

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Tehnologii Informaționale

RAPORT

Lucrarea de laborator nr.1

la Medii Interactive de Dezvoltare a Produselor Soft

A efectuat:

st.gr.TI-143

Dumbrava Alexandru

A verificat:

Cojocarui Svetlana

Chișinău 2016

Tema: Mediul intergrat C++ Builder

Scopul lucrării:

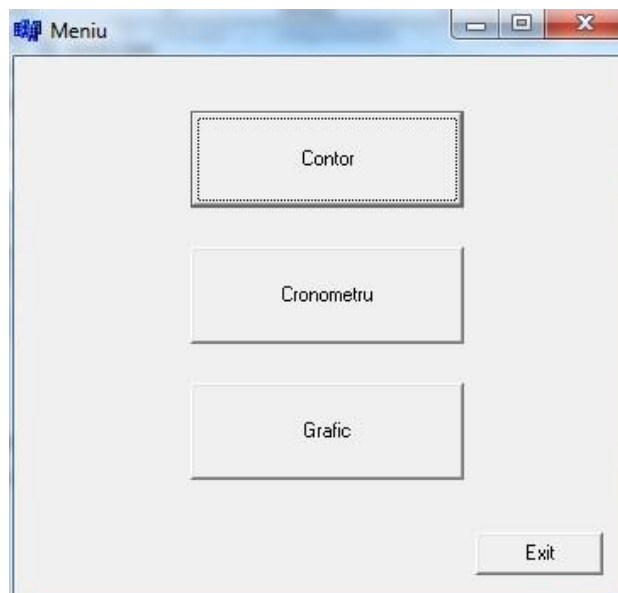
- Însușirea modului de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER . Realizarea unui program simplu care utilizează componente de tip *TButton*, *TEdit*, *Tlabel*, *RadioButton* etc.
- Însușirea modului de utilizare a componentei VCL **TTimer**. Însușirea modului de utilizare a funcțiilor de lucru cu timpul sistem. Realizarea unor aplicații de gestionare a resursei timp.
- 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).

Formularea condiției problemei (sarcina de lucru):

- Se elaborează un program pentru realizarea unui contor cu funcțiile incrementare/decrementare.
- Se elaborează un program pentru realizarea unui cronometru.
- Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu)

Implimentare task-uri:

De la inceput a fost creat o forma care este principala, cu ajutorul acestei forme putem accesa celelalte task-uri.



Primul butonului ne permite accesarea task-ului care realizeaza un contor.

Prin intermediul celui de-al doilea buton putem accesa o forma care realizeaza un cronometru.

În final cu ajutorul butonului trei putem vedea task-ul care realizeaza doua elemente: un bar graph si p diagram cu avans continuu.

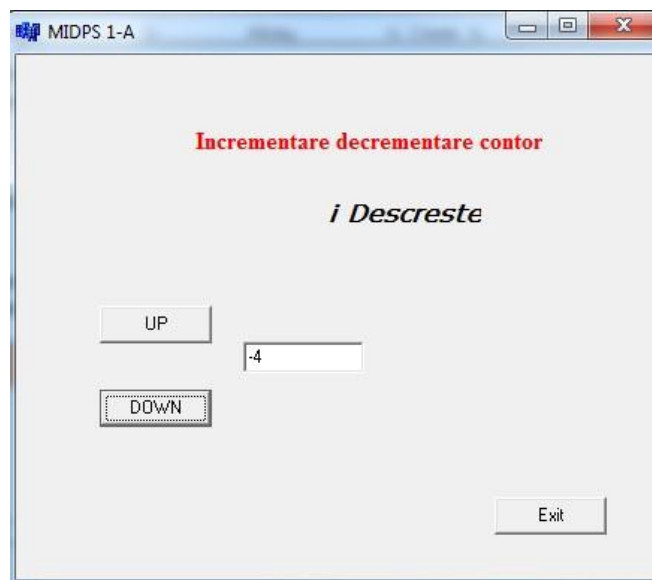
```

//-----
void __fastcall TFMenu::Button1Click(TObject *Sender)
{
    FMenu->Hide();
    FContor->ShowModal();
}
//-----
void __fastcall TFMenu::Button2Click(TObject *Sender)
{
    FMenu->Hide();
    FCronometru->ShowModal();
}
//-----
void __fastcall TFMenu::Button3Click(TObject *Sender)
{
    FMenu->Hide();
    FDiagram->ShowModal();
}
//-----
void __fastcall TFMenu::Button4Click(TObject *Sender)
{
    FMenu->Close();
}
//-----

```

In aceasta bucata de cod putem vedea realizarea fuctionalului formeii *Menu*, atunci cind facem klik pe unul dintre cele trei butoane, forma *Menu* devine invizibila si neascesibila pina cind nu inchidem unul din cele trei task-uri.

