

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Automatica și Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.1

la disciplina:

Medii Interactive de Dezvoltare a Produselor Soft

A efectuat: st. gr. TI-145

A verificat: lect.univ.

Zubenco Dmitri.

Cojanu I.

Chișinău 2016

Lucrarea de Laborator Nr.1

Mediu 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).

Facilitățile mediului C++Builder

Borland C++ Builder este un mediu de programare vizual, orientat pe obiecte, 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 sciind 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.

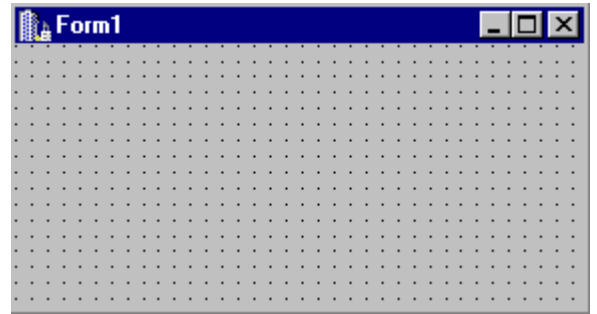
1 IDE (Mediul de Dezvoltare Integrat)

Elementele mediului integrat de dezvoltare sunt:

- Meniu principal (Main Menu);
- Forma (Form);
- Editorul de cod (Code Editor);
- Bara cu instrumente (Toolbar);
- Paleta cu componente (Component Palette);
- Tabelul cu proprietăți ale obiectelor (Object Inspector);
- Administratorul de program (Program Manager).

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.

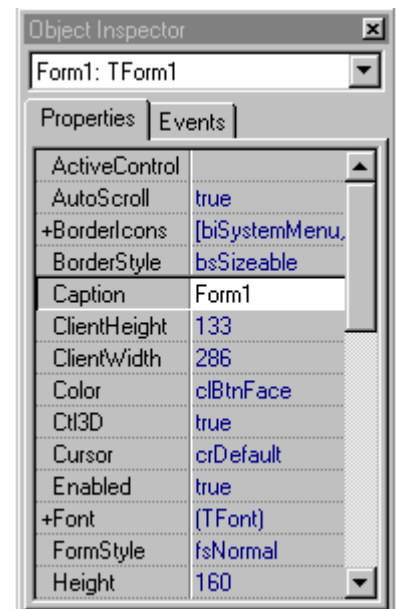


Proprietăți, metode, evenimente

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.



