

Ministerul Educatiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Tehnologii informationale

# RAPORT

Lucrare de laborator Nr. 1

*Medii interactive de dezvoltare a produselor soft*

*A efectuat:*

*st.gr.TI-143  
Gaina Cristin*

*A verificat:*

*lector universitar  
Cojocaru S.*

Chişinău 2016

## MEDIUL INTEGRAT C++ BUILDER

### 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).

### Indicații teoretice

#### Facilitățile mediului C++Builder

Borland C++Builder este un mediu de programare vizual, orientat obiect, pentru dezvoltarea rapidă de aplicații (**RAD**) cu scop general și aplicații client/server pentru Windows și WindowsNT. Folosind C++Builder se pot crea aplicații Windows eficiente știind un minim de cod. Facilitățile semnificative oferite de acestea sunt prezentate succint în cele ce urmează.

#### Înalta productivitate a mediului de dezvoltare

Aceasta este favorizată de principalele instrumente furnizate de mediul de dezvoltare integrat (**IDE**) C++Builder și anume :

- *Visual Form Designer*;
- *Object Inspector*;
- *Component Palette*;
- *Project Manager*;
- *Code Editor*;
- *Debugger*.

Acestea dau posibilitatea utilizatorului să dezvolte rapid aplicații având totodată un control complet asupra codului și resurselor.

#### Proiectare drag-and-drop

Utilizatorul poate crea aplicații prin simpla *tragere* (drag and drop) a componentelor din *Component Palette* pe *Form designer* urmată de setarea proprietăților din *Object Inspector*. *Handler-ele* de evenimente sunt automat create, iar codul lor este complet accesibil. Acest mod de proiectare a unei aplicații nu restricționează în nici un fel accesul programatorului la codul sursă, o aplicație putând fi scrisă și fără a folosi componente vizuale.

#### Proprietăți, metode, evenimente

Dezvoltarea rapidă a aplicațiilor înseamnă suport pentru proprietățile, metodele și evenimentele obiectelor (*PME*). Proprietățile permit setarea ușoară a caracteristicilor componentelor. Metodele execută acțiuni asupra obiectelor. Evenimentele permit ca aplicația să

răspundă la mesajele Windows, sau la schimbări de stare a obiectelor. Folosirea modelului PME furnizează un robust și intuitiv mediu de dezvoltare pentru aplicațiile Windows.

### **C++Builder Help**

Mediul C++Builder oferă un ghid practic, care conține peste 3000 de pagini de documentație despre IDE, VCL, baze de date și tehnici de programare.

### **Codurile sursă pentru VCL**

Mediul C++Builder pune la dispoziție codurile sursă pentru *VCL – Visual Component Library*, furnizând astfel o unică privire înăuntrul modului în care lucrează C++Builder. VCL furnizează peste 100 de componente reutilizabile care ajută programatorul să construiască aplicații robuste într-un timp scurt. Aceste componente pot fi modificate pentru a corespunde necesităților din cele mai diverse. C++Builder –ul include o suită completă de controale Windows: TreeView, Trackbars, ProgressBars, toolbars, Rich Edit, ListViews, ImageLists, StatusBars etc. Totodată C++Builder include suport pe 32 de biți pentru numele lungi de fișiere, multi-threading și Win API.

#### **Anexa:**

##### **a)cpp**

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit2.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm2 *Form2;  
int i=0;  
//-----  
__fastcall TForm2::TForm2(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm2::BitBtn2Click(TObject *Sender)  
{  
    i--;  
    Edit1->Text=i;  
}  
//-----
```

```

void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
    i++;
    Edit1->Text=i;
}
//-----

void __fastcall TForm2::BitBtn3Click(TObject *Sender)
{
    Close();
}
//-----
//-----

#ifndef Unit2H
#define Unit2H
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.Buttons.hpp>
//-----
class TForm2 : public TForm
{
__published: // IDE-managed Components
    TLabel *Label1;
    TLabel *Label2;
    TEdit *Edit1;
    TBitBtn *BitBtn1;
    TBitBtn *BitBtn2;
    TBitBtn *BitBtn3;
    TLabel *Label3;
    void __fastcall BitBtn2Click(TObject *Sender);
    void __fastcall BitBtn1Click(TObject *Sender);
    void __fastcall BitBtn3Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm2(TComponent* Owner);
};
//-----
extern PACKAGE TForm2 *Form2;
//-----
#endif

```

**b)**

