INFORMATICA APLICATA

in ingineria electrica

Traian Turc

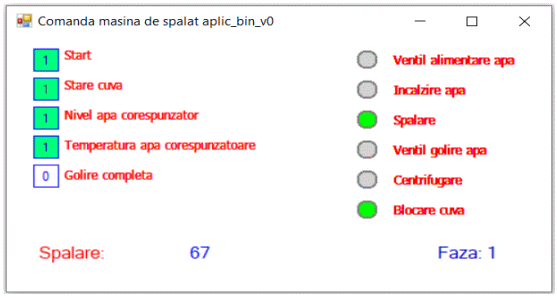
Introducere

Cui i se adreseaza cartea ?

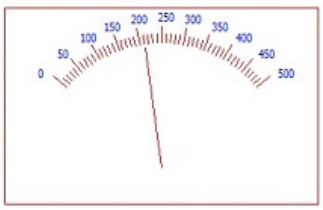
Cartea se adreseaza studentilor de la facultatile de inginerie electrica. Domeniile abordate in cadrul acestei carti sunt domenii din informatica ce au legatura cu domeniul ingineriei electrice in general, si in special cu domeniile automatizarilor, energeticii, tehnologiei informatice, ingineriei medicale. Elementele comune acestor domenii il reprezinta semnalele electrice. Cartea isi propune utilizarea informaticii pentru achizitia, analiza, prelucrarea si afisarea diverselor semnale electrice. Avand in vedere diversitatea semnalelor electrice, cartea propune utilizarea diverselor tipuri de date si structuri de date pentru pastrarea informatiilor referitoare la semnalele electrice. Se va pune accent pe afisarea datelor, utilizandu-se astfel elemente de programare obiect pentru realizarea de instrumente virtuale. Pentru achizitia datelor se va folosi un sistem de achizitie conectat pe USB care foloseste un protocol proprietar bazat pe siruri de caractere. Pentru salvarea datelor se vor vor fi folosite fisiere pe post de support de date.

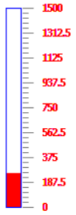
Domeniile abordate

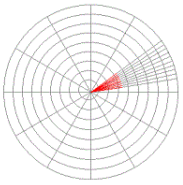
In sistemele electrice, in special in sistemele electronice actuale, majoritatea semnalelor electrice sunt semnale digitale, motiv pentru care prima parte a cartii se ocupa de formatul de date binar respective de operatii binare.Pe baza elementelor binare studiate, se propune o aplicatie pentru controlul unei masini de spalat care foloseste pentru inceput numai intrari si iesiri digitale. Sunt simulate intrarile folosind butoane, este implementata logica de functionare si sunt afisate comenzile sub forma de led-uri

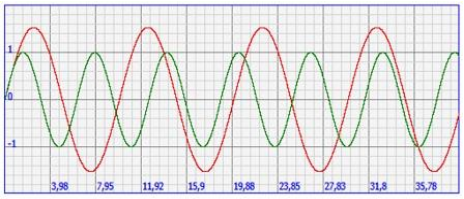


Afisarea semnalelor electrice constituie un element esential in ingineria electrica. Afisarea acestora se realizeaza in general cu echipamente electrice costisitoare, motiv pentru care sunt propuse o serie de aplicatii in care se realizeaza diverse instrumente virtuale pentru afisarea semnalelor electrice. In vederea realizarii instrumentelor virtuale, se vor utiliza elemente de programare obiect. Vor fi realizate atat instrumente pentru afisarea valorilor instantanee cat si pentru afisarea evolutiei in timp a acestora.

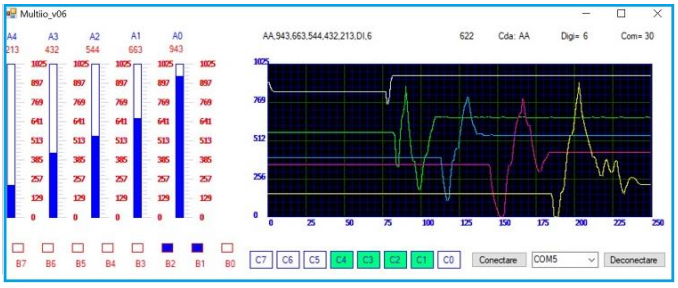








Pentru achizitia datelor se va folosi un sistem de achizitie conectat pe USB care permite citirra a 5 intrari analogice, 4 intrari digitale si 8 iesiri digitale



Operatii binare

Reprezentare binara, hexazecimala

In sistemele de calcul informatiile sunt codificate binar. Aceste informatii pot fi reprezentate binar, zecimal, hexazecimal sau sub diferite alte formate.

Vom analiza in continuare diferite aplicatii pentru conversia datelor si afisarea lor sub diferite formate.

- **Afisarea unui int sub forma binara**

Prin impartiri repetate cu 2 obtinem digitii ce reprezinta valoarea in binar.

```
// Programul afiseaza valoarea binara a unui intereg
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
            system("TITLE Afisare int sub forma binar");
            system("COLOR F9");
            int n,i;  // n:numarul de convertit
            cout << "\n\tProgramul afiseaza valoarea binara a unui intereg " ;
            cout << "\n\n\tIntroduceti un numar intreg: " ;
            cin >> n;
            cout << " \n\tValoarea binara afisata in ordine inversa este:\n\n\t";
   if (n > 0) {
    for(i=0; i < 32 ; i++) {
      cout << n%2;
      n = n/2;
    }
   }
   else {
    cout << "\n\n\tIntroduceti un numar pozitiv\n" << endl;
   }
            cin.ignore();
            cin.get();
   return 0;
}
```

Pentru a afisa bitii in ordine directa, va trebui sa memoram acesti biti pentru a fi afisati la sfarsit.

```
#include "stdafx.h"
#include < iostream >
using namespace std;
int main(void)
{
            system("TITLE Afisare int sub forma binar");
            system("COLOR F9");
            int n,i;  // number to convert to binary
```

```
        char val_b[32];
        cout << "\n\tProgramul afiseaza valoarea binara a unui intereg " ;
        cout << "\n\n\tIntroduceti un numar intreg: " ;
        cin >> n ;
        for (i=0; i < 31; i++){
                if (n%2==0)
                        val_b[30-i]='0';
                else
                        val_b[30-i]='1';
                n=n/2;
        }
        cout << "\n\t Valoarea binara este:";
        for (i=0; i < 31; i++)
        cout << val_b[i] ;
        cin.ignore();
        cin.get();
  return 0;
}
```

- **Afisarea unui int sub forma hexa**

Pentru a afisa sub forma hexa zecimala, vom utiliza operatorul **<< hex** care inclus in cadrul instructiunii **cout <<** va forta afisarea valorilor in format hexa.

```
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Afisare int sub forma hexa");
        system("COLOR F9");
        int n;
        cout << "\n\tProgramul afiseaza valoarea hexa a unui intereg " ;
        cout << "\n\n\tIntroduceti un numar intreg: " ;
        cin >> n;
        cout << " \n\tValoarea hexa este:"<< hex << n;
        cin.ignore();
        cin.get();
  return 0;
}
```

Dupa utilizarea operatorului **<< hex** trebuie utilizat operatorul **<< dec** pentru a reveni la afisarea sub forma zecimala.

```
// Programul afiseaza valoarea hexa a unui intereg
#include "stdafx.h"
#include < iostream >
using namespace std;
int main(void)
{
	system("TITLE Afisare int sub forma hexa");
	system("COLOR F9");
	int n;
	cout << "\n\tProgramul afiseaza valoarea hexa a unui intereg " ;
	cout << "\n\n\tIntroduceti un numar intreg: " ;
	cin >> n;
	cout << " \n\tValoarea hexa este: "<< hex << n;
	cout << " \n\tValoarea lui n este: "<< n;
	cout << " \n\tValoarea zecimala este:"<< dec << n;
	cin.ignore();
	cin.get();
   return 0;
}
```

Afisarea in format zecimal, binar, hexa sau in orice alta baza se poate face si prin utilizarea functiei **itoa**

```
// Conversia unui intreg intr-un sir reprezentand valoarea in diferite baze de numeratie
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Afisarea in diferite baze de numeratie");
	system("COLOR F9");
	int n;
	char buffer [33];
	cout << "\n\n\tIntroduceti un numar intreg: " ;
	cin >> n;
	itoa(n,buffer,10);//functia pentru conversia unui intreg intr-un sir
	cout << "\n\n\tValoarea zecimala a numarului este: " << buffer;
	itoa(n,buffer,2);
	cout << "\n\n\tValoarea binara a numarului este: " << buffer;
	itoa(n,buffer,16);
	cout << "\n\n\tValoarea hexa a numarului este: " << buffer;
	cin.ignore();
	cin.get();
 return 0;
}
```

Operatori binari

Deseori este nevoie sa folosim valori binare sau e nevoie sa convertim in format binar diferite date de diverse tipuri. C++ permite diverse operatii binare, operatii ce le vom folosi in continuare.

• AND logic -- operatorul &

Operatia AND este deseori folosita pentru a realizeaza o masca pentru extragerea anumitor biti dintr-un operand. Operatorul folosit este **&**

```
// Programul realizeaza operatia AND
// Se calculeaza 0xffff AND 0x5555  adica  1111111111111111 & 0101010101010101
// Rezultatul este 0x5555 adica 0101010101010101
// Operatia realizeaza o masca pentru extragerea anumitor biti dintr-un operand
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;

int main(void)
{
	system("TITLE AND logic ");
	system("COLOR F9");
	unsigned short a = 0xFFFF;	// = 1111111111111111
	unsigned short b = 0x5555;	// = 0101010101010101
	cout <<"\n\n\tProgramul calculeaza 0xffff AND 0x5555";
	cout <<"\n\n\t0xffff & 0x5555=" << hex << ( a & b );  // rezultat "5555" adica 0101010101010101
	cin.ignore();
	cin.get();
}
```

Operatia AND este deseori folosita pentru a realizeaza o masca pentru extragerea anumitor biti dintr-un operand. Operatorul folosit este **&**

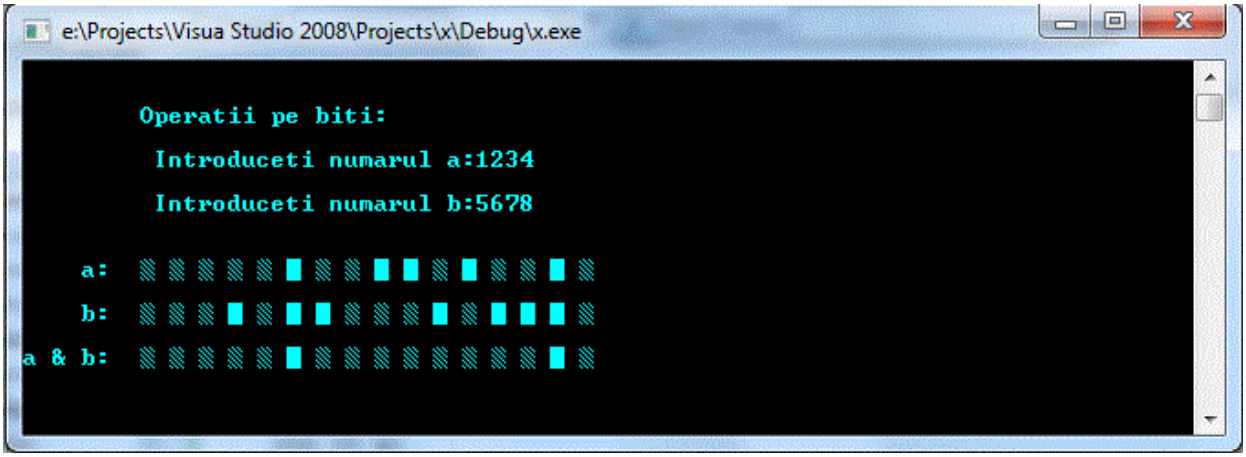
Pentru exemplificare, vom realiza in continuare o aplicatie care realizeaza functia logica & intre doi operanzi si sub forma binara atat operanzii cat si rezultatul. Aplicatia se bazeaza pe functiile afis_binar respectiv afis_bin

Programul principal fiind:

```
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	unsigned int a,b;
	cout << " \n\n\tOperatii pe biti: ";
	cout <<"\n\n\t Introduceti numarul a:";
	cin >> a;
```

```
        cout <<"\n\t Introduceti numarul b:";
        cin >> b;
        cout<<"\n\n    a:\t";
        afis_bin(a,16);
        cout<<"\n\n    b:\t";
        afis_bin(b,16);
        cout<<"\n\na & b:\t";
        afis_bin(a&b,16);
        cin.ignore();
        cin.get();
        return 0;
}
```

Dupa rularea programului in "Command Prompt" se afiseaza:



Daca se doreste introducerea operanzilor in format hexa, programul devine:

```
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        unsigned int a,b;
        cout << " \n\n\tOperatii pe biti: ";
        cout <<"\n\n\t Introduceti numarul a:";
        cin >> hex >> a;
        cout <<"\n\t Introduceti numarul b:";
        cin >> hex >> b;
        cout<<"\n\n    a:\t";
        afis_bin(a,16);
        cout<<"\n\n    b:\t";
```

```
	afis_bin(b,16);
	cout<<"\n\na & b:\t";
	afis_bin(a&b,16);
	cin.ignore();
	cin.get();
	return 0;
}
```

- **SAU logic -- operatorul |**

Operatia SAU este deseori folosita pentru a realizeaza o setare a anumitor biti dintr-un operand. Operatorul folosit este **|**

```
// Programul realizeaza operatia OR
// Se calculeaza 0xaaaa OR 0x5555  adica  1010101010101010 & 0101010101010101
// Rezultatul este 0xffff adica 1111111111111111
// Operatia realizeaza o setare anumitor biti dintr-un operand
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;

int main(void)
{
	system("TITLE SAU logic ");
	system("COLOR F9");
	unsigned short a = 0xaaaa;	// = 1010101010101010
	unsigned short b = 0x5555;	// = 0101010101010101
	cout <<"\n\n\tProgramul calculeaza 0xaaaa AND 0x5555";
	cout <<"\n\n\t0xaaaa & 0x5555=" << hex << ( a | b ); // rezultat "ffff" adica 1111111111111111
	cin.ignore();
	cin.get();
}
```

- **SAU exclusiv - operatorul ^**

Operatia SAU exclusiv este deseori folosita pentru complementarea anumitor biti dintr-un operand. Operatorul folosit este **^**

```
// Programul realizeaza operatia SAU EXCLUSIV
// Se calculeaza 0x5555 SAU EXLLUSIV 0xffff adica 0101010101010101 ^ 1111111111111111
// Rezultatul este 0xaaaa adica 1010101010101010
// Operatia complementeaza bitii primului operand
#include "stdafx.h"
#include < iostream >
```

```
#include < string >
using namespace std;

int main(void)
{
	system("TITLE SAU EXCLUSIV ");// Titlul ferestrei consola
	system("COLOR F9");			// Fundal alb caractere albastre
	unsigned short a = 0x5555;	// = 0101010101010101
	unsigned short b = 0xFFFF;	// = 1111111111111111
	cout <<"\n\n\tProgramul calculeaza 0x5555 SAU EXLLUSIV 0xffff";
	cout <<"\n\n\t0x5555 ^ 0xffff=" << hex << ( a ^ b );	// rezultat "aaaa" adica 1010101010101010
	cin.ignore();
	cin.get();
}
```

- **NOT - operatorul ~**

Operatia NOT realizeaza complementarea bitilor dintr-un operand. Operatorul folosit este **~**

```
// Programul realizeaza operatia NOT
// Se calculeaza NOT 0xaaaa  adica  NOT 1010101010101010
// Rezultatul este 0x5555 adica 0101010101010101
// Operatia realizeaza complementarea bitilor dintr-un operand
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;

int main(void)
{
	system("TITLE NOT logic ");
	system("COLOR F9");
	unsigned short a = 0xaaaa;	// = 1010101010101010
	cout <<"\n\n\tProgramul calculeaza NOT 0xaaaa ";
	cout <<"\n\n\t NOT 0xaaaa= " << hex << ~a ;		// rezultat "5555" adica 0101010101010101
	cin.ignore();
	cin.get();
}
```

- **Delasare dreapta - operatorul >>**

n >> p;

Deplaseaza spre dreapta cu p pozitii a bitilor ce compun in binar valoarea n. Pe pozitia cea mai semnificativa se pune 0. O deplasare spre dreapta cu 1 pozitie este echivalenta cu o impartire cu 2 cu 2. Astfel 24 << 3 = 3

Sa realizam o aplicatie care deplaseaza dreapta cu doua pozitii valoarea 64. Dupa rularea programului ar trebui sa obtinem valoarea 16.

```
// Deplasare dreapta
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
            system("TITLE Deplasare dreapta");
            system("COLOR F9");
            cout << "\n\n\tDeplasare dreapta";
            unsigned short int n=64;
            int i;
            cout << "\n\n\tValoarea initiala a lui n: "<< n;
                        n = n >> 2 ;
            cout << "\n\n\tValoarea lui n dupa deplasarea cu doua pozitii dreapta: "<< n;
            cin.ignore();
            cin.get();

 return 0;
}
```

Afisarea sub forma binara este mult mai simpla daca se utilizeaza operatorul de siftare dreapta

```
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
            system("TITLE Afisare int sub forma binar");
            system("COLOR F9");
  int n;
    cout << "\n\tProgramul afiseaza valoarea binara a unui intereg " ;

    cout << "\n\n\tIntroduceti un numar intreg: " ;
              cin >> n;
    cin.ignore();
    // print binary with leading zeros
              cout << "\n\tValoarea binara este: : ";
    for (int i=31; i>=0; i--) {
      int bit = ((n>>i) & 1);
      cout << bit;
    }
    cin.get();
   return 0;
}
```

- **Delasare stanga - operatorul <<**

n << p ;

Deplaseaza spre stanga cu p pozitii a bitilor ce compun in binar valoarea n. Pe pozitia cea mai nesemnificativa se pune 0. O deplasare spre stanga cu 1 pozitie este echivalenta cu o inmultire cu 2. Astfel 3 << 2 =12

Sa realizam o aplicatie care deplaseaza stanga cu doua pozitii valoarea 7. Dupa rularea programului ar trebui sa obtinem valoarea 28.

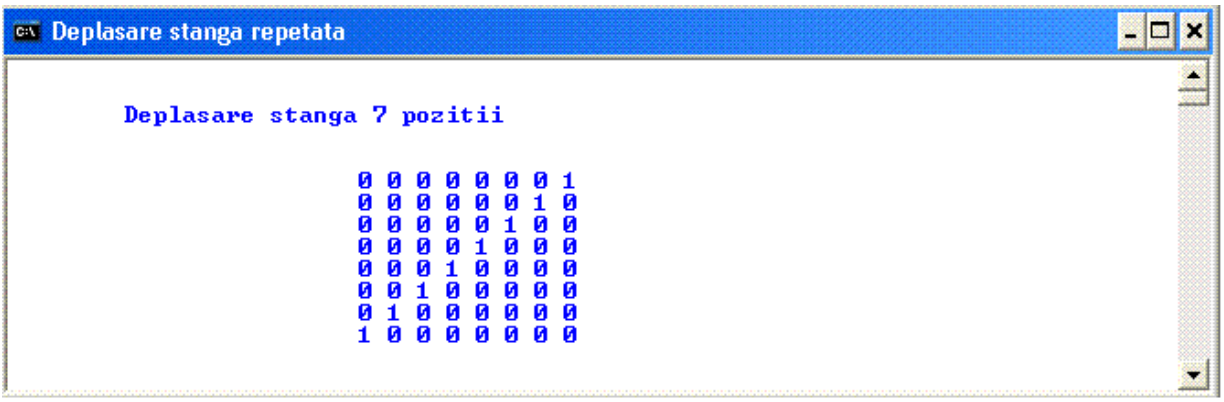
```
// Deplasare stanga
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Deplasare stanga");
	system("COLOR F9");
	cout << "\n\n\tDeplasare stanga";
	unsigned short int n=7;
	int i;
	cout << "\n\n\tValoarea initiala a lui n: "<< n;
		n = n << 2 ;
	cout << "\n\n\tValoarea lui n dupa deplasarea cu doua pozitii stanga: "<< n;
	cin.ignore();
	cin.get();

 return 0;
}
```

Sa utilizam operatorul de siftare stanga pentru a deplasa stanga valoarea 1 de 7 ori si sa afisam in binar valoarea obtinuta dupa fiecare deplasare.

```
// Deplasare stanga repetata
#include "stdafx.h"
#include < iostream >
using namespace std;
void af_binar(unsigned int );

int main(void)
{
	system("TITLE Deplasare stanga repetata");
	system("COLOR F9");
	cout << "\n\n\tDeplasare stanga 7 pozitii \n\n\n\t\t\t";
	unsigned short int n=1;
	int i;
	for(i=0; i < 8; i++) {
```

```
                    af_binar(n);
                    n = n << 1 ;
          }
          cin.ignore();
          cin.get();

 return 0;
}
// Afisarea bitilor ce corespund valorii parametrului u
void af_binar(unsigned int u)
{
 int j;
    for (int j=7; j>=0; j--) {
      int bit = ((u >> j) & 1);
      cout << bit<<" ";
    }
 cout << "\n\t\t\t\t";
}
```

Rulam programul si obtinem:



Utilizand operatiile de siftare precum si operatiile logice, vom realiza in continuare afisarea unui int sub forma hexa fara a folosi operatorul **<< hex**

Pentru a afisa sub forma hexa zecimala, vom face siftari dreapta cu cate 4 biti si vom interpreta ultimii patru biti utilizand masca 0xF adica 00001111.

```
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
          system("TITLE Afisare int sub forma hexa");
          system("COLOR F9");
  int n;
```

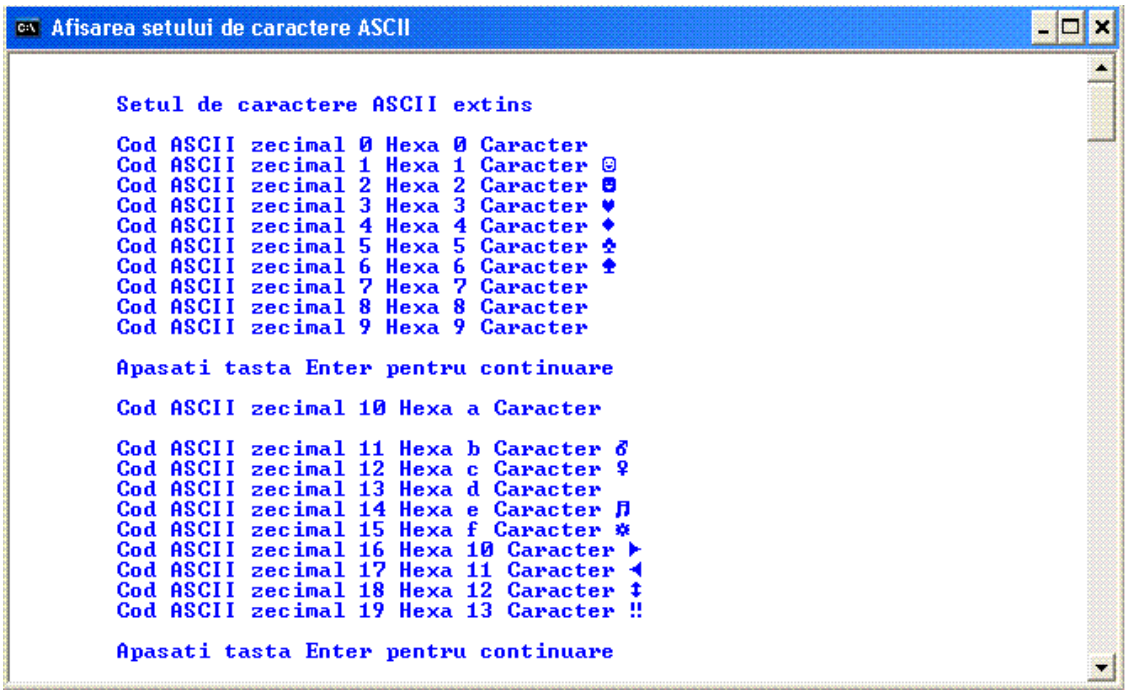
```
        char* c_hex="0123456789ABCDEF";
  cout << "\n\tProgramul afiseaza valoarea hexa a unui intereg " ;
  cout << "\n\n\tIntroduceti un numar intreg: " ;
        cin >> n;
        cout << " \n\tValoarea hexa este:\n\n\t";
   for (int i=2*sizeof(int) - 1; i>=0; i--) {
     cout << c_hex[((n >> i*4) & 0xF)];
   }
         cin.ignore();
         cin.get();
  return 0;
}
```

- **Coduri ASCII**

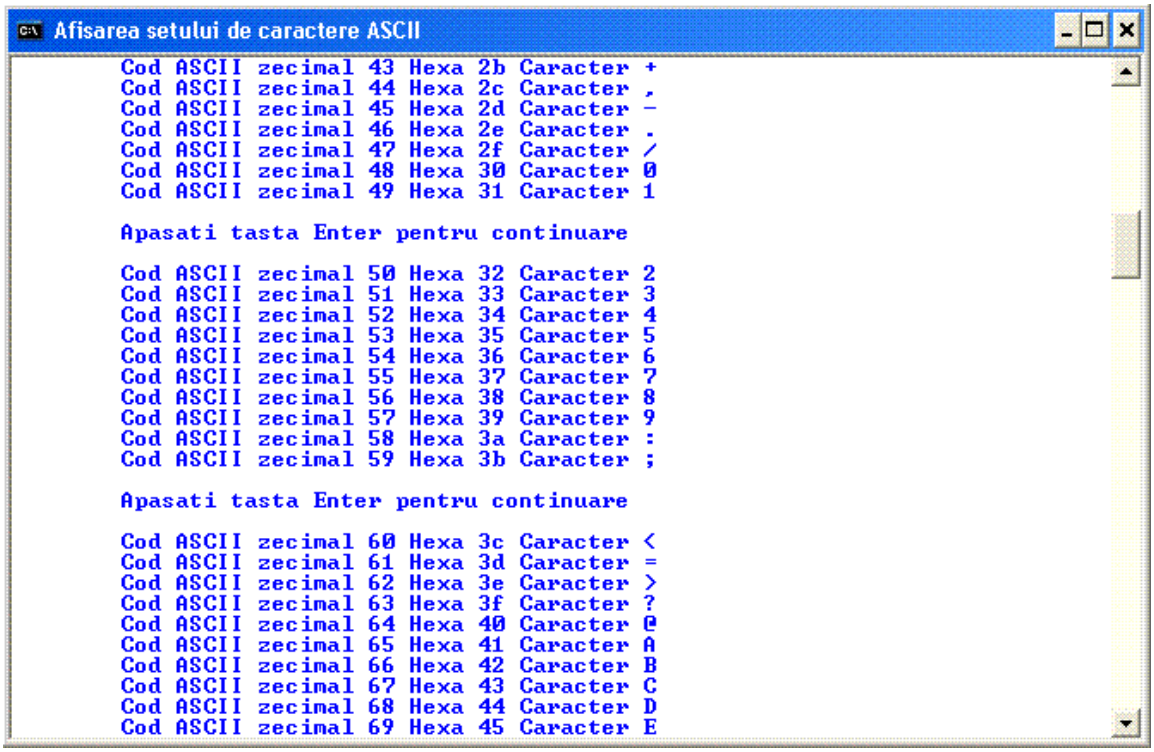
Functia **char(i)** converteste o valoare intreaga intre 0 si 255 intr-un caracter ASCII.
Urmatoarea aplicatie foloseste functia **char(i)** si afiseaza setul extins de caractere ASCII

```
//  Afisarea setului de caractere ASCII
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Afisarea setului de caractere ASCII");
        system("COLOR F9");
        int i ;
        cout << "\n\n\tSetul de caractere ASCII extins\n";
        for(i= 0;i < 256; i++){
                if ((i >0) && (i % 10 == 0)){
                        cout << "\n\n\tApasati tasta Enter pentru continuare ";
                        cin.get();
    }
                cout <<"\n\tCod ASCII zecimal " << dec << i <<" Hexa "<< hex << i << " Caracter " << char(i);
  }
        cout << "\n\n\tPentru a termina apasati tasta Enter";
        cin.get();
 return 0;
}
```

Dupa rularea aplicatiei, obtinem :



Apasand tasta Enter in continuare obtinem intreg setul extins de caractere ASCII



Pentru a face conversia inversa din caracter ASCII-int se procedeaza astfel:

```
//  Conversia  unui caracter ASCII intr-o valoare de tip int
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
            system("TITLE Conversia  ASCII int ");
            system("COLOR F9");
            char car;
            int val_ascii;
            cout << "\n\n\tIntroduceti un caracter :";
            cin.get(car);
            val_ascii = static_cast< int >(car);
            // Se poate atribui lui val_ascii  direct car
            // fara a fi nevoie de conversiastatic_cast (car) astfel:
            // val_ascii = car;
            cout << "\n\n\tValoarea ascii in zecimal a caracterului este: "<< val_ascii;
            cout << "\n\n\tValoarea ascii in hexa a caracterului este: "<< hex << val_ascii;
            cin.ignore();
            cin.get();
 return 0;
}
```

Cunoscand modul de codificare al caracterelor (codul ASCII) sa realizam un program cere un sir de caractere ce contine litere mari si mici, dupa care converteste literele mari in litere mici si afiseaza sirul rezultat.

```
// Programul cere un sir de caractere care contine litere mari si mici.
// Converteste literele mari in litere mici si afiseaza sirul.
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
void conv_lit(char s[]);

int main(void)
{
system("TITLE Conversie litere mari in litere mici");
system("COLOR F9");
char s[25];
int l;
cout<<"\n\tProgramul cere un sir de caractere cu litere mari si mici";
cout<<"\n\tSe va afisa sirul convertit in litere mici";
```

```
cout<<"\n\n\tIntroduceti un sir de caractere cu \n\tformat din litere mari si mici : ";cin >> s;
conv_lit(s);
cout << "\n\n\tSirul transformat in litere mici este : " << s;
cin.ignore();
cin.get();
return 0;
}
void conv_lit(char s[25])
{
int i;
i=0;
while(s[i]!=0)
{
if( (s[i] >= 65) && (s[i] <= 90) ) s[i]=s[i]+32;
i++;
}
return;
}
```

Operatii binare in spatiul system

- **Afisare grafica valori binare**

Bazandu-ne pe operatorii binari studiati si aplicatiile realizate vom realiza simularea grafica a operatiilor binare. Vom afisa bitii ce reprezinta valoarea unei variabile prin mici dreptunghiuri. Pentru inceput vom realiza o aplicatie de tipul CLR Windows Forms Application are afiseaza secvente de dreptunghiuri pe ecran.

