

MINISTERUL EDUCAȚIEI ȘI TINERETULUI AL REPUBLICII MOLDOVA  
UNIVERSITATEA TEHNICA A MOLDOVEI  
Facultatea Calculatoare, Informatică și Microelectronică

# RAPORT

LA LUCRAREA DE LABORATOR NR. 1

Disciplina : Medii interactive de dezvoltare a produselor soft

TEMA: „C++ Builder”

Elaborat de:	st.gr.TI-142 Ursachi Daniel
Controlat de:	Irina Cojanu

Chișinău 2016

## Scopul lucrării:

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

- 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).

## Sarcina propusă:

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

Se vor utiliza următoarele obiecte (în afara formei):

- două butoane (Button 1 și 2) pentru incrementarea (UP) respectiv decrementarea (DOWN) a unei variabile întregi **i** ;
- un buton (Button 3) pentru ieșirea din program (Exit);
- o casetă de editare (Edit1) unde se va afișa valoarea variabilei **i**;
- două etichete (Label1 și 2) pentru afișarea textului „**Incrementare decrementare contor.**” Respectiv a **sensului de variație a variabilei i din caseta Edit1**;
- în caption-ul formei se va afișa 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 și în *Caption*-ul căreia se va afișa textul „**MIDPS**”;
- patru butoane (*Button 1, 2, 3, 4*) cu următoarele funcții:
  - Button1 – pornirea cronometrului( **Caption Start**);
  - Button2 – oprirea cronometrului( **Caption Stop**);
  - Button3 – inițializarea cronometrului( **Caption Zero**);
  - Button4 – ieșirea din program (**Caption Exit**).
- două timere (*Timer1* și *Timer2*) cu următoarele funcții
  - Timer1 (*Interval=1000 ms*) utilizat la afișarea timpului curent;
  - Timer2 (*Interval=100 ms*) utilizat pentru cronometru;
- două casete de editare (*Edit1* și *Edit2*) utilizate pentru :
  - Edit1 - afișarea datei și orei curente;

- Edit2 - afișarea timpului cronometrat;
- două etichete (*Label1* și *Label2*) cu Caption-ul conform figurii 2.4

*Observații:*

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

În timpul execuției programului forma va avea aspectul din figura 4.3

**3)** Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și 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 afișa textul „**MIDPS**”;
- trei butoane (*Button 1, 2, 3*) cu următoarele funcții:
  - Buton1 – activarea afișării în diagramă și în bargraf ( *Caption* **Start**);
  - Buton2 – oprirea afișării în diagramă și în bargraf ( *Caption* **Stop**);
  - Buton3 – ieșirea din program (*Caption* **Exit**).

### **Programul 1:**

Inceputul programarii în mediul C++ Builder începe cu crearea elementelor:

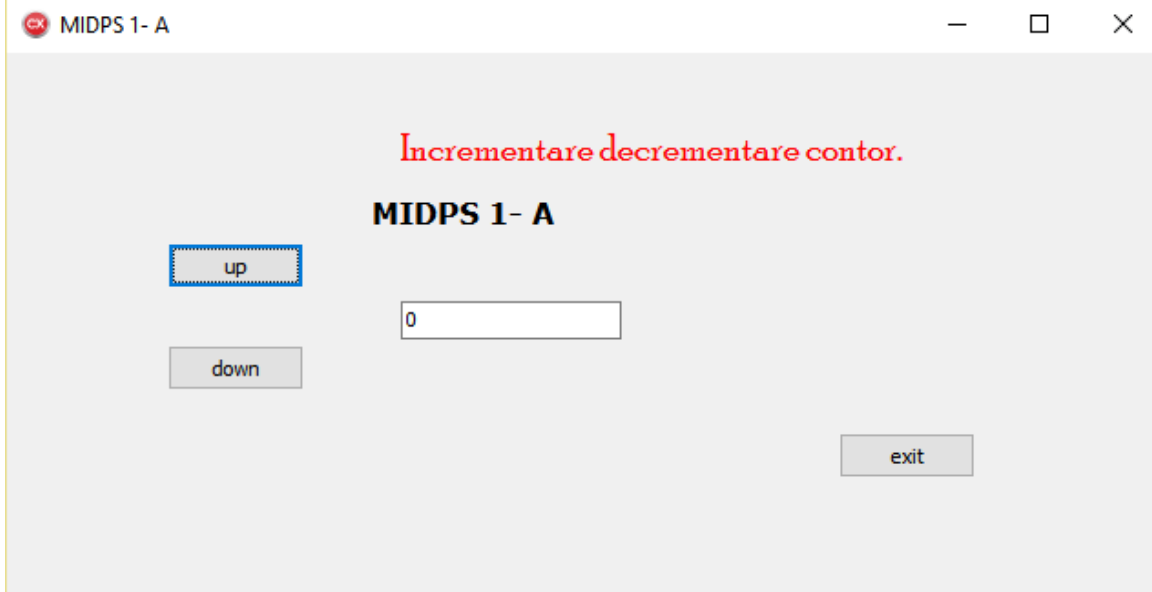
1. 2 label-uri ce reprezintă un sir de caractere-text, ce are nume, caption = textul afișat, și opțiuni de font
2. 3 butoane cu anumite denumiri
3. Un edit1 care reprezintă un form pentru introducere/afisare a datelor  
Aceste elemente sunt create și modificate doar prin interfața grafică.
4. Adăugarea Hinturilor