Efectuarea programelor din sarcina lucrării în mediul C++ Builder

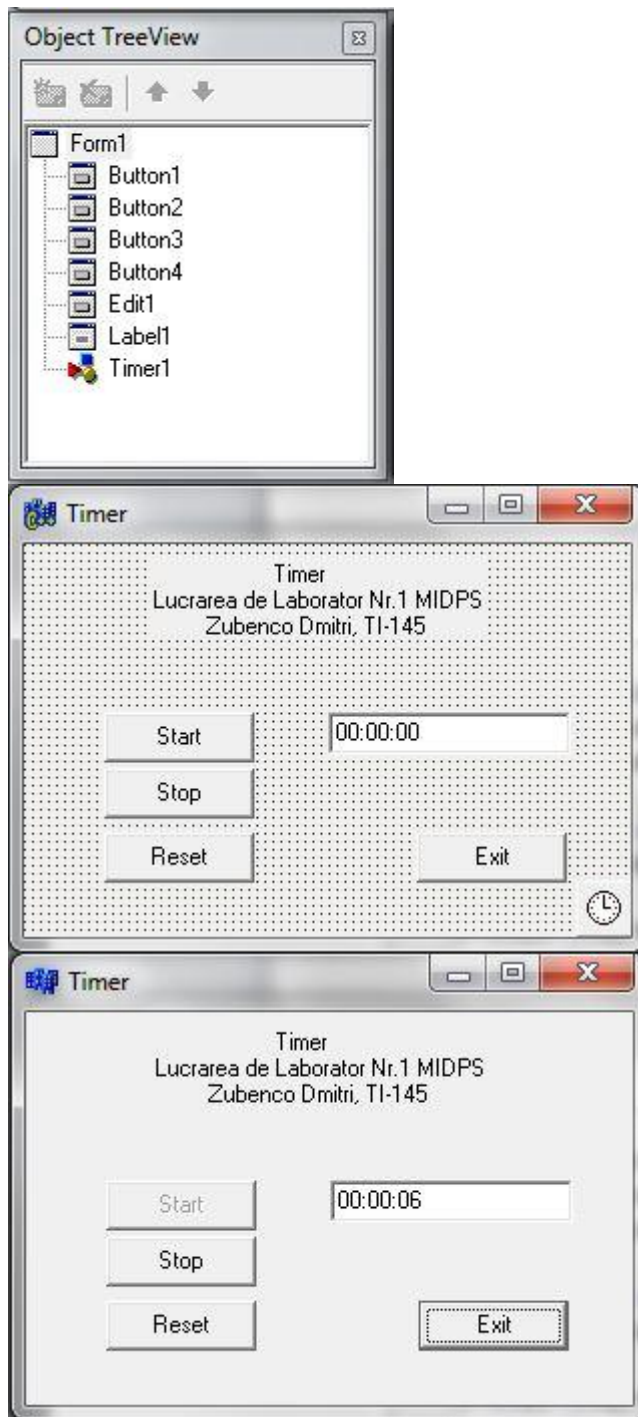
a) timer

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
    Timer1->Enabled=False;  
}  
int i=0,x1=0,x2=0,x3=0,x4=0,x5=0;  
//-----  
  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    Close();  
}  
//-----  
void __fastcall TForm1::Timer1Timer(TObject *Sender)  
{  
    i++;  
    if(i==10){ x1++; i=0; }  
    if(x1==6) { x2++; i=0; x1=0; }  
    if(x2==10) { x3++; x2=0; }  
    if(x3==6) { x4++; x3=0; x2=0; }  
    if(x4==10) { x5++; x4=0; }  
    if(x5==2 && x4==4) { x5=0; x4=0; i=0; }  
  
    Edit1->Text=IntToStr(x5)+IntToStr(x4)+":"+IntToStr(x3)+IntToStr(x2)+":"+IntToStr(x1)+IntToStr(i);  
}  
//-----  
void __fastcall TForm1::Button2Click(TObject *Sender)  
{  
    Timer1->Enabled=True;  
    Button2->Enabled=False;  
    Button3->Enabled=True;  
}  
//-----  
void __fastcall TForm1::Button3Click(TObject *Sender)  
{  
    Timer1->Enabled=False;  
    Button2->Enabled=True;  
    Button3->Enabled=False;  
}  
//-----  
void __fastcall TForm1::Button4Click(TObject *Sender)  
{  
    i=0;x1=0;x2=0;x3=0;x4=0;x5=0;  
    Edit1->Text="00:00:00";
```

```

}
//-----

```



b) grafic

```

//-----

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

#include "Unit1.h"
//-----

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

```

```

int width;
int height;
int x;
int y;

TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    stop->Enabled = false;
    //srand(time(NULL));
}
//-----

void __fastcall TForm1::PaintBox1Paint(TObject *Sender)
{
    PaintBox1->Canvas->Pen->Color = clBlack;
    PaintBox1->Canvas->Brush->Color = clBlack;
    PaintBox1->Canvas->Brush->Style = bsCross;
    PaintBox1->Canvas->Rectangle(0, 0, PaintBox1->Width, PaintBox1->Height);

}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Close();
}
//-----
void DrawLine()
{
    y = (height / 2.0) + (rand() % 41 - 20);
    Form1->PaintBox1->Canvas->LineTo(++x, y);
    Form1->Panel2->Height = y;
}
void __fastcall TForm1::stopClick(TObject *Sender)
{
    Timer1->Enabled = false;
    start->Enabled = true;
    stop->Enabled = false;