Deschidem un nou proiect Windows Forms Application intitulat **"secv_dr"** pe care plasam un obiect de tip button numit button1 caruia ii schimbam atributul text in "Start" si un obiect timer pe caruia ii setam intervalul la 50. De data aceasta nu setam proprietatea "Enabled" la "true", o lasam "false".

Initializam variabilele i,h,w, obiectele "Desen" si "Pensula" in zona "#pragma region" pentru a asigura domeniul de vizibilitate in proceduri le stasate diferitelor evenimente.

Completam deci #pragma region cu :

```
        static int i, w, h; // h, v dimensiunile unui dreptunghi
        static System::Drawing::Graphics^ Desen;
        static System::Drawing::SolidBrush^ Pensula;
```

Pe evenimentul click al obiectului button1 punem procedura:

```
        Desen = this->CreateGraphics();
        Pensula=gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);
```

```
        Desen->Clear(System::Drawing::Color(this->BackColor));
        w=this->Size.Width/60;
        h=this->Size.Height/20;
        i=2*w;
        this->timer1->Enabled=true;
```

Se observa ca validarea "timer-ului" se face pe evenimentul click al obiectului button1
Completam procedura deschisa pe evenimentul Tick al obiectului timer1 cu :

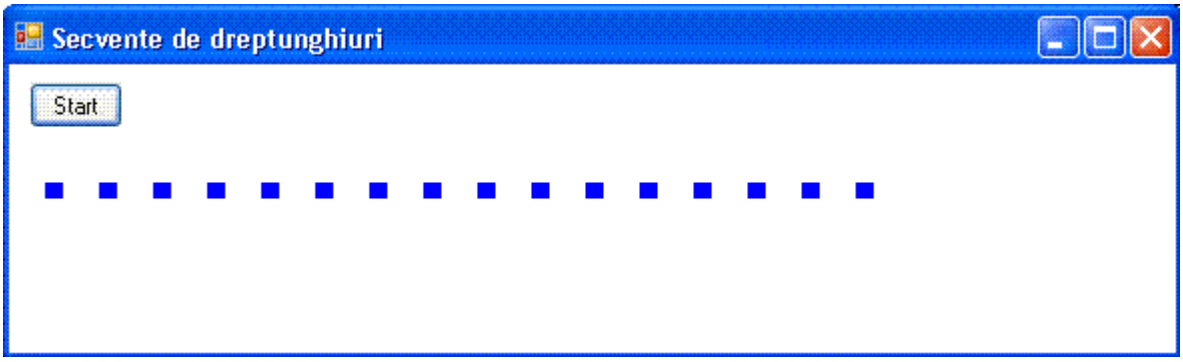
```
        Desen->FillRectangle(Pensula, i, this->Size.Height/3, w,h);
        if (i <= this->Size.Width-3*w)
                i+=3*w;
        else {
                i=2*w;
                Desen->Clear(System::Drawing::Color(this->BackColor));
        }
```

C#

```
namespace secv_dr
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    static int i, w, h; // h, v dimensiunile unui dreptunghi
    static System.Drawing.Graphics Desen;
    static System.Drawing.SolidBrush Pensula;
    static System.Drawing.Pen Creion_blu;
    private void button1_Click(object sender, EventArgs e)
    {
      Desen = this.CreateGraphics();
      Pensula = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
      Creion_blu = new System.Drawing.Pen(System.Drawing.Color.Blue);
      w = this.Size.Width / 60;
      h = this.Size.Height / 20;
      this.timer1.Enabled = true;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      Desen.FillRectangle(Pensula, i, this.Size.Height/3, w,h);
      if (i <= this.Size.Width - 3 * w)
```

```
        {
          i += 3 * w;
        }
        else
        {
          i = 2 * w;
          Desen.Clear(this.BackColor);
        }
      }
    }
}
```

Dupa rularea aplicatiei, obtinem secvente dinamice de dreptunghiuri:



Urmatoarea aplicatie isi propune sa converteasca o valoare numerica in binar dupa care sa afiseze aceasta valoare sub forma de dreptunghiuri pline pentru "1" si dreptunghiuri goale pentru "0"

Deschidem un nou proiect Windows Forms Application intitulat **"binar"** pe care plasam un obiect de tip button numit button1 caruia ii schimbam atributul text in "Start".

Completam deci #pragma region cu :

```
        static unsigned int n; // numarul ce va fi convertit in binar si afisat grafic
        static int i, w, h; // h, v dimensiunile unui dreptunghi
        static System::Drawing::Graphics^ Desen;
        static System::Drawing::SolidBrush^ Pensula;
        static System::Drawing::Pen^ Creion_blu ;
```

Pe evenimentul click al obiectului button1 punem procedura:

```
        Desen = this->CreateGraphics();
        Pensula=gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);
```

```
        Creion_blu=gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
        Desen->Clear(System::Drawing::Color(this->BackColor));
        w=this->Size.Width/96;
        h=this->Size.Height/20;
        i=2*w;
        n=0xaaaaaaaa;
        int val_b[32];
        for (i=0; i < 32; i++){
                if (n%2==0)
                        val_b[31-i]=0;
                else
                        val_b[31-i]=1;
                n=n/2;
        }
        int x=2*w;
        for (i=0;i < 32;i++){
        if (val_b[i]==1)
                Desen->FillRectangle(Pensula, x, this->Size.Height/3, w,h);
        else
                Desen->DrawRectangle(Creion_blu, x, this->Size.Height/3, w,h);
        x+=3*w;
        }
```

C#

```
namespace binar
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      static System.Drawing.Graphics Desen;
      static System.Drawing.SolidBrush Pensula;
      static System.Drawing.Pen Creion_blu;
      static System.UInt32 n; // numarul ce va fi convertit in binar si afisat grafic
      static int i, w, h; // h, v dimensiunile unui dreptunghi
      int [ ] val_b = new int [32];
      private void button1_Click(object sender, EventArgs e)
      {
         Desen = this.CreateGraphics();
                Pensula=new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
                Creion_blu=new System.Drawing.Pen(System.Drawing.Color.Blue);
                Desen.Clear(this.BackColor);
                w=this.Size.Width/96;
                h=this.Size.Height/20;
         int x = 2 * w;
```

```
                i=2*w;
                n=0xaaaaaaaa;
                for (i=0; i < 32; i++){
                        if (n%2==0)
                                val_b[31-i]=0;
                        else
                                val_b[31-i]=1;
                        n=n/2;
                }
                for (i=0;i < 32;i++){
                 if (val_b[i]==1)
                        Desen.FillRectangle(Pensula, x, this.Size.Height/3, w,h);
                 else
                        Desen.DrawRectangle(Creion_blu, x, this.Size.Height/3, w,h);
                 x+=3*w;
        }
     }
   }
}
```

Aplicatia poate fi simplificata, folosind operatorul de siftare. Pe evenimentul click al obiectului button1 vom plasa noua procedura :

```
        Desen = this->CreateGraphics();
        Pensula=gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);
        Creion_blu=gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
        Desen->Clear(System::Drawing::Color(this->BackColor));
        w=this->Size.Width/96;
        h=this->Size.Height/20;
        int x=this->Size.Width-10*w;
        n=0x00aa;
        for (i=31; i >= 0; i--){
        int bit=((n >>(31-i)) & 1);
        if (bit==1)
                Desen->FillRectangle(Pensula, x, this->Size.Height/3, w,h);
        else
                Desen->DrawRectangle(Creion_blu, x, this->Size.Height/3, w,h);
        x-=3*w;
        }
```

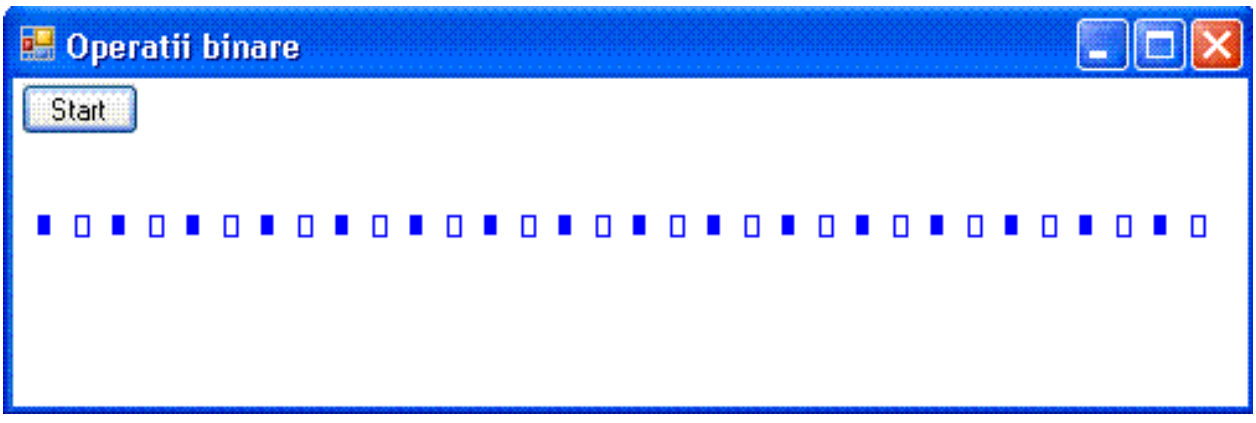
Daca realizam un nou proiect **"binar_v1"** si inlocuim numai procedura de pe evenimentul click al obiectului button1 aplicatia are acelasi rezultat. Observam ca nu a mai fost necesar sa introducem un vector care sa pastreze bitii ce compun numarul afisat.

In C# aplicatia devine::

```
namespace binar_v1
{
  public partial class Form1 : Form
```

```
  {
    public Form1()
    {
      InitializeComponent();
    }
    static System.UInt32 n; // numarul ce va fi convertit in binar si afisat grafic
    static int i, w, h; // h, v dimensiunile unui dreptunghi
    static System.Drawing.Graphics Desen;
    static System.Drawing.SolidBrush Pensula;
    static System.Drawing.Pen Creion_blu;
    private void button1_Click(object sender, EventArgs e)
    {
      Desen = this.CreateGraphics();
      Pensula = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
      Creion_blu = new System.Drawing.Pen(System.Drawing.Color.Blue);
      Desen.Clear(this.BackColor);
      w = this.Size.Width / 96;
      h = this.Size.Height / 20;
      int x = this.Size.Width - 10 * w;
      n = 0xaaaaaaaa;
      for (i = 31; i >= 0; i--)
      {
       System.UInt32 bit = ((n >> (31 - i)) & 1);
       if (bit == 1)
         Desen.FillRectangle(Pensula, x, this.Size.Height / 3, w, h);
       else
       Desen.DrawRectangle(Creion_blu, x, this.Size.Height / 3, w, h);
       x -= 3 * w;
      }
    }
  }
}
```

Dupa rularea aplicatiei, obtinem valoarea in binar afisata grafica a numarului 0xaaaaaaaa:



- **Afisare grafica valori ASCII**

Ne propunem sa realizam o aplicatie care asteapta tiparirea unui caracter, il converteste in cod ASCII si il afiseaza sub forma grafica.

Deschidem un nou proiect Windows Forms Application intitulat **"binar_v2"** pe care plasam un obiect de tip textBox numit textBox1 si doua obiecte de tip label : label1 si label2 .

Completam #pragma region cu :

```
        static unsigned short int n; // numarul ce va fi convertit in binar si afisat grafic
        static int i, w, h; // h, v dimensiunile unui dreptunghi
        static System::Drawing::Graphics^ Desen;
        static System::Drawing::SolidBrush^ Pensula;
        static System::Drawing::Pen^ Creion_blu ;
        static System::String^ txt;
```

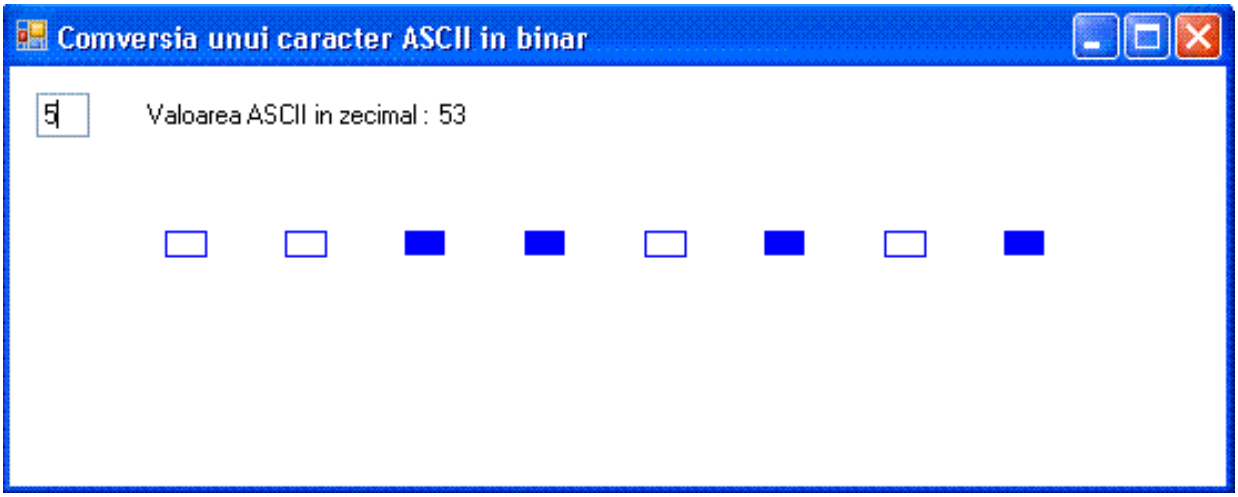
Completam procedura deschisa pe evenimentul "TextChanged" al butonului textBox1 cu :

```
        Desen = this->CreateGraphics();
        Pensula=gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);
        Creion_blu=gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
        Desen->Clear(System::Drawing::Color(this->BackColor));
        w=this->Size.Width/30;
        h=this->Size.Height/20;
        if (System::String::Compare(this->textBox1->Text ,System::String::Empty)){
                txt=this->textBox1->Text;
                char c=txt[0];
                c=System::Convert::ToChar(c);
                n=System::Convert::ToByte(c);
                this->label2->Text=System::Convert::ToString(n);
                int x=this->Size.Width-6*w;
                for (i=7; i >= 0; i--){
                        int bit=((n >> (7-i)) & 1);
                        if (bit==1)
                                Desen->FillRectangle(Pensula, x, this->Size.Height/3, w,h);
                        else
                                Desen->DrawRectangle(Creion_blu, x, this->Size.Height/3, w,h);
                        x-=3*w;
                }
  }
```

C#

```
namespace binar_v2
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      static int n; // numarul ce va fi convertit in binar si afisat grafic
      static int i, w, h; // h, v dimensiunile unui dreptunghi
      static System.Drawing.Graphics Desen;
      static System.Drawing.SolidBrush Pensula;
      static System.Drawing.Pen Creion_blu;
      static System.String txt;
      private void textBox1_TextChanged(object sender, EventArgs e)
      {
         Desen = this.CreateGraphics();
                  Pensula=new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
                  Creion_blu=new System.Drawing.Pen(System.Drawing.Color.Blue);
                  Desen.Clear(this.BackColor);
                  w=this.Size.Width/30;
                  h=this.Size.Height/20;
                  if (this.textBox1.Text.Length>0){
                              txt=this.textBox1.Text;
                              char c=txt[0];
                              c=System.Convert.ToChar(c);
                              n=System.Convert.ToByte(c);
                              this.label2.Text=System.Convert.ToString(n);
                              int x=this.Size.Width-6*w;
            for (i = 7; i >= 0; i--)
            {
               int bit = ((n >> (7 - i)) & 1);
               if (bit == 1)
                  Desen.FillRectangle(Pensula, x, this.Size.Height / 3, w, h);
               else
                  Desen.DrawRectangle(Creion_blu, x, this.Size.Height / 3, w, h);
               x -= 3 * w;
            }
         }
      }
   }
}
```

Rulam aplicatia , tastam cifra 5 si obtinem:



- **Afisare grafica shift**

Folosind operatiile de deplasare stanga respectiv dreapta ,ne propunem sa realizam o aplicatie care simuleaza un joc de lumini.

Deschidem un nou proiect Windows Forms Application intitulat **"binar_v3"** . Plasam numai un singur obiect de tip timer numit timer1.

Completam #pragma region cu :

```
static unsigned short int n=0x5555; // numarul ce va fi deplasat stanga dreapta si afisat
static int i, x, w, h; // h, v dimensiunile unui dreptunghi
static int sem=0;
static System::Drawing::Graphics^ Desen;
static System::Drawing::SolidBrush^ Pensula;
static System::Drawing::Pen^ Creion_blu ;
```

Completam procedura deschisa pe evenimentul "Paint" al form-ului cu :

```
Desen = this->CreateGraphics();
Pensula=gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);
Creion_blu=gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
Desen->Clear(System::Drawing::Color(this->BackColor));
w=this->Size.Width/50;
h=this->Size.Height/20;
```

Completam procedura deschisa pe evenimentul "Tick" al timer-ului cu :

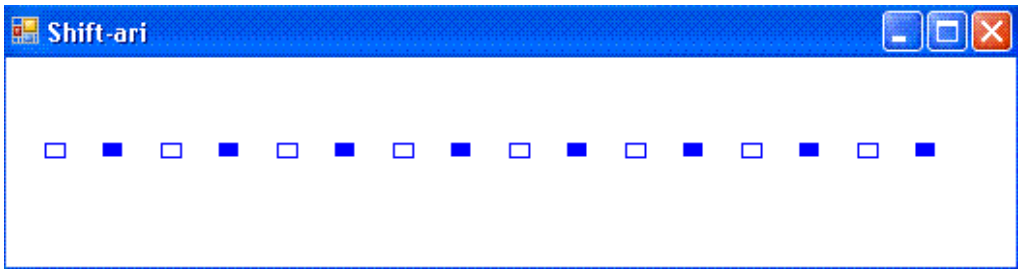
```
        Desen->Clear(System::Drawing::Color(this->BackColor));
        x=this->Size.Width-6*w;
        if (sem==0){
                n=n << 1;
                sem=1;
        }else{
                n=n >> 1;
                sem=0;
        }
        for (i=15; i >= 0; i--){
        int bit=((n >> (15-i)) & 1);
        if (bit==1)
                Desen->FillRectangle(Pensula, x, this->Size.Height/3, w,h);
        else
                Desen->DrawRectangle(Creion_blu, x, this->Size.Height/3, w,h);
        x-=3*w;
        }
```

C#

```
namespace binar_v3
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
            int i; // numarul ce va fi deplasat stanga dreapta si afisat
                  int n, w, h; // h, v dimensiunile unui dreptunghi
                  int sem;
         System.Drawing.Graphics Desen;
         System.Drawing.SolidBrush Pensula;
         System.Drawing.Pen Creion_blu ;
      private void Form1_Load(object sender, EventArgs e)
      {
         Desen = this.CreateGraphics();
                  Pensula=new System.Drawing.SolidBrush(Color.Blue);
                  Creion_blu=new System.Drawing.Pen(Color.Blue);
                  w=this.Size.Width/50;
                  h=this.Size.Height/20;
         n=0x5555; // numarul ce va fi deplasat stanga dreapta si afisat
```

```
        sem=0;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        Desen.Clear(this.BackColor);
            int x=this.Size.Width-6*w;
            if (sem==0){
                    n=n << 1;
                    sem=1;
            }else{
                    n=n >> 1;
                sem=0;
            }
            for (i=15; i >= 0; i--){
               int bit=((n >> (15-i)) & 1);
               if (bit==1)
                        Desen.FillRectangle(Pensula, x, this.Size.Height/3, w,h);
               else
                        Desen.DrawRectangle(Creion_blu, x, this.Size.Height/3, w,h);
               x-=3*w;
            }
    }
  }
}
```

Rulam aplicatia, obtinem:



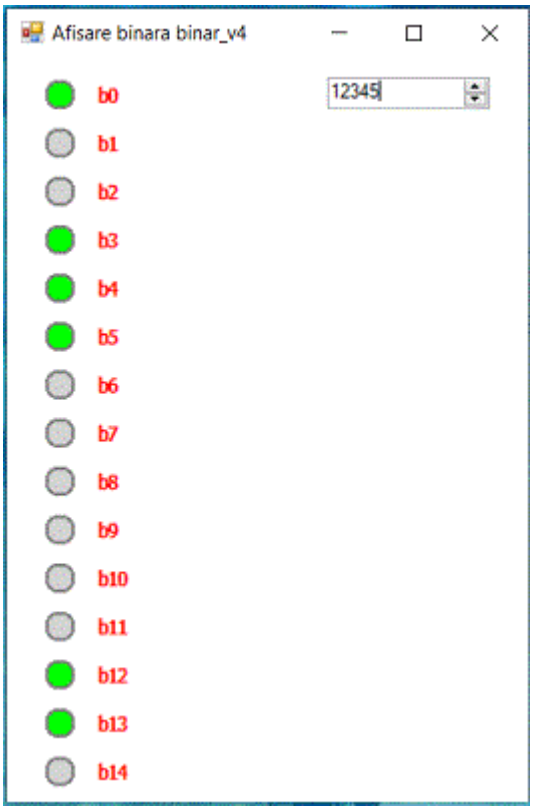
- **Conversia zecimal - binara**

Sa realizm aum o functie care afiseaza binar un numar intreg : conversie zecimal - binara

```
namespace binar_v4
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
```

```
    }
    System.Drawing.Graphics desen;
    System.Drawing.Pen creion;
    System.Drawing.SolidBrush pens_verde;
    System.Drawing.SolidBrush pens_rosie;
    System.Drawing.SolidBrush pens_gri;
    System.Drawing.Font font_nina;
    int x0, y0, w;
    string[] nume_b = new string[16] {"b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7",
                                      "b8", "b9", "b10","b11", "b12", "b13", "b14","b15"};
    private void afis_bin(int px0,int py0,int bw,int nrb, UInt64 n)
    {
      int x = px0 + bw;
      int y = py0 + bw;
      int i;
      for (i = nrb - 1; i >= 0; i--)
      {
        desen.DrawEllipse(creion, x - 1, y - 1, bw + 2, bw + 2);
        desen.DrawString(nume_b[nrb-i-1], font_nina, pens_rosie, x+2*bw, y);
        System.UInt64 bit = ((n >> (nrb - i - 1)) & 1);
        if (bit == 1)
          desen.FillEllipse(pens_verde, x, y, bw, bw);
        else
          desen.FillEllipse(pens_gri, x, y, bw, bw);
        y += 2 * bw;

      }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      creion = new System.Drawing.Pen(System.Drawing.Color.Gray,2);
      pens_verde = new System.Drawing.SolidBrush(System.Drawing.Color.Lime);
      pens_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
      pens_gri = new System.Drawing.SolidBrush(System.Drawing.Color.LightGray);
      font_nina = new System.Drawing.Font("Nina", 10);
      x0 = 10;
      y0 = 0;
      w = 15;
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      afis_bin(x0, y0, w, 16, 0);
    }
    private void numericUpDown1_ValueChanged(object sender, EventArgs e)
    {
      UInt64 val = System.Convert.ToUInt64(this.numericUpDown1.Value);
      afis_bin(x0, y0, w, 16, val);
    }
  }
}
```

Rulam aplicatia, obtinem:



