

Ministerul Educației al Republicii Moldova  
Universitatea Tehnică a Moldovei

# RAPORT

Lucrare de laborator Nr.1

*Tema:* MEDIUL INTEGRAT C++ BUILDER

A efectuat:

st. gr. TI-141  
Bulat Alexandru

A verificat:

Cojanu Irina

Chișinău 2016

## TEMA: MEDIUL INTEGRAT C++ BUILDER

**Scopul lucrării:** de a studia MEDIUL INTEGRAT C++ BUILDER

### Sarcina 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

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 scriind 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.

### Codul programului:

1. Am avut de creat cu ajutorul Borland C++ 3 programe.
2. Am inceput totul de la crearea unei programe care incrementează și decrementează un număr, folosind butoanele, meseje și Label.

Anexa1:

```
#include <vcl.h>
```

```
#pragma hdrstop
```

```
#include "lab1A.h"
```

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

```
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
```

```
TForm1 *Form1;
```

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

```
__fastcall TForm1::TForm1(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
    Edit1->Text = 1;
```

```
}
```

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

```
{
```

```
    Close();
```

```
}
```

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

```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{
```

```
    //Transformam textul din Edit1 in integer, iar daca
```

```
    //acesta nu-i numar, atunci va fi returnat 0
```

```

    int i = Edit1->Text.ToIntDef(0);

    Edit1->Text = ++i;

    Label2->Caption = "i a fost incrementat";
}

//-----

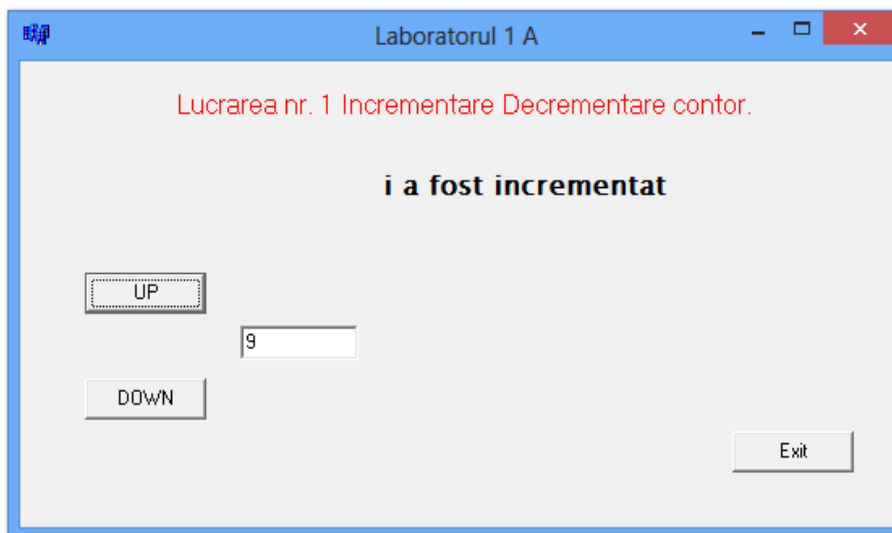
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    int i = Edit1->Text.ToIntDef(0);

    Edit1->Text = --i;

    Label2->Caption = "i a fost decrementat";
}

//-----

```



3) Se elaborează un program pentru realizarea unui cronometru.

**Anexa 2:**

```

//-----

```

```

#include <vcl.h>

```

```

#pragma hdrstop

```

```

#include "lab1Bp3.h"

```

```

#include <stdio.h>

//-----

#pragma package(smart_init)

#pragma resource "*.dfm"

TForm1 *Form1;

//-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    //completam deodata Edit1 cu ora curenta, ca sa nu fie pauza

    char buff[20];

    getdate(&date_);
    gettime(&time_);

    sprintf(buff, "%02d-%02d-%04d %02d:%02d:%02d",
        date_.da_day, date_.da_mon, date_.da_year,
        time_.ti_hour, time_.ti_min, time_.ti_sec);

    Edit1->Text = (AnsiString)buff;

    //setam timerul la 00:00:00

    timer_.ti_min = 0;

    timer_.ti_sec = 0;

    timer_.ti_hund = 0;

    //afisam in Edit2 00 min 00 sec 0 zec, din simpla dorinta

    printTimerToEdit2();

    //dezactivam timerul pentru inceput

    Timer2->Enabled = false;

```

```

}

//-----

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

    Button1->Enabled = false; //butonul start

    Button3->Enabled = false; //butonul reset
}

//-----

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

    Button1->Enabled = true;

    Button3->Enabled = true;
}

//-----

void __fastcall TForm1::ResetBtnClick(TObject *Sender)
{
    //setam timerul la 00:00:00

    timer_.ti_min = 0;

    timer_.ti_sec = 0;

    timer_.ti_hund = 0;

    printTimerToEdit2();
}

//-----

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