}
//-----
void __fastcall TForm1::startClick(TObject *Sender)
{
    PaintBox1->Repaint();
    PaintBox1->Canvas->Pen->Color = clRed;
    width = Form1->PaintBox1->Width;
    height = Form1->PaintBox1->Height;
    x = 0;
    PaintBox1->Canvas->MoveTo(0, height / 2.0);
    stop->Enabled = true;
    start->Enabled = false;
    Timer1->Enabled = true;

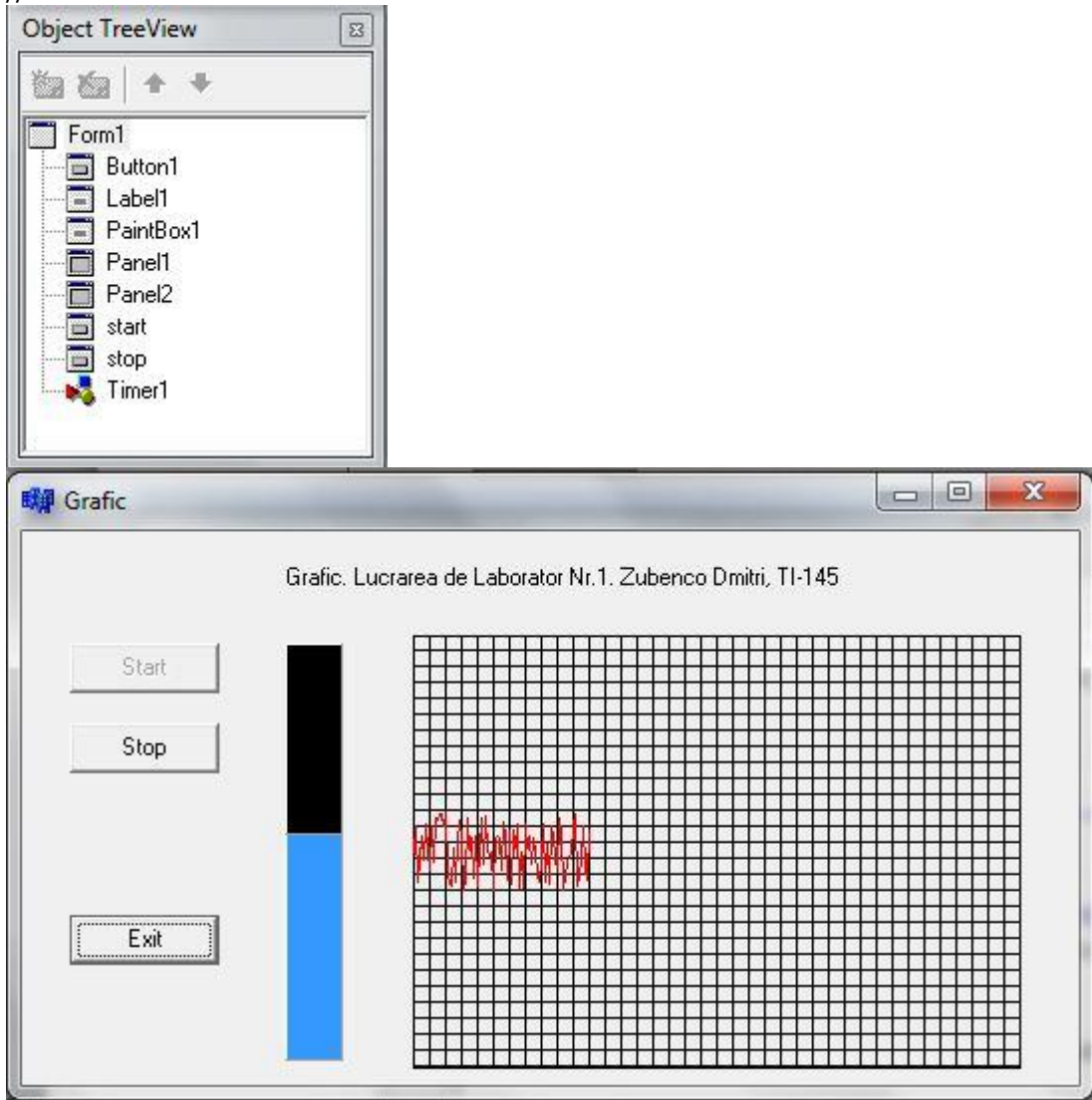
}
//-----

```

```

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    if(x == width)
        stopClick(Sender);
    else DrawLine();
}
//-----

```



c) counter with increment/decrement