- **Conversia binar zecimala**

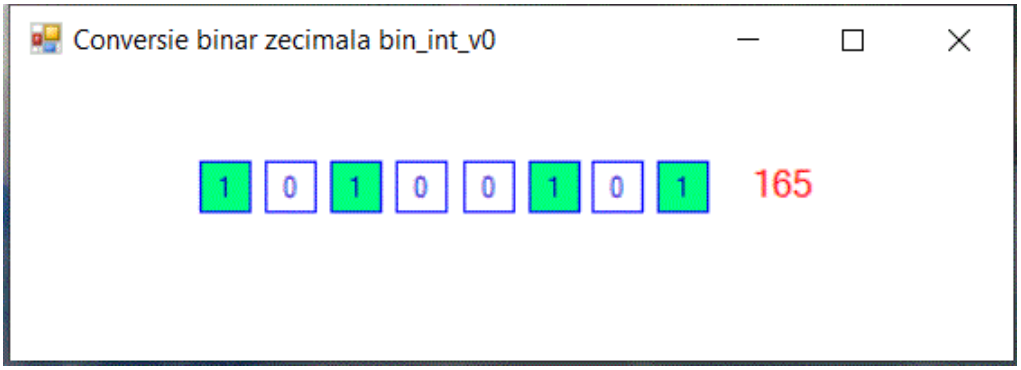
Sa realizm aum o functie care afiseaza binar un numar intreg : conversie binar - zecimala
Vom introduce o valoare binara pe care aplicatia o converteste intr-o valoare zecimala de tip int.
Pentru a introduce valorile binare vom folosi elemente de tip checkBox carora le setam corespunzator proprietatile:
FlatApparence - stabilim culorile CheckedBack
Apparence - Buton
FlatStyle - Flat
Text - Cu textul "0"
ForeColor - cu culoarea dorita

```
namespace bin_int_v0
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      int nr;
      private void afis_int()
      {
         nr=0;
         if (this.checkBox1.Text == "1")
```

```
        nr = nr + 1;
    if (this.checkBox2.Text == "1")
        nr = nr + 2;
    if (this.checkBox3.Text == "1")
        nr = nr + 4;
    if (this.checkBox4.Text == "1")
        nr = nr + 8;
    if (this.checkBox5.Text == "1")
        nr = nr + 16;
    if (this.checkBox6.Text == "1")
        nr = nr + 32;
    if (this.checkBox7.Text == "1")
        nr = nr + 64;
    if (this.checkBox8.Text == "1")
        nr = nr + 128;
    this.label1.Text = nr.ToString();
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (this.checkBox1.Text == "0")
        this.checkBox1.Text = "1";
    else
        this.checkBox1.Text = "0";
    afis_int();
}
private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
    if(this.checkBox2.Text == "0")
        this.checkBox2.Text = "1";
    else
        this.checkBox2.Text = "0";
    afis_int();
}
private void checkBox3_CheckedChanged(object sender, EventArgs e)
{
    if (this.checkBox3.Text == "0")
        this.checkBox3.Text = "1";
    else
        this.checkBox3.Text = "0";
    afis_int();
}
private void checkBox4_CheckedChanged(object sender, EventArgs e)
{
    if (this.checkBox4.Text == "0")
        this.checkBox4.Text = "1";
    else
        this.checkBox4.Text = "0";
    afis_int();
}
private void checkBox5_CheckedChanged(object sender, EventArgs e)
{
```

```
        if (this.checkBox5.Text == "0")
            this.checkBox5.Text = "1";
        else
            this.checkBox5.Text = "0";
        afis_int();
    }
    private void checkBox6_CheckedChanged(object sender, EventArgs e)
    {
        if (this.checkBox6.Text == "0")
            this.checkBox6.Text = "1";
        else
            this.checkBox6.Text = "0";
        afis_int();
    }
    private void checkBox7_CheckedChanged(object sender, EventArgs e)
    {
        if (this.checkBox7.Text == "0")
            this.checkBox7.Text = "1";
        else
            this.checkBox7.Text = "0";
        afis_int();
    }
    private void checkBox8_CheckedChanged(object sender, EventArgs e)
    {
        if (this.checkBox8.Text == "0")
            this.checkBox8.Text = "1";
        else
            this.checkBox8.Text = "0";
        afis_int();
    }
  }
}
```

Rulam aplicatia, obtinem:



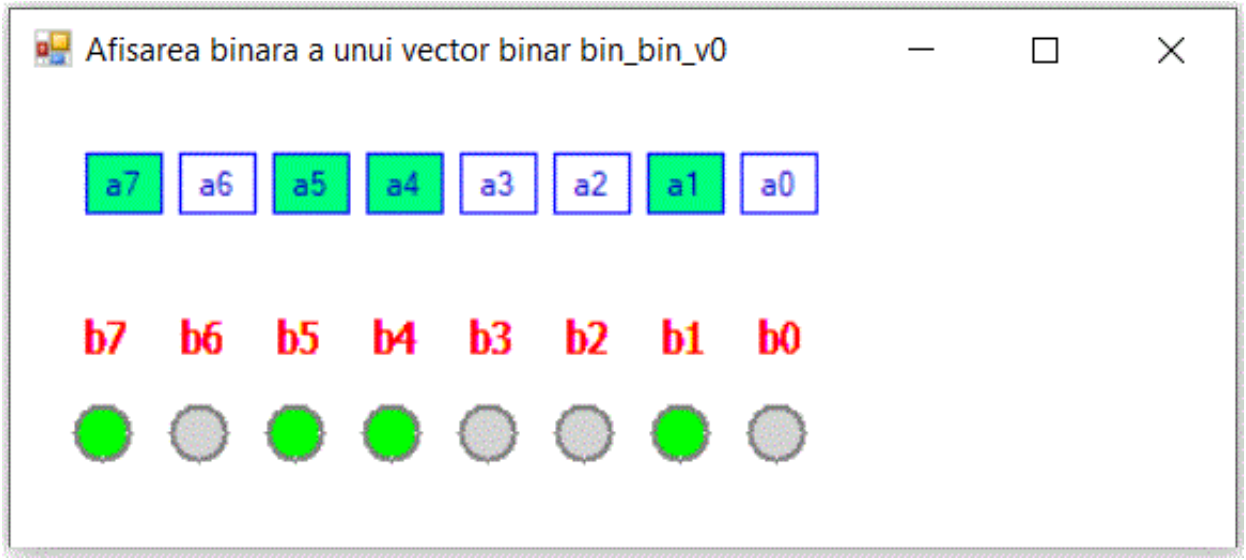
- **Afisarea binara a unui vector binar**

In cazul in care avem informatii binare pastrate intr-un vector binar, vom realiza o aplicatie care afseaza grafic continutul vectorului.

```
namespace bin_bin_v0
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics desen;
      System.Drawing.Pen creion;
      System.Drawing.SolidBrush pens_verde;
      System.Drawing.SolidBrush pens_rosie;
      System.Drawing.SolidBrush pens_gri;
      System.Drawing.Font font_nina;
      int x0, y0, w;
      const int nr_b = 8;
      string[] nume_b = new string[] {"b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"};
      bool[] val_bin = new bool[nr_b];
      private void afis_bin(int px0, int py0, int bw, int nrb, bool[] b)
      {
         int x = px0 + bw;
         int y = py0 + bw;
         int i;
         for (i = nrb - 1; i >= 0; i--)
         {
            desen.DrawEllipse(creion, x - 1, y - 1, bw + 2, bw + 2);
            desen.DrawString(nume_b[i].ToString(), font_nina, pens_rosie, x, y-2*bw);
            if (b[i])
               desen.FillEllipse(pens_verde, x, y, bw, bw);
            else
               desen.FillEllipse(pens_gri, x, y, bw, bw);
            x += 2 * bw;

         }
      }
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion = new System.Drawing.Pen(System.Drawing.Color.Gray, 2);
         pens_verde = new System.Drawing.SolidBrush(System.Drawing.Color.Lime);
         pens_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
         pens_gri = new System.Drawing.SolidBrush(System.Drawing.Color.LightGray);
         font_nina = new System.Drawing.Font("Nina", 12);
         x0 = 7;
```

```
        y0 = 100;
        w = 18;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        if (this.checkBox1.Checked)
            val_bin[0]=true;
        else
            val_bin[0] = false;
        if (this.checkBox2.Checked)
            val_bin[1] = true;
        else
            val_bin[1] = false;
        if (this.checkBox3.Checked)
            val_bin[2] = true;
        else
            val_bin[2] = false;
        if (this.checkBox4.Checked)
            val_bin[3] = true;
        else
            val_bin[3] = false;
        if (this.checkBox5.Checked)
            val_bin[4] = true;
        else
            val_bin[4] = false;
        if (this.checkBox6.Checked)
            val_bin[5] = true;
        else
            val_bin[5] = false;
        if (this.checkBox7.Checked)
            val_bin[6] = true;
        else
            val_bin[6] = false;
        if (this.checkBox8.Checked)
            val_bin[7] = true;
        else
            val_bin[7] = false;
        afis_bin(x0, y0, w, nr_b, val_bin);
    }
  }
}
```

Rulam aplicatia, obtinem:



- **Functii logice**

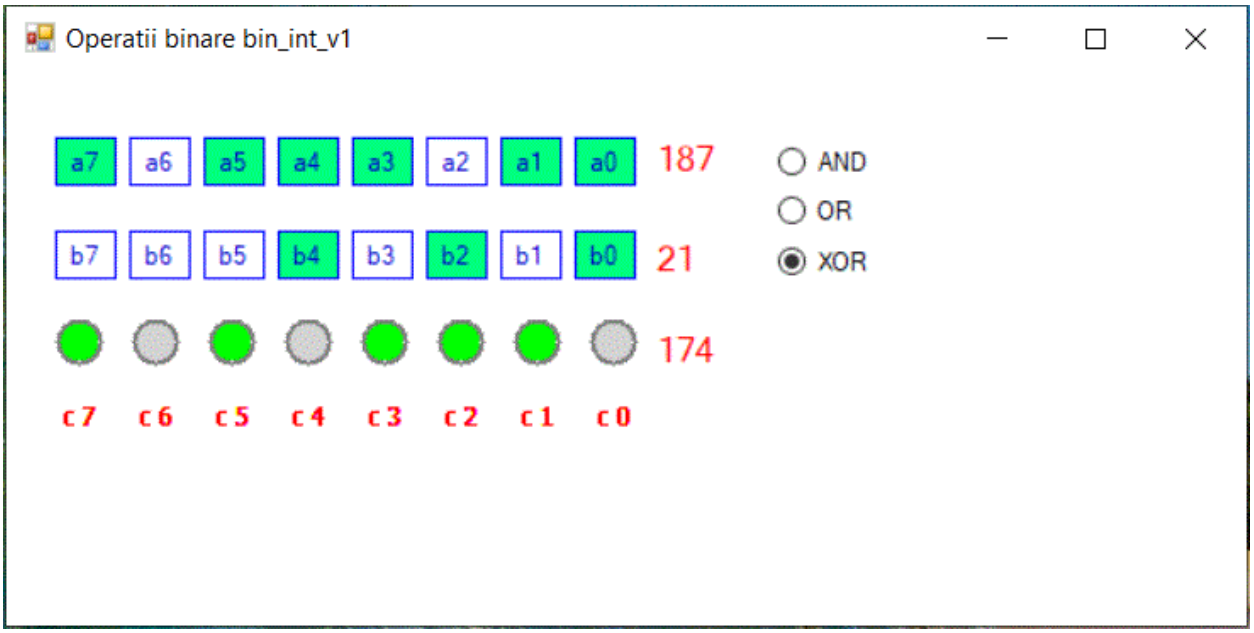
Sa utilizam cateva functii logice intr-o aplicatie grafica in spatiul System

```
namespace bin_int_v1
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      System.Drawing.Graphics desen;
      System.Drawing.Pen creion;
      System.Drawing.SolidBrush pens_verde;
      System.Drawing.SolidBrush pens_rosie;
      System.Drawing.SolidBrush pens_gri;
      System.Drawing.Font font_nina;
      int x0, y0, w;
      int nr_a = 0, nr_b = 0, nr_c = 0;
      UInt64 nr;
      string[] nume_b = new string[] {"b0", "b1", "b2",
                                                                  "b3", "b4", "b5", "b6", "b7"};
      private void afis_bin(int px0, int py0, int bw, int nrb, UInt64 n)
      {
         int x = px0 + bw;
         int y = py0 + bw;
         int i;
```

```
        for (i = nrb - 1; i >= 0; i--)
        {
            desen.DrawEllipse(creion, x - 1, y - 1, bw + 2, bw + 2);
            desen.DrawString(nume_b[i], font_nina, pens_rosie, x , y + 2 * bw);
            System.UInt64 bit = ((n >> (i)) & 1);
            if (bit == 1)
                desen.FillEllipse(pens_verde, x, y, bw, bw);
            else
                desen.FillEllipse(pens_gri, x, y, bw, bw);
            x += 2 * bw;

        }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        desen = this.CreateGraphics();
        creion = new System.Drawing.Pen(System.Drawing.Color.Gray, 2);
        pens_verde = new System.Drawing.SolidBrush(System.Drawing.Color.Lime);
        pens_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
        pens_gri = new System.Drawing.SolidBrush(System.Drawing.Color.LightGray);
        font_nina = new System.Drawing.Font("Nina", 10);
        x0 = 10;
        y0 = 100;
        w = 18;
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        nr_a=0;
        if (this.checkBox1.Checked)
            nr_a = nr_a + 1;
        if (this.checkBox2.Checked)
            nr_a = nr_a + 2;
        if (this.checkBox3.Checked)
            nr_a = nr_a + 4;
        if (this.checkBox4.Checked)
            nr_a = nr_a + 8;
        if (this.checkBox5.Checked)
            nr_a = nr_a + 16;
        if (this.checkBox6.Checked)
            nr_a = nr_a + 32;
        if (this.checkBox7.Checked)
            nr_a = nr_a + 64;
        if (this.checkBox8.Checked)
            nr_a = nr_a + 128;
        this.label1.Text = nr_a.ToString();

        nr_b = 0;
        if (this.checkBox9.Checked)
            nr_b = nr_b + 1;
        if (this.checkBox10.Checked)
            nr_b = nr_b + 2;
        if (this.checkBox11.Checked)
```

```
            nr_b = nr_b + 4;
        if (this.checkBox12.Checked)
            nr_b = nr_b + 8;
        if (this.checkBox13.Checked)
            nr_b = nr_b + 16;
        if (this.checkBox14.Checked)
            nr_b = nr_b + 32;
        if (this.checkBox15.Checked)
            nr_b = nr_b + 64;
        if (this.checkBox16.Checked)
            nr_b = nr_b + 128;
        this.label2.Text = nr_b.ToString();

                                    nr_c = 0;
        if(this.radioButton1.Checked)
            nr_c = nr_a & nr_b;
        if (this.radioButton2.Checked)
            nr_c = nr_a | nr_b;
        if (this.radioButton3.Checked)
            nr_c = nr_a ^ nr_b;
        this.label3.Text = nr_c.ToString();
        nr = Convert.ToUInt64(nr_c);
        afis_bin(x0, y0, w, 8, nr);
      }
    }
}
```

Rulam aplicatia, obtinem:



- **Aplicatie care foloseste valori binare**

Utilizam elementele anterioare pentru a schita o aplicatie care comanda o masina de spalat.
Vom plasa 5 elemente de tip checkBox pentru a simula intrarile configurate ca in aplicatia anterioara
Vom plasa 3 elemente de tip Label
Afisarea comenzilor se va face folosind o functie de tipul "afis_bin" care afiseaza valori binare sub forma de led-uri.
Valoarea binara ce urmeaza a fi afisata va fi pastrata intr-un vector de tip binar in care se vor memora starile comenzilor.
Pentru a lansa orice comanda, e suficient sa se seteze valoarea binara corespunzatoare din vectorul de tip bool[]: "leduri"
Pentru afisare se apeleaza procedura afis_bin(int px0, int py0, int bw, int nrb, bool[] n) care are ca parametri pozitia la care se afiseaza (px0, py0), latimea bw, nr de biti afisati nrb si vectorul n de tip bool[] in care se pastreaza valoarea binara a bitilor de afisat.

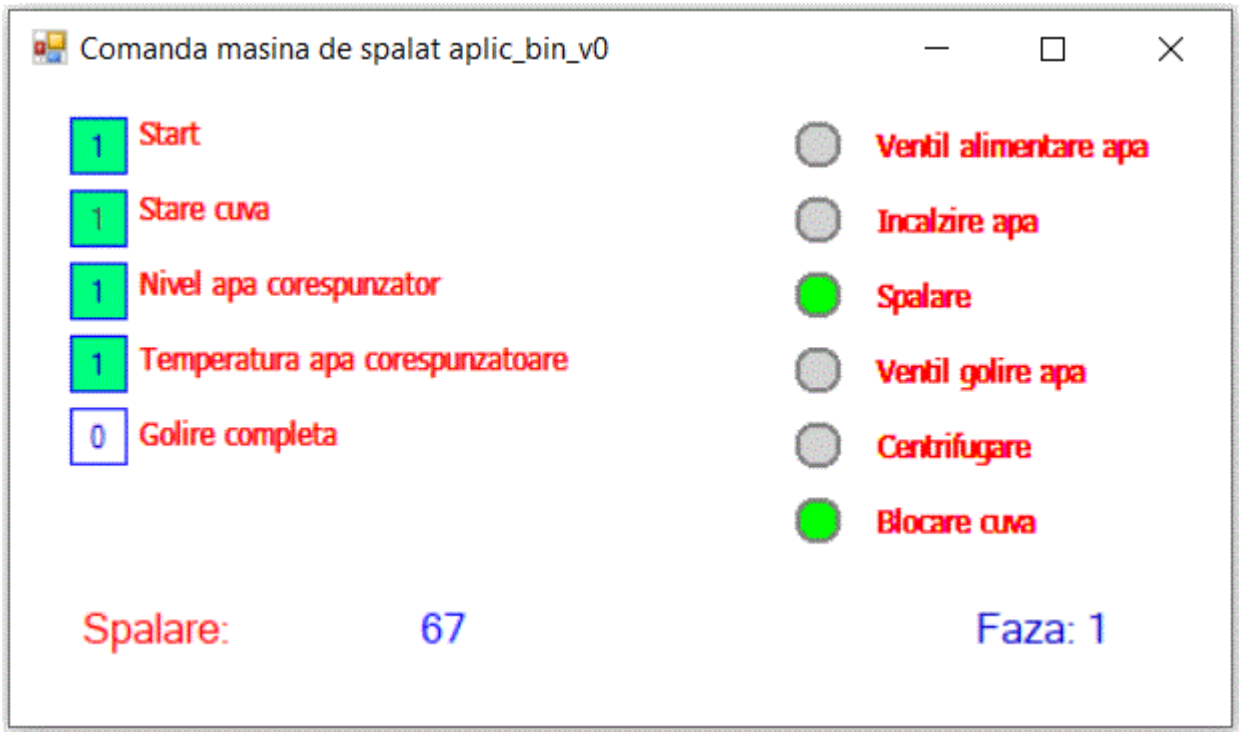
```
namespace aplic_bin_v0
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics desen;
      System.Drawing.Pen creion;
      System.Drawing.SolidBrush pens_verde;
      System.Drawing.SolidBrush pens_rosie;
      System.Drawing.SolidBrush pens_gri;
      System.Drawing.Font font_nina;
      int i, x0, y0, w, contor_sp = 0, faza = 0;
      const int nr_in = 5; //numarul intrari (butoane, contacte)
      // numele intrarilor
      string[] nume_in = new string[] {"Start", "Stare cuva", "Nivel apa corespunzator",
                              "Temperatura apa corespunzatoare", "Golire completa"};

      const int nr_out = 6; //Numarul de iesiri (actionari)
      //UInt64 val;
      // numele Iesirilor
      string[] nume_out = new string[] {"Ventil alimentare apa", "Incalzire apa", "Spalare", "Ventil golire apa",
"Centrifugare", "Blocare cuva"};

      // Afisarea iesirilor sub forma de led-uri
      bool[] leduri = new bool[nr_out];
      private void afis_bin(int px0, int py0, int bw, int nrb, bool[] n)
      {
         int x = px0 + bw;
         int y = py0 + bw;
         int i;
         for (i = 0; i < nrb; i++)
         {
```

```
            desen.DrawEllipse(creion, x - 1, y - 1, bw + 2, bw + 2);
            desen.DrawString(nume_out[i], font_nina, pens_rosie, x + 2 * bw, y);
            if (n[i] == true)
                desen.FillEllipse(pens_verde, x, y, bw, bw);
            else
                desen.FillEllipse(pens_gri, x, y, bw, bw);
            y += 2 * bw;
        }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        desen = this.CreateGraphics();
        creion = new System.Drawing.Pen(System.Drawing.Color.Gray, 2);
        pens_verde = new System.Drawing.SolidBrush(System.Drawing.Color.Lime);
        pens_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
        pens_gri = new System.Drawing.SolidBrush(System.Drawing.Color.LightGray);
        font_nina = new System.Drawing.Font("Nina", 10);
        x0 = 300;
        y0 = 0;
        w = 15;
        for (i = nr_out - 1; i >= 0; i--)
        {
            leduri[i] = false;
        }
        this.checkBox1.Checked = false;
        this.checkBox1.Text = "0";
        this.checkBox2.Checked = false;
        this.checkBox2.Text = "0";
        this.checkBox3.Checked = false;
        this.checkBox3.Text = "0";
        this.checkBox4.Checked = false;
        this.checkBox4.Text = "0";
        this.checkBox5.Checked = false;
        this.checkBox5.Text = "0";
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        i = 0;
        int y = 10;
        for (i = 0; i < nr_in; i++)
        {
            desen.DrawString(nume_in[i], font_nina, pens_rosie, 50, y);
            y += 30;
        }
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        afis_bin(x0, y0, w, nr_out, leduri);
        if (this.checkBox1.Checked) // Start
        {
```

```
            this.checkBox1.Text = "1";
            if (this.checkBox2.Checked)// Cuva
            {
               this.checkBox2.Text = "1";
               leduri[5] = true;
               this.checkBox2.Enabled = false;
               if (faza == 0)
               {
                  this.label1.Text = "Masina porinita";
                  leduri[0] = true;// Ventil apa
               }
               if (this.checkBox3.Checked) // Nivel apa
               {
                  this.checkBox3.Text = "1";
                  leduri[0] = false;// Ventil apa
                  leduri[1] = true;// Incalzire apa
                  if (this.checkBox4.Checked) // Temperatura corespunzatoare
                  {
                     this.checkBox4.Text = "1";
                     leduri[1] = false;// Incalzire apa
                     if (faza == 0)
                     {
                        this.label1.Text = "Spalare:";
                        leduri[2] = true;// Spalare
                        contor_sp = 100;
                        faza = 1;
                     }
                  }
               }
            }
            else
            {
               this.label1.Text = "Cuva deschisa";
            }
         }
         else
         {
            this.label1.Text = "Masina oprita";
            leduri[0] = false;
            leduri[1] = false;
            leduri[2] = false;
            leduri[3] = false;
            leduri[4] = false;
         }

         if ((contor_sp == 0) && (faza == 1))
         {
            leduri[2] = false;// Spalare
            this.label1.Text = "Golire apa";
            leduri[3] = true;// Golire apa
            faza = 2;
         }
```

```
        if (faza == 2)
        {
           if (this.checkBox5.Checked) // Golire completa
           {
              this.checkBox5.Text = "1";
              this.label1.Text = "Centrifugare:";
              leduri[3] = false;// Golire apa
              leduri[4] = true;// Centrifugare
              contor_sp = 50;
              faza = 3;
           }
        }

        if ((contor_sp == 0) && (faza == 3))
        {
           faza = 0;
           leduri[0] = false;
           leduri[1] = false;
           leduri[2] = false;
           leduri[3] = false;
           leduri[4] = false;
           leduri[5] = false;
           this.checkBox2.Enabled = true;
           this.checkBox1.Checked = false;
           this.checkBox1.Text = "0";
           this.checkBox2.Checked = false;
           this.checkBox2.Text = "0";
           this.checkBox3.Checked = false;
           this.checkBox3.Text = "0";
           this.checkBox4.Checked = false;
           this.checkBox4.Text = "0";
           this.checkBox5.Checked = false;
           this.checkBox5.Text = "0";
           this.label1.Text = "Gata spalare";
        }
        if (contor_sp > 0)
           contor_sp--;
        this.label2.Text = contor_sp.ToString();
        this.label3.Text = "Faza: "+faza.ToString();
     }
           // gata procedura timer1_Tick
   }
}
```

Rulam aplicatia, obtinem:



Elemente de programare orientata obiect

Concepte de baza ale programarii orientata pe obiecte

S-au utilizat pe parcursul capitolelor anterioare diverse de tipuri de date. Pentru a utiliza o anumita variabila, in prealabil a trebuit precizat tipul variabilei, dupa care se putea trece la initializarea si utilizarea ei. Astfel pentru a utiliza o variabila de tip intreg i, am declarat **int i;** In momentul precizarii tipului variabilei (int), s-a precizat un concept general in care se incadreaza variabila cu alte cuvinte s-a precizat clasa de care apartine variabila. Clasele implementeaza tipuri de date. Am definit si am utilizat apoi variabile mai complexe prin intermediul structurilor de date. Structurile de date ne-au permis sa definim entitati complexe numite obiecte. Obiectele sunt componente software care modeleaza fenomene din lumea reala. Obiectele care reprezinta aceeasi idee sau concept sunt de acelasi tip si pot fi grupate în clase. Obiectele inglobeaza datele cat si procedurile care actioneaza asupra datelor. Dupa declararea unei variabile, se poate face initializarea ei.

Astfel dupa declaratia **int i;** se poate face initializarea variabilei i astfel: **i=0;** . Identic se procedeaza si cu obiectele. Pentru a crea un obiect, se declarara clasa, apoi se face instantierea clasei. Un obiect este o instanta a unei clase.

- **Definirea conceptelor**

In capitolele anterioare au mai fost prezentate notiunile si conceptele utilizate in POO, acum insa suntem in masura sa definim complet conceptele utilizate in OOP.

Obiect

Obiectul reprezinta un ansamblu de date si functii. Datele se numesc "date membru" sau proprietati. Functiile se

numesc "functii membru" si au rolul de a actiona asupra datelor inglobate in obiecte. Functiile membru sunt compuse din metode si evenimente. Proprietatile, metodele si evenimentele sunt membrii unui obiect. Interfata de acces la obiect este realizata numai prin intermediul metodelor.

Proprietate

Proprietatea reprezinta un atribut al unui obiect care defineste una dintre caracteristicile sale.

Metoda

Metoda reprezinta actiunea pe care o poate executa un obiect. Utilizatorul unui obiect are acces la date numai prin intermediul metodelor obiectului, iar metoda are acces implicit la membrii unui obiect.

Eveniment

Evenimentul este o un membru al unei clase ce permite clasei sau obiectelor clasei sa faca notificari celorlalte obiecte asupra unor schimbari petrecute la nivelul starii. Evenimentului i se asociaza o referinta catre o functie necunoscuta careia i se precizeaza doar antetul, continutul functiei urmand a fi completat cu un program de raspuns la acel eveniment. Evenimentul reprezinta deci o actiune recunoscuta de alte obiecte. Tratarea evenimentului se materializeaza prin scrierea un program ca raspuns la evenimentul generat. Evenimentele pot fi externe (generate de actiuni ale utilizatorului) sau interne (generate printr-un cod de program sau de sistem).

Clasa

O clasa reprezinta definitia unui anumit tip de obiect. In cadrul unei clase sunt definite proprietatile si metodele obiectului. Folosirea claselor permite gestionarea mai multor obiecte de acelasi tip. Clasa este doar un termen abstract, care defineste caracteristicile unui obiect.

Programare orientata pe obiecte OOP

Programarea orientata pe obiecte OOP ("Object Oriented Programming") este programararea care utilizeaza obiecte. OOP permite modelarea obiectelor, proprietatilor si a relatiilor dintre ele. OOP ofera posibilitatea descompunerii programului în componente deschizand calea reutilizarii codului.

- **Principiile OOP**

Incapsularea – contopirea datelor cu codul
Mostenirea - posibilitatea de a extinde o clasa prin adaugarea de noi functionalitati
Polimorfismul – într-o ierarhie de clase obtinuta prin mostenire, o metoda poate avea implementari diferite la nivele diferite in acea ierarhie;

Aplicatii OOP

Aplicatiile OOP reusesc sa atinga un nivel ridicat de complexitate prin reutilizarea codului scris anterior. Odata definite clasele, acestea pot fi reutilizate in diverse aplicatii fara nici un fel de modificare sau adaptare. Problema cea mai complicata este definirea claselor astfel incat ele sa fie utile in diverse aplicatii.

