

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

Tema: «Mediu intergrat C++ Builder»

A efectuat: st. gr. TI-145

A verificat: lect.univ.

Pușcaș Iana.

Cojanu I.

Chișinău 2015

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 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.

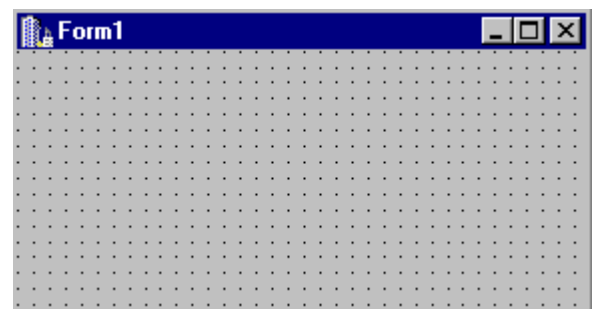
1 IDE (Mediul de Dezvoltare Integrat)

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.

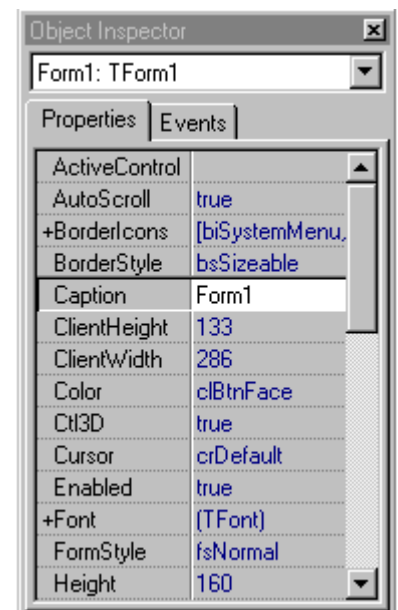


Proprietăți, metode, evenimente

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

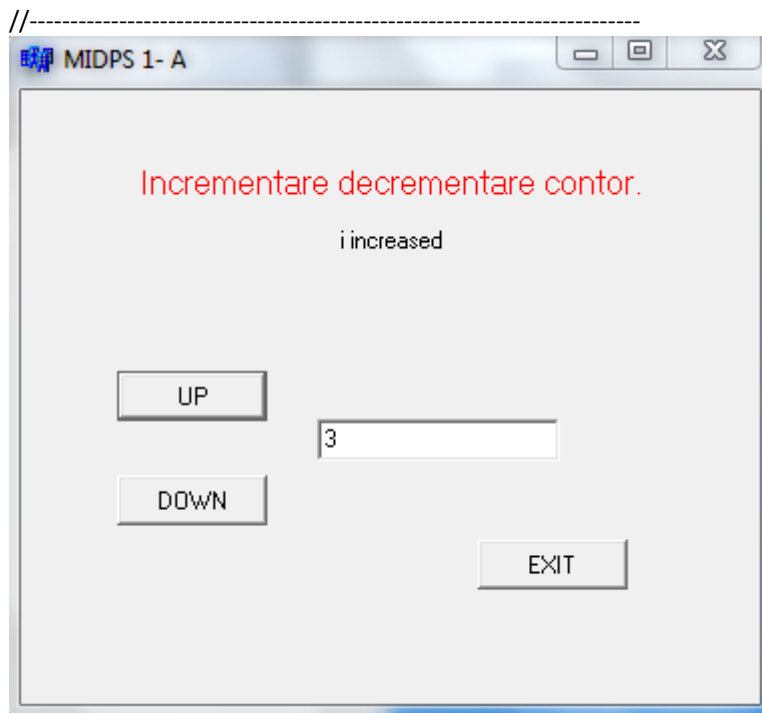
Mediul C++Builder oferă un ghid practic, care conține peste 3000 de pagini de documentație despre IDE, VCL, baze de date și tehnici de programare.