```
//-----

#include <vcl.h>
#include <dos.h>
#include <stdio.h>
#pragma hdrstop

#include "Lab12.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
int m = 0, s = 0, z = 0;
int day = 0, mon = 0, year = 0, hour = 0, min = 0, sec = 0;
int i = 0;
TForm1 *Form1;
struct date d;
struct time t;
int k = 0;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    Timer1->Enabled=true;
    Timer2->Enabled=false;
    b1->Enabled=true;
    b2->Enabled=false;
    b3->Enabled=false;
    Edit2->Text="0 min 0 sec 0 zec";

    getdate(&d);
    gettime(&t);
    char buf[40];

    sprintf ( buf, 40, "%d-%d-%d %d:%d:%d",
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::b1Click(TObject *Sender)
{
    b1->Enabled=false;
    b2->Enabled=true;
    b3->Enabled=false;
```

```

        Timer2->Enabled=true;

    }
    //-----
    void __fastcall TForm1::b2Click(TObject *Sender)
    {
        b1->Enabled=false;
        b2->Enabled=false;
        b3->Enabled=true;
        Timer2->Enabled=false;
    }
    //-----
    void __fastcall TForm1::ExitClick(TObject *Sender)
    {
        Close();
    }
    //-----
    void __fastcall TForm1::b3Click(TObject *Sender)
    {
        b1->Enabled=true;
        b2->Enabled=false;
        b3->Enabled=false;
        Timer2->Enabled=false;
        i = z = s = m = 0;
        Edit2->Text="0 min 0 sec 0 zec";
    }
    //-----
    void __fastcall TForm1::Timer2Timer(TObject *Sender)
    {
        char buf[40];

        Timer2->Tag=i;
        if( i<100){
            z = i; }
        if(i == 10){
            z = 0;
            s++;
            i = 0;
        }
        if(s == 60){
            m++;
            s = 0;
        }

        snprintf ( buf, 40, "%d min %d sec %d zec", m,s,z);
        Edit2->Text=(AnsiString)buf;
        i++;
        //Edit2->Text=Timer2->Tag;
    }
    //-----

```

```

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{

    getdate(&d);
    gettime(&t);
    char buff[40];

    sprintf ( buff, 40, "%d-%d-%d %d:%d:%d",
d.da_day,d.da_mon,d.da_year,t.ti_hour,t.ti_min,t.ti_sec);
    Edit1->Text=(AnsiString)buff;

}
//-----
.....
//-----

#ifndef Lab12H
#define Lab12H
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.Buttons.hpp>
#include <Vcl.ExtCtrls.hpp>
//-----
class TForm1 : public TForm
{
__published: // IDE-managed Components
    TBitBtn *b1;
    TBitBtn *b2;
    TBitBtn *b3;
    TBitBtn *Exit;
    TTimer *Timer1;
    TTimer *Timer2;
    TLabel *Label1;
    TLabel *Label2;
    TEdit *Edit1;
    TEdit *Edit2;
    void __fastcall b1Click(TObject *Sender);
    void __fastcall b2Click(TObject *Sender);
    void __fastcall ExitClick(TObject *Sender);
    void __fastcall b3Click(TObject *Sender);
    void __fastcall Timer2Timer(TObject *Sender);
    void __fastcall Timer1Timer(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm1(TComponent* Owner);

```

```
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
```

**c)**

```
//-----

#include <vcl.h>
#include <dos.h>
#include <stdio.h>
#pragma hdrstop

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int x = 0;
int i = 0;
struct date d;
struct time t;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{

    TRect dreptunghi;
    TRect sursa,destinatie;
    PaintBox1->Canvas->Pen->Color=clRed;
    dreptunghi=Rect(464,128,464+129,128+129);
    PaintBox1->Repaint();

    sursa=Rect(0,0,100,150);
    destinatie=Rect(100,0,200,150);

    sursa=Rect(464,128,464+129,128+129);
    PaintBox1->Canvas->CopyRect(destinatie,PaintBox1->Canvas,sursa);
    //Rect(0,0,50,50);
    //PaintBox1->Canvas->FillRect(dreptunghi);
    //PaintBox1->Canvas->Brush->Color = clAqua;
    // PaintBox1->Canvas->FloodFill(464,128,clBlack,fsBorder);
```



```

    Timer1->Enabled=true;
    Timer2->Enabled=false;
    getdate(&d);
    gettime(&t);
    char buf[40];

    snprintf ( buf, 40, "%d-%d-%d %d:%d:%d",
d.da_day,d.da_mon,d.da_year,t.ti_hour,t.ti_min,t.ti_sec);
    Edit1->Text=(AnsiString)buf;

    Start->Enabled=true;
    Stop->Enabled=false;
    Continue->Enabled=false;