- **Definirea claselor**

In capitolul referitor la "Clase" am creat o clasa numita "ap_electric" . Vom crea o aplicatie "Windows Forms Application" in care se va defini clasa "ap_electric". Aplicatia va clacula puterea consumata de un aparat electric prin instantierea clasei "ap_electric" si invocarea metodei vezi_p().

Generam un nou proiect de tipul Windows Forms Application numit **"oop_v0"** .
Plasam doua obiecte de tip NumericUpDown, numite NumericUpDown1 respectiv NumericUpDown2, necesare pentru introducerea valorilor tensiunii si curentului aparatului electric.
Plasam un obiect de tip button numit button1 apoi schimbam proprietatea "Text" in "Calculeaza P=U*I".
Plasam un obiect de tip label numit label1 pentru afisarea puterii nominale a aparatului electric.

Dupa cum am precizat, partea cea mai dificila o reprezinta realizarea claselor. Cu cat clasa este mai generala, cu atat ea va putea fi reutilizata de mai multe ori in diverse aplicatii. Vom realiza deci clasa "ap_electric" de forma:

```
class ap_electric {
	private:
		double u_n;
		double i_n;
	public:
		void set_u(double);
		int vezi_u() const;
		void set_i(double);
		double vezi_i() const;
		double vezi_p() const;
};
```

Vom plasa codul ce reprezinta clasa ap_electric in sectiunea "Header Files" in fisierul "stdafx.h".
Fisierul "stdafx.h" va arata astfel :

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
#pragma once

// TODO: reference additional headers your program requires here
class ap_electric {
	private:
		double u_n;
		double i_n;
	public:
		void set_u(double);
		int vezi_u() const;
		void set_i(double);
		double vezi_i() const;
		double vezi_p() const;
};
```

- **Realizarea functiilor membru**

Dupa cum se observa, in cadrul claselor s-a definit numai prototipul functiilor membru. Functiile membru vor fi definite in cadrul fisierului "stdafx.cpp". Continutul acestui fisier se poate vedea mai jos:

```
#include "stdafx.h"
void ap_electric::set_u(double tens) {
	if (tens < 0)
		u_n=0;
	else
```

```
                u_n=tens;
}
int ap_electric::vezi_u() const {
        return u_n;
}
void ap_electric::set_i(double crnt){
        if (crnt < 0)
                i_n=0;
        else
                i_n=crnt;
}
double ap_electric::vezi_i() const{
        return i_n;
}
double ap_electric::vezi_p() const{
        return i_n*u_n;
}
```

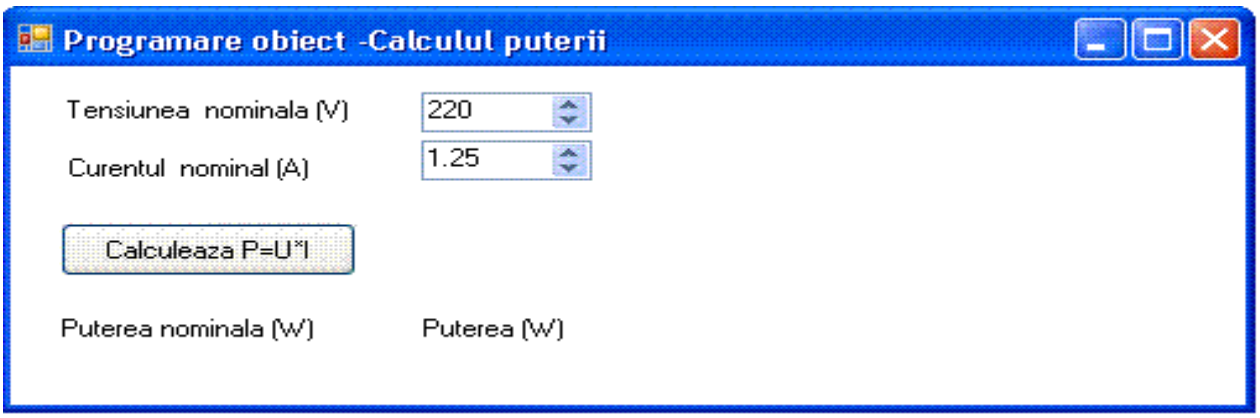
- **Crearea obiectelor**

Dupa ce clasa ap_electric a fost definita, prin instantierea acestei clase se obtin obiecte. Pentru a crea obiectul a vom introduce instructiunea: **ap_electric a;** Aceasta instructiune creaza obiectul "a" din clasa "ap_electric"

Pe evenimentul click al butonului button1 vom plasa:

```
        ap_electric a;
        a.set_u(System::Convert::ToDouble(this->numericUpDown1->Value));
        a.set_i(System::Convert::ToDouble(this->numericUpDown2->Value));
        this->label1->Text=System::Convert::ToString(a.vezi_p());
```

Rulam aplicatia, setam valorile pentru tensiune si curent , apasam butonul "Calculeaza P=U*I" si obtinem:



In procedura de sus s-a creat obiectul "a" prin instantierea clasei "ap_electric". Valoarea curentului si a tensiunii au fost setate prin invocarea metodei "set_u" respectiv "set_i". Valoarea puterii nominale a fost calculata prin invocarea metodei "vezi_p".

Folosind metoda de sus obiectul "a" este creat in memoria statica. Pentru a crea un obiect in memoria dinamica

va trebui sa declaram "a" de tip pointer adica **ap_electric* a;** iar pentru crearea obiectului "a" sa folosim peratorul **new** operator, care face alocarea memoriei in mod dinamic. Vezi aplicatia**"oop_v01"**.

Daca aparatul electric a fost definit sub forma: **ap_electric* a;** , procedura plasata pe evenimentul click al butonului button1 va deveni:

```
ap_electric* a;
a=new ap_electric;
a->set_u(System::Convert::ToDouble(this->numericUpDown1->Value));
a->set_i(System::Convert::ToDouble(this->numericUpDown2->Value));
this->label1->Text=System::Convert::ToString(a->vezi_p());
delete a;
```

Dupa cum se observa "a" este un pointer si in acest caz nu mai folosim operatorul "." ci operatoriul "->".

Se observa de asemenea ca "a" este un pointer negestionat (a fost definit cu ap_electri* a). Pentru declararea lui "a" s-a utilizat operatorul "*". Dupa ce nu mai este necesar obiectul "a" el trebuie sters cu **delete a**

Daca dorim sa utilizam un pointer gestionat (sa folosim operatorul ^) pentru a nu mai fi nevoie sa stergem obiectul creat dupa ce nu mai este util va trebui sa definim mai intai clasa "ap_electric" de tip gestionat si abia apoi sa creem un obiect "a".

Clasa "ap_electric" va putea fi definita de tip "gestionat" prin declararea ei de tip "ref" (reference) astfel:

```
ref class ap_electric {
        private:
                double u_n;
                double i_n;
        public:
                void set_u(double);
                int vezi_u();
                void set_i(double);
                double vezi_i();
                double vezi_p();
};
```

Utilizand clasa gestionata **ref class ap_electric**, "a" va fi un pointer gestionat de tipul: **ap_electric^ a;**. Crearea obiectului "a" se va face cu **gcnew** si nu va mai trebui sters dupa ce nu mai este util. Vezi aplicatia **"oop_v02"**.

Procedura plasata pe evenimentul click al butonului button1 va deveni:

```
ap_electric^ a;
a=gcnew ap_electric;
a->set_u(System::Convert::ToDouble(this->numericUpDown1->Value));
a->set_i(System::Convert::ToDouble(this->numericUpDown2->Value));
this->label1->Text=System::Convert::ToString(a->vezi_p());
```

C#

Aplicatia oop_v0

```
namespace oop_v0
{
  public partial class Form1 : Form
```

```
  {
    public Form1()
    {
      InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
                    ap_electric a = new ap_electric();
                    a.set_u(System.Convert.ToDouble(this.numericUpDown1.Value));
                    a.set_i(System.Convert.ToDouble(this.numericUpDown2.Value));
                    this.label1.Text=System.Convert.ToString(a.vezi_p());
    }
  }
  public class ap_electric
  {
    private double u_n;
    private double i_n;
    public  void set_u(double tens)
    {
      if (tens < 0)
        u_n = 0;
      else
        u_n = tens;
    }

     public double vezi_u()
    {
      return u_n;
    }

     public void set_i(double crnt)
    {
      if (crnt < 0)
        i_n = 0;
      else
        i_n = crnt;
    }

     public double vezi_i()
    {
      return i_n;
    }


     public double vezi_p()
        {
      return i_n * u_n;
    }
  }
}
```

Obiecte grafice realizate in Windows Forms Application

La fel cum au fost create clase si obiecte in aplicatiile anterioare, vom realiza in continuare clase si obiecte grafice folosind Windows Forms Application. Aceste obiecte afiseaza elemente grafice pe form-ul deschis.

• Realizarea unui obiect grafic sub forma a doua dreptunghiuri concentrice "drcon"

Vom realiza in continuare o alicatie grafica in care este utilizat un obiect pe care il vom denumi "drc" obiect care este reprezentat grafic sub forma a doua dreptunghiuri concentrice, are numeroase metode printre care cea mai importanda este metoda "desenez" metoda cu care se afiseaza dreptunghiulexterior si metoda setdr cu care desenam un dreptunghi interior concentric de diverse dimensiuni.

Generam un nou proiect de tipul Windows Forms Application numit **"oop_00"**.

Metoda "setdr" va fi invocata la intervale regulate de timp mentru a putea afisa dreptunghiul interior de dimensiuni variabile. Vom plasa deci un obiect de tip timer numit timer1 si setam proprietatea interval la 10 ms si Enabled la "true".

Vom scrie in C# aplicatia care contine clasa "drcon".

C# Aplicatia **"oop_00"**

- obiectul: dreptunghiuri concentrice

```
namespace oop_00
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public drcon drc;
      System.Random nr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         drc = new drcon();
         drc.init_dr(100, 100, 100, 75);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         drc.desenez(desen, creion_albastru);
      }
```

```
    private void timer1_Tick(object sender, EventArgs e)
    {
      drc.sterg(desen, radiera);
      drc.setdr(desen, creion_rosu, nr.Next(100-3), nr.Next(75-3));
    }
  }
  public class drcon
  {
    int x0,y0,w0,h0,w,h;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawRectangle(creion_a, x0, y0, w0, h0);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w0 - 1, h0 - 1);
    }

    public void setdr(System.Drawing.Graphics zona_des, System.Drawing.Pen creion,int w,int h)
    {
      int x = System.Convert.ToInt16(System.Convert.ToDouble(x0) + System.Convert.ToDouble(w0-w) / 2);
      int y = System.Convert.ToInt16(System.Convert.ToDouble(y0) + System.Convert.ToDouble(h0-h) / 2);
      zona_des.DrawRectangle(creion, x+1, y+1, w, h);

    }
    public void init_dr(int pozx, int pozy, int lat, int inalt)
    {
      x0 = pozx;
      y0 = pozy;
      w0 = lat;
      h0 = inalt;
    }
  }
}
```

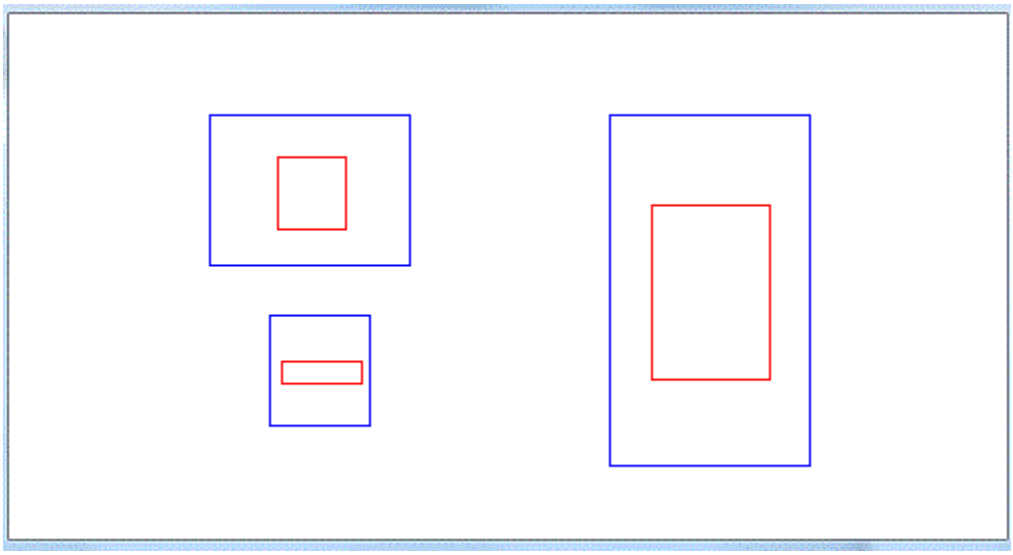
Putem simplifica clasa "drcon" prin utilizarea unui constructor. Inacest caz numai este nevoie de metoda "init_dr"aceasta fiind inlocuita de constructorul **public drcon(int pozx, int pozy, int lat, int inalt)**. In acest caz instantierea clase "drcon" se va face: **drc = new drcon(100, 100, 100, 75);**. Am renuntat de asemenea la metodele "desenez" si "sterg", acestea fiind preluate de metoda "setdr";

```
namespace oop_10
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public System.Drawing.Pen creion_albastru;
```

```
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public drcon drc;
      System.Random nr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         drc = new drcon(100, 100, 100, 75);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         drc.setdr(desen, creion_rosu, creion_albastru, radiera, nr.Next(100 - 3), nr.Next(75 - 3));
      }
      public class drcon
      {
      int x0, y0, w0, h0;
      public void setdr(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, System.Drawing.Pen
creion_a, System.Drawing.Brush rad, int w, int h)
      {
         zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w0 - 1, h0 - 1);
         zona_des.DrawRectangle(creion_a, x0, y0, w0, h0);
         int x = System.Convert.ToInt16(System.Convert.ToDouble(x0) + System.Convert.ToDouble(w0 - w) / 2);
         int y = System.Convert.ToInt16(System.Convert.ToDouble(y0) + System.Convert.ToDouble(h0 - h) / 2);
         zona_des.DrawRectangle(creion, x + 1, y + 1, w, h);
      }
      public drcon(int pozx, int pozy, int lat, int inalt)
      {
          x0 = pozx;
          y0 = pozy;
          w0 = lat;
          h0 = inalt;
      }
   }
 }
}
```

C# Aplicatia **"oop_01"**

- mai multe obiecte : dreptunghiuri concentrice



```
namespace oop_01
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public drcon drc1;
      public drcon drc2;
      public drcon drc3;
      System.Random nr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         drc1 = new drcon(100, 50, 100, 75);
         drc2 = new drcon(300, 50, 100, 175);
         drc3 = new drcon(130, 150, 50, 55);
      }
```

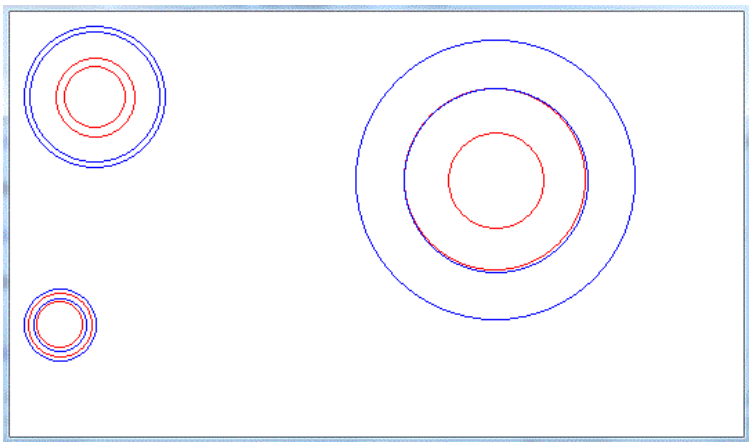
```
    private void timer1_Tick(object sender, EventArgs e)
    {
      drc1.setdr(desen, creion_rosu, creion_albastru, radiera, nr.Next(100 - 3), nr.Next(75 - 3));
      drc2.setdr(desen, creion_rosu, creion_albastru, radiera, nr.Next(100 - 3), nr.Next(175 - 3));
      drc3.setdr(desen, creion_rosu, creion_albastru, radiera, nr.Next(50 - 3), nr.Next(55 - 3));
    }
  }
    public class drcon
    {
      int x0, y0, w0, h0;

      public void setdr(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, System.Drawing.Pen
creion_a, System.Drawing.Brush rad, int w, int h)
      {
        zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w0 - 1, h0 - 1);
        zona_des.DrawRectangle(creion_a, x0, y0, w0, h0);
        int x = System.Convert.ToInt16(System.Convert.ToDouble(x0) + System.Convert.ToDouble(w0 - w) / 2);
        int y = System.Convert.ToInt16(System.Convert.ToDouble(y0) + System.Convert.ToDouble(h0 - h) / 2);
        zona_des.DrawRectangle(creion, x + 1, y + 1, w, h);

      }
      public drcon(int pozx, int pozy, int lat, int inalt)
      {
        x0 = pozx;
        y0 = pozy;
        w0 = lat;
        h0 = inalt;
      }
    }
  }
```

C# Aplicatia **"oop_02"**

- obiect : cercuri concentrice

```
namespace oop_02
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public System.Drawing.Pen creion_albastru;
    public System.Drawing.Pen creion_rosu;
    public System.Drawing.SolidBrush radiera;
    public ccon cc;
```

```
    System.Random nr;
    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
      creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
      radiera = new System.Drawing.SolidBrush(this.BackColor);
      nr = new System.Random();
      cc = new ccon();
      cc.init_c(100, 100, 100);
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      cc.desenez(desen, creion_albastru);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      cc.sterg(desen, radiera);
      cc.setc(desen, creion_rosu, nr.Next(100 - 3));
      cc.setc(desen, creion_albastru, nr.Next(100 - 3));
      cc.setc(desen, creion_rosu, nr.Next(100 - 3));
    }
  }
  public class ccon
  {
    int x0, y0, r0, r;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawEllipse(creion_a, x0, y0, r0+2, r0+2);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillEllipse(rad, x0 + 1, y0 + 1, r0 - 1, r0 - 1);
    }

    public void setc(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, int r)
    {
      int x = System.Convert.ToInt16(System.Convert.ToDouble(x0) + System.Convert.ToDouble(r0 - r) / 2);
      int y = System.Convert.ToInt16(System.Convert.ToDouble(y0) + System.Convert.ToDouble(r0 - r) / 2);
      zona_des.DrawEllipse(creion, x + 1, y + 1, r, r);

    }
    public void init_c(int pozx, int pozy, int raza)
    {
      x0 = pozx;
      y0 = pozy;
      r0 = raza;
    }
  }
}
```



C# Aplicatia **"oop_03"**

- mai multe obiecte : cercuri concentrice

```
namespace oop_03
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public ccon cc1;
      public ccon cc2;
      public ccon cc3;
      System.Random nr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         cc1 = new ccon();
         cc1.init_c(10, 10, 100);
         cc2 = new ccon();
         cc2.init_c(250, 20, 200);
         cc3 = new ccon();
         cc3.init_c(10, 200, 50);
      }
```

```
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      cc1.desenez(desen, creion_albastru);
      cc2.desenez(desen, creion_albastru);
      cc3.desenez(desen, creion_albastru);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      cc1.sterg(desen, radiera);
      cc1.setc(desen, creion_rosu, nr.Next(100 - 3));
      cc1.setc(desen, creion_albastru, nr.Next(100 - 3));
      cc1.setc(desen, creion_rosu, nr.Next(100 - 3));
      cc2.sterg(desen, radiera);
      cc2.setc(desen, creion_rosu, nr.Next(200 - 3));
      cc2.setc(desen, creion_albastru, nr.Next(200 - 3));
      cc2.setc(desen, creion_rosu, nr.Next(200 - 3));
      cc3.sterg(desen, radiera);
      cc3.setc(desen, creion_rosu, nr.Next(50 - 3));
      cc3.setc(desen, creion_albastru, nr.Next(50 - 3));
      cc3.setc(desen, creion_rosu, nr.Next(50 - 3));
    }
  }
  public class ccon
  {
    int x0, y0, r0, r;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawEllipse(creion_a, x0, y0, r0 + 2, r0 + 2);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillEllipse(rad, x0 + 1, y0 + 1, r0 - 1, r0 - 1);
    }

    public void setc(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, int r)
    {
      int x = System.Convert.ToInt16(System.Convert.ToDouble(x0) + System.Convert.ToDouble(r0 - r) / 2);
      int y = System.Convert.ToInt16(System.Convert.ToDouble(y0) + System.Convert.ToDouble(r0 - r) / 2);
      zona_des.DrawEllipse(creion, x + 1, y + 1, r, r);

    }
    public void init_c(int pozx, int pozy, int raza)
    {
      x0 = pozx;
      y0 = pozy;
      r0 = raza;
    }
  }
}
```

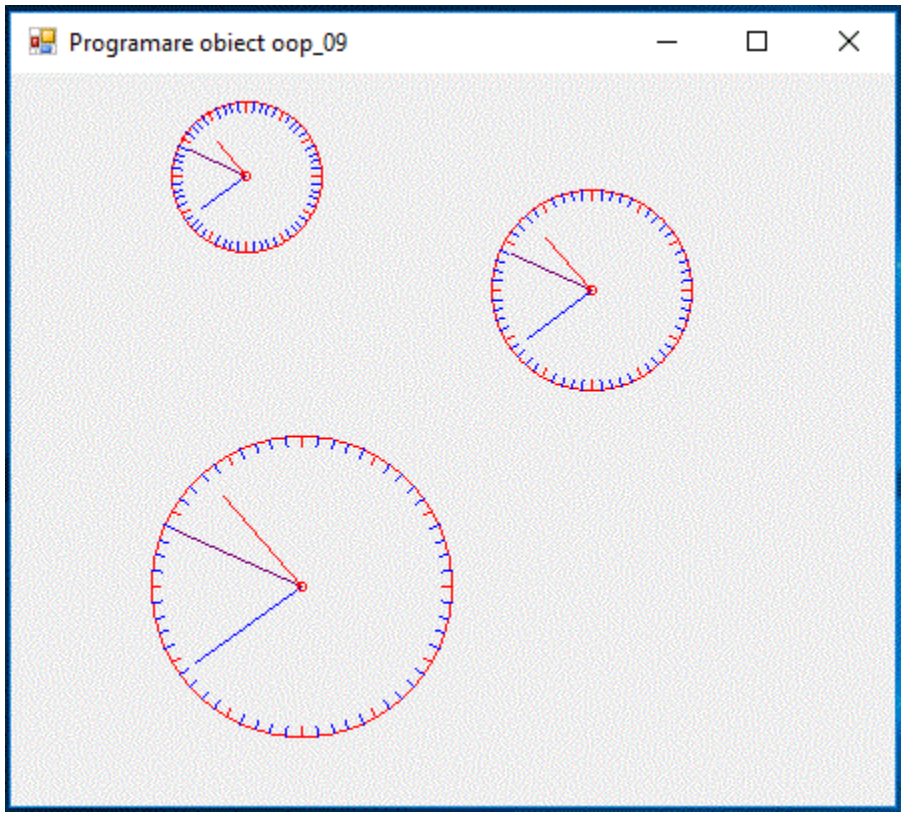
C# Aplicatia **"ceas"**

Vom raliza in continuare clasa numita ceas dupa care vom instantia clasa ceas si vom plasa mai multe obiecte "ceas":

```
namespace oop_09
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics Desen;
      System.Drawing.SolidBrush Pens_back;
      public ceas ceas1;
      public ceas ceas2;
      public ceas ceas3;
      public class ceas
      {
         int x0;
         int y0;
         int w;

         public void des_ceas(System.Drawing.Graphics zona_des,System.Drawing.SolidBrush radiera)
         {
            int x = x0+w / 5;
            int y = y0+ w / 5;
            int d = w / 2;
            int x1 = x + d / 2;
            int y1 = y + d / 2;
            double g;
            System.Drawing.Pen c_min = new System.Drawing.Pen(System.Drawing.Color.Blue);
            System.Drawing.Pen c_5min = new System.Drawing.Pen(System.Drawing.Color.Red);
            zona_des.DrawEllipse(c_5min, x, y, d, d);
            for (g = 0; g < 360; g += 6)
            {
               double r = 2 * System.Math.PI * g / 360;
               int x2 = Convert.ToInt16(x1 + (d / 2) * System.Math.Cos(r));
               int y2 = Convert.ToInt16(y1 - (d / 2) * System.Math.Sin(r));
               zona_des.DrawLine(c_min, x1, y1, x2, y2);
               if (g % 30 == 0)
                  zona_des.DrawLine(c_5min, x1, y1, x2, y2);
               else
                  zona_des.DrawLine(c_min, x1, y1, x2, y2);
            }
            zona_des.FillEllipse(radiera, x + 5, y + 5, d - 10, d - 10);
         }

         public void setval(System.Drawing.Graphics zona_des,System.Drawing.SolidBrush radiera)
```

```
        {
            System.Drawing.Pen c_sec = new System.Drawing.Pen(System.Drawing.Color.Purple);
            System.Drawing.Pen c_min = new System.Drawing.Pen(System.Drawing.Color.Blue);
            System.Drawing.Pen c_ore = new System.Drawing.Pen(System.Drawing.Color.Red);
            int x = x0+w / 5;
            int y = y0+w / 5;
            int d = w / 2;
            int x1 = x + d / 2;
            int y1 = y + d / 2;
            int i ,x3,y3;
            double r;
            //secundar
            i = 90 - 6 * System.DateTime.Now.Second;
            r = (2 * System.Math.PI * i) / 360;
            x3 = Convert.ToInt16(x1 + (d / 2 - 6) * System.Math.Cos(r));
            y3 = Convert.ToInt16(y1 - (d / 2 - 6) * System.Math.Sin(r));
            zona_des.FillEllipse(radiera, x + 5, y + 5, d - 10, d - 10);
            zona_des.DrawLine(c_sec, x1, y1, x3, y3);
            //minutar
            i = 90 - 6 * System.DateTime.Now.Minute;
            r = (2 * System.Math.PI * i) / 360;
            x3 = Convert.ToInt16(x1 + (d / 2 - 10) * System.Math.Cos(r));
            y3 = Convert.ToInt16(y1 - (d / 2 - 10) * System.Math.Sin(r));
            zona_des.DrawLine(c_min, x1, y1, x3, y3);
            //ora
            i = 90 - (30 * System.Convert.ToInt16(System.DateTime.Now.Hour) +
System.Convert.ToInt16(System.DateTime.Now.Minute) / 2);
            r = (2 * System.Math.PI * i) / 360;
            x3 = Convert.ToInt16(x1 + (d / 2 - 15) * System.Math.Cos(r));
            y3 = Convert.ToInt16(y1 - (d / 2 - 15) * System.Math.Sin(r));
            zona_des.DrawLine(c_ore, x1, y1, x3, y3);
            zona_des.DrawEllipse(c_ore, x1 - 2, y1 - 2, 4, 4);
        }
        public void init_ceas(int pozx, int pozy, int lat)
        {
            x0 = pozx;
            y0 = pozy;
            w = lat;
        }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        Pens_back = new System.Drawing.SolidBrush(this.BackColor);
        Desen = this.CreateGraphics();
        ceas1 = new ceas();
        ceas1.init_ceas(50, 10, 150);
        ceas2 = new ceas();
        ceas2.init_ceas(10, 125, 300);
        ceas3 = new ceas();
        ceas3.init_ceas(200, 50, 200);
    }
```

