Ministerul Educației Tineretului și Sportului

Universitatea Tehnică a Moldovei

Catedra Tehnologii Informaționale

Raport

La lucrarea de laborator nr.1 la

Medii interactive de dezvoltare a produselor soft

A efectuat st.gr 144: Roșca Andrei

A verificat: doctor, conf. univ: Cojanu Irina

Chișinau 2016

Tema: MEDIUL INTEGRAT C++ BUILDER

Scopul lucrarii:

a) Însuşirea 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) Însuşirea modului de utilizare a componentei VCL **TTimer.** Însuşirea modului de utilizare a funcţiilor de lucru cu timpul sistem. Realizarea unor aplicaţii de gestionare a resursei timp.

c) Însuşirea modului de utilizare a componentelor VCL **TPaintBox** şi **TPanel.** Însuşirea modului de utilizare a principalelor funcţii grafice ale mediului C++BUILDER . Realizarea unor elemente pentru afişarea grafică a informaţiei (diagramă şi bargraf).

Notiuni teoretice:

Componenta **TTimer** se găseşte în **Component Palette** (*pagina System*) .

Obiectul de acest tip permite execuţia in cadrul aplicaţiei a unor funcţii la intervale specificate. În context Windows obiectul TTimer lansează către aplicaţie mesaje la intervale prestabilite.

O particularitate faţă de componentele utilizate în lucrarea precedentă constă in faptul ca acest obiect nu are corespondent grafic pe formă în momentul execuţiei programului.

Componenta **TPaintBox** se găseşte în **Component Palette** (*pagina System*) . Obiectul de acest tip furnizează o componentă *TCanvas* care permite desenarea în interiorul unui dreptunghi, prevenind depăşirea marginilor acestuia.

Codul sursa (a):

Counter.cpp

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

#include <vcl.h>
#pragma hdrstop
//---------------------------------------------------------------------------
USEFORM("CounterInteractor.cpp", FirstTask);
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                Application->Initialize();
                Application->CreateForm(__classid(TFirstTask),
&FirstTask);
```

```
                Application->Run();
        }
        catch (Exception &exception)
        {
                Application->ShowException(&exception);
        }
        catch (...)
        {
                try
                {
                        throw Exception("");
                }
                catch (Exception &exception)
                {
                        Application->ShowException(&exception);
                }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

CounterInteractor.h

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

#ifndef CounterInteractorH
#define CounterInteractorH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include "sButton.hpp"
#include "sEdit.hpp"
#include "sLabel.hpp"
#include <ComCtrls.hpp>
//---------------------------------------------------------------------------
class TFirstTask : public TForm
{
__published:    // IDE-managed Components
        TsButton *upButton;
        TsButton *downButton;
        TsEdit *counterEditor;
        TsLabel *taskLabel;
        TsLabel *editorLabel;
        TsButton *exitButton;
        TUpDown *upDownControl;
        void __fastcall onUpButtonClick(TObject *Sender);
        void __fastcall onDownButtonClick(TObject *Sender);
```

```
        void __fastcall onExitButtonClick(TObject *Sender);
        void __fastcall onUpDownControlChange(TObject *Sender,
          bool &AllowChange, short NewValue, TUpDownDirection Direction);
private:    // User declarations
        int counter;
public:          // User declarations
        __fastcall TFirstTask(TComponent* Owner);
        void printCounter();
};
//---------------------------------------------------------------------------
---
extern PACKAGE TFirstTask *FirstTask;
//---------------------------------------------------------------------------
---
#endif
```

CounterInteractor.cpp

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

#include <vcl.h>
#pragma hdrstop

#include "CounterInteractor.h"
//---------------------------------------------------------------------------
---
#pragma package(smart_init)
#pragma link "sButton"
#pragma link "sEdit"
#pragma link "sLabel"
#pragma resource "*.dfm"
TFirstTask *FirstTask;
//---------------------------------------------------------------------------
---
__fastcall TFirstTask::TFirstTask(TComponent* Owner)
        : TForm(Owner)
{
        printCounter();
        this->counter = 0;
}
//---------------------------------------------------------------------------
---
void __fastcall TFirstTask::onUpButtonClick(TObject *Sender)
{
        □>counter;
        printCounter();
}
//---------------------------------------------------------------------------
---
void __fastcall TFirstTask::onDownButtonClick(TObject *Sender)
```

```
{
        --this->counter;
        printCounter();
}
//---------------------------------------------------------------------------
---
void TFirstTask::printCounter() {
      counterEditor->Text = this->counter;
}
void __fastcall TFirstTask::onExitButtonClick(TObject *Sender)
{
        Close();
}
//---------------------------------------------------------------------------
---

void __fastcall TFirstTask::onUpDownControlChange(TObject *Sender,
      bool &AllowChange, short NewValue, TUpDownDirection Direction)
{
        this->counter = NewValue;
        printCounter();
}
//---------------------------------------------------------------------------
---
```