    PaintBox1->Canvas->MoveTo(0,60);
    /*
    TRect dreptunghi;
    PaintBox1->Canvas->Brush->Style = bsHorizontal;
    dreptunghi=Rect(0,0,400,400);
    PaintBox1->Canvas->FillRect(dreptunghi);
    PaintBox1->Repaint();
    PaintBox1->Canvas->Pen->Color=clRed;
    PaintBox1->Canvas->Pen->Width=1;
    //PaintBox1->Canvas->Pen->Style=psDash;
    PaintBox1->Canvas->Ellipse(0,0,400,400);
    TRect sursa,destinatie;
    sursa=Rect(0,0,100,150);
    PaintBox1->Canvas->CopyRect(destinatie,PaintBox1->Canvas,sursa); */

}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{

    getdate(&d);
    gettime(&t);
    char buff[40];

    snprintf ( buff, 40, "%d-%d-%d %d:%d:%d",
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::ExitClick(TObject *Sender)
{
    Close();
}
//-----
void __fastcall TForm1::StartClick(TObject *Sender)
{
    Start->Enabled=false;

```

```

Continue->Enabled=false;
Stop->Enabled=true;
Timer2->Enabled=true;

PaintBox1->Canvas->Pen->Color=clRed;
i = 0;
x = 0;

PaintBox1->Canvas->MoveTo(0,60);
Panel2->Height=60;

PaintBox1->Canvas->FloodFill(50,50,clBtnFace,fsBorder);
PaintBox1->Repaint();

//PaintBox1->Canvas->LineTo(10,70);
}
//-----
void __fastcall TForm1::StopClick(TObject *Sender)
{
    Start->Enabled=true;
    Stop->Enabled=false;
    Timer2->Enabled=false;
    Continue->Enabled=true;
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    x = rand()%100;
    i = i++;
    //Label1->Caption=x;
    PaintBox1->Canvas->LineTo(i,x);

    Panel2->Height=x;

    if(i==129){ //129
        Start->Enabled=true;
        Stop->Enabled=false;
        Continue->Enabled=false;
        Timer2->Enabled=false;
        Timer2->Tag = 0;

        i = 0;
        x = 0;
        //Timer2->Tag=0;

    }
}

```

```

}
//-----

void __fastcall TForm1::ContinueClick(TObject *Sender)
{
    Timer2->Enabled=true;

    Start->Enabled=false;
    Stop->Enabled=true;
    Continue->Enabled=false;

}
//-----

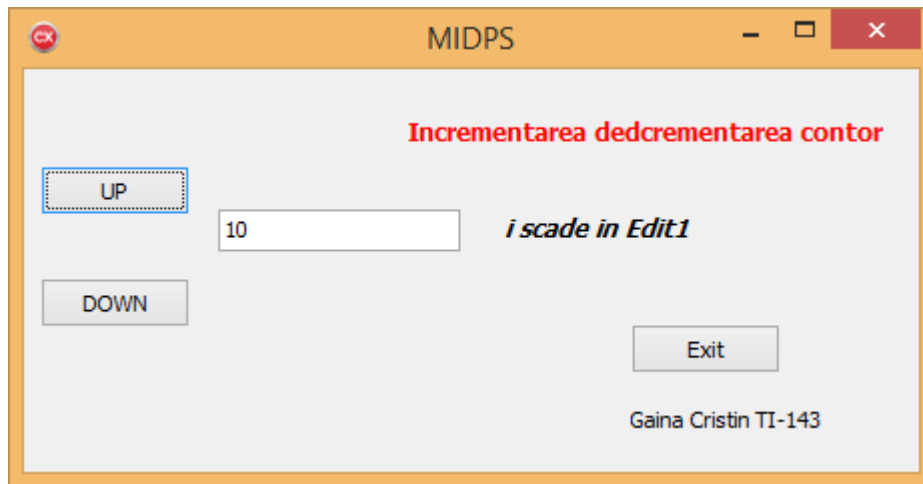
.....
//-----

#ifdef Unit1H
#define Unit1H
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.Buttons.hpp>
#include <Vcl.ExtCtrls.hpp>
//-----
class TForm1 : public TForm
{
__published: // IDE-managed Components
    TBitBtn *Start;
    TBitBtn *Stop;
    TBitBtn *Exit;
    TTimer *Timer1;
    TTimer *Timer2;
    TLabel *Label1;
    TLabel *Label2;
    TEdit *Edit1;
    TPaintBox *PaintBox1;
    TPanel *Panel1;
    TPanel *Panel2;
    TBitBtn *Continue;
    void __fastcall Timer1Timer(TObject *Sender);
    void __fastcall ExitClick(TObject *Sender);
    void __fastcall StartClick(TObject *Sender);
    void __fastcall StopClick(TObject *Sender);
    void __fastcall Timer2Timer(TObject *Sender);
    void __fastcall ContinueClick(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----