```
    private void timer1_Tick(object sender, EventArgs e)
    {
      ceas1.setval(Desen, Pens_back);
      ceas2.setval(Desen, Pens_back);
      ceas3.setval(Desen, Pens_back);
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      ceas1.des_ceas(Desen, Pens_back);
      ceas2.des_ceas(Desen, Pens_back);
      ceas3.des_ceas(Desen, Pens_back);
    }
  }
}
```



Obiecte cu parametrii tablouri

C# Aplicatia **"Oop_instr_10"**

- Afisaj grafic x-t

```
namespace Oop_instr_10
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public afisor_xt instr;
      System.Random nr;
      int np = 35;
      int v_max = 300;
      static float[] valori = new float[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         instr = new afisor_xt();
         instr.init_ins(10, 10, np*10, 200, v_max);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         instr.desenez(desen, creion_albastru);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         int nr_max, val_max;

         nr_max = np;
         val_max = v_max;
         instr.sterg(desen, radiera);
         Array.Resize(ref valori, nr_max + 1);
```

```
        for(int i=1;i<=nr_max;i++){
          valori[i]=nr.Next(val_max);
        }
        instr.setval(desen, creion_rosu,valori,nr_max);
      }
    }

    public class afisor_xt
    {
      float x0;
      float y0;
      float w;
      float h;
      float val_max;
      public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
      {
        zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      }
      public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
      {
        zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
      }

      public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] vals, int nrv)
      {
        float val_v,val;
        val_v = 0;
        for(int i=1;i<=nrv;i++){
          val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
          zona_des.DrawLine(creion, x0 +(i-1)*10, y0+val_v, x0 +i*10, y0+val);
          val_v=val;
        }
      }
      public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
      {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        h = inalt;
        val_max = vmax;
      }
    }
}
```

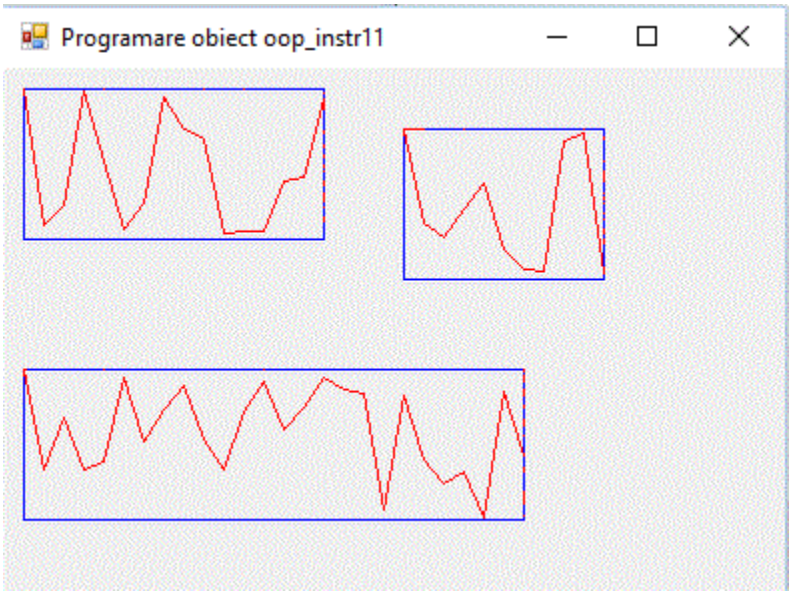
C# Aplicatia **"Oop_instr_11"**

- mai multe obiecte : Afisaj grafic x-t

```
namespace Oop_instr_11
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public afisor_xt instr_1;
      public afisor_xt instr_2;
      public afisor_xt instr_3;
      System.Random nr;
      int np1 = 10;// numarul de puncte afisate la distanta 10 pixeli
      int np2 = 25;
      int np3 = 15;
      int val_max1 = 300;
      int val_max2 = 400;
      int val_max3 = 100;

      static float[] valori = new float[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
```

```
    creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
    creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
    radiera = new System.Drawing.SolidBrush(this.BackColor);
    nr = new System.Random();
    instr_1 = new afisor_xt();
    instr_1.init_ins(200, 30, np1*10, 75, val_max1);
    instr_2 = new afisor_xt();
    instr_2.init_ins(10, 150, np2*10, 100, val_max2);
    instr_3 = new afisor_xt();
    instr_3.init_ins(10, 10, np3*10, 75, val_max3);
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    instr_1.desenez(desen, creion_albastru);
    instr_2.desenez(desen, creion_albastru);
    instr_3.desenez(desen, creion_albastru);

}

private void timer1_Tick(object sender, EventArgs e)
{
    int nr_max, val_max;
    // instrumentul 1
    nr_max = np1;
    val_max = val_max1;
    instr_1.sterg(desen, radiera);
    Array.Resize(ref valori, nr_max + 1);
    for (int i = 1; i <= nr_max; i++)
    {
        valori[i] = nr.Next(val_max);
    }
    instr_1.setval(desen, creion_rosu, valori, nr_max);
    // instrumentul 2
    nr_max = np2;
    val_max = val_max2;
    instr_2.sterg(desen, radiera);
    Array.Resize(ref valori, nr_max + 1);
    for (int i = 1; i <= nr_max; i++)
    {
        valori[i] = nr.Next(val_max);
    }
    instr_2.setval(desen, creion_rosu, valori, nr_max);
    // instrumentul 3
    nr_max = np3;
    val_max = val_max3;
    instr_3.sterg(desen, radiera);
    Array.Resize(ref valori, nr_max + 1);
    for (int i = 1; i <= nr_max; i++)
    {
        valori[i] = nr.Next(val_max);
    }
```

```
        instr_3.setval(desen, creion_rosu, valori, nr_max);
      }
    }

    public class afisor_xt
    {
      float x0;
      float y0;
      float w;
      float h;
      float val_max;
      public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
      {
        zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      }
      public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
      {
        zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
      }

      public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] vals, int nrv)
      {
        float val_v, val;
        val_v = 0;
        for (int i = 1; i <= nrv; i++)
        {
          val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
          zona_des.DrawLine(creion, x0 + (i - 1) * 10, y0 + val_v, x0 + i * 10, y0 + val);
          val_v = val;
        }
      }
      public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
      {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        h = inalt;
        val_max = vmax;
      }
    }
}
```



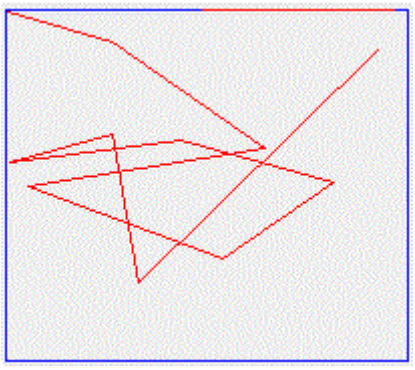
C# Aplicatia **"Oop_instr_30"**

- Afisaj grafic x-y

```
namespace Oop_instr_30
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      Int32  nr_max;
      float  x0,y0,val_maxx, val_maxy,w,h;
      System.Random nr;
      static float[] valorix = new float[0];
      static float[] valoriy = new float[0];
      public afisor_xy instr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         nr_max = 10;
         val_maxx = 300;
         val_maxy = 500;
         x0 = 10;
```

```
        y0 = 10;
        w = 200;
        h = 175;
        instr = new afisor_xy();
        instr.init_ins(x0, y0, w, h, val_maxx,val_maxy);
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        instr.desenez(desen, creion_albastru);
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        instr.sterg(desen, radiera);
        Array.Resize(ref valorix, nr_max + 1);
        Array.Resize(ref valoriy, nr_max + 1);
        for (int i = 1; i <= nr_max; i++)
        {
            valorix[i] = nr.Next(System.Convert.ToInt32(val_maxx));
            valoriy[i] = nr.Next(System.Convert.ToInt32(val_maxy));
        }
        instr.setval(desen, creion_rosu, valorix, valoriy, nr_max);
    }
}
public class afisor_xy
{
    float x0,y0,w,h,val_maxx,val_maxy;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
        zona_des.DrawRectangle(creion_a, x0, y0, w, h);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
        zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
    }

    public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] valsx, float[] valsy,
int nrv)
    {
        float val_vx, valx, val_vy, valy;
        val_vx = 1;
        val_vy = 1;
        for (int i = 1; i <= nrv; i++)
        {
            valx = System.Convert.ToInt16(System.Convert.ToDouble(valsx[i]) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_maxx))); //scalare
            valy = System.Convert.ToInt16(System.Convert.ToDouble(valsy[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_maxy))); //scalare

            zona_des.DrawLine(creion, x0 + val_vx, y0 + val_vy, x0 + valx, y0 + valy);
            val_vx = valx;
            val_vy = valy;
```

```
        }
    }
    public void init_ins(float pozx, float pozy, float lat, float inalt, float vmaxx, float vmaxy)
    {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        h = inalt;
        val_maxx = vmaxx;
        val_maxy = vmaxy;

    }
  }
}
```



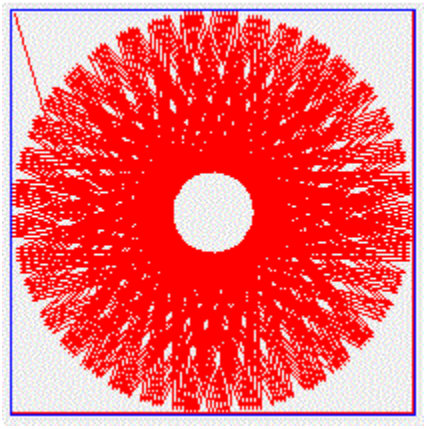
C# Aplicatia **"Oop_instr_31"**

- Afisaj grafic x-y functia sin(i) cos(k*i)

```
namespace Oop_instr_31
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public System.Drawing.Pen creion_albastru;
    public System.Drawing.Pen creion_rosu;
    public System.Drawing.SolidBrush radiera;
    Int32 nr_max;
    float x0, y0, val_maxx, val_maxy, w, h;
    System.Random nr;
    static float[] valorix = new float[0];
    static float[] valoriy = new float[0];
    public afisor_xy instr;

    private void Form1_Load(object sender, EventArgs e)
```

```
    {
        desen = this.CreateGraphics();
        creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
        creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
        radiera = new System.Drawing.SolidBrush(this.BackColor);
        nr = new System.Random();
        nr_max = 500;
        val_maxx = 100;
        val_maxy = 100;
        x0 = 10;
        y0 = 10;
        w = 200;
        h = 200;
        instr = new afisor_xy();
        instr.init_ins(x0, y0, w, h, val_maxx, val_maxy);
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        instr.desenez(desen, creion_albastru);
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        instr.sterg(desen, radiera);
        Array.Resize(ref valorix, nr_max + 1);
        Array.Resize(ref valoriy, nr_max + 1);

        int k = System.Convert.ToInt32(nr.Next(15));
        double f = 10;
        //double f = nr.Next(100);
        // Lisajou
        //int k = 7;
        //double f = 0.01*nr.Next(10);


        for (int i = 1; i <= nr_max; i++)
        {
            //valorix[i] = System.Convert.ToInt32(val_maxx));
            valorix[i] = System.Convert.ToInt32(50*(1-Math.Sin(f*i)));

            valoriy[i] = System.Convert.ToInt32(50*(1-Math.Cos(f*k * i)));
        }
        instr.setval(desen, creion_rosu, valorix, valoriy, nr_max);
    }

}
public class afisor_xy
{
    float x0, y0, w, h, val_maxx, val_maxy;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
        zona_des.DrawRectangle(creion_a, x0-1, y0-1, w+2, h+2);
    }
```

```
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillRectangle(rad, x0 , y0 , w, h);
    }

    public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] valsx, float[] valsy,
int nrv)
    {
      float val_vx, valx, val_vy, valy;
      val_vx = 1;
      val_vy = 1;
      for (int i = 1; i <= nrv; i++)
      {
        valx = System.Convert.ToInt16(System.Convert.ToDouble(valsx[i]) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_maxx))); //scalare
        valy = System.Convert.ToInt16(System.Convert.ToDouble(valsy[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_maxy))); //scalare

        zona_des.DrawLine(creion, x0 + val_vx, y0 + val_vy, x0 + valx, y0 + valy);
        val_vx = valx;
        val_vy = valy;

      }
    }
    public void init_ins(float pozx, float pozy, float lat, float inalt, float vmaxx, float vmaxy)
    {
      x0 = pozx;
      y0 = pozy;
      w = lat;
      h = inalt;
      val_maxx = vmaxx;
      val_maxy = vmaxy;

    }
  }
}
```



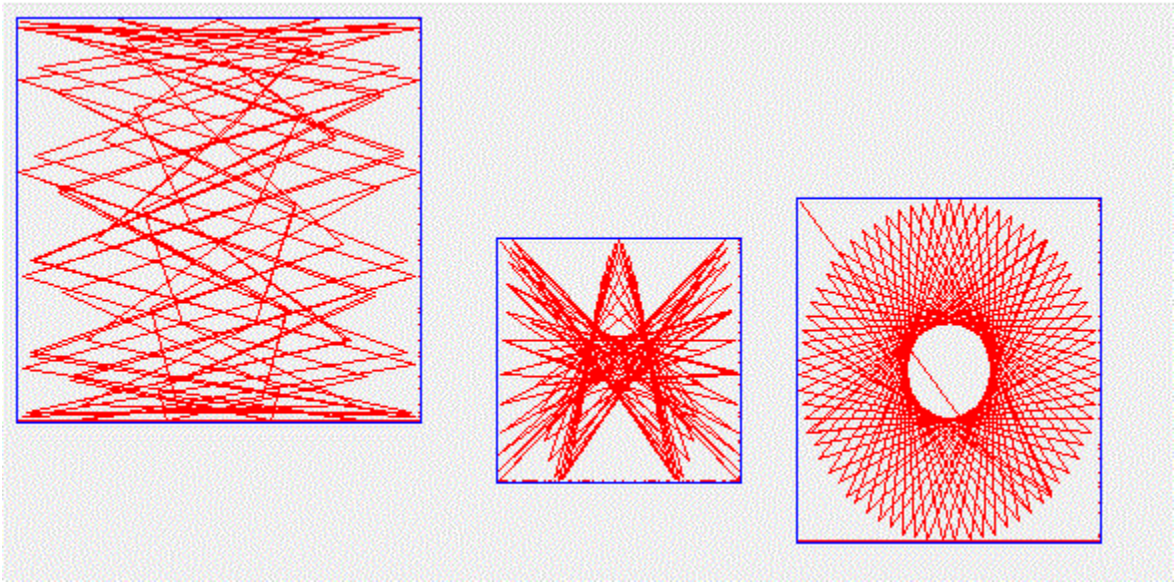
C# Aplicatia **"Oop_instr_32"**

- mai multe obiecte : Afisaj grafic x-y

```
namespace Oop_instr_32
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public System.Drawing.Pen creion_albastru;
    public System.Drawing.Pen creion_rosu;
    public System.Drawing.SolidBrush radiera;
    Int32 nr_max;
    float x0, y0, val_maxx, val_maxy, w, h;
    System.Random nr;
    static float[] valorix = new float[0];
    static float[] valoriy = new float[0];
    public afisor_xy instr1,instr2,instr3;
    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
      creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
      radiera = new System.Drawing.SolidBrush(this.BackColor);
      nr = new System.Random();
      nr_max = 500;
      val_maxx = 100;
      val_maxy = 100;
      x0 = 10;
      y0 = 10;
      w = 200;
      h = 200;
      instr1 = new afisor_xy();
      instr1.init_ins(x0, y0, w, h, val_maxx, val_maxy);
      instr2 = new afisor_xy();
      instr2.init_ins(400, 100, 150, 170, val_maxx, val_maxy);
      instr3 = new afisor_xy();
      instr3.init_ins(250, 120, 120, 120, val_maxx, val_maxy);
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      instr1.desenez(desen, creion_albastru);
      instr2.desenez(desen, creion_albastru);
      instr3.desenez(desen, creion_albastru);
    }

    private void timer1_Tick(object sender, EventArgs e)
```

```
    {
      int i, k;
      Array.Resize(ref valorix, nr_max + 1);
      Array.Resize(ref valoriy, nr_max + 1);
      double f = 10;
      k = System.Convert.ToInt32(1 + nr.Next(15));
      for (i = 1; i <= nr_max; i++)
      {
        valorix[i] = System.Convert.ToInt32(50 * (1 - Math.Sin(f * i)));
        valoriy[i] = System.Convert.ToInt32(50 * (1 - Math.Cos(f * k * i)));
      }
      instr1.sterg(desen, radiera);
      instr1.setval(desen, creion_rosu, valorix, valoriy, nr_max);
      k = System.Convert.ToInt32(1 + nr.Next(15));
      for (i = 1; i <= nr_max; i++)
      {
        valorix[i] = System.Convert.ToInt32(50 * (1 - Math.Sin(f * i)));
        valoriy[i] = System.Convert.ToInt32(50 * (1 - Math.Cos(f * k * i)));
      }
      instr2.sterg(desen, radiera);
      instr2.setval(desen, creion_rosu, valorix, valoriy, nr_max);
      k = System.Convert.ToInt32(1 + nr.Next(15));
      for (i = 1; i <= nr_max; i++)
      {
        valorix[i] = System.Convert.ToInt32(50 * (1 - Math.Sin(f * i)));
        valoriy[i] = System.Convert.ToInt32(50 * (1 - Math.Cos(f * k * i)));
      }
      instr3.sterg(desen, radiera);
      instr3.setval(desen, creion_rosu, valorix, valoriy, nr_max);
    }
  }
  public class afisor_xy
  {
    float x0, y0, w, h, val_maxx, val_maxy;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawRectangle(creion_a, x0 - 1, y0 - 1, w + 2, h + 2);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillRectangle(rad, x0, y0, w, h);
    }

    public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] valsx, float[] valsy,
int nrv)
    {
      float val_vx, valx, val_vy, valy;
      val_vx = 1;
      val_vy = 1;
      for (int i = 1; i <= nrv; i++)
      {
```

```
            valx = System.Convert.ToInt16(System.Convert.ToDouble(valsx[i]) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_maxx))); //scalare
            valy = System.Convert.ToInt16(System.Convert.ToDouble(valsy[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_maxy))); //scalare

            zona_des.DrawLine(creion, x0 + val_vx, y0 + val_vy, x0 + valx, y0 + valy);
            val_vx = valx;
            val_vy = valy;

          }
      }
      public void init_ins(float pozx, float pozy, float lat, float inalt, float vmaxx, float vmaxy)
      {
          x0 = pozx;
          y0 = pozy;
          w = lat;
          h = inalt;
          val_maxx = vmaxx;
          val_maxy = vmaxy;

      }
   }
}
```



Realizarea de clase care descriu instrumente virtuale

Instrumente virtuale pentru valori instantanee

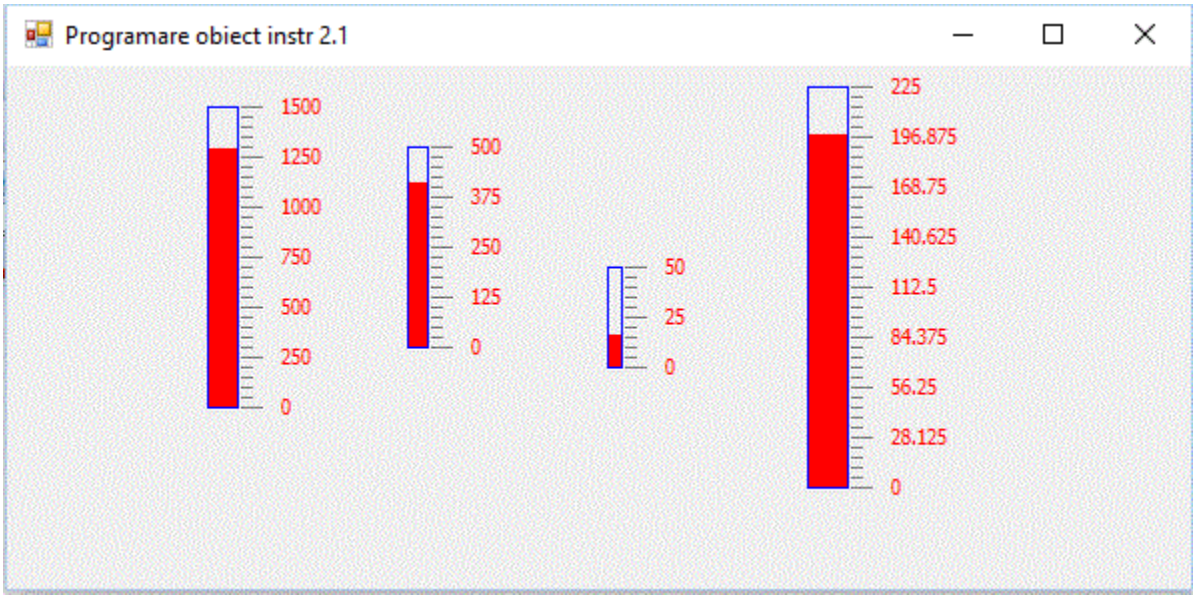
- Instrument virtual - termometru

C# Aplicatia **"Oop_instr_20"**

```
namespace Oop_instr_20
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_gri;
      public System.Drawing.SolidBrush radiera;
      public System.Drawing.SolidBrush pensula_rosie;
      public System.Drawing.Font font_nina;
      public termo instr;
      public System.Random nr;

      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_gri = new System.Drawing.Pen(System.Drawing.Color.Gray);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         pensula_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
         font_nina = new System.Drawing.Font("Nina", 8);
         nr = new System.Random();
         instr = new termo();
         instr.init_ins(100, 20, 10, 150, 1500);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         instr.desenez(desen, creion_albastru, creion_gri, pensula_rosie,font_nina);
      }
```

```
    private void timer1_Tick(object sender, EventArgs e)
    {
      instr.sterg(desen, radiera);
      instr.setval(nr.Next(1500), desen, pensula_rosie);
    }
  }

  public class termo
  {
    float x0;
    float y0;
    float w;
    float h;
    float val_max;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a, System.Drawing.Pen
creion_gr,System.Drawing.SolidBrush pens_r,System.Drawing.Font font_ni)
    {
      zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      for (int j = 0; j <= h; j += 5)// desenez gradatii
      {
        if (j % 25 == 0)
        {
          zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0 + w + 12, y0 + j);
          zona_des.DrawString(System.Convert.ToString(val_max - j * val_max / h), font_ni, pens_r, x0 + w + 20,
y0 + j - 7);
        }
        else
        {
          zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0 + w + 7, y0 + j);
        }
      }



    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.SolidBrush rad)
    {
      zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
    }

    public void setval(float val, System.Drawing.Graphics zona_des, System.Drawing.SolidBrush pens_r)
    {
      val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
      zona_des.FillRectangle(pens_r, x0 + 1, y0+h-val, w - 1, val);

    }
    public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
    {
      x0 = pozx;
      y0 = pozy;
```

```
            w = lat;
            h = inalt;
            val_max = vmax;
        }
    }
}
```



C# Aplicatia **"Oop_instr_21"**

- mai multe obiecte : Mai multe obiecte de tip : instrument virtual

```
namespace Oop_instr_21
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public System.Drawing.Pen creion_albastru;
        public System.Drawing.Pen creion_gri;
        public System.Drawing.SolidBrush radiera;
        public System.Drawing.SolidBrush pensula_rosie;
```

```
    public System.Drawing.Font font_nina;
    public termo instr_1;
    public termo instr_2;
    public termo instr_3;
    public termo instr_4;
    public System.Random nr;
    private void Form1_Load(object sender, EventArgs e)
    {
        desen = this.CreateGraphics();
        creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
        creion_gri = new System.Drawing.Pen(System.Drawing.Color.Gray);
        radiera = new System.Drawing.SolidBrush(this.BackColor);
        pensula_rosie = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
        font_nina = new System.Drawing.Font("Nina", 8);
        nr = new System.Random();
        instr_1 = new termo();
        instr_1.init_ins(100, 20, 15, 150, 1500);
        instr_2 = new termo();
        instr_2.init_ins(200, 40, 10, 100, 500);
        instr_3 = new termo();
        instr_3.init_ins(300, 100, 7, 50, 50);
        instr_4 = new termo();
        instr_4.init_ins(400, 10, 20, 200, 225);
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        instr_1.desenez(desen, creion_albastru, creion_gri, pensula_rosie, font_nina);
        instr_2.desenez(desen, creion_albastru, creion_gri, pensula_rosie, font_nina);
        instr_3.desenez(desen, creion_albastru, creion_gri, pensula_rosie, font_nina);
        instr_4.desenez(desen, creion_albastru, creion_gri, pensula_rosie, font_nina);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        instr_1.sterg(desen, radiera);
        instr_1.setval(nr.Next(1500), desen, pensula_rosie);
        instr_2.sterg(desen, radiera);
        instr_2.setval(nr.Next(500), desen, pensula_rosie);
        instr_3.sterg(desen, radiera);
        instr_3.setval(nr.Next(50), desen, pensula_rosie);
        instr_4.sterg(desen, radiera);
        instr_4.setval(nr.Next(225), desen, pensula_rosie);
    }
}
public class termo
{
    float x0;
    float y0;
    float w;
    float h;
    float val_max;
```

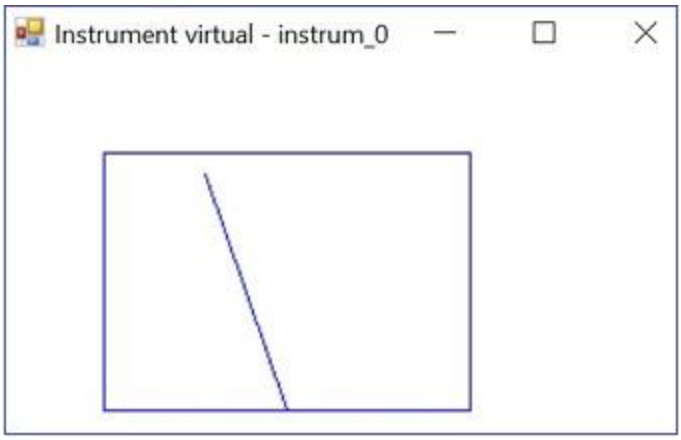
```
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a, System.Drawing.Pen
creion_gr, System.Drawing.SolidBrush pens_r, System.Drawing.Font font_ni)
    {
      zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      for (int j = 0; j <= h; j += 5)// desenez gradatii
      {
        if (j % 25 == 0)
        {
          zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0 + w + 12, y0 + j);
          zona_des.DrawString(System.Convert.ToString(val_max - j * val_max / h), font_ni, pens_r, x0 + w + 20,
y0 + j - 7);
        }
        else
        {
          zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0 + w + 7, y0 + j);
        }
      }

    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.SolidBrush rad)
    {
      zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
    }

    public void setval(float val, System.Drawing.Graphics zona_des, System.Drawing.SolidBrush pens_r)
    {
      val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
      zona_des.FillRectangle(pens_r, x0 + 1, y0 + h - val, w - 1, val);

    }
    public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
    {
      x0 = pozx;
      y0 = pozy;
      w = lat;
      h = inalt;
      val_max = vmax;
    }
  }
}
```

- ## Instrument virtual - voltmetru

Vom crea o clasa instrument de masura analogic denumita "voltm" dupa care vom realza un obiect prin instantierea clasei voltm

Pornim de la aplicatia clasica ce simuleaza un voltmetru.

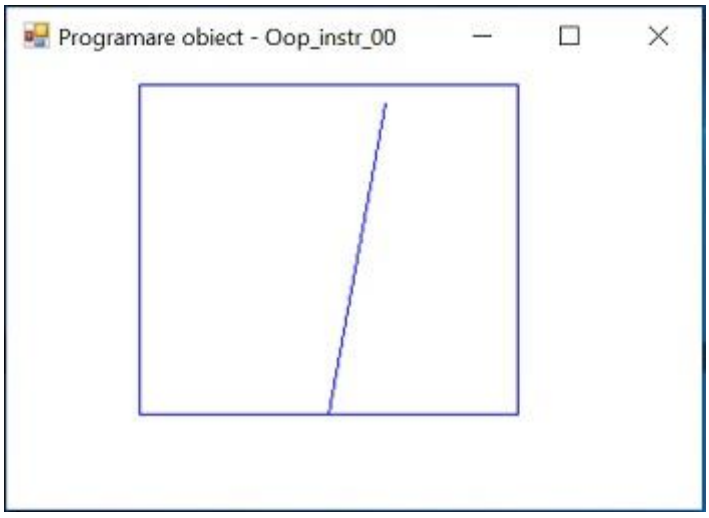
```
namespace instrum_0
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.SolidBrush radiera;
      System.Random nr;
      float x0=50;
      float y0=40;
      float w=140;
      float h=100;
      float val_max=220;
      float val;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         desen.FillRectangle(radiera, x0 + 1, y0 + 1, w - 1, h - 1);
         desen.DrawRectangle(creion_albastru, x0, y0, w, h);
         val = nr.Next(System.Convert.ToInt16(val_max));
         val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_max))); //scalare
         desen.DrawLine(creion_albastru, (x0 + w / 2), h + y0, val + x0, y0 + 10);

      }
   }
}
```



Creem acum o clasa instrument de masura analogic denumita "voltm" si o instantiem.

