

FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA
UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT
LUCRAREA DE LABORATOR#1

Mediul integrat C++ Builder

Autor:
Victor Talpa

lector asistent:
Irina COJANU
lector superior:
Svetlana COJOCARU

Lucrarea de Laborator Nr.1

Mediul 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 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.

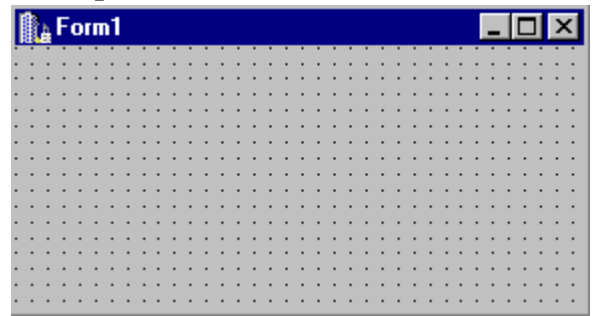
1 IDE (Mediul de Dezvoltare Integrat)

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-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.

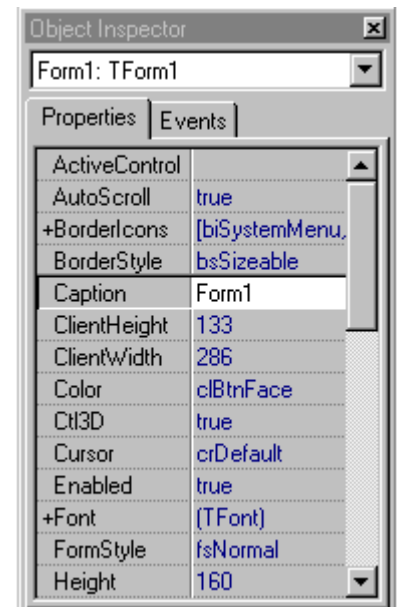


Proprietăți, metode, evenimente

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 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)

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

#include "Unit1.h"

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

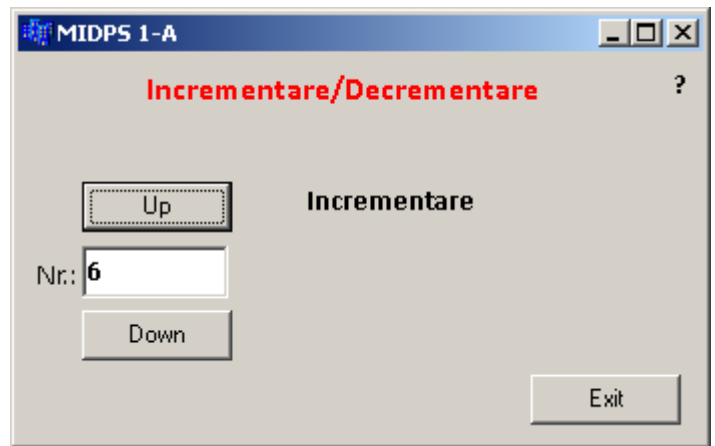
TForm1 *Form1;

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    status->Caption = "";
}

void __fastcall TForm1::upButtonClick(TObject *Sender)
{
    textBox->Text = textBox->Text.ToInt() + 1;
    status->Caption = "Incrementare";
}

void __fastcall TForm1::downButtonClick(TObject *Sender)
{
    textBox->Text = textBox->Text.ToInt() - 1;
    status->Caption = "Decrementare";
}

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



B)

```
#include <vc1.h>
#include <iostream>
#include <sstream>
#pragma hdrstop

#include "Unit1.h"
#include "dos.h"

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

using namespace std;

TForm1 *Form1;

void PrintTime();

struct date d;
struct time t;

struct countdown
{
    int minutes;
    int seconds;
    int zecimala;
};

countdown myTime;

__fastcall TForm1::TForm1(TComponent* Owner) : TForm(Owner)
{
    myTime.minutes = 0;
    myTime.seconds = 0;
    myTime.zecimala = 0;
    PrintTime();
    Timer2->Enabled = false;
    stopButton->Enabled = false;
}

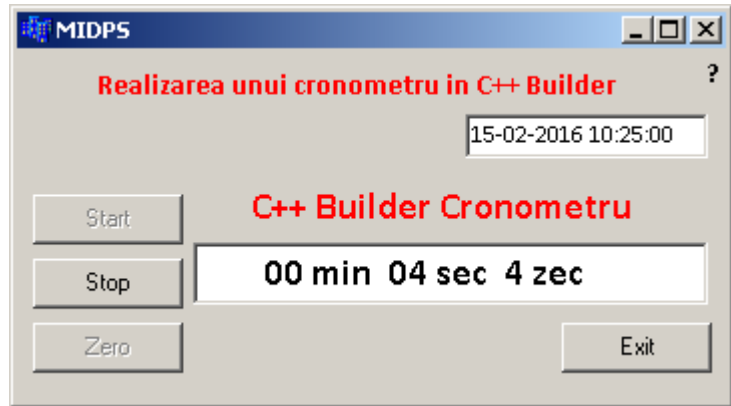
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    char buf[20];
    getdate(&d);
    gettime(&t);
    sprintf(buf, "%02d-%02d-%04d
%02d:%02d:%02d", d.da_day, d.da_mon, d.da_year, t.ti_hour, t.ti_min, t.ti_sec);
    textbox1->Text = (AnsiString)buf;
}

void PrintTime()
{
    stringstream timeFormat;
    timeFormat << " ";
    timeFormat << ((myTime.minutes < 10) ? "0" : "") << myTime.minutes << " min ";
    timeFormat << ((myTime.seconds < 10) ? "0" : "") << myTime.seconds << " sec ";
    timeFormat << /*((myTime.zecimala < 10) ? "0" : "") <<*/ myTime.zecimala << " zec";
    string temp = timeFormat.str();
    Form1->textbox2->Text = (AnsiString)temp.c_str();
}

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

void __fastcall TForm1::startButtonClick(TObject *Sender)
{
    Timer2->Enabled = true;
    startButton->Enabled = false;
    zeroButton->Enabled = false;
    stopButton->Enabled = true;
}

void __fastcall TForm1::stopButtonClick(TObject *Sender)
{
    Timer2->Enabled = false;
    startButton->Enabled = true;
    zeroButton->Enabled = true;
    stopButton->Enabled = false;
}
```



```
void __fastcall TForm1::zeroButtonClick(TObject *Sender)
{
    myTime.minutes = 0;
    myTime.seconds = 0;
    myTime.zecimals = 0;
    PrintTime();
}

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    if(myTime.zecimals == 9)
    {
        myTime.zecimals = 0;
        if(myTime.seconds == 59)
        {
            myTime.seconds = 0;
            myTime.minutes++;
        }
        else myTime.seconds++;
    }
    else myTime.zecimals++;
    PrintTime();
}
```