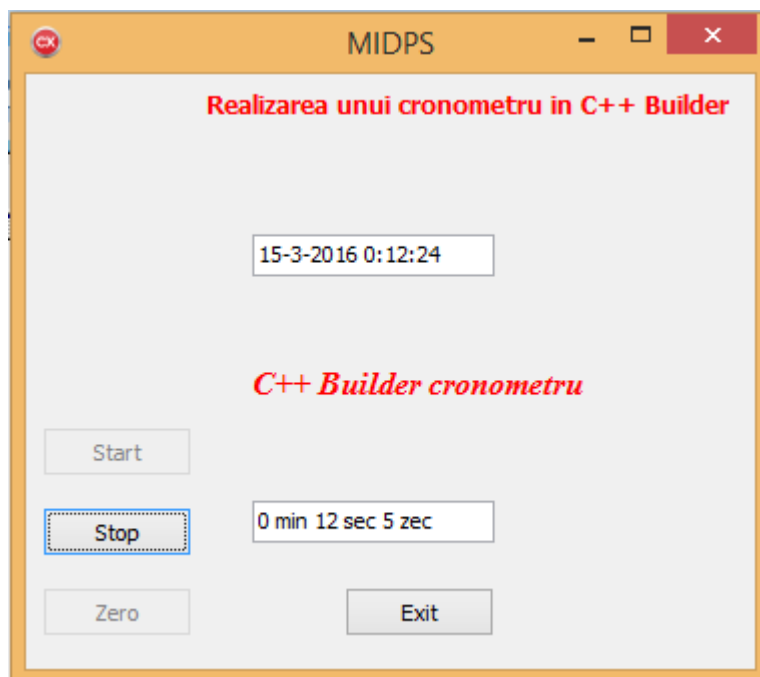
```

```
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

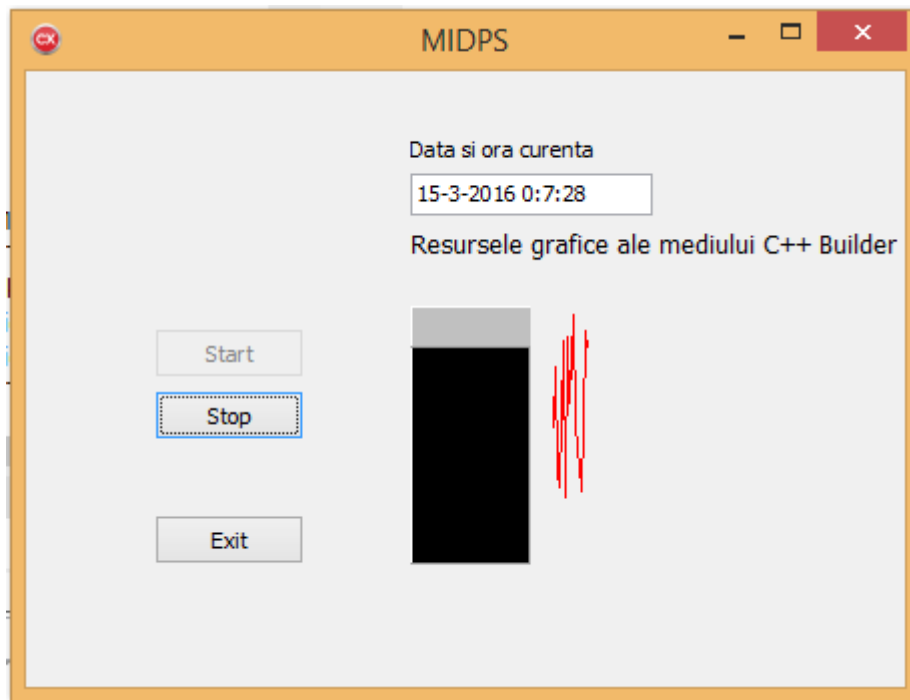
Rezultatele aplicatiei:



Aplicatia 1.



Aplicatia2.



**Aplicatai 3.**

**Concluzie:**

Efectuind aceasta lucrare de laborator am facut o mica experienta in crearea aplicatiilor grafice folosind limbajul Borland C++. Am insusit modul de utilizare a componentei VCL TTimer, utilizarea funcțiilor de lucru cu timpul sistem. Modul de utilizare a componentelor VCL TPaintBox și TPanel, utilizare a principalelor funcții grafice ale mediului C++BUILDER .