```
namespace Oop_instr_00
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.SolidBrush radiera;
      public voltm instr;
      System.Random nr;

      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         instr = new voltm();
         instr.init_ins(100,100,100,75,1500);
      }
     private void Form1_Paint(object sender, PaintEventArgs e)
      {
         instr.desenez(desen, creion_albastru);
      }
      private void timer1_Tick(object sender, EventArgs e)
      {
         instr.sterg(desen, radiera);
         instr.setval(nr.Next(1500),desen, creion_albastru);
      }
   }
```

```
  public class voltm
  {
    float x0;
    float y0;
    float w;
    float h;
    float val_max;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawRectangle(creion_a, x0, y0, w, h);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillRectangle(rad, x0+1, y0+1, w-1, h-1);
    }

    public void setval(float val, System.Drawing.Graphics zona_des, System.Drawing.Pen creion)
    {
      val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_max))); //scalare
      zona_des.DrawLine(creion, (x0 + w / 2), h + y0, val + x0, y0 + 10);

    }
    public void init_ins(float pozx, float pozy, float lat,float inalt,float vmax)
    {
      x0=pozx;
      y0=pozy;
      w=lat;
      h=inalt;
      val_max = vmax;
    }
  }
}
```



Putem rescrie clasa "voltm" folosind un constructor. Aplicatia devine:

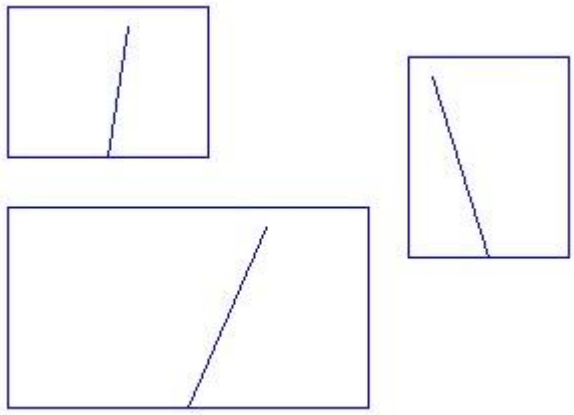
```
namespace Oop_instr_000
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.SolidBrush radiera;
      public voltm instr;
      System.Random nr;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         instr = new voltm(100, 100, 100, 75, 1500);

      }

     private void Form1_Paint(object sender, PaintEventArgs e)
      {
         instr.desenez(desen, creion_albastru);
      }
      private void timer1_Tick(object sender, EventArgs e)
      {
         instr.sterg(desen, radiera);
         instr.setval(nr.Next(1500), desen, creion_albastru);
      }
   }
   public class voltm
   {
      float x0;
      float y0;
      float w;
      float h;
      float val_max;
      public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
      {
         zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      }
      public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
      {
         zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
      }
```

```
    public void setval(float val, System.Drawing.Graphics zona_des, System.Drawing.Pen creion)
    {
      val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_max))); //scalare
      zona_des.DrawLine(creion, (x0 + w / 2), h + y0, val + x0, y0 + 10);

    }
    public voltm(float pozx, float pozy, float lat, float inalt, float vmax)
    {
      x0 = pozx;
      y0 = pozy;
      w = lat;
      h = inalt;
      val_max = vmax;
    }
  }
}
```

Pe baza clasei "voltm" creata anterior, vom realiza mai multe obiecte prin instantierea clasei voltm

```
namespace Oop_instr_01
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public System.Drawing.Pen creion_albastru;
    public System.Drawing.SolidBrush radiera;
    public voltm instr_1;
    public voltm instr_2;
    public voltm instr_3;
    System.Random nr;
    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
      radiera = new System.Drawing.SolidBrush(this.BackColor);
      nr = new System.Random();
      instr_1 = new voltm();
      instr_1.init_ins(100, 50, 100, 75, 1500);
      instr_2 = new voltm();
      instr_2.init_ins(300, 75, 80, 100, 20);
      instr_3 = new voltm();
      instr_3.init_ins(100, 150, 180, 100, 400);
    }
```

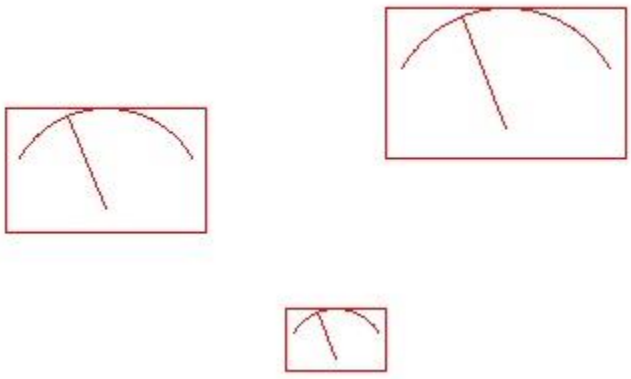
```
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      instr_1.desenez(desen, creion_albastru);
      instr_2.desenez(desen, creion_albastru);
      instr_3.desenez(desen, creion_albastru);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      instr_1.sterg(desen, radiera);
      instr_1.setval(nr.Next(1500), desen, creion_albastru);
      instr_2.sterg(desen, radiera);
      instr_2.setval(nr.Next(20), desen, creion_albastru);
      instr_3.sterg(desen, radiera);
      instr_3.setval(nr.Next(400), desen, creion_albastru);
    }
  }
  public class voltm
  {
    float x0;
    float y0;
    float w;
    float h;
    float val_max;
    public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
    {
      zona_des.DrawRectangle(creion_a, x0, y0, w, h);
    }
    public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
    {
      zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
    }

    public void setval(float val, System.Drawing.Graphics zona_des, System.Drawing.Pen creion)
    {
      val = System.Convert.ToInt16(System.Convert.ToDouble(val) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(val_max))); //scalare
      zona_des.DrawLine(creion, (x0 + w / 2), h + y0, val + x0, y0 + 10);

    }
    public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
    {
      x0 = pozx;
      y0 = pozy;
      w = lat;
      h = inalt;
      val_max = vmax;
    }
  }
}
```



Vom modifica clasa instrument de masura analogic denumita "voltm" si vom realiza mai multe obiecte prin instantierea clasei voltm nou creata

```
namespace Oop_instr_02
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics Desen;
      System.Drawing.Pen Creion_rosu;
      System.Drawing.SolidBrush Pens_blu;
      System.Drawing.SolidBrush Pens_back;
      public voltm voltm1;
      public voltm voltm2;
      public voltm voltm3;

      System.Random nr;
      int alfa;

      public class voltm
      {
         int x0;
         int y0;
         int w;
         int h;

         public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion,
System.Drawing.SolidBrush radiera, int alfa_gr)
         {
```

```
            // alfa_gr unghiul in grade
            int xc = x0 + w / 2;
            int yc = y0 + w / 2;
            int raza = w / 2;
            zona_des.FillEllipse(radiera, x0, y0, w, w);

            double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
            int x = System.Convert.ToInt16(xc + raza * System.Math.Cos(alfa_r));
            int y = System.Convert.ToInt16(yc - raza * System.Math.Sin(alfa_r));
            zona_des.DrawRectangle(creion, xc - raza, yc - raza, 2 * raza, 5 * raza / 4);
            zona_des.DrawArc(creion, xc - raza, yc - raza, 2 * raza, 2 * raza, -30, -120);
            zona_des.DrawLine(creion, x, y, xc, yc);
        }

        public void init_voltm(int pozx, int pozy, int lat, int inalt)
        {
            x0 = pozx;
            y0 = pozy;
            w = lat;
            h = inalt;
        }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        Creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
        Pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
        Pens_back = new System.Drawing.SolidBrush(this.BackColor);
        Desen = this.CreateGraphics();
        nr = new System.Random();
        voltm1 = new voltm();
        voltm1.init_voltm(10, 100, 100, 75);
        voltm2 = new voltm();
        voltm2.init_voltm(200, 50, 120, 55);
        voltm3 = new voltm();
        voltm3.init_voltm(150, 200, 50, 175);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        voltm1.setval(Desen, Creion_rosu, Pens_back, alfa);
        voltm2.setval(Desen, Creion_rosu, Pens_back, alfa);
        voltm3.setval(Desen, Creion_rosu, Pens_back, alfa);
        alfa -= 7;
        if (alfa < 40)
            alfa = 140;
    }
  }
}
```

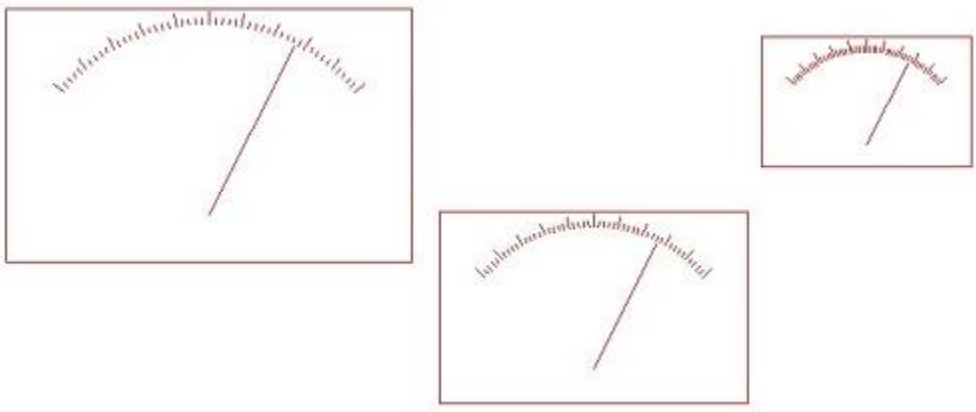


Modificam din nou clasa instrument de masura analogic denumita "voltm" dupa care realizam mai multe obiecte prin instantierea clasei voltm.

```
namespace Oop_instr_03
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics Desen;
      System.Drawing.Pen Creion_rosu;
      System.Drawing.SolidBrush Pens_blu;
      System.Drawing.SolidBrush Pens_back;
      public voltm voltm1;
      public voltm voltm2;
      public voltm voltm3;
      System.Random nr;
      int alfa;
      public class voltm
      {
         int x0;
         int y0;
         int w;
         public void desen_voltm(System.Drawing.Graphics zona_des, System.Drawing.Pen creion,
System.Drawing.SolidBrush radiera)
         {
            int lg = 5;
            int x1, x2, y1, y2;
            int xc = x0 + w / 2;
            int yc = y0 + w / 2;
            int raza = w / 2;
            int nrd;
```

```
            // alfa_gr unghiul in grade
            double alfa_gr = 40;
            nrd = 0;
            while (alfa_gr <= 140)
            {
               double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
               if (nrd % 5 == 0)
               {
                  x1 = System.Convert.ToInt16(xc + raza * System.Math.Cos(alfa_r));
                  y1 = System.Convert.ToInt16(yc - raza * System.Math.Sin(alfa_r));
               }
               else
               {
                  x1 = System.Convert.ToInt16(xc + (raza - lg) * System.Math.Cos(alfa_r));
                  y1 = System.Convert.ToInt16(yc - (raza - lg) * System.Math.Sin(alfa_r));

               }
               x2 = System.Convert.ToInt16(xc + (raza - 2 * lg) * System.Math.Cos(alfa_r));
               y2 = System.Convert.ToInt16(yc - (raza - 2 * lg) * System.Math.Sin(alfa_r));
               zona_des.DrawLine(creion, x1, y1, x2, y2);
               alfa_gr += 2;
               nrd++;
            }
            zona_des.DrawRectangle(creion, xc - raza, yc - raza - 2, 2 * raza, 5 * raza / 4);
         }
         public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion,
System.Drawing.SolidBrush radiera, int alfa_gr)
         {
            // alfa_gr unghiul in grade
            int lg = 5;
            int xc = x0 + w / 2;
            int yc = y0 + w / 2;
            int raza = w / 2;
            zona_des.FillPie(radiera, x0 + 2 * lg-1, y0 + 2 * lg-1, w - 4 * lg+2, w - 4 * lg+2, 10, -180);
            double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
            int x = System.Convert.ToInt16(xc + (raza-2*lg) * System.Math.Cos(alfa_r));
            int y = System.Convert.ToInt16(yc - (raza-2*lg) * System.Math.Sin(alfa_r));
            zona_des.DrawLine(creion, x, y, xc, yc);
            alfa_gr = 40;
            zona_des.DrawRectangle(creion, xc - raza, yc - raza-2, 2 * raza, 5 * raza / 4);
         }

         public void init_voltm(int pozx, int pozy, int lat)
         {
            x0 = pozx;
            y0 = pozy;
            w = lat;
         }
      }
      private void Form1_Load(object sender, EventArgs e)
      {
         Creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
```

```
        Pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
        Pens_back = new System.Drawing.SolidBrush(this.BackColor);
        Desen = this.CreateGraphics();
        nr = new System.Random();
        voltm1 = new voltm();
        voltm1.init_voltm(10, 10, 290);
        voltm2 = new voltm();
        voltm2.init_voltm(320, 155, 220);
        voltm3 = new voltm();
        voltm3.init_voltm(550, 30, 150);
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        voltm1.desen_voltm(Desen, Creion_rosu, Pens_back);
        voltm2.desen_voltm(Desen, Creion_rosu, Pens_back);
        voltm3.desen_voltm(Desen, Creion_rosu, Pens_back);
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        voltm1.setval(Desen, Creion_rosu, Pens_back, alfa);
        voltm2.setval(Desen, Creion_rosu, Pens_back, alfa);
        voltm3.setval(Desen, Creion_rosu, Pens_back, alfa);
        alfa -= 7;
        if (alfa < 40)
            alfa = 140;
    }
  }
}
```

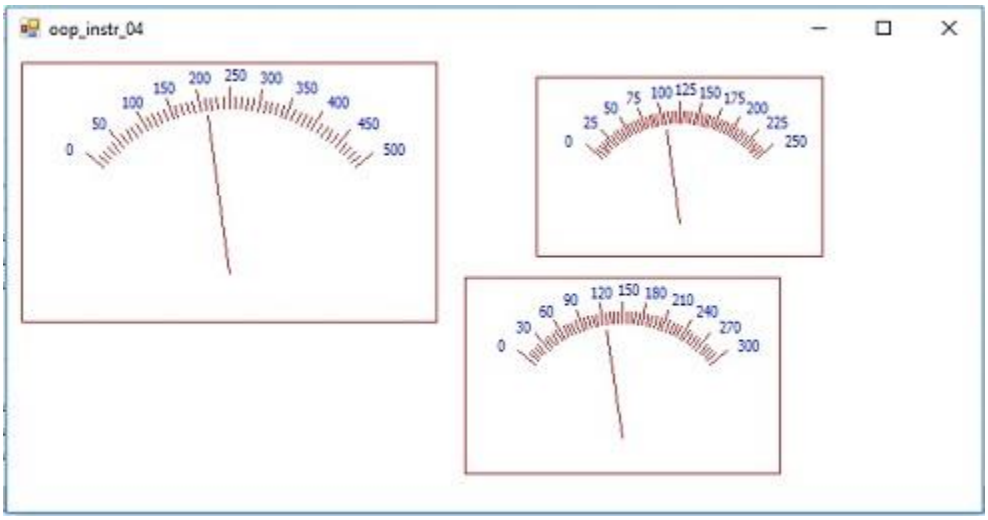


Adugam clasei voltm facilitati pentru afisarea valorilor numerice si totodata vom adauga un constructor, ne mai fiind necesara metoda init_voltm. Pentru a usura folosirea clasei, s-au scos facilitatile pentru schimbarea culorilor acestea fiind fixate in cadrul clasei. S-a mai introdus parametrul val_max in vederea afisarii valorilor numerice pe voltmetru. Metoda "setval" are ca parametru "val" adica valoarea parametrului de afisat (intre 0 si val_max), fiind astfel inlocuit parametrul "alfa".

```
namespace Oop_instr_04
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics Desen;

      public voltm voltm1;
      public voltm voltm2;
      public voltm voltm3;

      System.Random nr;
      double u1,u2,u3;
      double um1 = 500;
      double um2 = 300;
      double um3 = 250;
      public class voltm
      {
         int x0;
         int y0;
         int w;
         double vm;
         System.Drawing.Pen creion= new System.Drawing.Pen(System.Drawing.Color.Red);
         System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
         System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
         System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

         public void desen_voltm(System.Drawing.Graphics zona_des)
         {
            int lt = 15;
            int lg = 22;
            int x1, x2, xt, y1, y2, yt;
            int xc = x0 + w / 2;
            int yc = y0 + w / 2;
            int raza = w / 2;
            int nrd;
            int val_a=0;
            // alfa_gr unghiul in grade
            double alfa_gr = 140;
            nrd = 0;
            while (alfa_gr >=40)
            {
               double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
               if (nrd % 5 == 0)
               {
                  x1 = System.Convert.ToInt16(xc + (raza-lt) * System.Math.Cos(alfa_r));
                  y1 = System.Convert.ToInt16(yc - (raza-lt) * System.Math.Sin(alfa_r));
                  xt = System.Convert.ToInt16(xc-5 + raza * System.Math.Cos(alfa_r));
```

```
            yt = System.Convert.ToInt16(yc - raza * System.Math.Sin(alfa_r));
            zona_des.DrawString(System.Convert.ToString(val_a), font_ni, pens_blu, xt, yt);
            val_a = val_a + System.Convert.ToInt16(vm /10);

          }
          else
          {
            x1 = System.Convert.ToInt16(xc + (raza - lg) * System.Math.Cos(alfa_r));
            y1 = System.Convert.ToInt16(yc - (raza - lg) * System.Math.Sin(alfa_r));
          }
          x2 = System.Convert.ToInt16(xc + (raza - 2 * lt) * System.Math.Cos(alfa_r));
          y2 = System.Convert.ToInt16(yc - (raza - 2 * lt) * System.Math.Sin(alfa_r));
          zona_des.DrawLine(creion, x1, y1, x2, y2);
          alfa_gr -= 2;
          nrd++;
        }
        zona_des.DrawRectangle(creion, xc - raza, yc - raza - 2, 2 * raza, 5 * raza / 4);
      }
      public void setval(System.Drawing.Graphics zona_des, double val)
      {
        int alfa_gr = 140 - System.Convert.ToInt16(100 * val / vm); ;//unghiul in grade

        int lg = 17;
        int xc = x0 + w / 2;
        int yc = y0 + w / 2;
        int raza = w / 2;
        zona_des.FillPie(radiera, x0 + 2 * lg - 1, y0 + 2 * lg - 1, w - 4 * lg + 2, w - 4 * lg + 2, 10, -180);
        double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
        int x = System.Convert.ToInt16(xc + (raza - 2 * lg) * System.Math.Cos(alfa_r));
        int y = System.Convert.ToInt16(yc - (raza - 2 * lg) * System.Math.Sin(alfa_r));
        zona_des.DrawLine(creion, x, y, xc, yc);
        alfa_gr = 40;
        zona_des.DrawRectangle(creion, xc - raza, yc - raza - 2, 2 * raza, 5 * raza / 4);
      }

      public voltm(int pozx, int pozy, int lat, double val_max)
      {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        vm = val_max;
      }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
      Desen = this.CreateGraphics();
      nr = new System.Random();
      voltm1 = new voltm(10, 10, 290,um1);
      voltm2 = new voltm(320, 160, 220,um2);
      voltm3 = new voltm(370, 20, 200,um3);
    }
```

```
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      voltm1.desen_voltm(Desen);
      voltm2.desen_voltm(Desen);
      voltm3.desen_voltm(Desen);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      if (u1 > um1)
        u1 = 0;
      voltm1.setval(Desen, u1);
      u1 += 10;
      if (u2 > um2)
        u2 = 0;
      voltm2.setval(Desen, u2);
      u2 += 25;
      if (u3 > um3)
        u3 = 0;
      voltm3.setval(Desen, u3);
      u3 += 25;
    }
  }
}
```



In loc sa desenam gradatii si valori am putea incarca o imagine a unui voltmetu.

```
namespace Oop_instr_06
{
  public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }
    System.Drawing.Graphics Desen;
    System.Random nr;
    public voltm voltm1;
    double u1,um=250;
    private void Form1_Load(object sender, EventArgs e)
    {
        Desen = this.CreateGraphics();
        nr = new System.Random();
        //voltm1 = new voltm(10, 10, um,"e:\\voltm.jpg");
        voltm1 = new voltm(10, 10, um, "voltm.jpg");
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        if (u1 > um)
            u1 = 0;
        voltm1.setval(Desen, u1);
        u1 += 10;
    }
// Clasa voltmetru
    public class voltm
    {
        System.Drawing.Pen creion = new System.Drawing.Pen(System.Drawing.Color.Red);
        System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 14);
        System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
        System.Drawing.Bitmap img;
        System.Drawing.Bitmap ims;
        int x0,y0,w,h;
        double vm;
        public void setval(System.Drawing.Graphics zona_des, double val)
        {
            int alfa_gr = 140 - System.Convert.ToInt16(100 * val / vm); ;//unghiul in grade
            int lg = 5;
            int xc = w / 2;
            int yc = h-20;
            int raza = 3*w / 5;
            img = new Bitmap(ims);
            Graphics g = Graphics.FromImage(img);
            double alfa_r = 2 * System.Math.PI * (alfa_gr) / 360;// unghiul in radiani
            int x = System.Convert.ToInt16(xc + (raza - 2 * lg) * System.Math.Cos(alfa_r));
            int y = System.Convert.ToInt16(yc - (raza - 2 * lg) * System.Math.Sin(alfa_r));
            g.DrawLine(creion, x, y, xc, yc);
            alfa_gr = 40;
            g.DrawString(val.ToString(),font_ni, pens_blu,0,0);
            zona_des.DrawImage(img, x0, y0);
        }
        public voltm(int pozx, int pozy, double vmax, string fisier_img)
        {
```

```
            x0 = pozx;
            y0 = pozy;
            vm = vmax;
            ims = new Bitmap(fisier_img);
            w = ims.Width;
            h = ims.Height;
        }
    }
  // gata clasa voltmetru
  }
}
```

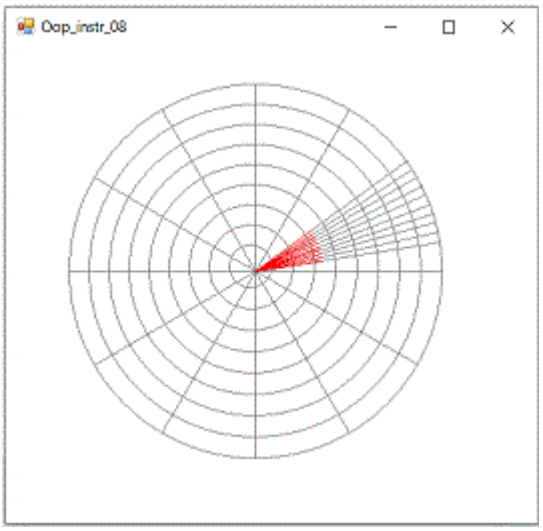
- ## Instrument virtual - anemometru

Vom realiza acum un instrument pentru masurarea directiei si intensitatii vantului.

