2->Text = (AnsiString)buf;
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::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::Button4Click(TObject *Sender)

```

```

{
Close();
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
cronometru.incCronos();
char buf[30];
sprintf(buf,"%02d min : %02d sec : %02d zec",cronometru.getM(),
cronometru.getS(),cronometru.getZ());
Edit2->Text = (AnsiString)buf;
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Timer2->Enabled = true;
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Timer2->Enabled = false;
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
cronometru.zeroCronos();
char buf[30];
sprintf(buf,"%02d min : %02d sec : %02d zec",cronometru.getM(),
cronometru.getS(),cronometru.getZ());
Edit2->Text = (AnsiString)buf;
}

//-----

```

c) Realizarea unor elemente pentru afișarea grafică a informației (diagramă și bargraf). (Fig. C)

```

//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "stdio.h"
#include "dos.h"
#include "stdlib.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
struct date d;
struct time t;
int xx=5;
//-----

```

```

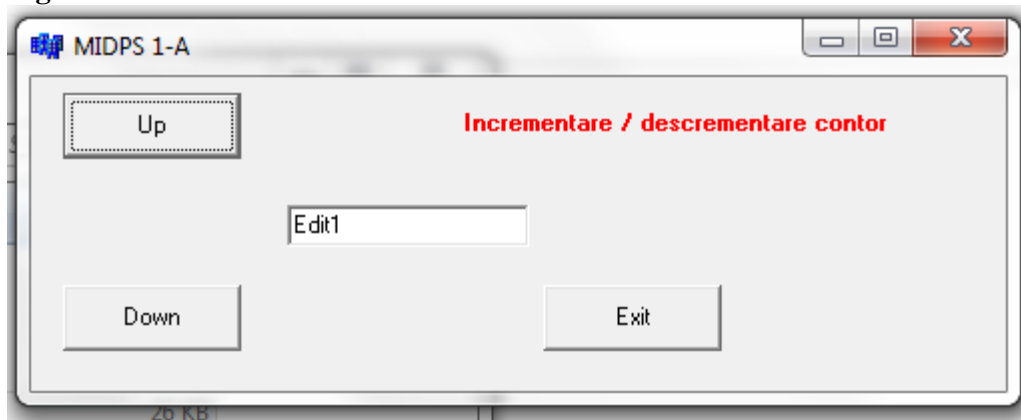
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
Timer2->Enabled = false;
PaintBox1->Canvas->MoveTo(0,150);
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
char buf[20];
getdate(&d);
gettime(&t);
sprintf(buf,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
t.ti_hour,t.ti_min,t.ti_sec);
Edit1->Text=(AnsiString)buf;
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
PaintBox1->Canvas->Brush->Color = clGray;
PaintBox1->Canvas->Pen->Color = clGray;
PaintBox1->Canvas->Brush->Style = bsCross;
PaintBox1->Canvas->Rectangle(0,0,PaintBox1->Width,PaintBox1->Height);

PaintBox1->Canvas->FloodFill(PaintBox1->Left+5,PaintBox1->Top+5,clBlack,fsBorder);
Panel2->Height = rand() % 150 + 50;
PaintBox1->Canvas->Pen->Color=clRed;
PaintBox1->Canvas->Pen->Width = 1;
PaintBox1->Canvas->LineTo(xx,rand() % 150 + 50);
xx += 3;
if(xx > PaintBox1->Width){
xx = 0;
PaintBox1->Canvas->MoveTo(0,150);
PaintBox1->Repaint();
}
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Timer2->Enabled = true;
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Timer2->Enabled = false;
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}
//-----