Codul sursa (b):

secondTask.cpp

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

#include <vcl.h>
#pragma hdrstop
//---------------------------------------------------------------------------
---
USEFORM("Unit1.cpp", TimerForm);
//---------------------------------------------------------------------------
---
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                 Application->Initialize();
                 Application->CreateForm(__classid(TTimerForm),
&TimerForm);
                 Application->Run();
        }
        catch (Exception &exception)
        {
                 Application->ShowException(&exception);
        }
```

```
        catch (...)
        {
                 try
                 {
                         throw Exception("");
                 }
                 catch (Exception &exception)
                 {
                         Application->ShowException(&exception);
                 }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

Unit1.cpp

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

#include <vcl.h>
#pragma hdrstop

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

TTimerForm *TimerForm;

static struct time &operator +=( struct time &lhs, int hundredsOfASecond)
{
        lhs.ti_hund += hundredsOfASecond;
        if (lhs.ti_hund >= 100) {
                ++lhs.ti_sec;
                lhs.ti_hund = 0;
        }
        if (lhs.ti_sec >= 60) {
                ++lhs.ti_min;
                lhs.ti_sec = 0;
                if (lhs.ti_min >= 60) {
                        ++lhs.ti_hour;
                        lhs.ti_min = 0;
                        if (lhs.ti_hour >= 24)
```

```
                                lhs.ti_hour = 0;
                }
        }
        return lhs;
}

//---------------------------------------------------------------------------
__fastcall TTimerForm::TTimerForm(TComponent* Owner)
        : TForm(Owner)
{
        Timer1->Enabled = false;
        Timer2->Enabled = false;
        stopButton->Enabled = false;
}
//---------------------------------------------------------------------------
void __fastcall TTimerForm::onExitButtonClick(TObject *Sender)
{
        Close();
}
//---------------------------------------------------------------------------

void __fastcall TTimerForm::onStartButtonClick(TObject *Sender)
{
        Timer1->Enabled = true;
        Timer2->Enabled = true;
        stopButton->Enabled = true;
        startButton->Enabled = !stopButton->Enabled;
        zeroButton->Enabled = startButton->Enabled;
}
//---------------------------------------------------------------------------

void __fastcall TTimerForm::onStopButtonClick(TObject *Sender)
{
        Timer1->Enabled = false;
        Timer2->Enabled = false;
        stopButton->Enabled = false;
        startButton->Enabled = !stopButton->Enabled;
        zeroButton->Enabled = startButton->Enabled;
}
//---------------------------------------------------------------------------

void __fastcall TTimerForm::onFirstTimerCall(TObject *Sender)
{
        static struct time stopwatchTime = {0};
        stopwatchTime += 10;

        const int MAX_BUFFER_SIZE = 1024;
        char buffer[MAX_BUFFER_SIZE] = {0};
        sprintf(buffer, "%02d:%02d:%02d:%02d", stopwatchTime.ti_hour,
```