- Se elaborează un program pentru realizarea unui contor cu funcțiile incrementare/decrementare.



Realizarea acestui task se incepe de la declararea unei variabile de tip `int` care va reprezenta valoarea curenta a contorului. Dupa care pe forma se adauga doua butoane care permita marirea sau micsoararea valorii contorului. Se mai adauga si un Edit prin intermediul caruia putem vedea valoarea contorului la fiecare modificare a lui. In final a fost adaugat un buton „Exit” care permite inchiderea acestei forme. La fel pe forma au mai fost adaugate doua label caption-ul unui din aceste se modifica la fiecare schimbare a valorii contorului.

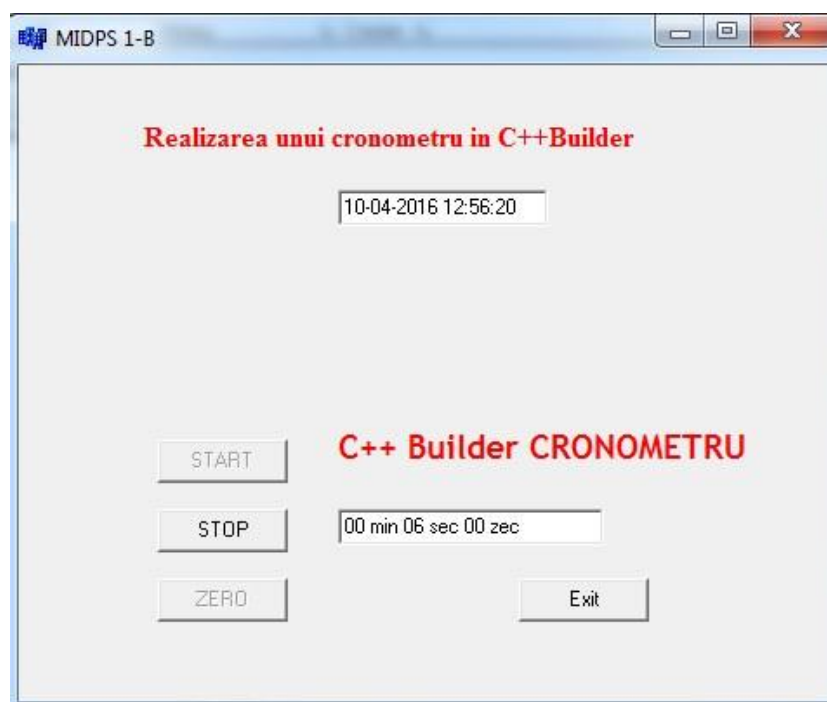
```

int count=0;
//-----
__fastcall TFContor::TFContor(TComponent* Owner)
    : TForm(Owner)
{
    Label2->Visible=false;
    Edit1->Text=AnsiString(count);
}
//-----
void __fastcall TFContor::Button1Click(TObject *Sender)
{
    count++;
    Label2->Visible=true;
    Label2->Caption="i Creste";
    Edit1->Text=AnsiString(count);
}
//-----
void __fastcall TFContor::Button2Click(TObject *Sender)
{
    count--;
    Label2->Visible=true;
    Label2->Caption="i Descreste";
    Edit1->Text=AnsiString(count);
}
//-----
void __fastcall TFContor::Button3Click(TObject *Sender)
{
    FContor->Close();
    FMenu->Show();
}
//-----

```

Aceast fragment de cod explica realizarea functionala a formei *FContor*, crearea unui variabile *count* de tip *int*, crearea a doua componente butoane *Button1* si *Button2* si functiile care permit incrementarea si decrementarea valori *count* respectiv. Afisarea valorii curente cu ajutorul componentei *TEdit* si un *TLabel* unde caption se modifica la incrementare/decrementare.

- Se elaborează un program pentru realizarea unui cronometru.



