Ministerul Educaţiei şi Tineretului al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Catedra Automatica şi Tehnologii Informaţionale

Raport

MIDPS

Lucrarea de laborator Nr. 1

Tema: Borland C++ Builder

A efectuat: st. grupei TI-142

*Croitoru Ionel*

A verificat : *Cojanu Irina*

Chişinău 2015

**Scopul lucrării:**

De studiat bazele și principiile de creare a aplicațiilor pe baza platformei C++ Builder.

1. 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.

1. 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.
2. 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).

**Sarcina propusă:**

1.Se elaboreaza un contor ce poate fi dirijat cu butoane

Se vor utiliza următoarele obiecte (în afara formei):

* + două butoane (Button 1 şi 2) pentru incrementarea (UP) respectiv decrementarea (DOWN) a unei variabile întregi **i** ;
  + un buton (Button 3) pentru ieţirea din program (Exit);
  + o casetă de editare (Edit1) unde se va afişa valoarea variabilei **i**;
  + două etichete (Label1 şi 2) pentru afişarea textului „**Incrementare decrementare contor.”** Respectiv a **sensului de variaţie a variabilei i din caseta Edit1;**
  + în caption-ul formei se va afişa textul **„ MIDPS 1- A”;**
* fiecare obiect va avea hint-ul activ completat corespunzător

2.Se elaborează un program pentru realizarea unui cronometru

* o formă (*Form1*) pe care sunt dispuse celelalte obiecte şi în *Caption*-ul căreia se va afişa textul

**„MIDPS”;**

* patru butoane (*Button 1, 2, 3 , 4*) cu următoarele funcţii:
  + Button1 – pornirea cronometrului( Caption **Start**);
  + Button2 – oprirea cronometrului( Caption **Stop**);
  + Button3 – iniţializarea cronometrului( Caption **Zero**);
  + Button4 – ieşirea din program (Caption **Exit**).
* două timere (*Timer1* şi *Timer2*) cu următoarele funcţii
  + Timer1 (*Interval=1000 ms*) utilizat la afişarea timpului curent;
  + Timer2 (*Interval=100 ms*) utilizat pentru cronometru;
* două casete de editare (*Edit1* si *Edit2*) utilizate pentru :
  + Edit1 - afisarea datei si orei curente;

 Edit2 - afişarea timpului cronometrat;

* două etichete (*Label1* si *Label2*) cu Caption-ul conform figurii 2.4

*Observaţii*:

* din primele trei butoane, la un un moment dat va fi activ unul singur;
* fiecare obiect va avea *hint*-ul activ completat corespunzător;

În timpul execuţiei programului forma va avea aspectul din figura 4.3

**3)** Se elaboreazăun programpentru realizarea a două elemente de afişare (bargraf şi diagramă cuavans continuu) pentru care forma arată ca în figura 4.4 pe care sunt dispuse următoarele obiecte:

* o formă (*Form1*) în *Caption*-ul căreia se va afişa textul **„MIDPS;**
* trei butoane (*Button 1, 2, 3* ) cu următoarele funcţii:
  + Buton1 – activarea afişării în diagramă şi în bargraf ( Caption **Start**);
  + Buton2 – oprirea afişării în diagramă şi în bargraf ( Caption **Stop**);
  + Buton3 – ieşirea din program (Caption **Exit**).

**Programul 1:**

Se vor utiliza următoarele obiecte (în afara formei):

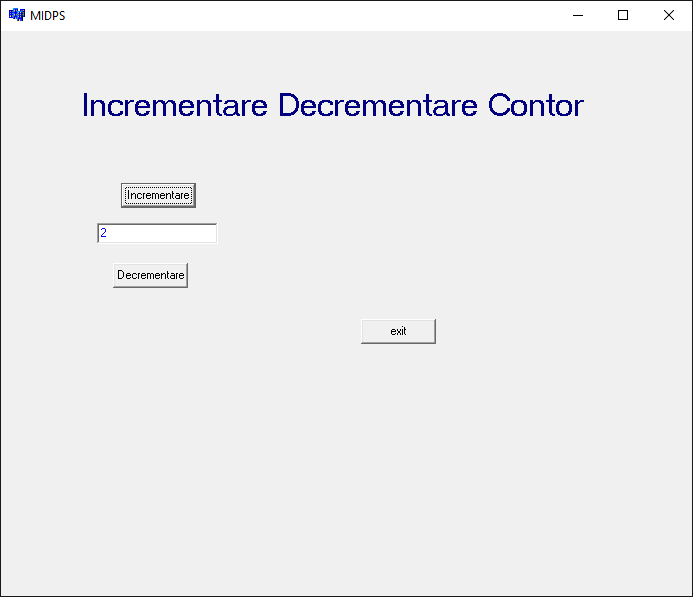
* 1. două butoane (Button 1 şi 2) pentru incrementarea (UP) respectiv decrementarea (DOWN) a unei variabile întregi **i** ;
  2. un buton (Button 3) pentru ieţirea din program (Exit);
  3. o casetă de editare (Edit1) unde se va afişa valoarea variabilei **i**;