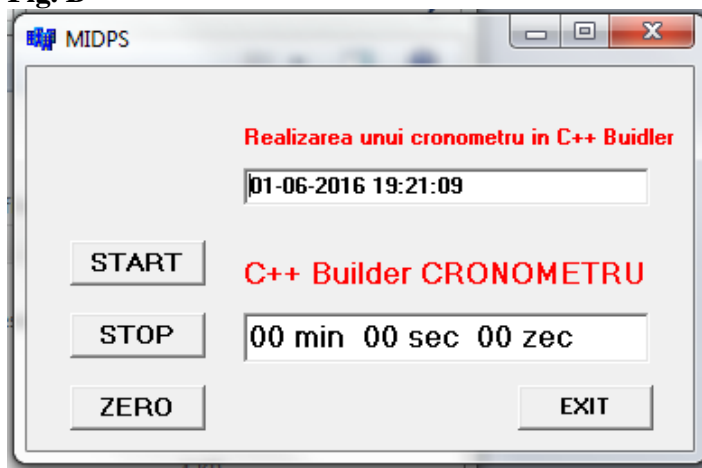
```

3.3. Rezultate(imagini)

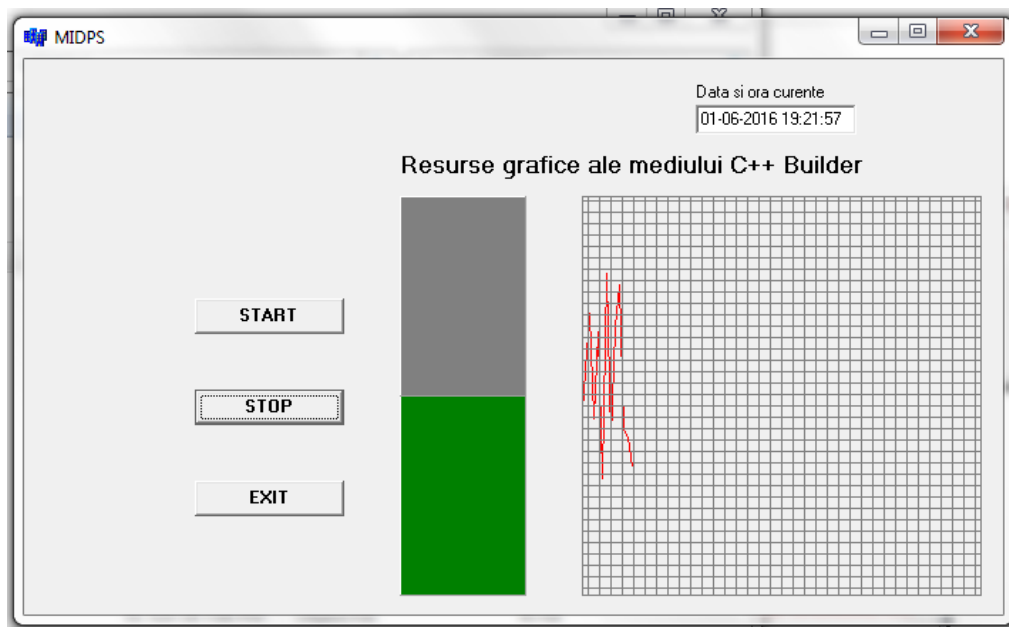
1. Fig. A



2. Fig. B



3. Fig. C



Concluzie

În urma efectuării laboratorului la tema: "*Mediul Integrat C++ Builder*", am însușit modul de utilizare a celor mai importante componente ale Mediului Integrat C++ BUILDER.

Am elaborat un program care utilizează componente de tip *TButton*, *TEdit*, *TLabel*, *RadioButton*. Programul include o fereastră cu caption-ul "MIDPS 1-A", care conține 3 butoane:

Up – incrementarea contorului;

Down - decrementarea contorului;

Exit – închide fereastra.

Fereastra mai include o casetă de editare unde se afișează valoarea variabilei **i**, două etichete, prima(albastru), afișează textul "*Incrementare/decrementare contor*", incrementarea sau decrementarea contorului în dependență de butonul apăsător.

De asemenea am elaborat un program care utilizează componenta de tip *VCL TTimer*, pentru realizarea unui cronometru. Programul include o fereastră cu caption-ul "MIDPS", care conține 4 butoane:

Start – pornirea cronometrului;

Stop - oprirea cronometrului;

Zero – inițializarea cronometrului;

Exit - închide fereastra.

Fereastra include două timer-e, pentru afișarea timpului curent și pentru cronometru, două etichete, care afișează textele: "**Realizarea unui cronometru în C++ Builder**", "**C++ Builder Cronometru**", corespunzătoare celor două casete de editare de mai jos. La fel am utilizat componentele de tip *VCL TPaintBox* și *TPanel*, pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu). Programul include o fereastră cu caption-ul "MIDPS", care conține 3 butoane:

Start – activarea afișării în diagramă și în bargraf;

Stop - oprirea afișării în diagramă și în bargraf;

Exit - închide fereastra.

De asemenea, fereastra include două timer-e, pentru afișarea timpului curent și pentru intervalul de afișare în diagramă și în bargraf, o casetă de editare, care afișează data și ora curentă, și două etichete care afișează textele: *"Data si ora curenta"*, *"Resurse grafice ale mediului C++ Builder"*.

În concluzie, am însușit modul de utilizare a componentelor de bază. În această lucrare am pus în practică IDE-ul C++Builder pentru crearea ferestrelor în Windows cu diverse funcții, ca incrementarea decrementarea, cronometru și bargraf.

Am lucrat cu **TCanvas**, **TPaintBox**, **TTimer** și alte componente a C++Builder-ului. Astfel am realizat un bargraf cu ajutorul funcțiilor MoveTo(int x, int y), LineTo (int x, int y).

Bibliografie

1. Îndrumar metodic pentru lucrările de laborator la MIDPS