### **COD**

În cod, creăm o variabilă *int i* ce o egalăm inițial cu 0 .

1. În primul `__fastcall`, care este declansat la rularea programului, includem sub forma de text valoarea *i*-ului în Edit 1
2. La apăsarea butonului up, prin evenimentul `OnClick`, creăm o funcție , în care introducem incrementarea variabilei *i* și atribuirea valorii noi a lui *i* în edit1. La aceeași acțiune de apăsare s atribuie un alt sir pentru afișarea în label2, ce indică ultima schimbare efectuată.
3. La tastarea butonului 2 – down efectuăm aceeași acțiune ca și la butonul up, cu excepția incrementării ce este schimbată cu decrementarea, la fel și textul din label 2.
4. La tastarea butonului 3 – exit folosim funcția `Close()` ce execută ieșirea din program
5. Am efectuat hinturile prin interfața grafică accesăm câmpul `Hint` unde introducem denumirea și câmpul `Show Hint` unde avem nevoie de valoarea `True`.

## Screen program 1:



### Programul 2:

Crearea elementelor:

1. 4 butoane
2. 2 timer-uri ce sunt percepute ca elemente externe si se acceseaza din librarie dos-ului, sentindu-i intervalul de 1000 = 1 sec
3. 2 label-uri
4. 2 edit-uri
5. Hint-uri