```

//-----

#include <vcl.h>
#pragma hdrstop

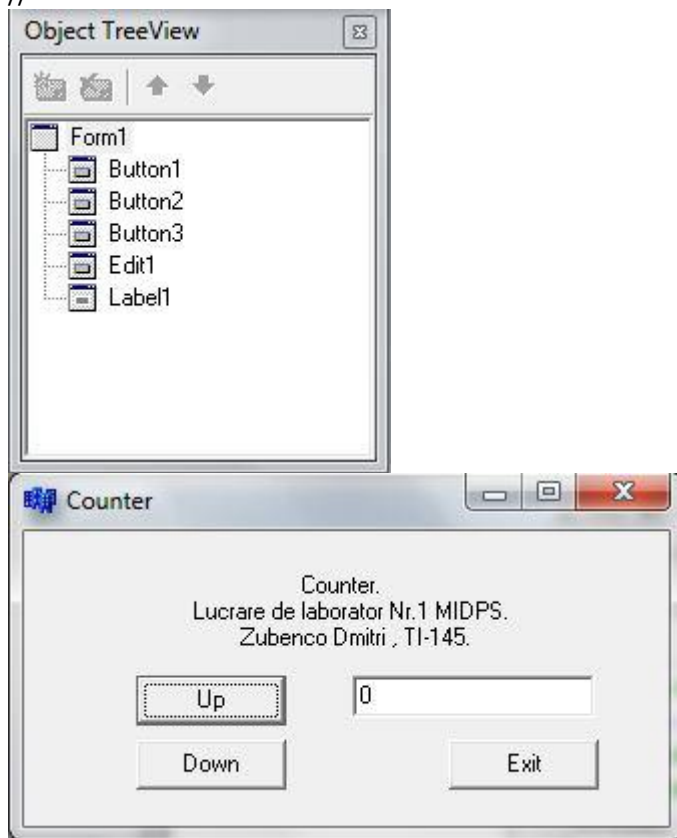
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}

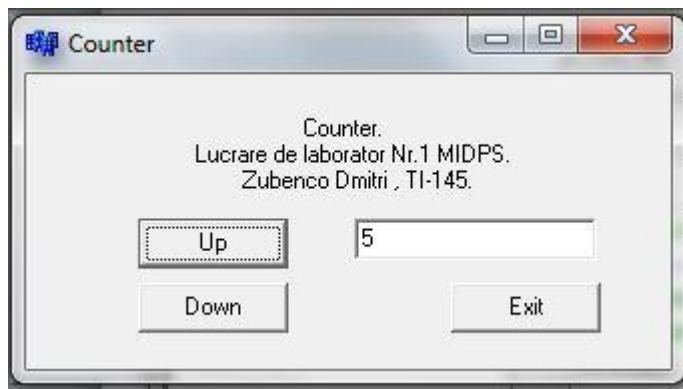
```

```
//-----
int x=0;
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Close();
}
//-----

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

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





Concluzie

În urma realizării acestei lucrări de laborator, am făcut cunoștință cu mediul de dezvoltare C++ Builder, am aflat cum funcționează componentele TButton, Ttimer, Label, Edit, PaintBox, Panel etc. Am însușit modul de utilizare a principalelor funcții grafice ale mediului C++BUILDER . Aceste funcții facilitează lucrul programatorului și crește productivitatea. Am aflat cum funcționează funcțiile de lucru cu timpul sistem. C++ Builder este destul de performant și conține tot de ce e nevoie pentru a crea o fereastra ce stă la baza unui program.