```
                stopwatchTime.ti_min,
                stopwatchTime.ti_sec, stopwatchTime.ti_hund);

        stopwatchEditor->Text = buffer;
}

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

void __fastcall TTimerForm::onTimerTimerCall(TObject *Sender)
{
        const int MAX_BUFFER_SIZE = 1024;
        struct date currentDate;
        struct time currentTime;
        getdate(&currentDate);
        gettime(&currentTime);

        char buffer[MAX_BUFFER_SIZE] = {0};
        sprintf(buffer, "%02d-%02d-%4d %02d:%02d:%02d",
currentDate.da_mon,
                currentDate.da_day, currentDate.da_year,
currentTime.ti_hour,
                currentTime.ti_min, currentTime.ti_sec);

    timeEditor->Text = buffer;
}
//---------------------------------------------------------------------------

void __fastcall TTimerForm::onZeroButtonClick(TObject *Sender)
{
        stopwatchEditor->Text = "00:00:00:00";
        //timeEditor->Text = "";
}
//---------------------------------------------------------------------------
```

Unit1.h

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

#ifndef Unit1H
#define Unit1H
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include "sButton.hpp"
```

```
#include "sEdit.hpp"
#include "sLabel.hpp"
#include <ExtCtrls.hpp>
//---------------------------------------------------------------------------
---
class TTimerForm : public TForm
{
__published:     // IDE-managed Components
        TsEdit *timeEditor;
        TsEdit *stopwatchEditor;
        TsButton *startButton;
        TsButton *stopButton;
        TsButton *zeroButton;
        TsLabel *taskLabel;
        TsLabel *stopwatchLabel;
        TsButton *exitButton;
        TTimer *Timer1;
        TTimer *Timer2;

        void __fastcall onExitButtonClick(TObject *Sender);
        void __fastcall onStartButtonClick(TObject *Sender);
        void __fastcall onStopButtonClick(TObject *Sender);
        void __fastcall onFirstTimerCall(TObject *Sender);
        void __fastcall onTimerTimerCall(TObject *Sender);
        void __fastcall onZeroButtonClick(TObject *Sender);
private:   // User declarations
public:          // User declarations
        __fastcall TTimerForm(TComponent* Owner);
};
//---------------------------------------------------------------------------
---
extern PACKAGE TTimerForm *TimerForm;
//---------------------------------------------------------------------------
---
#endif
```

Histogram.cpp

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

#include <vcl.h>
#pragma hdrstop
//---------------------------------------------------------------------------
---
USEFORM("Unit2.cpp", Form2);
//---------------------------------------------------------------------------
---
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                 Application->Initialize();
```

```
                Application->CreateForm(__classid(TForm2), &Form2);
                Application->Run();
        }
        catch (Exception &exception)
        {
                Application->ShowException(&exception);
        }
        catch (...)
        {
                try
                {
                        throw Exception("");
                }
                catch (Exception &exception)
                {
                        Application->ShowException(&exception);
                }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

Codul sursa (c):

Unit2.h

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

#ifndef Unit2H
#define Unit2H
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//---------------------------------------------------------------------------
class TForm2 : public TForm
{
__published:     // IDE-managed Components
        TPaintBox *PaintBox1;
        TPanel *Panel1;
        TPanel *Panel2;
        TButton *Button1;
        TButton *Button2;
```