```

```

}

//-----

void __fastcall TForm1::DateTimer1Timer(TObject *Sender)
{
    char buff[20];

    getdate(&date_);

    gettime(&time_);

    sprintf(buff, "%02d-%02d-%04d %02d:%02d:%02d",

        date_.da_day, date_.da_mon, date_.da_year,

        time_.ti_hour, time_.ti_min, time_.ti_sec);

    Edit1->Text = (AnsiString)buff;
}

//-----

void __fastcall TForm1::TimeTimer2Timer(TObject *Sender)
{
    if (++timer_.ti_hund == 10)
    {
        timer_.ti_hund = 0;

        if (++timer_.ti_sec == 60)
        {
            timer_.ti_sec = 0;

            if (++timer_.ti_min == 60)
            {
                timer_.ti_min = 0;
            }
        }

        printTimerToEdit2();
    }

}

//-----

```

```

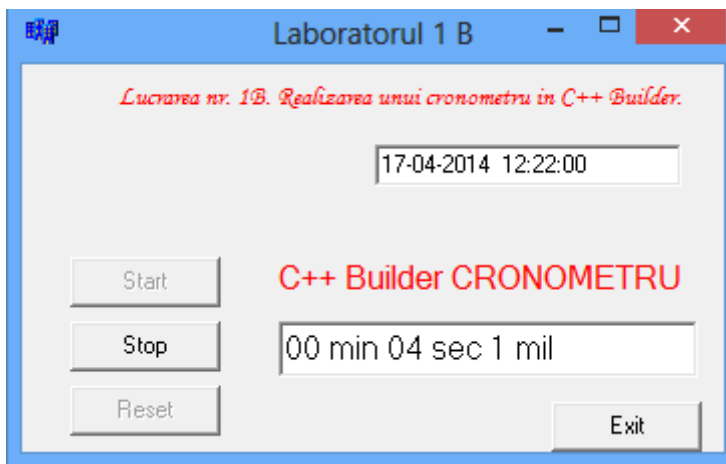
void TForm1::printTimerToEdit2() const
{
    char buff[20];

    sprintf(buff, "%02d min %02d sec %d zec",
        timer_.ti_min, timer_.ti_sec, timer_.ti_hund);

    Edit2->Text = (AnsiString)buff;
}

//-----

```



4).Se elaborează un program pentru realizarea a două elemente de afișare

### Anexa 3

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "lab1C.h"
#include <stdio.h>
#include <time.h>

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{

```



```

        DrawTimer->Enabled = false;
        BlackPanel->Height = WhitePanel->Height;
        StartButton->Enabled = true;
        StopButton->Enabled = false;
        srand(time(0));
    }
    //-----

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

void __fastcall TForm1::StartButtonClick(TObject *Sender)
{
    DrawTimer->Enabled = true;
    StartButton->Enabled = false;
    StopButton->Enabled = true;
}
//-----

void __fastcall TForm1::StopButtonClick(TObject *Sender)
{
    DrawTimer->Enabled = false;
    StartButton->Enabled = true;
    StopButton->Enabled = false;
}
//-----

void __fastcall TForm1::DateTimerActive(TObject *Sender)
{
    char buff[20];
    getdate(&date_);
    gettime(&time_);
    sprintf(buff, "%02d-%02d-%04d %02d:%02d:%02d",
        date_.da_day, date_.da_mon, date_.da_year,
        time_.ti_hour, time_.ti_min, time_.ti_sec);

    DateEdit->Text = (AnsiString)buff;
}
//-----

void __fastcall TForm1::DrawTimerActive(TObject *Sender)
{
    PaintBox->Repaint();
}

```

```

        DrawBackGrid();
        ChangePanels();
        DrawStatusLines();
    }
    //-----

```

```

void TForm1::DrawBackGrid() const
{
    PaintBox->Canvas->Pen->Color = clBlack;
    PaintBox->Canvas->Pen->Width = 1;
    PaintBox->Canvas->Pen->Style = psSolid;

    for (int x = 0; x < PaintBox->Width; x += 10)
    {
        PaintBox->Canvas->MoveTo(x, 0);
        PaintBox->Canvas->LineTo(x, PaintBox->Height);
    }

    for (int y = 0; y < PaintBox->Height; y += 10)
    {
        PaintBox->Canvas->MoveTo(0, y);
        PaintBox->Canvas->LineTo(PaintBox->Width, y);
    }
}
//-----

```

