

Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatica și Microelectronică
Catedra Automatică și Tehnologii Informaționale

Raport

Disciplina: Medii interactive de dezvoltare a produselor soft

Lucrare de laborator Nr. 1

A efectuat: st.gr. TI-144

Olaru Cristian

A verificat: lect. sup.

Cojocaru Svetlana

Chisinau 2016

Lucrare de laborator Nr. 1

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 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.

Efectuarea Lucrării :

A)

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

#include "a.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int i=0;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Form1->Caption="MIDPS 1-A";
    Edit1->Text="0";
    Label2->Caption="";
}
//-----

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

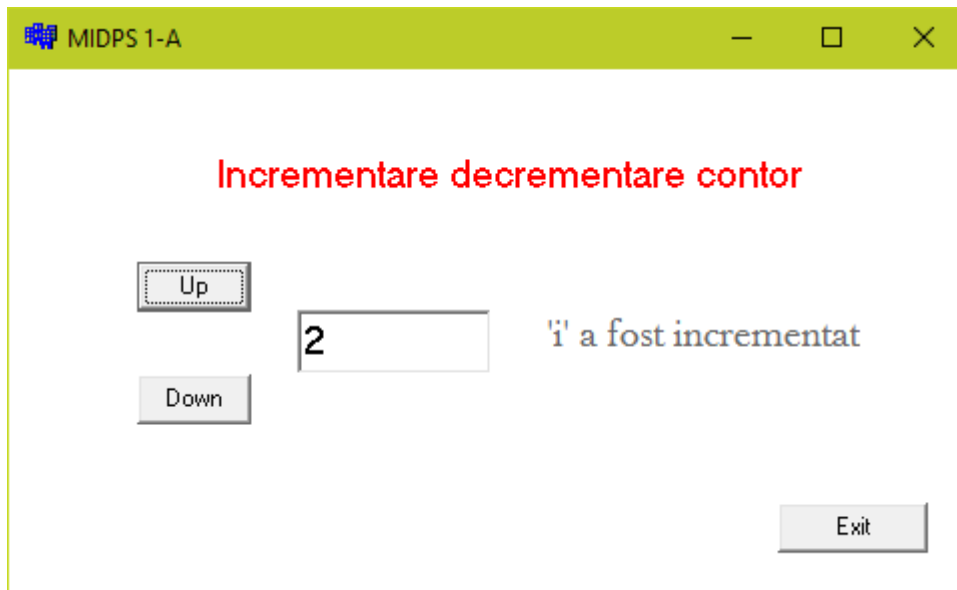
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    i=Edit1->Text.ToInt();
    Label2->Caption=" 'i' a fost incrementat";
    i++ ;
    Edit1->Text=(i);
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    i=Edit1->Text.ToInt();
    Label2->Caption=" 'i' a fost decrementat";
    i--;
```

```
Edit1->Text=(i);  
}  
//-----
```

Rezultat:

Incrementare:



Incrementare decrementare contor

Up

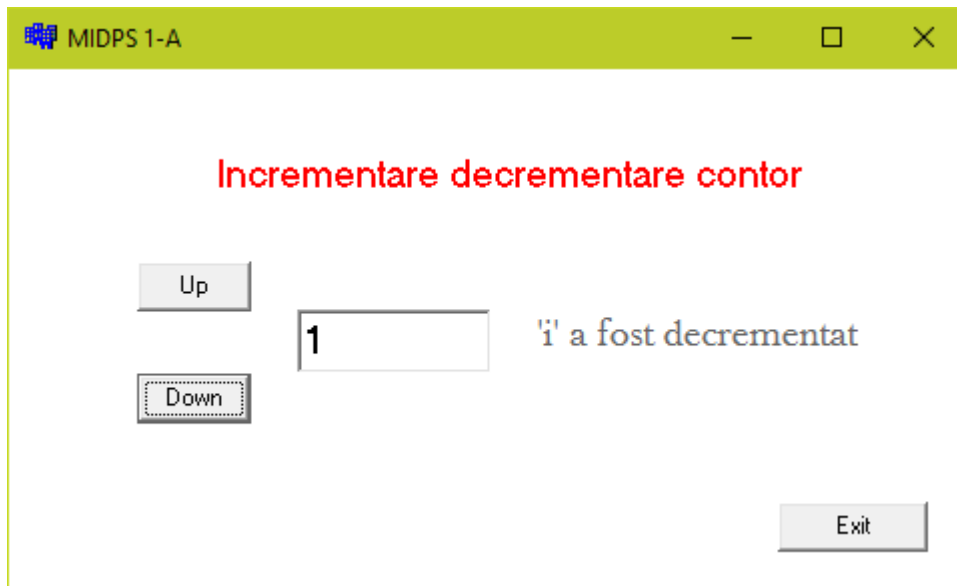
Down

2

i' a fost incrementat

Exit

Decrementare:



Incrementare decrementare contor

Up

Down

1

i' a fost decrementat

Exit

B)

```
//-----

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

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int zec=0,min,sec;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Timer1->Enabled=false;
    Form1->Caption="MIDPS";
    Edit2->Text=" ";
    Edit1->Text="00 min 00sec 00zec ";
}
//-----

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

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    zec++;
    if(zec==10)
    {
        zec=0;
        sec++;
    }
    if(sec==60)
    {
        zec=0;sec=0;
        min++;}
    Edit1->Text=AnsiString(min)+" min "+ AnsiString(sec)+" sec "+ AnsiString(zec)+" zec ";
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Timer1->Enabled=false;
}
```