```
        TButton *Button3;
        TTimer *Timer1;
        TTimer *Timer2;
        TEdit *Edit1;
        TLabel *Label1;
        TLabel *Label2;
        void __fastcall Timer1Timer(TObject *Sender);
        void __fastcall PaintBox1Paint(TObject *Sender);
        void __fastcall Timer2Timer(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
private:   // User declarations
public:          // User declarations
        __fastcall TForm2(TComponent* Owner);
};
//---------------------------------------------------------------------------
---
extern PACKAGE TForm2 *Form2;
//---------------------------------------------------------------------------
---
#endif
```

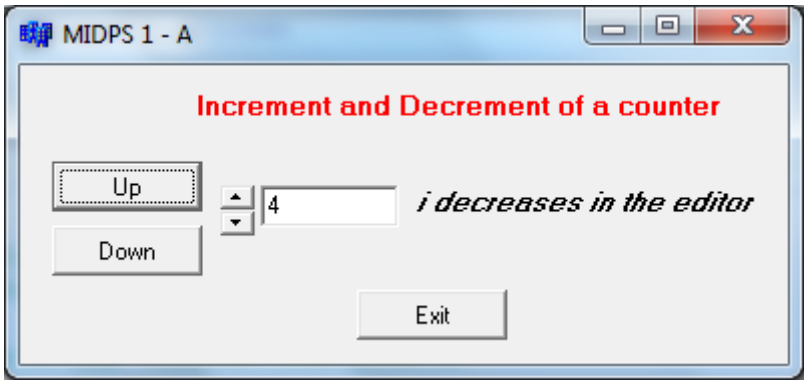
Unit2.cpp

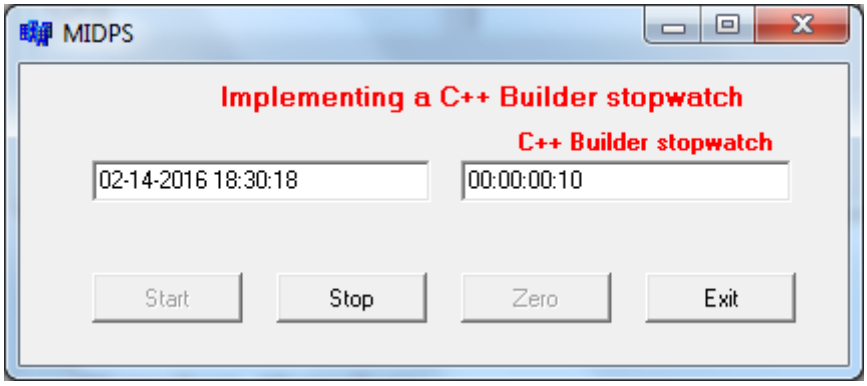
```
//---------------------------------------------------------------------------
---

#include <vcl.h>
#pragma hdrstop
#include <dos.h>
#include "Unit2.h"
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
//---------------------------------------------------------------------------
---
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
//---------------------------------------------------------------------------
---
__fastcall TForm2::TForm2(TComponent* Owner)
        : TForm(Owner)
{

srand(time(NULL));
Timer2->Enabled = false;


}
//---------------------------------------------------------------------------
---
```

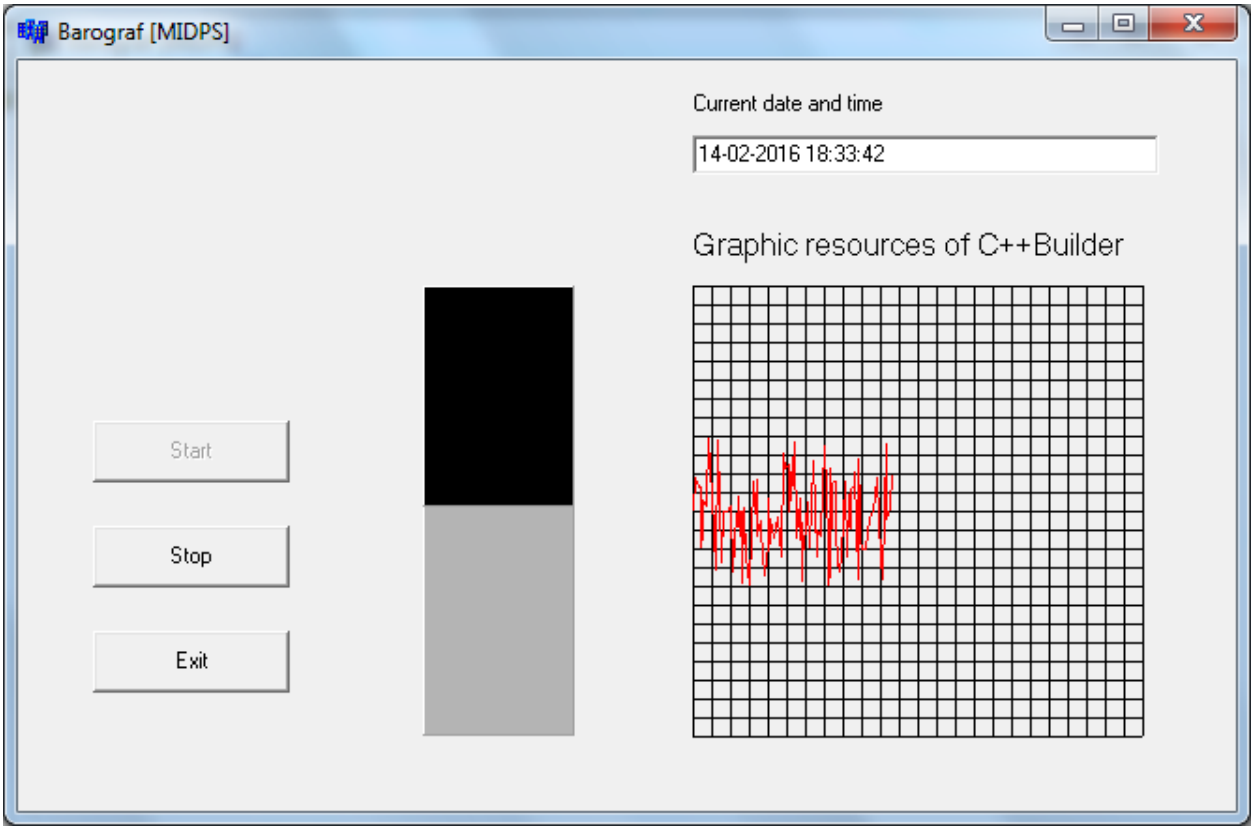
```
void __fastcall TForm2::Timer1Timer(TObject *Sender)
{
   struct time t;
   struct date d;