La realizarea acestui task principalul element este timer-ul. Aici au fost folosite 2 componente timer unul cu intervalul 1s. si al doilea cu intervalul 100ms. Primul timer cu intervalul de 1sec este folosit pentru a afisa data si ora curenta, si faptul ca el la fiecare secunda genereaza un event ne permite sa actualizam ora si data. Ora si data se citesc din sistem cu ajutorul functiilor getdate si gettime. Al doilea timer se foloseste pentru a crea un cronometru, valorile cronometrului sunt zeci de secunda, secunde si minute, aceste 3 valori sunt pastrate in 3 variabile integer si in dependenta de evenimentul timer-ului se incrementeaza. Mai jos se poate observa mai detaliat in ce mod a fost realizata pedepplin functionalitatea cronometrului.

```
int zec;
int sec;
int min;

//-----
__fastcall TFChronometru::TFChronometru(TComponent* Owner)
    : TForm(Owner)
{
    zec=0;
    sec=0;
    min=0;
    char buf[20];
    sprintf(buf,"%02d min %02d sec %02d zec",min,sec,zec);
    Edit2->Text=(AnsiString)buf;
}

//-----
void __fastcall TFChronometru::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);
    Edit1->Text=(AnsiString)buf;
}

//-----

void __fastcall TFChronometru::FormCreate(TObject *Sender)
{
    Edit1->Clear();
    Timer1->Interval=1000;
    Timer2->Interval=100;
    Timer2->Enabled=false;
}

//-----

void __fastcall TFChronometru::ExitClick(TObject *Sender)
{
    FCronometru->Close();
    FMenu->Show();
}

//-----
```

```

void __fastcall TFChronometru::Timer2Timer(TObject *Sender)
{
    zec++;
    if(zec==10){
        zec=0;
        sec++;
    }
    if(sec==60){
        sec=0;
        min++;
    }
    char buf[20];
    sprintf(buf,"%02d min %02d sec %02d zec",min,sec,zec);
    Edit2->Text=(AnsiString)buf;
}
//-----

void __fastcall TFChronometru::STARTClick(TObject *Sender)
{
    Timer2->Enabled=true;
    START->Enabled=false;
    STOP->Enabled=true;
    ZERO->Enabled=false;
}
//-----

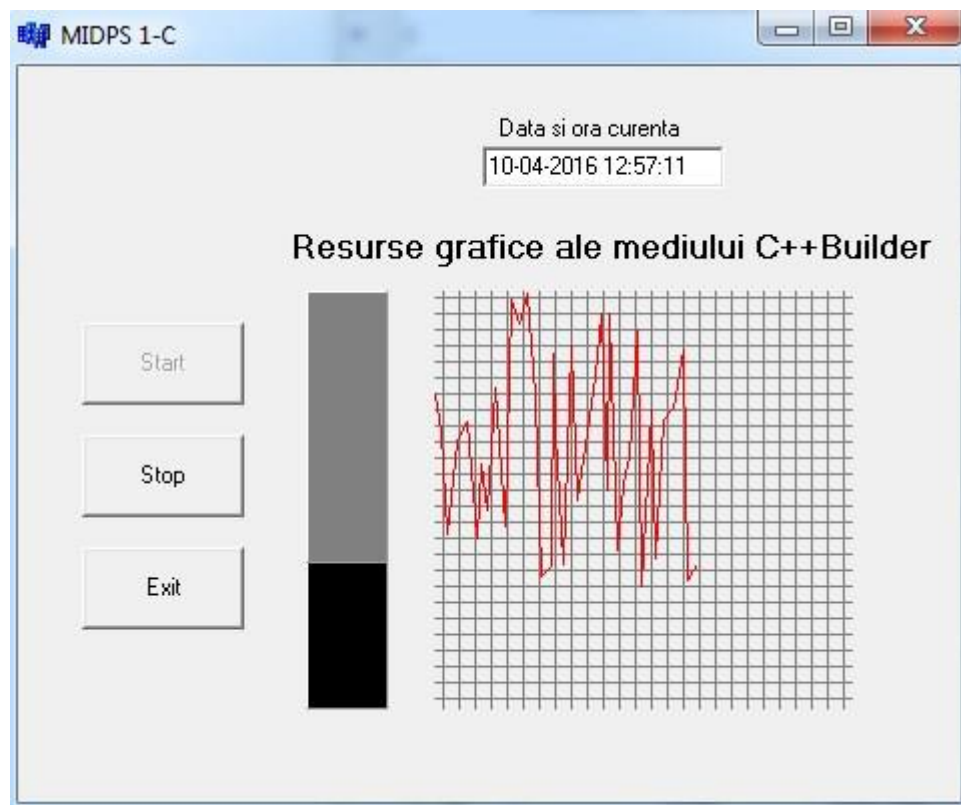
void __fastcall TFChronometru::STOPClick(TObject *Sender)
{
    START->Enabled=true;
    STOP->Enabled=false;
    ZERO->Enabled=true;
    Timer2->Enabled=false;
}
//-----

void __fastcall TFChronometru::ZEROClick(TObject *Sender)
{
    START->Enabled=true;
    STOP->Enabled=false;
    ZERO->Enabled=false;
    zec=0;
    sec=0;
    min=0;

    char buf[20];
    sprintf(buf,"%02d min %02d sec %02d zec",min,sec,zec);
    Edit2->Text=(AnsiString)buf;
}
//-----