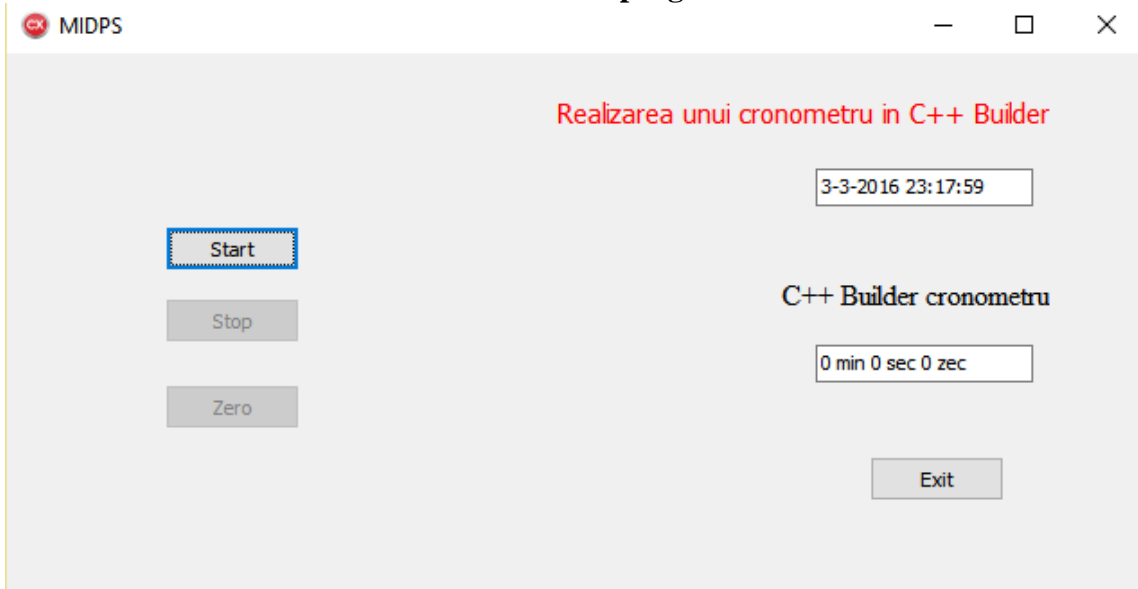
### COD

In cod, introducem 2 structuri necesare pentru data si timp si variabilele necesare

1. La rularea programului setam doar 2 dintre butoane sa fie Enabled – exit si start, pornim primul timer, indicam textul default in Edit2, utilizam functiile getdate, gettime in structurile create, dupa care cream un bufer de tip char, in care convertam toate valorile int returnate de dos despre timp, data, in format text. Apoi introducem baloare buf-erului in Edit 1
2. In functia Timer-ului 1 efectuam aceeasi operatie, care automat se pentrece la intervalul de 1 sec setat.
3. In cazul testarii butonului start, setam innacesibil acest buton, si setam accesibil stop, pornind timer-ul 2
4. In timer-ul 2 efectuam un crnonometru manula, atribuim tagului de timer o variabila, astfel in cit se acceseaza aceasta functie cu interval de care avem nevoie, folosim acea variabila ce se incrementeaza la fiecare iteratie, si in cazul in care e = 100, se egaleaza cu 0 so se incrementeaza contorul cu secunde, insa cind si acest este = 60 la fel se egaleaza cu 0, incrementindu-se contorul minutelor. Dupa aceasta procedura se converteaza rezultatul in textul de format necesar si se atribuie edit-ului 2
5. La accesarea butonului Stop, oprim timerul 2 si setam toate butoanele inaccesibile cu exceptia exit-ului si zero,

6. La accesarea butonul Zero, egalam toate contoarele de timp de la timerul 2 cu 0, paralel lasind accesibil doar butonul exit si startul.
7. Butonul exit proceseaza inchiderea programului
8. Setam hint-urile

### Screen program 2:



### Programul 3:

In cazul programului 3 am modificat neinstemnat functionalul programului propus in indrumar, pe motiv de a avea posibilitatea unei pauze in timpul executarii, si continuarii acestei executii, precum si resetarea pentru inceperea de la inceput a executiei.

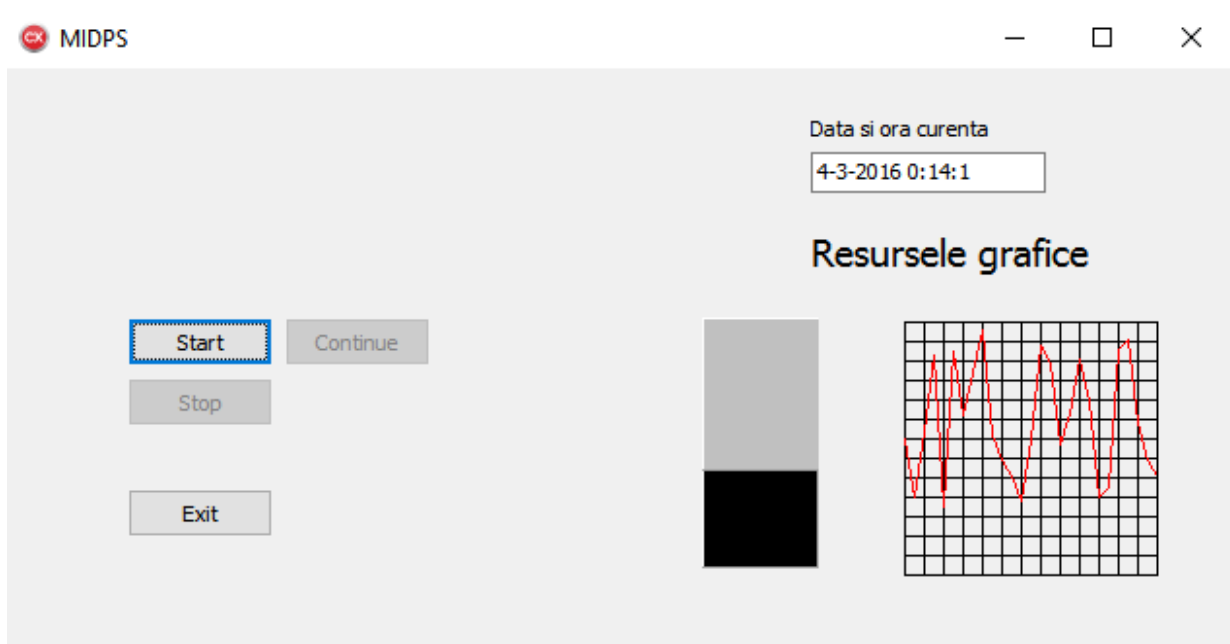
1. Creat 4 butoane
2. 2 timer-uri
3. 2 label-uri cu text
4. Un edit
5. Un panel format din 2 elemente 2tpanel ce reprezinta un dreptunghi de o anumita culoare(in cazul obiectivului nostru)
6. Un element PaintBox, care ne permite desenarea