```

void TForm1::ChangePanels() const
{
    BlackPanel->Height = rand()%WhitePanel->Height;
}
//-----

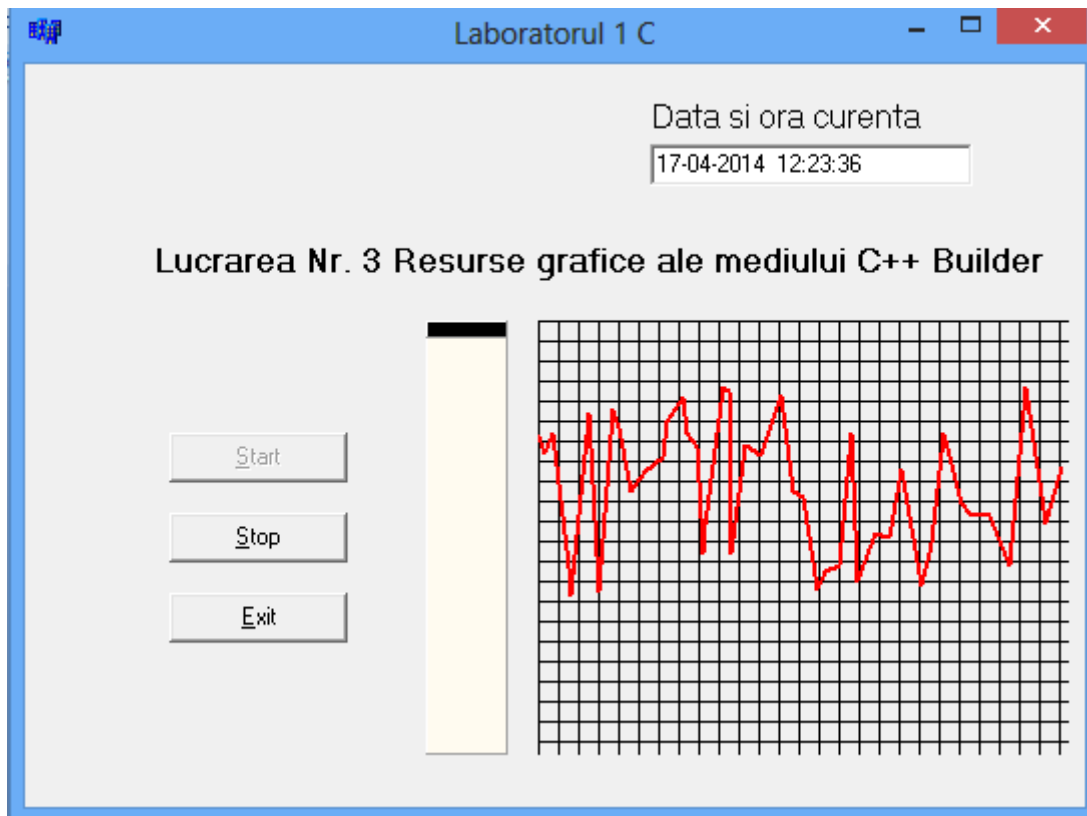
```

```

void TForm1::DrawStatusLines() const
{
    PaintBox->Canvas->Pen->Color = clRed;
    PaintBox->Canvas->Pen->Width = 2;
    PaintBox->Canvas->Pen->Style = psSolid;

    PaintBox->Canvas->MoveTo(0, rand()%(PaintBox->Height)/2+30);
    for (int x = rand()%10+1; x < PaintBox->Width; x += rand()%10+1)
    {
        PaintBox->Canvas->LineTo(x, rand()%(PaintBox->Height)/2+30);
    }
}
//-----

```



**Concluzie:** In urma efectuării lucrării de laborator nr.1 am lucrat in MEDIUL INTEGRAT C++ BUILDER unde am facut cunostinta cu o noua metoda de a crea anumite programe prin modelarea lor cu unele ustensile deja gata.

Pe parcursul lucrării am realizat 3 programe , una din programe este folosirea operatiilor de Incrementare si decrementare a unui numar, scopul celei de a doua programe este de a afisa oora curenta si de a ne folosi de un cronometru, pentru a ne folosi de el am introdus butoanele start,stop si reset cee ace ne permite sa operam cu cronometrul.ultimul program a avut ca scop sa ne permita lucrul cu graficul si a afişa ora curentă .Datorita programului 3 am vazut resursele grafice ale programului C++ BUILDER

**Bibliografie:**

Lectii video

Indicatii metodice <http://moodle.ati.utm.md/mod/assign/view.php?id=987>