```

- Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu)



Pentru realizarea bargraph-ului și diamei a fost creat un timer cu un interval de 500ms de fiecare dată generând un număr random. Numărul acesta este salvat într-o variabilă de tip `int` și după care el este reprezentat pe bargraph și pe diagramă. Bargraphul de fapt sunt două componente `panel` care se suprapun și când este generat o valoare random `panel`-ul care este deasupra își schimbă dimensiunile. Pe diagramă valoare reprezentată pe bargraph este valoare pe axa „y”, iar axa „x” tot variază în dependență de un număr generat care poate fi 1, 2, 3 aceasta ne permite la fiecare generare a unui număr random diagramă să se deplaseze și treptat să fie afișată diagramă pe toată dimensiunea `PaintBox`-ului. Când toată suprafața `PaintBox`-ului a fost umplută se folosește funcția `CopyRect` pentru a deplasa diagramă în stînga.

```
struct date d;
struct time t;

int x=0;
int y=0;

//-----
__fastcall TFDiagram::TFDiagram(TComponent* Owner)
: TForm(Owner)
{
    std::srand(std::time(NULL));
    y=rand()% 151;
    Panel2->Height=y;
    PaintBox1->Canvas->MoveTo(0,y);
}
```



```

void __fastcall TFDiagram::Timer1Timer(TObject *Sender)
{
    getdate(&d);
    gettime(&t);
    char buf[20];
    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 TFDiagram::ExitClick(TObject *Sender)
{
    FDiagram->Close();
    FMenu->Show();
}
//-----

void __fastcall TFDiagram::StartClick(TObject *Sender)
{
    Timer2->Enabled=true;
    Start->Enabled=false;
    Stop->Enabled=true;
}
//-----

void __fastcall TFDiagram::StopClick(TObject *Sender)
{
    Timer2->Enabled=false;
    Start->Enabled=true;
    Stop->Enabled=false;
}
//-----

void __fastcall TFDiagram::Timer2Timer(TObject *Sender)
{
    y = rand() % 151;
    x+= (rand() % 5) + 1;
    Panel2->Height = y;
    PaintBox1->Canvas->LineTo(x, y);
    if (x > 180){
        TRect sursa,destinatie;
        sursa=Rect(0,0,209,209);
        destinatie=Rect(-(207- x),0,211- (209 - x),209);
        PaintBox1->Canvas->CopyRect(destinatie,PaintBox1->Canvas,sursa);
        x -= (209 - x);
    }
    PaintBox1->Canvas->MoveTo(x, y);
}
//-----

```



```

void __fastcall TFDiagram::PaintBox1Click(TObject *Sender)
{
    TRect dreptunghi;
    PaintBox1->Canvas->Pen->Color=clRed;
    PaintBox1->Canvas->Pen->Width=1;
    PaintBox1->Canvas->Brush->Style=bsCross;
    PaintBox1->Canvas->Brush->Color=clGray;
    PaintBox1->Canvas->FloodFill(10,10,clGray,fsBorder);
}
//-----

void __fastcall TFDiagram::FormCreate(TObject *Sender)
{
    Timer2->Enabled=false;
    Panel2->Enabled=false;
}
//-----

```

Concluzie:

In urma efectuării acestei lucrări de laborator au fost capătate noi abilități în lucrul cu IDE C++ Builder. Au fost folosite mai multe componente de control care permit crearea interfețelor grafice rapid și simplu.

Am capatat depreinderi in lucrul cu butoane, casete de editare, “label” – uri etc. Am folosit componenta Timer pentru a afisa data si ora curenta si pentru a efectua niste operatii la un interval anumit de timp. Timer-ul ne-a ajutat in crearea cronometrului si a unui bargraph pe care erau reprezentate niste numere generate random.

La general putem spune ca C++ Builder este un mediu care permite crearea interfețelor grafice mult mai rapid și simplu.

Link-ul catre repozitoriu:

<https://github.com/DumbravaAlexandru/MIDPS>