C)

```
#include <vcl.h>
#include <stdio.h>
#include <time.h>
#pragma hdrstop
#include "Unit1.h"
#include "dos.h"
#pragma package(smart_init)
#pragma resource "*.dfm"

TForm1 *Form1;

struct date d;
struct time t;

int width, height, x, y;

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

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);
    timebox->Text=(AnsiString)buf;
}

void __fastcall TForm1::startButtonClick(TObject *Sender)
{
    paintBox->Repaint();
    paintBox->Canvas->Pen->Color = clRed;
    width = Form1->paintBox->Width;
    height = Form1->paintBox->Height;
    x = 0;
    Form1->paintBox->Canvas->MoveTo(0, height / 2.0);
    stopButton->Enabled = true;
    startButton->Enabled = false;
    Timer2->Enabled = true;
}

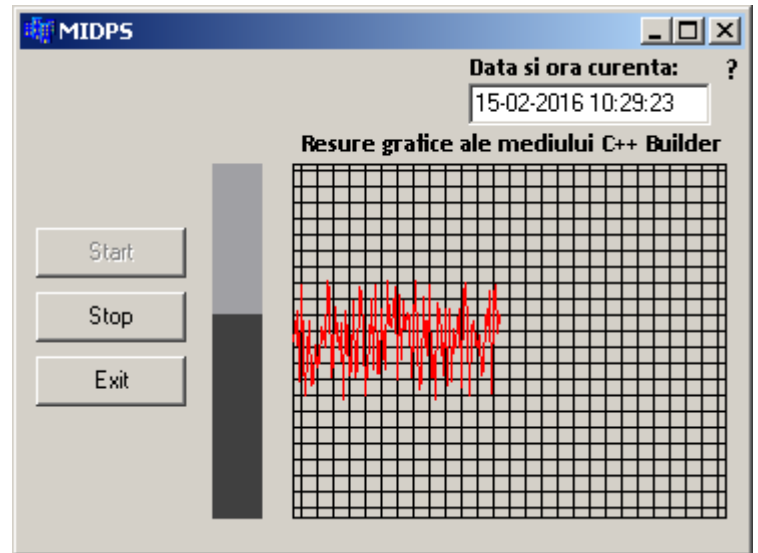
void __fastcall TForm1::paintBoxPaint(TObject *Sender)
{
    paintBox->Canvas->Pen->Color = clBlack;
    paintBox->Canvas->Brush->Color = clBlack;
    paintBox->Canvas->Brush->Style = bsCross;
    paintBox->Canvas->Rectangle(0, 0, paintBox->Width, paintBox->Height);
}

void __fastcall TForm1::stopButtonClick(TObject *Sender)
{
    Timer2->Enabled = false;
    startButton->Enabled = true;
    stopButton->Enabled = false;
}

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

void DrawLine()
{
    {
        y = (height / 2.0) + (rand() % 61 - 30);
        Form1->paintBox->Canvas->LineTo(++x, y);
        Form1->Panel12->Height = y;
    }
}

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    if(x == width)
        stopButtonClick(Sender);
    DrawLine();
}
```



Concluzie

În urma efectuării acestui laborator , am făcut cunoștință mediul de dezvoltare C++ Builder, am aflat cum funcționează componentele TButton, Ttimer, Label, Edit, PaintBox, Panel etc. Astfel de mediu de dezvoltare permite programatorului să reducă timpul pentru a dezvolta un program și crește productivitatea. Mediul de dezvoltare se ocupa de partea vizuală pe când programatorul se axează doar pe câteva detalii ale interfeței grafice și pe reprogramarea acestora în dependență de evenimente. C++ Builder este destul de performant și conține tot de ce e nevoie pentru a crea o fereastră ce stă la baza unui program. Însă în prezent sunt medii de dezvoltare mai performante și mai moderne, de exemplu ca Visual Studio care are mai multe facilități. Dacă C++ Builder era dezvoltat până în prezent, putea să ajunga la un nivel cu Visual Studio și alte medii de dezvoltare avansate.