### COD

1. La pornirea programului setam datele pentru PaintBox si repetam functia pentru ceas din programul trecut(o putem chiar copia[pasul 1]). Setam butoanele Start, Exit – accesibile. Mutam cursorul din PaintBox la 0,50 ce inseamna pe latura din stinga a dreptunghiului, aproximativ la mijlocul ei.
2. La apasarea butonului Start, el devine inaccesibil si accesibil devine stopul si porneste timerul 2. Prin functia FloodFill curatim ecranul Box-ului in cazul in care se ruleaza startul a 2-a data, prin 2 for-uri ce au pprogresia de 10px setam un grid ce e compus din linii paralel pe orizontala si verticala. Setam inaltimea Panelului interior – sur inaltimea egala cu x initial si cu moveto, ce reprezinta aceeasi inaltime pentru valoarea initiala a valorii random.

3. La accesarea Timerului 2 ce are intervalul 500 de apelare, de fiecare data folosim o variabila x ce va primi valori random in segmentul de inaltime a Box-ului, dupa care incrementam contorul i cu 5 care va fi folosit cu rolul de pas, pentru parcurgerea tabelului in dreapta pe orizontal, insa random-ul obtinut va obtine valori diferite ce vor crea o diagrama. Paralel inaltimea Panelului interior se va modifica fiind egal cu randomul, pentru a descrie evolutia sa avind rolul de baragraf. Acest for se va efectua pana valoarea i va fi egala cu latimea Box-ului.
4. Butonul Stop va opri timerul 2, pastrand imaginea „cardiogramei”, activizind butoanele Start pentru resetarea si inceperea unei noi proceduri sau butonul continuarii procedurii de la locul ramas.
5. Butonul Continue doar activeaza din nou timerul 2 ce pune in functiune graficul si baragraful, activizind doar butonul stop.
6. Butonul exit asigura iesirea din program
7. Setarea Hint-urilor.

### Screen program 3:



### Concluzii:

In aceasta lucrare de laborator am avut ocazia pentru prima data sa implementez cunostintele in cod paralel cu interfata grafica ce optimizeaza considerabil timpul si complexitatea efectuării unui program. In deosebi este usor setarea stilului sau a dimensiunilor programului, si functiile automate in cazul tastarii unui buton anumit, fara existenta necesitatii setarii manual sistemul Windows pentru citirea din firul de asteptare. Compilerul integrat la fel este foarte comod. In viziunea mea unicul minus Builderul C++ il reprezinta prin erori create daca initializam manual o functie de actiune in cod, spre exemplu OnClick, acest builder ne impune sa il facem din panelul de intrumente, altfel ne compileaza eroare. La fel pentru incepator mi-a luat ceva timp faptul ca salvam programul intr-un format propus, dupa care nu se deschidea, si era nevoie de a salva in format Project. In rest mi sa parut un mediu de programare foarte comod si simplu, totodata ce ofera un spectru larg optiuni in functionalitate.