```
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Timer1->Enabled=false;
    zec=sec=min=0;
    Edit1->Text="00 min 00 sec 00 zec";

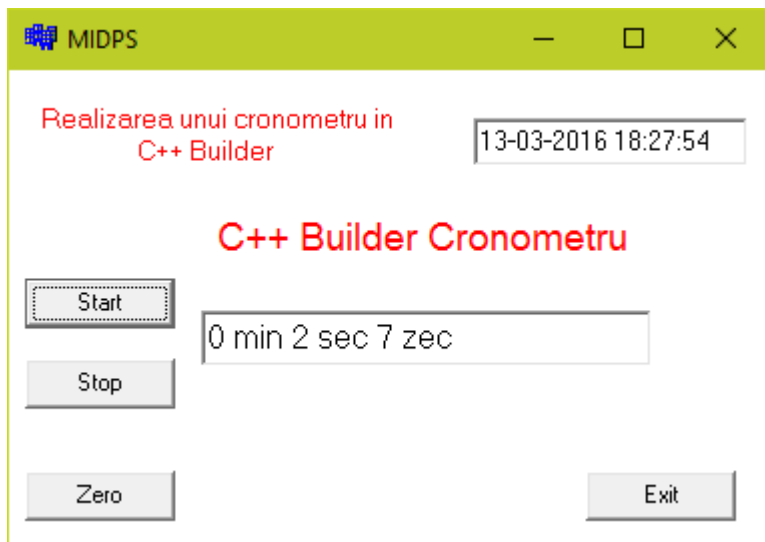
}
//-----

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

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    char buf[20];
    struct date d;
    struct time t;
    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);
    Edit2->Text=(AnsiString)buf;

}
//-----
```

Rezultat:



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 x0=0 ,y0=0,x1=0,y1=0;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    char  buf[20];
        struct date d;
        struct time t;
        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 __fastcall TForm1::Button1Click(TObject *Sender)
{
    PaintBox1->Repaint();
    PaintBox1->Canvas->Pen->Color = clRed;
    x1 = Form1->PaintBox1->Width;
    y1 = Form1->PaintBox1->Height;
    x0 = 0;
    Form1->PaintBox1->Canvas->MoveTo(0, y1 / 2.0);
    Button1->Enabled = false;
    Button2->Enabled = true;
    Timer2->Enabled = true;
}
//-----

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

```

}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Button1->Enabled = true;
    Button2->Enabled = false;
    Timer2->Enabled = false;

}
//-----

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);
}
//-----

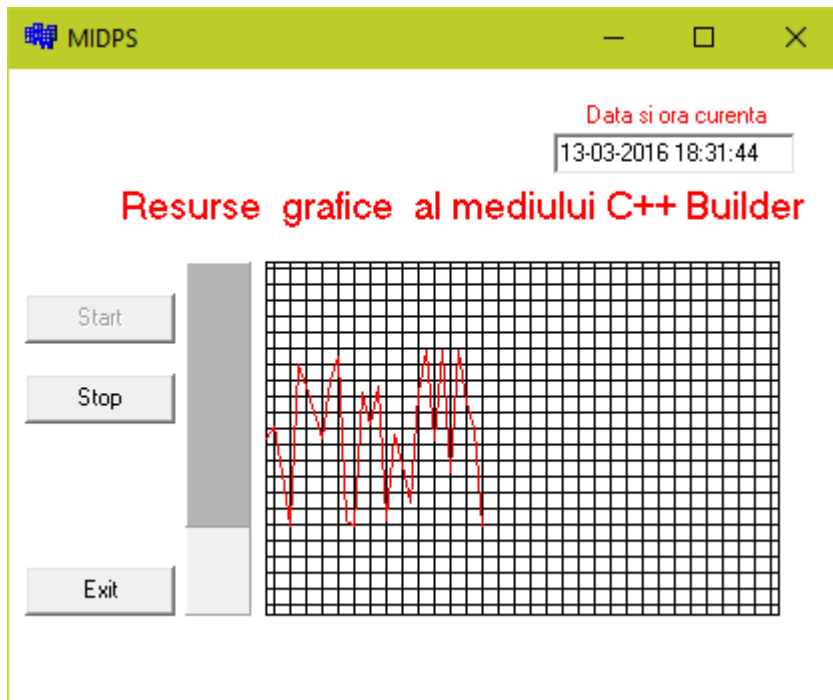
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    y0 = (y1 / 2.0) + (rand() % 91 - 45);
    x0 += 4 ;
    Form1->PaintBox1->Canvas->LineTo(x0, y0);
    Form1->Panel2->Height = y0;
    if(x0 > x1)
    {
        Button2Click(Sender);
    }
}
//-----

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

}
//-----

```


Rezultat:



Concluzie:

În urma efectuării acestui laborator, am făcut cunoștință mediul de dezvoltare C++ Builder, am înțeles cum funcționează componentele TButton, TTimer, Label, Edit, PaintBox, Panel etc. C++Builder include unelte care permit dezvoltarea vizuală bazată pe drag-and-drop, făcând programarea mai facilă prin implementarea unui GUI builder.

Bibliografie:

Îndrumar laborator MIDPS;
www.google.com.

Link spre repozitoriu:

<https://github.com/olarucristian/midps/tree/master/midps/Lab1>