- două etichete (Label1 şi 2) pentru afişarea textului „**Incrementare decrementare contor.”** Respectiv a **sensului de variaţie a variabilei i din caseta Edit1;**

- în caption-ul formei se va afişa textul **„ MIDPS 1- A”;**

- fiecare obiect va avea hint-ul activ completat corespunzător .

|  |
| --- |
| //---------------------------------------------------------------------------  #include <vcl.h>  #pragma hdrstop  #include "Unit1.h"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  TForm1 \*Form1;  int i = 0;  //---------------------------------------------------------------------------  \_\_fastcall TForm1::TForm1(TComponent\* Owner)  : TForm(Owner)  {}//---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button3Click(TObject \*Sender)  {  Close();  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button2Click(TObject \*Sender)  {  i--;  Edit1->Text=i;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button1Click(TObject \*Sender)  {  i++;  Edit1->Text=i;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Edit1Change(TObject \*Sender)  {  Edit1->Text=i;  }  //--------------------------------------------------------------------------- |
|  |



**Programul 2:**

Se elaborează un program pentru realizarea unui cronometru.

Se vor utiliza următoarele obiecte, evidenţiate în figura 4.2:

- o formă (*Form1*) pe care sunt dispuse celelalte obiecte şi în *Caption*-ul căreia se va afişa textul **„MIDPS”;**

* 1. patru butoane (*Button 1, 2, 3 , 4*) cu următoarele funcţii:
     1. Button1 – pornirea cronometrului( Caption **Start**);
     2. Button2 – oprirea cronometrului( Caption **Stop**);
     3. Button3 – iniţializarea cronometrului( Caption **Zero**);
     4. Button4 – ieşirea din program (Caption **Exit**).
  2. două timere (*Timer1* şi *Timer2*) cu următoarele funcţii
     1. Timer1 (*Interval=1000 ms*) utilizat la afişarea timpului curent;
     2. Timer2 (*Interval=100 ms*) utilizat pentru cronometru;
  3. două casete de editare (*Edit1* si *Edit2*) utilizate pentru :
     1. Edit1 - afisarea datei si orei curente;
     2. Edit2 - afişarea timpului cronometrat;
  4. două etichete (*Label1* si *Label2*) cu Caption-ul conform figurii 2.4

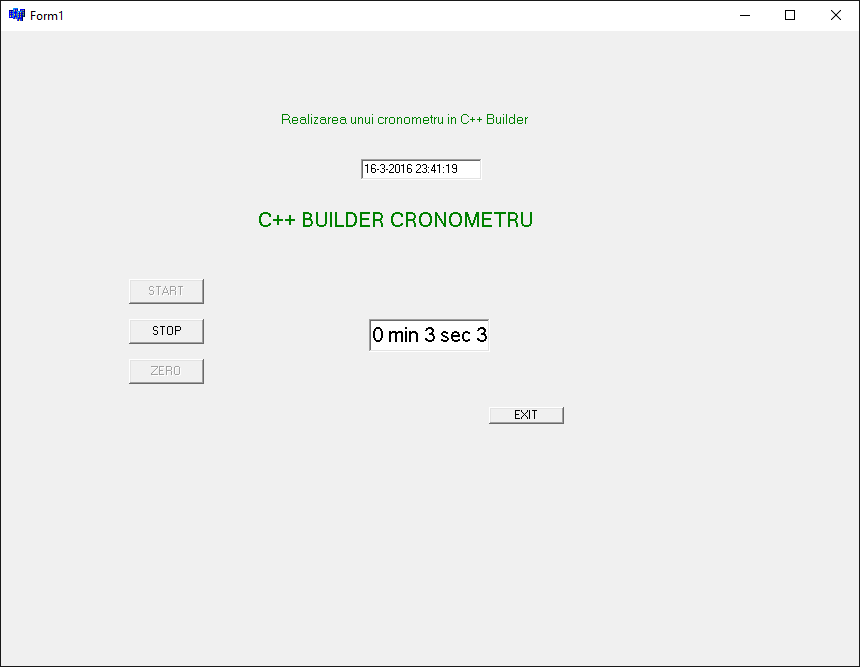
*Observaţii*:

- din primele trei butoane, la un un moment dat va fi activ unul singur;

* 1. fiecare obiect va avea *hint*-ul activ completat corespunzător;