```
namespace Oop_instr_07
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public anemom anemom1;
    public System.Random n;
    int pozx = 50, pozy = 30, diam = 280;
```

```
private void Form1_Load(object sender, EventArgs e)
{
   desen = this.CreateGraphics();
   n = new System.Random();
   anemom1 = new anemom(pozx, pozy, diam);
}

private void timer1_Tick(object sender, EventArgs e)
{
   anemom1.setval(desen, n.Next(diam / 2), n.Next(360));
}
// ----------Clasa anemometru ------------------

public class anemom
{
   int x0, y0, w, val;
   int r, rc, ds, d, xm, ym, xv, yv, xc, yc;
   double ar, a, ai;
   System.Drawing.Graphics zona_des;
   System.Drawing.Pen creion_a = new System.Drawing.Pen(System.Drawing.Color.Gray);
   System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);

   public void setval(System.Drawing.Graphics desen,int i_vant, int dir_v)
   {
      zona_des = desen;
      r = w / 2 - 1;
      ds = 2 * r / 9; // distanta dintre cercuri
      xc = x0+r;
      yc = y0+r;
      rc = 0;
      val = i_vant;
      ai = dir_v;
      // sterg
      zona_des.Clear(Color.White);

      // pun grid

      // cercuri concentrice
      rc = 0;
      for (d = 2 * r; d >= 0; d -= ds)
      {
         zona_des.DrawEllipse(creion_a, rc+x0, rc+y0, d, d);
         rc += ds / 2;
      }

      //raze in cerc

      for (a = 0; a < 360; a += 360 / 12)
      {
          ar = a * Math.PI / 180;
          xm = System.Convert.ToInt16(xc + r * Math.Sin(ar));
          ym = System.Convert.ToInt16(yc - r * Math.Cos(ar));
```

```
            zona_des.DrawLine(creion_a, xc, yc, xm, ym);
        }

        // fascicul de raze
        for (a = ai; a < (ai + 30); a += 360 / 120)
        {
            ar = a * Math.PI / 180;
            xm = System.Convert.ToInt16(xc + r * Math.Sin(ar));
            ym = System.Convert.ToInt16(yc - r * Math.Cos(ar));
            zona_des.DrawLine(creion_a, xc, yc, xm, ym);
            xv = System.Convert.ToInt16(xc + val * Math.Sin(ar));
            yv = System.Convert.ToInt16(yc - val * Math.Cos(ar));
            zona_des.DrawLine(creion_r, xc, yc, xv, yv);
        }
      }
      public anemom(int pozx, int pozy, int diam)
      {
        x0 = pozx;
        y0 = pozy;
        w = diam;
      }
    }
    // --------------Sfarsit clasa anemometru ------------------
  }
}
```



La rularea aplicatiei se observa faptul ca imaginea nu e fixa pe ecran. Imaginea "clipeste" din cauza stergerii repetate si redesenarii acesteia la fiecare invocare a metodei "set_val"

Pentru a elimina acest efect, vom crea la initializare o imagine de fundal **ims** reprezentand "grid-ul" iar la invocarea metodei "set_val" in loc de stergere, vom incarca imaginea **ims** in imaginea **img**, dupa care se va desenarea tot in imaginea **img** fascicolul de raze reprezentand directia si intensitatea vantului. Afisarea se va face doar cand imaginea e pregatita folosind declaratia: **zona_des.DrawImage(img, x0, y0);**

```
namespace Oop_instr_08
{
   public partial class Form1 : Form
   {

      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public anemom anemom1;
      public System.Random n;
      int pozx = 50, pozy = 30, diam = 300;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         n = new System.Random();
         anemom1 = new anemom(desen, pozx, pozy, diam);
      }
      private void timer1_Tick(object sender, EventArgs e)
      {
         anemom1.setval(n.Next(diam / 2), n.Next(360));
      }
   }
   // --------------Clasa anemometru ------------------
   public class anemom
   {
      int x0, y0, w, val;
      int r , rc , ds , d, xm, ym, xv, yv, xc, yc, i, j;
      double ar, a,ai;

      System.Drawing.Graphics zona_des;
      System.Drawing.Pen creion_a = new System.Drawing.Pen(System.Drawing.Color.Gray);
      System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
      System.Drawing.Bitmap img;
      System.Drawing.Bitmap ims;

      public void setval( int i_vant,int dir_v)
      {
         val = i_vant;
         ai = dir_v;
         if (w > 0)
         {
            // pun grid
            img = new Bitmap(ims);

            // fascicul de raze
            Graphics g = Graphics.FromImage(img);
            for (a = ai; a < (ai + 30); a += 360 / 120)
```

```
        {
            ar = a * Math.PI / 180;
            xm = System.Convert.ToInt16(xc + r * Math.Sin(ar));
            ym = System.Convert.ToInt16(yc - r * Math.Cos(ar));
            g.DrawLine(creion_a, xc, yc, xm, ym);
            xv = System.Convert.ToInt16(xc + val * Math.Sin(ar));
            yv = System.Convert.ToInt16(yc - val * Math.Cos(ar));
            g.DrawLine(creion_r, xc, yc, xv, yv);
        }
    }
    // afisez
    zona_des.DrawImage(img, x0, y0);
}
public anemom(System.Drawing.Graphics desen, int pozx, int pozy, int diam)
{
    x0 = pozx;
    y0 = pozy;
    w = diam;
    r = w / 2 - 1;
    ds = 2 * r / 9; // distanta dintre cercuri
    xc = r;
    yc = r;
    rc = 0;
    double ar, a;
    zona_des = desen;
    img = new Bitmap(w, w, zona_des);
    if (w > 0 )
    {
        ims = new Bitmap(w, w, zona_des);
        Graphics g = Graphics.FromImage(ims);

        // sterg imaginea

        for (j = 0; j < w; j++)
        {
            for (i = 0; i < w; i++)
            {
                ims.SetPixel(i, j, System.Drawing.Color.White);
            }
        }

        // cercuri concentrice

        rc = 0;
        for (d = 2 * r; d >= 0; d -= ds)
        {
            g.DrawEllipse(creion_a, rc, rc, d, d);
            rc += ds / 2;
        }

        //raze in cerc
```

```
            for (a = 0; a < 360; a += 360 / 12)
            {
               ar = a * Math.PI / 180;
               xm = System.Convert.ToInt16(xc + r * Math.Sin(ar));
               ym = System.Convert.ToInt16(yc - r * Math.Cos(ar));
               g.DrawLine(creion_a, xc, yc, xm, ym);
            }
         }
      }
      // --------------Sfarsit clasa anemometru ------------------
   }
}
```

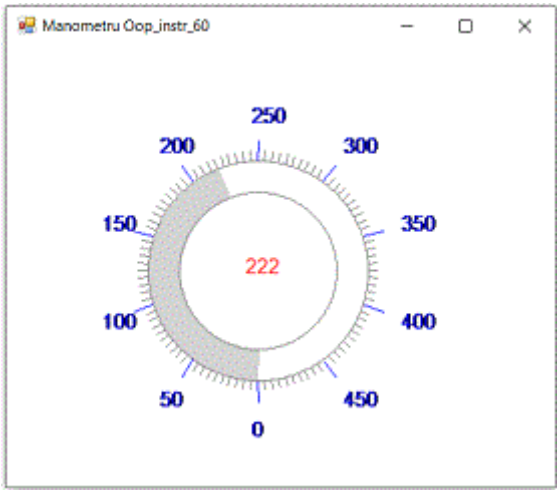
- ## Instrument virtual - manometru

Vom realiza acum un instrument pentru masurarea presiunii.

```
namespace Oop_instr_60
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public manom manom1;
      public System.Random n;
      int pozx = 50, pozy = 30, diam = 300, max=500;
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         n = new System.Random();
         manom1 = new manom(desen, pozx, pozy, diam, max);
      }
      private void timer1_Tick(object sender, EventArgs e)
      {
         manom1.setval(n.Next(diam / 2), n.Next(max));
      }
   }
   // --------------Clasa manometru ------------------

      public class manom
      {
         static int gr = 25; //grosimea arcului de cerc
         int x0, y0, xi, xe, xm, xgr, xgm, xtx, w, val_m, diam_i, diam_e, diam_m, diam_gr, diam_gm, diam_tx;
```

```
System.Drawing.Pen creion_lg = new System.Drawing.Pen(System.Drawing.Color.LightGray,gr);
System.Drawing.Pen creion_rad = new System.Drawing.Pen(System.Drawing.Color.White, gr);
System.Drawing.Pen creion_g = new System.Drawing.Pen(System.Drawing.Color.Gray);
System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Blue);
System.Drawing.Font font_arial=new System.Drawing.Font("Arial",12);
System.Drawing.SolidBrush pens_a = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
System.Drawing.SolidBrush pens_r = new System.Drawing.SolidBrush(System.Drawing.Color.Red);
System.Drawing.SolidBrush pens_rad = new System.Drawing.SolidBrush(System.Drawing.Color.White);
System.Drawing.Graphics zona_des;
System.Drawing.Bitmap img;
Graphics g;
public void setval(int i_vant, int val)
{
   int vl = val;
   int alfa = 360 * vl / val_m;
   if (w > 0)
   {
      // trasez arc de cerc

      g.DrawArc(creion_rad, xm, xm, diam_m, diam_m, 0, 365);
      g.DrawArc(creion_lg, xm, xm, diam_m, diam_m, 90, alfa);
      g.DrawEllipse(creion_g, xi, xi, diam_i, diam_i);
      g.DrawEllipse(creion_g, xe, xe, diam_e, diam_e);
      g.FillRectangle(pens_rad, (w - gr) / 2, (w - gr) / 2, 2 * gr, gr);
      g.DrawString(vl.ToString(), font_arial, pens_r, (w - gr) / 2, (w - gr) / 2);
   }
   // afisez imaginea

   zona_des.DrawImage(img, x0, y0);
}
public manom(System.Drawing.Graphics desen, int pozx, int pozy, int diam, int val_max)
{
   x0 = pozx; // pozitia pe x a instrumentului
   y0 = pozy; // pozitia pe y a instrumentului
   diam_i = diam - 7 * gr; // diametrul cercului interior
   diam_e = diam_i + 2 * gr; // diametrul cercului exterior
   diam_m = (diam_e + diam_i) / 2; // diametrul cercului mediu pentru arcul de cerc
   diam_gr = diam_e + 2 * gr / 3; // diametrul cercului pentru gradatii
   diam_gm = diam_e + 4 * gr / 3; // diametrul cercului pentru gradatii mari
   diam_tx = diam_e + 3 * gr; // diametrul cercului pentru texte
   w = diam; // latimea instrumentului
   val_m = val_max; // valoarea maxima afisata
   xi = (w - diam_i) / 2; // coordonata pentru cercul interior
   xe = (w - diam_e) / 2; // coordonata pentru cercul exterior
   xm = (w - diam_m) / 2; // coordonata pentru cercul mediu
   xgr = (w - diam_gr) / 2; // coordonata pentru cercul gradatiilor mici
   xgm = (w - diam_gm) / 2; // coordonata pentru cercul gradatiilor mari
   xtx = (w - diam_tx) / 2 - gr / 3; // coordonata pentru cercul valorilor
   zona_des = desen;
   img = new Bitmap(w, w, zona_des);
   if (w > 0)
   {
```

```
            g = Graphics.FromImage(img);
            int k = 0;
            double grd; // grade hexazecimale
            for (grd = 630; grd > 270; grd -= 3.6)
            {
               double rad = 2 * System.Math.PI * grd / 360;
               int x1 = Convert.ToInt16(xe + (diam_e / 2)+(diam_e / 2) * System.Math.Cos(rad));
               int y1 = Convert.ToInt16(xe + (diam_e / 2)-(diam_e / 2) * System.Math.Sin(rad));
               int x2 = Convert.ToInt16(xgr + (diam_gr / 2) + (diam_gr / 2) * System.Math.Cos(rad));
               int y2 = Convert.ToInt16(xgr + (diam_gr / 2) - (diam_gr / 2) * System.Math.Sin(rad));
               int x3 = Convert.ToInt16(xgm + (diam_gm / 2) + (diam_gm / 2) * System.Math.Cos(rad));
               int y3 = Convert.ToInt16(xgm + (diam_gm / 2) - (diam_gm / 2) * System.Math.Sin(rad));
               int x4 = Convert.ToInt16(xtx + (diam_tx / 2) + (diam_tx / 2) * System.Math.Cos(rad));
               int y4 = Convert.ToInt16(xtx + (diam_tx / 2) - (diam_tx / 2) * System.Math.Sin(rad));
               if (k % 10 != 0)
               {
                  g.DrawLine(creion_g, x1, y1, x2, y2);
               }
               else
               {
                  g.DrawLine(creion_r, x1, y1, x3, y3);
                  g.DrawString((k*val_m/100).ToString(), font_arial, pens_a, x4, y4);
               }
               k++;
            }
         }
      }
    // ----------Sfarsit clasa manometru ----------
  }
}
```

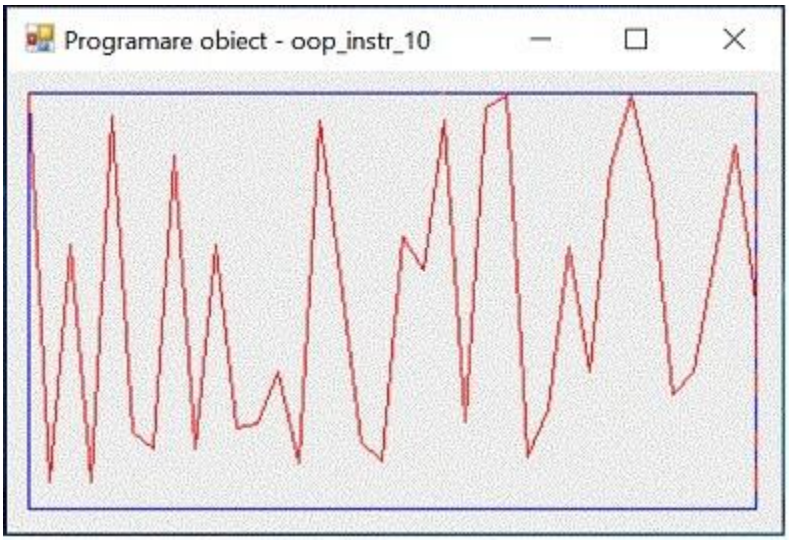


Instrumente virtuale pentru afisarea evolutiei in timp a marimilor electrice

Pornim de la clasa **afisor_xt** folosita in **Aplicatia C#** **"Oop_instr_10"** pentru afisare grafica in coordonate x-t.

```
namespace Oop_instr_10
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.SolidBrush radiera;
      public afisor_xt instr;
      System.Random nr;
      int np = 10;
      int v_max = 300;
      static float[] valori = new float[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         radiera = new System.Drawing.SolidBrush(this.BackColor);
         nr = new System.Random();
         instr = new afisor_xt();
         instr.init_ins(10, 10, np*10, 200, v_max);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         instr.desenez(desen, creion_albastru);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         int nr_max, val_max;

         nr_max = np;
         val_max = v_max;
         instr.sterg(desen, radiera);
         Array.Resize(ref valori, nr_max + 1);
          for(int i=1;i<=nr_max;i++){
            valori[i]=nr.Next(val_max);
```

```
        }
        instr.setval(desen, creion_rosu,valori,nr_max);
      }
    }

    public class afisor_xt
    {
      float x0;
      float y0;
      float w;
      float h;
      float val_max;
      public void desenez(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a)
      {
        zona_des.DrawRectangle(creion_a, x0, y0, w, h);
      }
      public void sterg(System.Drawing.Graphics zona_des, System.Drawing.Brush rad)
      {
        zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
      }

      public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion, float[] vals, int nrv)
      {
        float val_v,val;
        val_v = 0;
        for(int i=1;i<=nrv;i++){
          val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
          zona_des.DrawLine(creion, x0 +(i-1)*10, y0+val_v, x0 +i*10, y0+val);
          val_v=val;
        }
      }
      public void init_ins(float pozx, float pozy, float lat, float inalt, float vmax)
      {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        h = inalt;
        val_max = vmax;
      }
    }
}
```



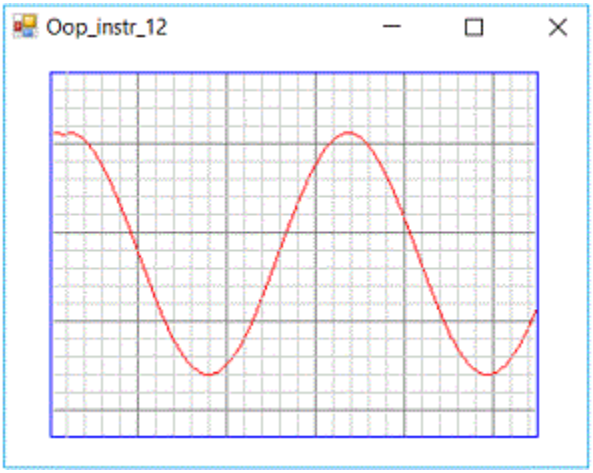
Afisare sinus

Utilizarea clasei afisor_xt pentru afisare grafica in coordonate x-t a functiei sin(x)

```
namespace Oop_instr_12
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_albastru;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.Pen creion_pic;
      public System.Drawing.Pen creion_gri_d;
     public System.Drawing.Pen creion_gri;
      public System.Drawing.SolidBrush radiera;
      public afisor_xt afisor_xt1;
      int pozx = 30, pozy = 10, n_maxx=273, n_maxy=205;
      double alfa = 0, fi = 0;
      static int[] valori = new int[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         Array.Resize(ref valori, n_maxx + 1);
         desen = this.CreateGraphics();
         creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue);
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         creion_pic = new System.Drawing.Pen(this.BackColor);
```

```
        creion_gri_d = new System.Drawing.Pen(System.Drawing.Color.LightGray);
        creion_gri = new System.Drawing.Pen(System.Drawing.Color.Gray);
        radiera = new System.Drawing.SolidBrush(this.BackColor);
        afisor_xt1 = new afisor_xt();
        afisor_xt1.init_ins(pozx,pozy,n_maxx,n_maxy);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
        fi=fi+0.3;
        alfa = fi;
        for (int i = 1; i <= n_maxx; i++)
        {
          alfa += 0.04;
          valori[i] = System.Convert.ToUInt16(n_maxy/6+((n_maxy/3)*( 1 - Math.Sin(alfa))));
        }
        afisor_xt1.setval(desen, creion_albastru, creion_rosu, creion_gri_d, creion_gri, creion_pic,radiera, valori,
n_maxx);
      }
      public class afisor_xt
      {
        int x0, y0, w, h,val_max, nr_max;
        public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_a, System.Drawing.Pen
creion_r, System.Drawing.Pen creion_grd, System.Drawing.Pen creion_gr, System.Drawing.Pen
pic,System.Drawing.SolidBrush radiera, int[] vals, int nrv)
        {
          int val_v, val ,i ,j;
          //zona_des.FillRectangle(radiera, x0 + 1, y0 + 1, w - 1, h - 1);
          //chenar
          zona_des.DrawRectangle(creion_a, x0, y0, w+1, h);
          val_v = System.Convert.ToInt16(System.Convert.ToDouble(vals[1]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max)));
          for (i = 1; i < nrv; i++)
          {
            //grid vertical
            if ((i + 1) % 10 == 0)
            {
              zona_des.DrawLine(creion_grd, i + x0, y0, i + x0, y0 + h);
              if ((i+1) % 50 == 0)
                zona_des.DrawLine(creion_gr, i + x0, y0, i + x0, y0 + h);
            }
            else
              zona_des.DrawLine(pic, x0 + 1, y0 + 1, x0 + 1, y0 + h - 2);
            // grid orizontal
            j = y0+10;
            while (j <= h)
            {
              if (j % 50 == 0)
                zona_des.DrawLine(creion_gr, i + x0, j, i+x0+1, j);
              else
                zona_des.DrawLine(creion_grd, i + x0, j, i+x0+1, j);
              j += 10;
```

```
            }

            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            zona_des.DrawLine(pic, x0 + i+1, y0+1, x0 + i+1, y0 + h-2);
            zona_des.DrawLine(creion_r, x0 + i, y0 + val_v, x0 + i+1, y0 + val);
            val_v = val;
          }
        }
        public void init_ins(int pozx, int pozy, int n_maxx, int n_maxy)
        {
          x0 = pozx;
          y0 = pozy;
          w = n_maxx;
          h = n_maxy;
          nr_max = n_maxx;
          val_max = n_maxy;
        }
      }
    }
}
```



Optimizarea clasei afisor_xt prin folosirea obiectului img

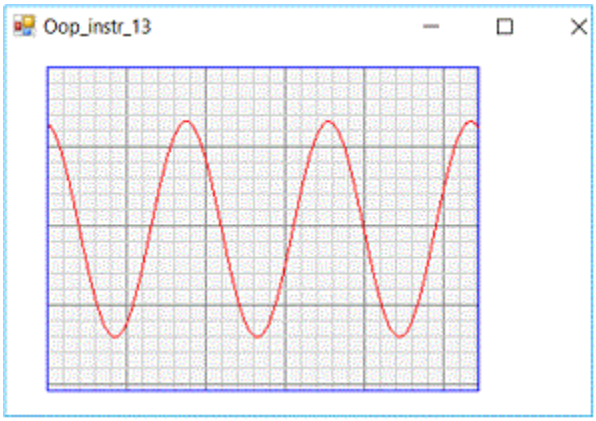
Pentru ptimizarea clasei afisor_xt, vom folosi obiectul imagine **img** in care vom crea imaginea graficului sub forma de puncte dupa care vom afisa imaginea img

```
namespace Oop_instr_13
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
```

```
}
public System.Drawing.Graphics desen;
public System.Drawing.Pen creion_rosu;
public afisor_xt afisor_xt1;
int pozx = 30, pozy = 10, n_maxx = 273, n_maxy = 205;
double alfa = 0;
double fi = 0;
static int[] valori = new int[0];
private void Form1_Load(object sender, EventArgs e)
{
   Array.Resize(ref valori, n_maxx + 1);
   desen = this.CreateGraphics();
   creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
   afisor_xt1 = new afisor_xt();
   afisor_xt1.init_ins(pozx, pozy, n_maxx, n_maxy);
}
private void timer1_Tick(object sender, EventArgs e)
{
   fi = fi + 0.3;
   alfa = fi;
   for (int i = 1; i <= n_maxx; i++)
   {
      alfa += 0.07;
      valori[i] = System.Convert.ToUInt16(n_maxy / 6 + ((n_maxy / 3) * (1 - Math.Sin(alfa))));
   }
   afisor_xt1.setval(desen,creion_rosu, valori, n_maxx);
}
public class afisor_xt
{
   int x0;
   int y0;
   int w;
   int h;
   int val_max,val_v;
   int nr_max;
   System.Drawing.Bitmap img;
   public void setval(System.Drawing.Graphics zona_des,System.Drawing.Pen creion_r, int[] vals, int nrv)
   {
      int val, i, j;
      img = new Bitmap(w,h,zona_des);
      // sterg imaginea
      for (j = 0; j < h; j++)
      {
         for (i = 0; i < w; i++)
         {
            img.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
         }
      }
      // grid
      for (j = 0; j < h; j++)
      {
         // grid orizontal
```

```
            if (j % 10 == 0)
            {
               for (i = 0; i < w; i++)
               {
                  if (j % 50 == 0)
                     img.SetPixel(i, j, System.Drawing.Color.Gray);
                  else
                     img.SetPixel(i, j, System.Drawing.Color.LightGray);
               }
            }
            else{
               // grid orizontal vertical
               for (i = 0; i < w; i++)
               {
                  if (i % 10 == 0)
                  {
                     if (i % 50 == 0)
                        img.SetPixel(i, j, System.Drawing.Color.Gray);
                     else
                        img.SetPixel(i, j, System.Drawing.Color.LightGray);
                  }
               }
            }
         }
         // afisare valoare sub forma de puncte
         /*
         for (i = 0; i < w; i++)
         {
            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            img.SetPixel(i, val, System.Drawing.Color.Red);
            if (val < h - 1)
               img.SetPixel(i, val + 1, System.Drawing.Color.Red);
         }
          */
         //chenar
         for (i = 0; i < w; i++)
         {
           img.SetPixel(i, 0, System.Drawing.Color.Blue);
           img.SetPixel(i, h-1, System.Drawing.Color.Blue);
         }
         for (j = 0; j < h; j++)
         {
            img.SetPixel(0,j, System.Drawing.Color.Blue);
            img.SetPixel(w - 1,j, System.Drawing.Color.Blue);
         }
         zona_des.DrawImage(img, x0, y0);
         // afisare valoare sub forma de linii
         val_v = System.Convert.ToInt16(System.Convert.ToDouble(vals[1]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max)));
         for (i = 1; i < w-1; i++)
         {
```

```
            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            zona_des.DrawLine(creion_r, x0 + i, y0 + val_v, x0 + i + 1, y0 + val);
            val_v = val;
          }
        }
        public void init_ins(int pozx, int pozy, int n_maxx, int n_maxy)
        {
          x0 = pozx;
          y0 = pozy;
          w = n_maxx;
          h = n_maxy;
          nr_max = n_maxx;
          val_max = n_maxy;
        }
      }
    }
}
```



Amplificarea si translatia

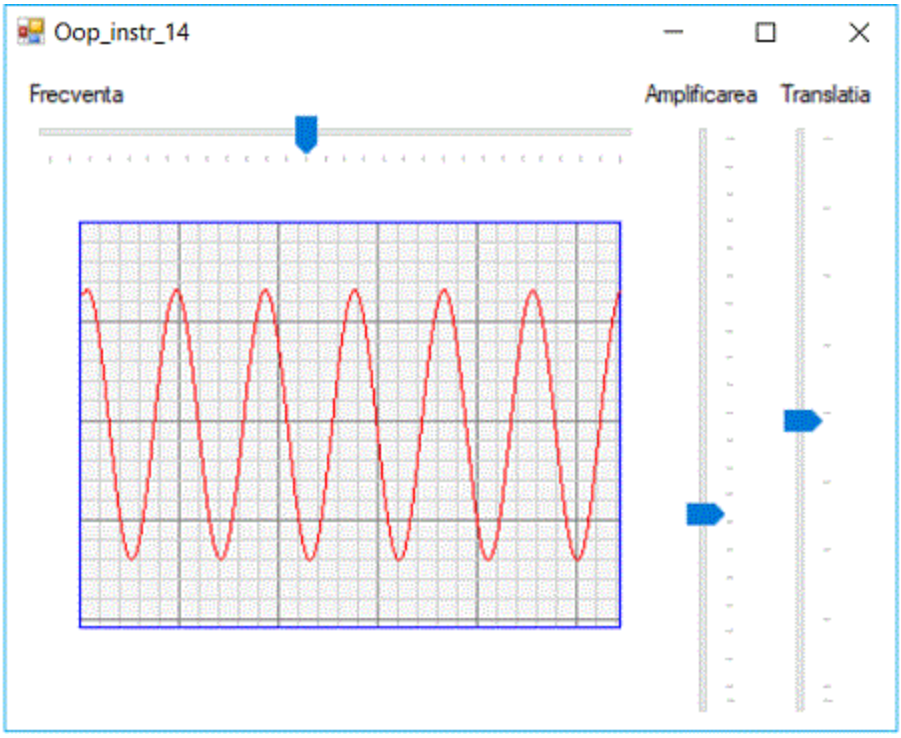
Vom utiliza in continuare clasa "afisor_xt" pentru afisare grafica in coordonate x-t, pentru afisarea functiei sin. Vom utiliza controale pentru modificarea amplitudinii, frecventei, deplasarii etc.

```
namespace Oop_instr_14
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
```

```
public System.Drawing.Pen creion_rosu;
public afisor_xt instrum1;
int pozx = 40, pozy = 80, n_maxx = 273, n_maxy = 205;
double alfa = 0;
double fi = 0;
static int[] valori = new int[0];
private void Form1_Load(object sender, EventArgs e)
{
   Array.Resize(ref valori, n_maxx + 1);
   desen = this.CreateGraphics();
   creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
   instrum1 = new afisor_xt();
   instrum1.init_ins(pozx, pozy, n_maxx, n_maxy);
}
private void Form1_Paint(object sender, PaintEventArgs e)
{
   this.trackBar1.Maximum = n_maxy/5;
   this.trackBar1.Minimum = -n_maxy/5;
   this.trackBar1.Value = 0;
   this.trackBar2.Maximum = n_maxy;
   this.trackBar2.Value = n_maxy / 3;
   this.trackBar3.Minimum = 1;
   this.trackBar3.Value = 14;
}
private void timer1_Tick(object sender, EventArgs e)
{
   fi = fi + 0.3;
   alfa = fi;
   int transl =-this.trackBar1.Value;
   int amplif = this.trackBar2.Value;
   int zero = n_maxy / 2;
   for (int i = 1; i <= n_maxx; i++)
   {
      alfa += System.Convert.ToDouble( this.trackBar3.Value) / 100;
      int f = System.Convert.ToInt32(transl + zero-amplif * Math.Sin(alfa));
      if((f<n_maxy)&&(f>=0))
         valori[i] = f;
      if (f > n_maxy)
         valori[i] = n_maxy-1;
      if(f< h; j++)
   {
      for (i = 0; i < w; i++)
      {
         img.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
      }
   }
   // grid
   for (j = 0; j < h; j++)
   {
      // grid orizontal
      if (j % 10 == 0)
      {
```

```
            for (i = 0; i < w; i++)
            {
               if (j % 50 == 0)
                  img.SetPixel(i, j, System.Drawing.Color.Gray);
               else
                  img.SetPixel(i, j, System.Drawing.Color.LightGray);
            }
         }
         else
         {
            // grid orizontal vertical
            for (i = 0; i < w; i++)
            {
               if (i % 10 == 0)
               {
                  if (i % 50 == 0)
                     img.SetPixel(i, j, System.Drawing.Color.Gray);
                  else
                     img.SetPixel(i, j, System.Drawing.Color.LightGray);
               }
            }
         }
      }
      // afisare valoare sub forma de puncte
      /*
      for (i = 0; i < w; i++)
      {
         val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
         img.SetPixel(i, val, System.Drawing.Color.Red);
         if (val < h - 1)
            img.SetPixel(i, val + 1, System.Drawing.Color.Red);
      }
       */
      //chenar
      for (i = 0; i < w; i++)
      {
         img.SetPixel(i, 0, System.Drawing.Color.Blue);
         img.SetPixel(i, h - 1, System.Drawing.Color.Blue);
      }
      for (j = 0; j < h; j++)
      {
         img.SetPixel(0, j, System.Drawing.Color.Blue);
         img.SetPixel(w - 1, j, System.Drawing.Color.Blue);
      }
      zona_des.DrawImage(img, x0, y0);
      // afisare valoare sub forma de linii
      val_v = System.Convert.ToInt16(System.Convert.ToDouble(vals[1]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max)));
      for (i = 1; i < w - 1; i++)
      {
```

```
            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            zona_des.DrawLine(creion_r, x0 + i, y0 + val_v, x0 + i + 1, y0 + val);
            val_v = val;
        }
    }
    public void init_ins(int pozx, int pozy, int n_maxx, int n_maxy)
    {
        x0 = pozx;
        y0 = pozy;
        w = n_maxx;
        h = n_maxy;
        nr_max = n_maxx;
        val_max = n_maxy;
    }
  }
  // Gata clasa afisor_xt
}
```



Viteza de afisare

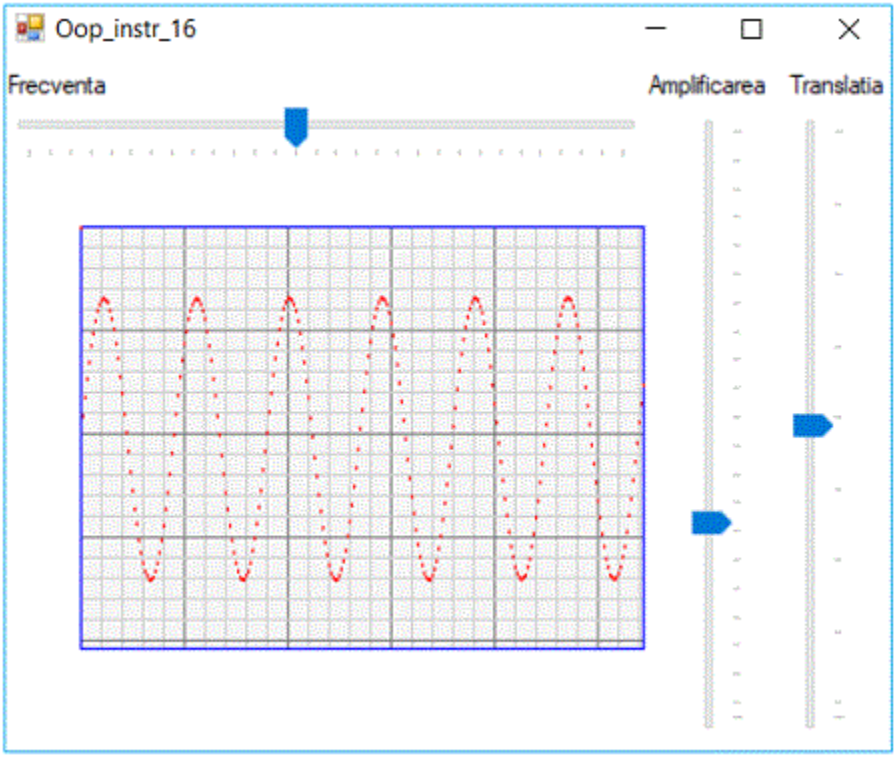
Vom modifica in continuare obiectul "afisor_xt" pentru afisare grafica in coordonate x-t, pentru a obtine o viteza mai mare de afisare. Pentru aceasta vom crea o imagine de findal care contine grid-ul. Dupa fiecare afisare a imaginii, nu vom mai sterge imaginea si nu vom mai realiza din nou grid-ul. Vom incarca doar imaginea care e pregatita in prealabil, imagine care contine numai grid-ul.

