

Ministerul Educatiei si Tineretului al Republicii Moldova
Universitatea Tehnica a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică

RAPORT

LA LUCRAREA DE LABORATOR NR. 1

Disciplina : MIDPS

TEMA: „Componentele mediului integrat C++ Builder.

Funcții pentru gestionarea resurselor grafice ale mediului C++ Builder”

A efectuat :

st. gr. TI-142 Donca Diana

A verificat:

Cojan Irina

Chișinău 2016

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.

Înaltă 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-urile* 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 mesaje 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.

Sarcina 1:

```
#include <vcl.h>
#pragma hdrstop
#include<stdio.h>
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent*
Owner)
: TForm(Owner)
{
//Edit1->Text=0;
Form1->Name="MIDPS";
Label1->Caption="Incrementare
decrementare contor" ;
Label1->Font->Color=clRed;
}

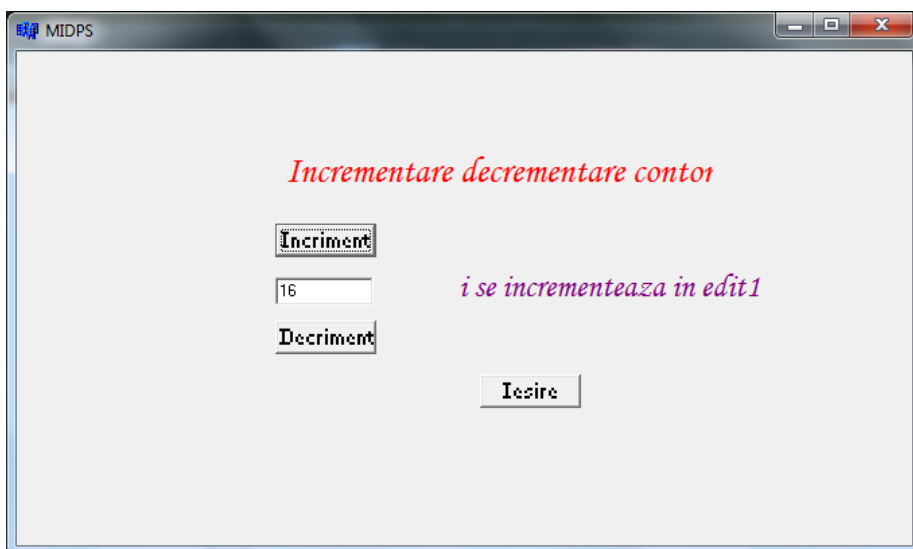
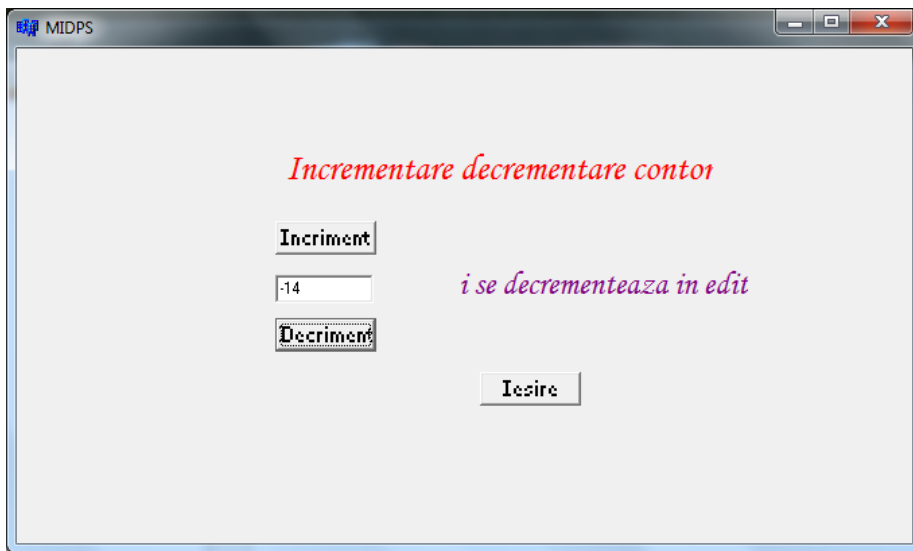
void __fastcall
TForm1::Button1Click(TObject *Sender)
{ char *s=Edit1->Text.c_str();
int i=atoi(s);
i=i+1;
Edit1->Text=i;
Label2->Caption="i se incrementeaza in
edit1";

}
//-----

void __fastcall
TForm1::Button2Click(TObject *Sender)
{ char *s=Edit1->Text.c_str();
int i=atoi(s);
i=i-1;
Edit1->Text=i;
Label2->Caption="i se decrementeaza in
edit";
}
//-----

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

Rezultatul rularii programului nr.1



Sarcina 2:

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

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

//-----
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
struct date d;
struct time t;

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

```
__fastcall TForm1::TForm1(TComponent*
Owner)
: TForm(Owner)
{ Button2->Enabled=false;
  Button3->Enabled=false;
}

//-----
//-----

void __fastcall
TForm1::Button1Click(TObject *Sender)
{ Timer1->Enabled=true;

  Button1->Enabled=false;
  Button2->Enabled=true;
  Button3->Enabled=false;
```

```

}
//-----

void __fastcall
TForm1::Button2Click(TObject *Sender)
{ Timer1->Enabled=false;
  Button1->Enabled=true;
  Button3->Enabled=true;
}
//-----

void __fastcall
TForm1::Button3Click(TObject *Sender)
{ Edit2->Text=0;
  Timer1->Tag=0;
  Button2->Enabled=false;
}
//-----