**Listingul programului**

|  |
| --- |
| //---------------------------------------------------------------------------  #include <vcl.h>  #include <dos.h>  #include <stdio.h>  #pragma hdrstop  #include "Unit1.h"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  int m = 0, s = 0, z = 0;  int day = 0, mon = 0, year = 0, hour = 0, min = 0, sec = 0;  int i = 0;  TForm1 \*Form1;  struct date d;  struct time t;  int k = 0;  //---------------------------------------------------------------------------  \_\_fastcall TForm1::TForm1(TComponent\* Owner)  : TForm(Owner)  {  Timer1->Enabled=true;  Timer2->Enabled=false;  Button1->Enabled=true;  Button2->Enabled=false;  Button3->Enabled=false;  Edit2->Text="0 min 0 sec 0 zec";  getdate(&d);  gettime(&t);  char buf[40];  snprintf ( buf, 40, "%d-%d-%d %d:%d:%d", 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::Timer1Timer(TObject \*Sender)  {  getdate(&d);  gettime(&t);  char buff[40];  snprintf ( buff, 40, "%d-%d-%d %d:%d:%d", d.da\_day,d.da\_mon,d.da\_year,t.ti\_hour,t.ti\_min,t.ti\_sec);  Edit1->Text=(AnsiString)buff;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Timer2Timer(TObject \*Sender)  {  char buf[40];  Timer2->Tag=i;  if( i<100){  z = i; }  if(i == 10){  z = 0;  s++;  i = 0;  }  if(s == 60){  m++;  s = 0;  }  snprintf ( buf, 40, "%d min %d sec %d zec", m,s,z);  Edit2->Text=(AnsiString)buf;  i++;  //Edit2->Text=Timer2->Tag;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button1Click(TObject \*Sender)  {  Button1->Enabled=false;  Button2->Enabled=true;  Button3->Enabled=false;  Timer2->Enabled=true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button2Click(TObject \*Sender)  {  Button1->Enabled=false;  Button2->Enabled=false;  Button3->Enabled=true;  Timer2->Enabled=false;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button4Click(TObject \*Sender)  {  Close();  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button3Click(TObject \*Sender)  {  Button1->Enabled=true;  Button2->Enabled=false;  Button3->Enabled=false;  Timer2->Enabled=false;  i = z = s = m = 0;  Edit2->Text="0 min 0 sec 0 zec";  }  **//---------------------------------------------------------------------------** |



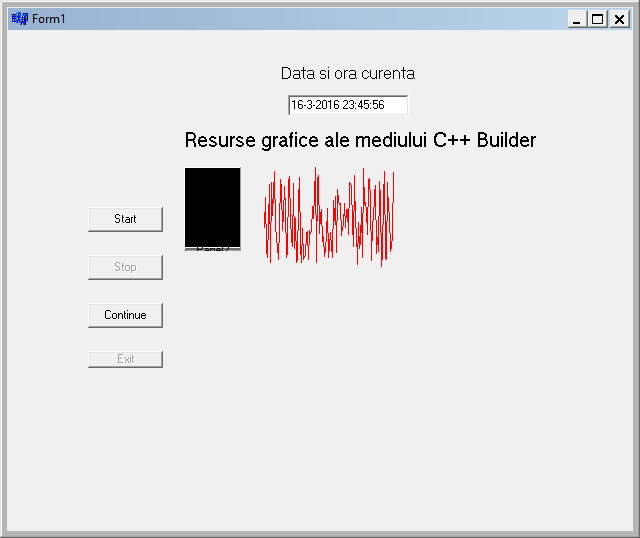
**Programul 3:**

Se elaborează un program pentru realizarea a două elemente de afişare (bargraf şi diagramă cu avans continuu) pentru care forma arată ca în figura 4.4 pe care sunt dispuse următoarele obiecte:

- o formă (*Form1*) în *Caption*-ul căreia se va afişa textul **„MIDPS;**

* 1. trei butoane (*Button 1, 2, 3* ) cu următoarele funcţii:
     1. Buton1 – activarea afişării în diagramă şi în bargraf ( Caption **Start**);
     2. Buton2 – oprirea afişării în diagramă şi în bargraf ( Caption **Stop**);
     3. Buton3 – ieşirea din program (Caption **Exit**).
  2. două timere (*Timer1* şi *Timer2*) cu următoarele funcţii
     1. Timer1 (*Interval=1000 ms*) utilizat la afişarea timpului curent;
     2. Timer2 (*Interval=500 ms*) pentru intervalul de afişare în diagramă şi în bargraf;
  3. o casetă de editare (*Edit1*) utilizată pentru afişarea datei si orei curente;
  4. două etichete (*Label1* si *Label2*) cu Caption-ul conform figurii 4.4