Efectuarea programelor din sarcina lucrării în mediul C++ Builder

a) incrementare

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma link "sButton"  
#pragma link "sEdit"  
#pragma link "sLabel"  
#pragma resource "*.dfm"  
TForm1 *Form1;  
int i = 0;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::sButton3Click(TObject *Sender)  
{  
    Close();  
}  
//-----  
void __fastcall TForm1::sButton1Click(TObject *Sender)  
{  
    sEdit1->Text = ++i;  
    sLabel2->Caption = "i increased";  
}  
//-----  
void __fastcall TForm1::sButton2Click(TObject *Sender)  
{  
    sEdit1->Text = --i;  
    sLabel2->Caption = "i decreased";  
}
```



b) cronometru

```
//-----

#include <vcl.h>
#pragma hdrstop

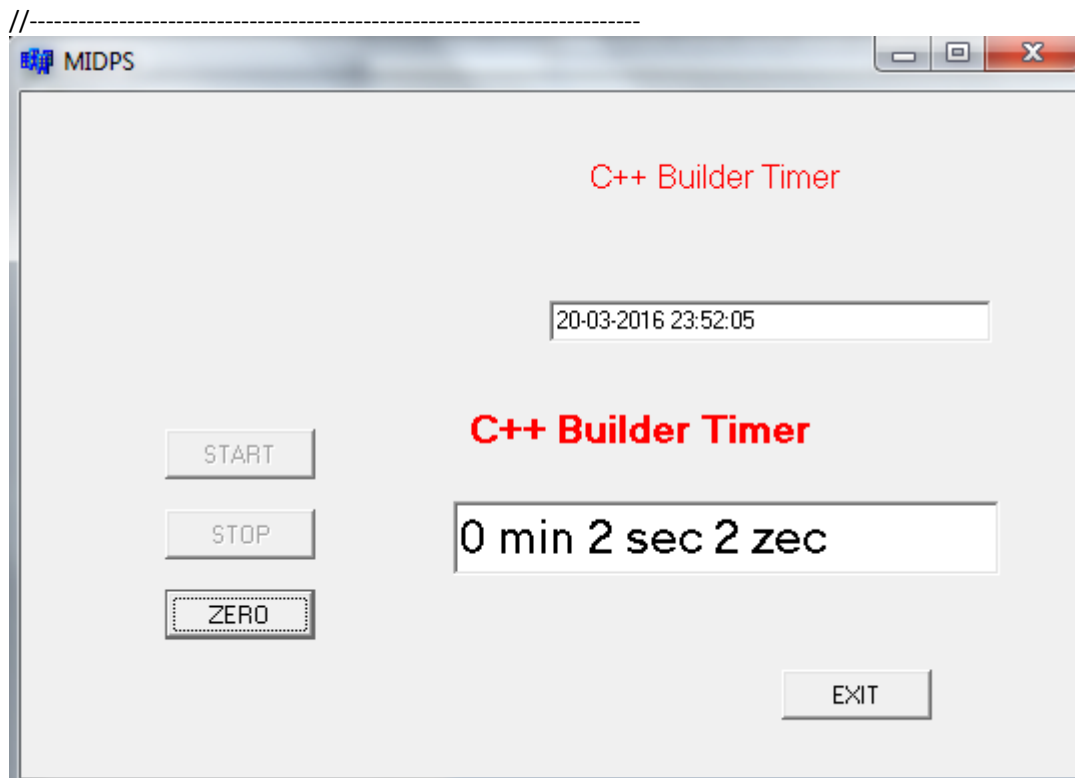
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma link "sButton"
#pragma link "sEdit"
#pragma link "sLabel"
#pragma resource "*.dfm"
#include <stdio.h>
#include <iostream>
#include <string.h>
using namespace std;
#include "dos.h"

TForm1 *Form1;
struct date d;
struct time t;
int i = 0;

//-----
__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);
sEdit2->Text=(AnsiString)buf;
```

```
}
//-----
void __fastcall TForm1::sButton1Click(TObject *Sender)
{
    Timer2->Enabled = true;
    sButton1->Enabled = false;
    sButton2->Enabled = true;
    sButton3->Enabled = false;
}
//-----
void __fastcall TForm1::sButton2Click(TObject *Sender)
{
    Timer2->Enabled = false;
    sButton1->Enabled = false;
    sButton2->Enabled = false;
    sButton3->Enabled = true;
}
//-----
void __fastcall TForm1::sButton3Click(TObject *Sender)
{
    i = 0;
    sEdit1->Text = "0 min 0 sec 0 zec";
    sButton1->Enabled = true;
    sButton2->Enabled = false;
    sButton3->Enabled = false;
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    i++;
    AnsiString m=i/600;
    AnsiString s=(i%600)/10;
    AnsiString z=i%10;
    sEdit1->Text = ""+m+" min "+s+" sec "+z+" zec";
}
//-----
void __fastcall TForm1::sButton4Click(TObject *Sender)
{
    Close();
}
```



c) grafic

```
//-----

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

#include "Unit1.h"
#include "Dos.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
struct time t;
struct date d;
int width;
int height;
int x;
int y;

TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Button2->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::Button3Click(TObject *Sender)
```

```
{
Close();
}
//-----
```

```
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 DrawLine()
```

```
{
y = (height / 2.0) + (rand() % Form1->PaintBox1->Height/4 - Form1->PaintBox1->Height/8);
Form1->PaintBox1->Canvas->LineTo(++x, y);
Form1->Panel1->Height = y;
}
```

```
void __fastcall TForm1::Button1Click(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);
Button2->Enabled = true;
Button1->Enabled = false;
Timer2->Enabled = true;

}
//-----
```

```
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```
{
Timer2->Enabled = false;
Button1->Enabled = true;
Button2->Enabled = false;

}
//-----
```

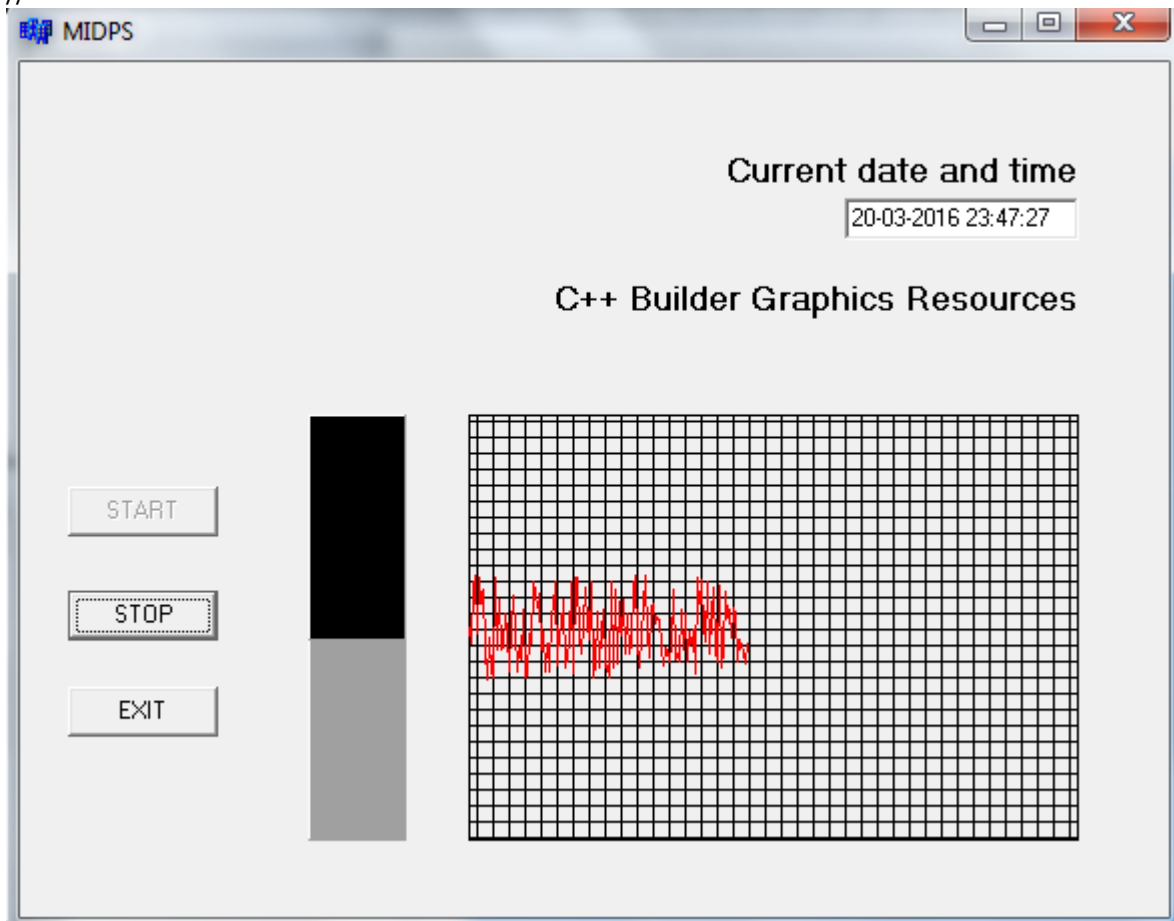
```
void __fastcall TForm1::Timer2Timer(TObject *Sender)
```



```

{
if(x == width)
    Button2Click(Sender);
else DrawLine();
}
//-----

```



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 fereastră ce stă la baza unui program.