   gettime(&t);
   getdate(&d);

   char buf[30];

   sprintf(buf, "%02d-%02d-%02d %02d:%02d:%02d", d.da_day, d.da_mon,
d.da_year,t.ti_hour,t.ti_min,t.ti_sec);
   Edit1->Text = buf;


}
//---------------------------------------------------------------------------
---//---------------------------------------------------------------------------
------}
//---------------------------------------------------------------------------
---
void __fastcall TForm2::PaintBox1Paint(TObject *Sender)
{
PaintBox1->Canvas->Pen->Color = clBlack;
PaintBox1->Canvas->Brush->Style = bsHorizontal;
for(int i = 0; i <= 240; i += 10){
PaintBox1->Canvas->MoveTo(0,i);
PaintBox1->Canvas->LineTo(240,i);
PaintBox1->Canvas->MoveTo(i,0);
PaintBox1->Canvas->LineTo(i,240);
}
}
//---------------------------------------------------------------------------
---
void __fastcall TForm2::Timer2Timer(TObject *Sender)
{
        static int x = 0;
        int sign = rand() % 2 == 0 ? 1 : -1;
        static int y = ((rand() % 2) * sign + 120);

        PaintBox1->Canvas->Pen->Color = clRed;
        PaintBox1->Canvas->Brush->Style = bsHorizontal;
        Panel1->Height = y;

        PaintBox1->Canvas->MoveTo(x , y);
        x =((rand() % 3) + x);
        y = ((rand() % 40) * sign + 120);

        PaintBox1->Canvas->LineTo(x,y);
}
//---------------------------------------------------------------------------
---
void __fastcall TForm2::Button1Click(TObject *Sender)
```

```
{
        Timer2->Enabled = true;
        Button1->Enabled = false;
}
//---------------------------------------------------------------------------
---
void __fastcall TForm2::Button2Click(TObject *Sender)
{
        Button1->Enabled = true;
        Timer2->Enabled = false;
}
//---------------------------------------------------------------------------
---
void __fastcall TForm2::Button3Click(TObject *Sender)
{
Form2->Close();
}
//---------------------------------------------------------------------------
---
```

Executind aplicatiile, obtinem rezultatul:







Concluzii: In urma efectuarii acestei lucrari de laborator au fost obtinute deprinderi practice de create a aplicatiilor cu interfata grafica utilizind Borland C++ Builder. Au fost studiate asa elemente grafice din VCL ca editoare, buttoane, label-uri, timere, painbox-uri. Am ajuns la concluzia ca astfel de medii interactive usureaza mult procesul de implementare a aplicatiilor.