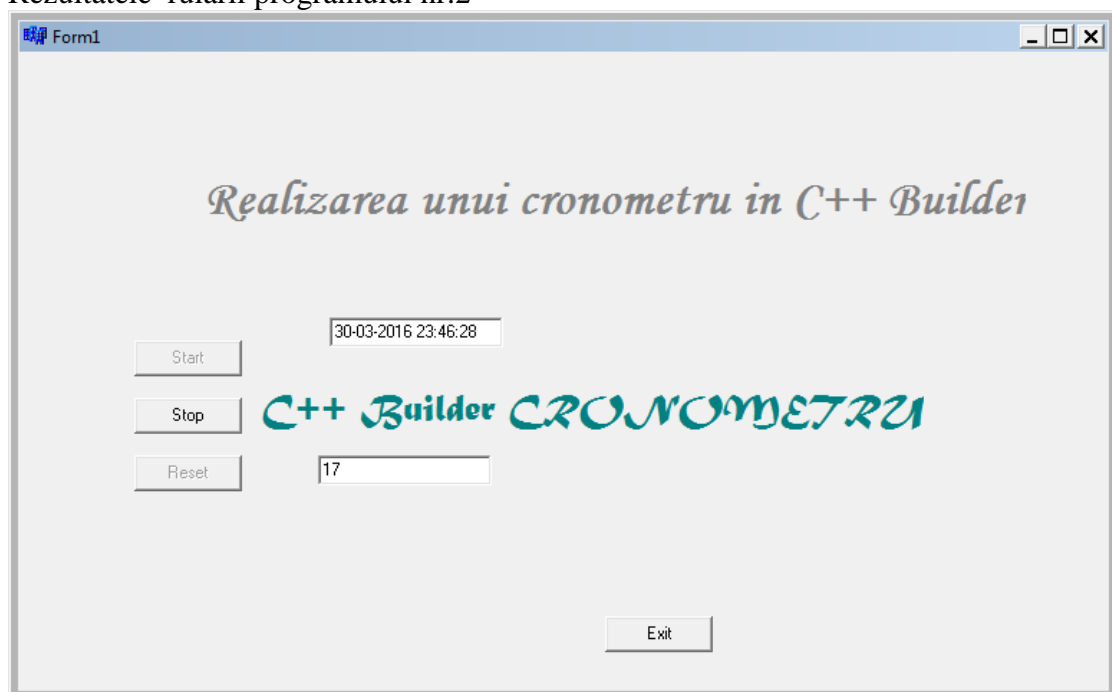
void __fastcall
TForm1::Button4Click(TObject *Sender)
{
  Close();
}

//-----
void __fastcall
TForm1::Timer1Timer(TObject *Sender)
{ int i;
  i=Timer1->Tag;
  i++;
  Timer1->Tag=i;
  Edit2->Text=Timer1->Tag;
}
//-----

void __fastcall
TForm1::Timer2Timer(TObject *Sender)
{
  char buf[20];
  getdate(&d);
  gettime(&t);
  sprintf(buf,"%02d-%02d-%4d
%02d:%02d:%02d",d.da_day,d.da_mon,d.d
a_year,
t.ti_hour,t.ti_min,t.ti_sec);
  Edit1->Text=(AnsiString)buf;
}
//-----

```

Rezultatele rularii programului nr.2



Sarcina 3:

```
//-----

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

#include "Unit1.h"
#include "dos.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
struct date data;
struct time timp;
int i, x;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    exit(1);
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Edit1->Clear();
    Timer1->Enabled=true;
    Timer2->Enabled=false;
    Timer1->Interval=1000;
    Timer2->Interval=500;
    Button2->Enabled=false;
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    char buffer[20];
    getdate(&data);
    gettime(&timp);
    sprintf(buffer,"%02d/%02d/%4d %2d:%02d:%02d", data.da_day, data.da_mon,
    data.da_year, timp.ti_hour, timp.ti_min, timp.ti_sec);

    Edit1->Text=(AnsiString)buffer;
}
//-----
```

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Button1->Enabled=false;
    Button2->Enabled=true;
    Timer2->Enabled=true;

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

    PaintBox1->Canvas->MoveTo(0,100);

    PaintBox1->Canvas->FloodFill(50,50,clBtnFace,fsBorder);
    PaintBox1->Repaint();
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Button2->Enabled=false;
    Button1->Enabled=true;
    Timer2->Enabled=false;
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    x = rand()%100;
    i = i++;
    PaintBox1->Canvas->LineTo(i,x);

    Panel2->Height=x;

    if(i==100)
    {
        Button1->Enabled=true;
        Button2->Enabled=false;
        Timer2->Enabled=false;
        Timer2->Tag = 0;

        i = 0;
        x = 0;
    }
}
//-----

```

Rezultatul rularii programului nr.3



Concluzie: În această lucrare de laborator am însușit modul de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER. Am realizat un program simplu care utilizează componente de tip *TButton*, *TEdit*, *Tlabel*, *RadioButton* etc. am însușit modul de utilizare a componentei VCL **Ttimer**, VCL **TPaintBox** și **TPanel**. Și am realizat o afișare grafică a informației: diagramă și bargraf.