|  |
| --- |
| //---------------------------------------------------------------------------  #include <vcl.h>  #include <dos.h>  #include <stdio.h>  #pragma hdrstop  #include "Unit1.h"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  TForm1 \*Form1;  int x = 0;  int i = 0;  struct date d;  struct time t;  //---------------------------------------------------------------------------  \_\_fastcall TForm1::TForm1(TComponent\* Owner)  : TForm(Owner)  {  TRect dreptunghi;  TRect sursa,destinatie;  PaintBox1->Canvas->Pen->Color=clRed;  dreptunghi=Rect(464,128,464+129,128+129);  PaintBox1->Repaint();  sursa=Rect(0,0,100,150);  destinatie=Rect(100,0,200,150);  sursa=Rect(464,128,464+129,128+129);  PaintBox1->Canvas->CopyRect(destinatie,PaintBox1->Canvas,sursa);  //Rect(0,0,50,50);  //PaintBox1->Canvas->FillRect(dreptunghi);  //PaintBox1->Canvas->Brush->Color = clAqua;  // PaintBox1->Canvas->FloodFill(464,128,clBlack,fsBorder);  Timer1->Enabled=true;  Timer2->Enabled=false;  getdate(&d);  gettime(&t);  char buf[40];  snprintf ( buf, 40, "%d-%d-%d %d:%d:%d", d.da\_day,d.da\_mon,d.da\_year,t.ti\_hour,t.ti\_min,t.ti\_sec);  Edit1->Text=(AnsiString)buf;  Button1->Enabled=true;  Button2->Enabled=false;  Button3->Enabled=false;  PaintBox1->Canvas->MoveTo(0,60);  /\*  TRect dreptunghi;  PaintBox1->Canvas->Brush->Style = bsHorizontal;  dreptunghi=Rect(0,0,400,400);  PaintBox1->Canvas->FillRect(dreptunghi);  PaintBox1->Repaint();  PaintBox1->Canvas->Pen->Color=clRed;  PaintBox1->Canvas->Pen->Width=1;  //PaintBox1->Canvas->Pen->Style=psDash;  PaintBox1->Canvas->Ellipse(0,0,400,400);  TRect sursa,destinatie;  sursa=Rect(0,0,100,150);  PaintBox1->Canvas->CopyRect(destinatie,PaintBox1->Canvas,sursa); \*/  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button1Click(TObject \*Sender)  {  Button1->Enabled=false;  Button3->Enabled=false;  Button2->Enabled=true;  Timer2->Enabled=true;  PaintBox1->Canvas->Pen->Color=clRed;  i = 0;  x = 0;  PaintBox1->Canvas->MoveTo(0,60);  Panel2->Height=60;  PaintBox1->Canvas->FloodFill(50,50,clBtnFace,fsBorder);  PaintBox1->Repaint();  //PaintBox1->Canvas->LineTo(10,70);  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button2Click(TObject \*Sender)  {  Button1->Enabled=true;  Button2->Enabled=false;  Timer2->Enabled=false;  Button3->Enabled=true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button4Click(TObject \*Sender)  {  Close();  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Button3Click(TObject \*Sender)  {  Timer2->Enabled=true;  Button1->Enabled=false;  Button2->Enabled=true;  Button3->Enabled=false;  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Timer2Timer(TObject \*Sender)  {  x = rand()%100;  i = i++;  //Label1->Caption=x;  PaintBox1->Canvas->LineTo(i,x);  Panel2->Height=x;  if(i==129){ //129  Button1->Enabled=true;  Button2->Enabled=false;  Button3->Enabled=false;  Timer2->Enabled=false;  Timer2->Tag = 0;  i = 0;  x = 0;  //Timer2->Tag=0;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)  {  getdate(&d);  gettime(&t);  char buff[40];  snprintf ( buff, 40, "%d-%d-%d %d:%d:%d", d.da\_day,d.da\_mon,d.da\_year,t.ti\_hour,t.ti\_min,t.ti\_sec);  Edit1->Text=(AnsiString)buff;  }  //--------------------------------------------------------------------------- |



***Concluzii:***

In urma efectuarii primului laborator la MIDPS am studiat bazele si principiile de creare a aplicatiilor pe baza platformei C++ Builder.Am insusit modul de utilizare a celor mai importante componente ale mediului itegrat C++ Builder.Am realizat 3 programe simple utilizand componetele de baza ale acestui mediu de dezvoltare.

Mediul de dezoltare C++ Builder reprezinta un instrument foarte rapid si efficient in elaborarea unor programe.Cu ajutor interfetii de lucru asa numita “WorkSpace” adica Forma in care lucram putem creea visual obiecte,butoane,texte,functii s.a. totul se face prin ajutor unui meniu destul de intuitiv astfel si pentru un utilizator de rand crearea unor aplicatii simple nu va fi foarte complicat.  
Codul programului se scrie in marea parte automat de programa,inginerului IT ii ramine doar sa completeze codul cu variabile,conditii,structure,functii s.a.  
In general am inteles ca Borland-ul reprezinta un mediu de programare foarte comod si simplu, totodata ce ofera un spectru larg optiuni in functionalitate.