```
namespace Oop_instr_16
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public System.Drawing.Pen creion_rosu;
      public System.Drawing.Bitmap im;
      public afisor_xt instrum1;
      int pozx = 40, pozy = 80, n_maxx = 273, n_maxy = 205;
      double alfa = 0;
      double fi = 0;
      static int[] valori = new int[0];

      private void Form1_Load(object sender, EventArgs e)
      {
         Array.Resize(ref valori, n_maxx + 1);
         desen = this.CreateGraphics();
         creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
         instrum1 = new afisor_xt();
         instrum1.init_ins(pozx, pozy, n_maxx, n_maxy);

         //-------- creare imagine fundal ----------------

         int  i, j;
         im = new Bitmap(n_maxx, n_maxy, desen);
         // sterg imaginea
         for (j = 0; j < n_maxy; j++)
         {
            for (i = 0; i < n_maxx; i++)
            {
               im.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
            }
         }
         // grid
         for (j = 0; j < n_maxy; j++)
         {
            // grid orizontal
            if (j % 10 == 0)
            {
               for (i = 0; i < n_maxx; i++)
               {
                  if (j % 50 == 0)
                     im.SetPixel(i, j, System.Drawing.Color.Gray);
                  else
                     im.SetPixel(i, j, System.Drawing.Color.LightGray);
               }
            }
```

```
        else
        {
          // grid orizontal vertical
          for (i = 0; i < n_maxx; i++)
          {
            if (i % 10 == 0)
            {
              if (i % 50 == 0)
                im.SetPixel(i, j, System.Drawing.Color.Gray);
              else
                im.SetPixel(i, j, System.Drawing.Color.LightGray);
            }
          }
        }
      }
      //chenar
      for (i = 0; i < n_maxx; i++)
      {
        im.SetPixel(i, 0, System.Drawing.Color.Blue);
        im.SetPixel(i, n_maxy - 1, System.Drawing.Color.Blue);
      }
      for (j = 0; j < n_maxy; j++)
      {
        im.SetPixel(0, j, System.Drawing.Color.Blue);
        im.SetPixel(n_maxx - 1, j, System.Drawing.Color.Blue);
      }

      // -------- gata imagine de fundal ----------
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      this.trackBar1.Maximum = n_maxy / 5;
      this.trackBar1.Minimum = -n_maxy / 5;
      this.trackBar1.Value = 0;
      this.trackBar2.Maximum = n_maxy;
      this.trackBar2.Value = n_maxy / 3;
      this.trackBar3.Minimum = 1;
      this.trackBar3.Value = 14;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      fi = fi + 0.3;
      alfa = fi;
      int transl = -this.trackBar1.Value;
      int amplif = this.trackBar2.Value;
      int zero = n_maxy / 2;
      for (int i = 1; i <= n_maxx; i++)
      {
        alfa += System.Convert.ToDouble(this.trackBar3.Value) / 100;
        int f = System.Convert.ToInt32(transl + zero - amplif * Math.Sin(alfa));
```

```
            if ((f < n_maxy) && (f >= 0))
               valori[i] = f;
            if (f > n_maxy)
               valori[i] = n_maxy - 1;
            if (f < 0)
               valori[i] = 0;
         }
         desen.DrawImage(im, pozx,pozy);
         instrum1.setval(desen, creion_rosu, valori, n_maxx);
      }
   }
   public class afisor_xt
   {
      int x0;
      int y0;
      int w;
      int h;
      int val_max;
      int nr_max;
      System.Drawing.Bitmap img;
      public void setval(System.Drawing.Graphics zona_des, System.Drawing.Pen creion_r, int[] vals, int nrv)
      {
         int val, i, j;
         img = new Bitmap(w, h, zona_des);

         // afisare valoare sub forma de puncte
         for (i = 0; i < w; i++)
         {
            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            img.SetPixel(i, val, System.Drawing.Color.Red);
            if (val < h - 1)
               img.SetPixel(i, val + 1, System.Drawing.Color.Red);
         }
         zona_des.DrawImage(img, x0, y0);
         // afisare valoare sub forma de linii
         /*
         val_v = System.Convert.ToInt16(System.Convert.ToDouble(vals[1]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max)));
         for (i = 1; i < w - 1; i++)
         {
            val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
            zona_des.DrawLine(creion_r, x0 + i, y0 + val_v, x0 + i + 1, y0 + val);
            val_v = val;
         }
         */

      }
      public void init_ins(int pozx, int pozy, int n_maxx, int n_maxy)
      {
         x0 = pozx;
```

```
            y0 = pozy;
            w = n_maxx;
            h = n_maxy;
            nr_max = n_maxx;
            val_max = n_maxy;
        }
    }
}
```



Clasa osciloscop

Crearea imaginii de fundal ar trebui sa faca parte din clasa afisor_xt. Vom crea o noua clasa numita **osciloscop** pe baza clasei afisor_xt la care adaugam crearea imaginii de fundal si afisarea valorilor numerice.

```
namespace Oop_instr_17
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public osciloscop osciloscop1;
```

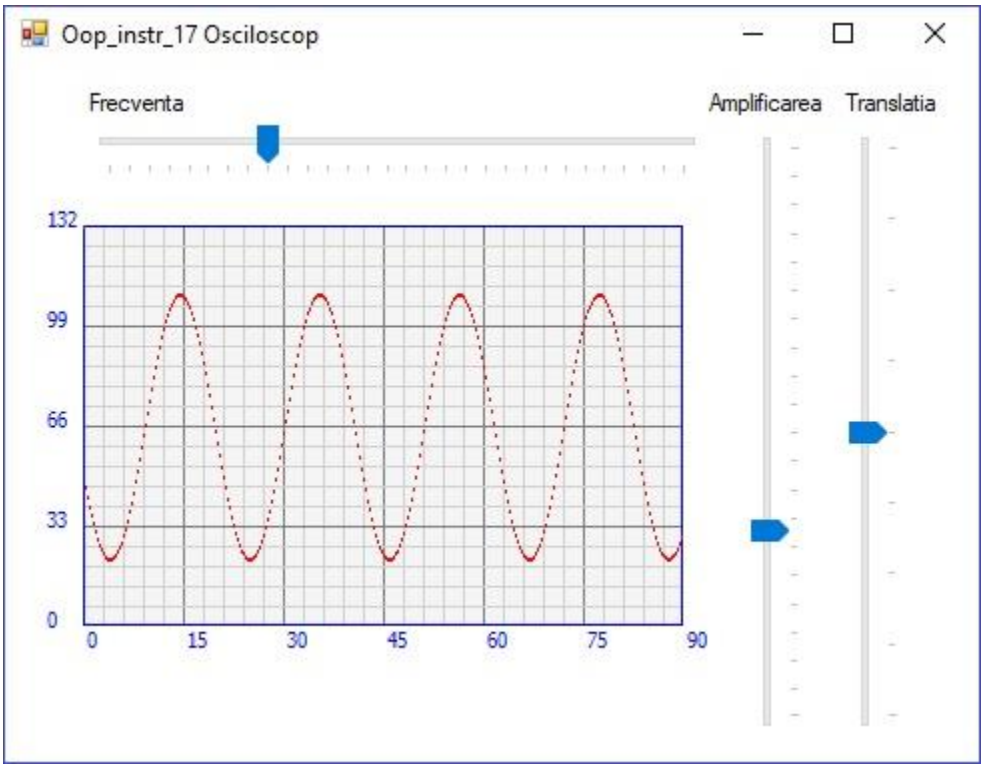
```
    int pozx = 40, pozy = 80, n_maxx = 300, n_maxy = 200;
    double alfa = 0;
    double fi = 0;
    static int[] valori = new int[0];

    private void Form1_Load(object sender, EventArgs e)
    {
        Array.Resize(ref valori, n_maxx + 1);
        desen = this.CreateGraphics();
        osciloscop1 = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy);

    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        this.trackBar1.Maximum = n_maxy / 5;
        this.trackBar1.Minimum = -n_maxy / 5;
        this.trackBar1.Value = 0;
        this.trackBar2.Maximum = n_maxy;
        this.trackBar2.Value = n_maxy / 3;
        this.trackBar3.Minimum = 1;
        this.trackBar3.Value = 14;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        fi = fi + 0.3;
        alfa = fi;
        int transl = -this.trackBar1.Value;
        int amplif = this.trackBar2.Value;
        int fr = this.trackBar3.Value;
        int zero = n_maxy / 2;
        for (int i = 1; i <= n_maxx; i++)
        {
            alfa += System.Convert.ToDouble(fr) / 100;
            int f = System.Convert.ToInt32(transl + zero - amplif * Math.Sin(alfa));
            if ((f < n_maxy) && (f >= 0))
                valori[i] = f;
            if (f > n_maxy)
                valori[i] = n_maxy - 1;
            if (f < 0)
                valori[i] = 0;
        }

        osciloscop1.setval(valori, n_maxx, amplif, fr);
    }
}

// --- Clasa osciloscop --------

public class osciloscop
{
    int x0;
```

```
    int y0;
    int w;
    int h;
    int val_max;
    int nr_max;
    System.Drawing.Graphics zona_des;
    System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
    System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
    System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
    System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

    System.Drawing.Bitmap img;
    System.Drawing.Bitmap ims;

    public void setval(int[] vals, int nrv, int ampl, int fr)
    {
      img = new Bitmap(ims);
      int val, i;

      // afisare grafic sub forma de puncte

      for (i = 0; i < w; i++)
      {
        val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
        img.SetPixel(i, val, System.Drawing.Color.Red);
        if (val < h - 1)
          img.SetPixel(i, val + 1, System.Drawing.Color.Red);
      }
      zona_des.DrawImage(img, x0, y0);
      zona_des.FillRectangle(radiera, x0, y0 + h, w + 20, 20);

      // Afisare valoi axa x

      for (i = 0; i <= w; i += 50)
      {
        val = System.Convert.ToInt16(System.Convert.ToDouble(i * fr / 30) *
(System.Convert.ToDouble(nr_max) / System.Convert.ToDouble(w))); //scalare
        zona_des.DrawString(val.ToString(), font_ni, pens_blu, x0 + i, y0 + h);
      }

      // Afisare valoi axa y

      zona_des.FillRectangle(radiera, x0 - 20, y0 - 10, 20, h + 20);
      for (i = 0; i <= h; i += 50)
      {
        val = System.Convert.ToInt16(System.Convert.ToDouble(i * ampl / 100) *
(System.Convert.ToDouble(val_max) / System.Convert.ToDouble(h))); //scalare
        zona_des.DrawString(val.ToString(), font_ni, pens_blu, x0 - 20, y0 + h - i - 10);
      }
    }
```

```
// Clasa osciloscop

public osciloscop(System.Drawing.Graphics desen, int pozx, int pozy, int n_maxx, int n_maxy)
{
   x0 = pozx;
   y0 = pozy;
   w = n_maxx;
   h = n_maxy;
   nr_max = n_maxx;
   val_max = n_maxy;
   zona_des = desen;
   int i, j;
   ims = new Bitmap(nr_max, n_maxy, zona_des);

   // sterg imaginea

   for (j = 0; j < val_max; j++)
   {
      for (i = 0; i < nr_max; i++)
      {
         ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
      }
   }
   // grid
   for (j = 0; j < val_max; j++)
   {

      // grid orizontal


      if (j % 10 == 0)
      {
         for (i = 0; i < nr_max; i++)
         {
            if (j % 50 == 0)
               ims.SetPixel(i, j, System.Drawing.Color.Gray);
            else
               ims.SetPixel(i, j, System.Drawing.Color.LightGray);
         }
      }
      else
      {

         // grid orizontal vertical

         for (i = 0; i < nr_max; i++)
         {
            if (i % 10 == 0)
            {
               if (i % 50 == 0)
                  ims.SetPixel(i, j, System.Drawing.Color.Gray);
               else
```

```
                    ims.SetPixel(i, j, System.Drawing.Color.LightGray);
                }
            }
        }
    }

    //chenar

    for (i = 0; i < n_maxx; i++)
    {
        ims.SetPixel(i, 0, System.Drawing.Color.Blue);
        ims.SetPixel(i, val_max - 1, System.Drawing.Color.Blue);
    }
    for (j = 0; j < val_max; j++)
    {
        ims.SetPixel(0, j, System.Drawing.Color.Blue);
        ims.SetPixel(nr_max - 1, j, System.Drawing.Color.Blue);
    }
  }
 }
 // --------- Gata clasa osciloscop ----------
}
```

Din cauza afisarii sub forma de puncte calitatea imaginii scade dar viteza este mult mai mare.



Optimizare clasa osciloscop

Modificam clasa osciloscop astfel incat sa pastram afisarea sub forma de puncte dar adaugam puncte suplimentare astfel incat graficul sa para continuu. De asemenea, din cauza faprului ca se afiseaza coordonate dinamice, trebuie stearsa zona in care se afiseaza coordonatele dupa care se afisez din nou. Daca nu se sterge zona, coordonatele se suprascriu, si devin neinteligibile. Din cauza acestei operatii coordonatele "clipesc" cand se afiseaza din cauza stergerii anterioare. Vom incerca sa le scriem direct pe imaginea cu graficul, evitand astfel fenomenul de clipire.

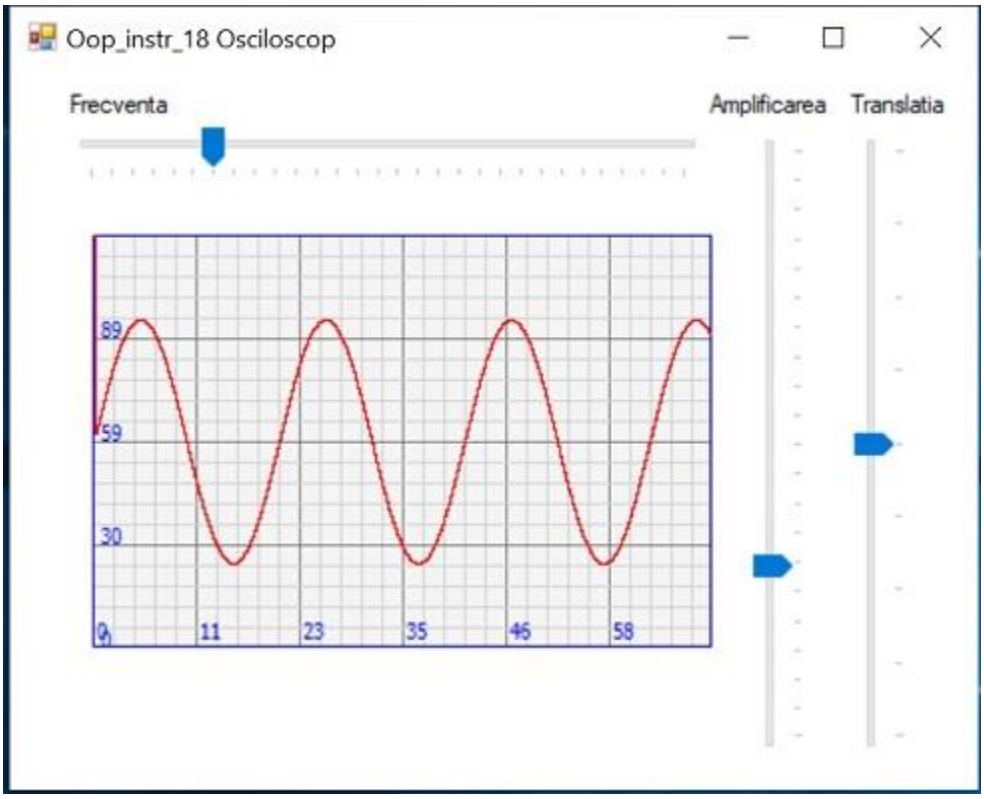
```
namespace Oop_instr_18
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public osciloscop osciloscop1;
      int pozx = 40, pozy = 80, n_maxx = 300, n_maxy = 200;
      double alfa = 0;
      double fi = 0;
      static int[] valori = new int[0];

      private void Form1_Load(object sender, EventArgs e)
      {
         Array.Resize(ref valori, n_maxx + 1);
         desen = this.CreateGraphics();
         osciloscop1 = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         this.trackBar1.Maximum = n_maxy / 5;
         this.trackBar1.Minimum = -n_maxy / 5;
         this.trackBar1.Value = 0;
         this.trackBar2.Maximum = n_maxy;
         this.trackBar2.Value = n_maxy / 3;
         this.trackBar3.Minimum = 1;
         this.trackBar3.Value = 14;
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         fi = fi + 0.3;
         alfa = fi;
         alfa = 0;
         int transl = -this.trackBar1.Value;
```

```
        int amplif = this.trackBar2.Value;
        int fr = this.trackBar3.Value;
        int zero = n_maxy / 2;
        for (int i = 1; i <= n_maxx; i++)
        {
           alfa += System.Convert.ToDouble(fr) / 100;
           int f = System.Convert.ToInt32(transl + zero - amplif * Math.Sin(alfa));
           if ((f < n_maxy) && (f >= 0))
              valori[i] = f;
           if (f > n_maxy)
              valori[i] = n_maxy - 1;
           if (f < 0)
              valori[i] = 0;
        }

        osciloscop1.setval(valori, n_maxx, amplif, fr);

     }
  }

  // --------------Clasa osciloscop ------------------

  public class osciloscop
  {
     int x0;
     int y0;
     int w;
     int h;
     int val_max;
     int nr_max;
     System.Drawing.Graphics zona_des;
     System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
     System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
     System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
     System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

     System.Drawing.Bitmap img;
     System.Drawing.Bitmap ims;

     public void setval(int[] vals, int nrv, int ampl, int fr)
     {
        img = new Bitmap(ims);
        int val,val_v, i, j;
        val_v = System.Convert.ToInt16(System.Convert.ToDouble(vals[0]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max)));

        // afisare grafic sub forma de puncte unite cu linii

        for (i = 0; i < w; i++)
        {
           val = System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) * (System.Convert.ToDouble(h) /
System.Convert.ToDouble(val_max))); //scalare
```

```
            if (val_v < val)
            {
               // unesc doua puncte cu o linie crescatoare
               for (j = val_v; j <= val; j++)
                  img.SetPixel(i, j, System.Drawing.Color.Red);
            }
            else
            {
               // unesc doua puncte cu o linie descrescatoare
               for (j = val; j <= val_v; j++)
                  img.SetPixel(i, j, System.Drawing.Color.Red);

            }
            val_v = val;
         }

         // gata afisare grafic sub forma de puncte

         // Afisare valori axa x

         Graphics g = Graphics.FromImage(img);

         for (i = 0; i <= w; i += 50)
         {
            val = System.Convert.ToInt16(System.Convert.ToDouble(i * fr / 30) *
(System.Convert.ToDouble(nr_max) / System.Convert.ToDouble(w))); //scalare
            g.DrawString(val.ToString(), font_ni, pens_blu, i, h-15);
         }

         // Afisare valori axa y


         for (i = 1; i <= h; i += 50)
         {
            val = System.Convert.ToInt16(System.Convert.ToDouble(i * ampl / 100) *
(System.Convert.ToDouble(val_max) / System.Convert.ToDouble(h))); //scalare
            g.DrawString(val.ToString(), font_ni, pens_blu, 2, h - i - 10);
         }
         zona_des.DrawImage(img, x0, y0);
      }
      public osciloscop(System.Drawing.Graphics desen, int pozx, int pozy, int n_maxx, int n_maxy)
      {
         x0 = pozx;
         y0 = pozy;
         w = n_maxx;
         h = n_maxy;
         nr_max = n_maxx;
         val_max = n_maxy;
         zona_des = desen;
         int i, j;
         ims = new Bitmap(nr_max, n_maxy, zona_des);
```

```
// sterg imaginea

for (j = 0; j < val_max; j++)
{
   for (i = 0; i < nr_max; i++)
   {
      ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
   }
}

// grid

for (j = 0; j < val_max; j++)
{

   // grid orizontal


   if (j % 10 == 0)
   {
      for (i = 0; i < nr_max; i++)
      {
         if (j % 50 == 0)
            ims.SetPixel(i, j, System.Drawing.Color.Gray);
         else
            ims.SetPixel(i, j, System.Drawing.Color.LightGray);
      }
   }
   else
   {

      // grid orizontal vertical

      for (i = 0; i < nr_max; i++)
      {
         if (i % 10 == 0)
         {
            if (i % 50 == 0)
               ims.SetPixel(i, j, System.Drawing.Color.Gray);
            else
               ims.SetPixel(i, j, System.Drawing.Color.LightGray);
         }
      }
   }
}
//chenar

for (i = 0; i < n_maxx; i++)
{
   ims.SetPixel(i, 0, System.Drawing.Color.Blue);
   ims.SetPixel(i, val_max - 1, System.Drawing.Color.Blue);
}
```

```
        for (j = 0; j < val_max; j++)
        {
           ims.SetPixel(0, j, System.Drawing.Color.Blue);
           ims.SetPixel(nr_max - 1, j, System.Drawing.Color.Blue);
        }
     }
  }
  // --------------Sfarsit clasa osciloscop ------------------
}
```



Pe baza experientei acumulate in perfectionarea clasei **osciloscop** sa incercam sa definitivam aceasta clasa astfel incat ea sa poata fi folosita in general pentru afisarea unei functii f(x) oarecare.

Se vor elimina parametrii **fr** si **amplif** acestia fiind un caz particular pentru afisarea unui sinus de amplitudine si frecventa variabila.

Pentru utilizarea mai simpla a acestei clase, se va modifica sensul axei y astfel ca atunci cand se va afisa o functie, nu mai trebuie afisata -functia, inversiunea sensului axei facandu-se in cadrul clasei. Pentru scalare, se vor aduga param **val_max** si **val_min**. Pe baza acestor valori se va calcula **amplif** si **transl**.

In concluzie, pentru utilizarea clasei, utilizatorul trebuie doar sa inscrie valorile penru afisat intr-un vector si sa precizeze valoarea maxima si minima de afisat. V-a trebi sa definim vectorul de tip double avand in vedere ca vectorul nu mai contine coordonata y a punctelor ce urmeaza a fi afisate, acesta continand efectiv valori ale finctiei ce urmeaza a fi reprezentata.

Vom relua deci afisarea functiei sinus dar fara butoanele pentru translatie, amplificare etc.

```
namespace Oop_instr_19
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public osciloscop oscil;
      int pozx = 40, pozy = 10, n_maxx = 300, n_maxy = 200;
      double val_max = 1.5, val_min=-1.5;
      static double[] valori = new double[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         Array.Resize(ref valori, n_maxx + 1);
         oscil = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         double fi = this.trackBar1.Value / 100.0;
         double alfa = 0;
         for (int i = 1; i <= n_maxx; i++)
         {
            alfa += 0.05;
            valori[i] = Math.Sin(alfa - fi);
         }
         oscil.setval(valori, n_maxx, Color.Red);
      }
   }

   // --------------Clasa osciloscop ------------------

   public class osciloscop
   {
      int x0, y0, w, h, val, val_v;
      double v_max, v_min;
      System.Drawing.Graphics zona_des;
      System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
      System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
      System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
      System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

      System.Drawing.Bitmap img;
      System.Drawing.Bitmap ims;

      public void setval(double[] vals, int nrv, System.Drawing.Color cul)
      {
         int i, j;
         img = new Bitmap(ims);
```

```
        // afisare grafic sub forma de puncte
        double amplif=(System.Convert.ToDouble(h) / System.Convert.ToDouble(v_max-v_min));
        double transl = v_min * amplif;
        val_v = System.Convert.ToInt16(h + transl- amplif* System.Convert.ToDouble(vals[0])) ; //scalare
        if (val_v >= h)
           val_v = h - 1;
        if (val_v <= 0)
           val_v = 1;
        for (i = 0; i < w; i++)
        {
           val = System.Convert.ToInt16(h + transl - amplif * System.Convert.ToDouble(vals[i]));  //scalare
           if (val >= h)
              val = h - 1;
           if (val <= 0)
              val = 1;
           if (val_v < val)
           {
              // unesc doua puncte cu o linie crescatoare
              for (j = val_v; j <= val; j++)
              {
                 img.SetPixel(i, j, cul);

              }
           }
           else
           {
              // unesc doua puncte cu o linie descrescatoare
              for (j = val; j <= val_v; j++)
              {
                 img.SetPixel(i, j, cul);
              }
           }
           val_v = val;
        }
        Graphics g = Graphics.FromImage(img);

        //valori axa x

        for (i = 50; i <= w; i += 50)
        {
           val = System.Convert.ToInt16(System.Convert.ToDouble(i) * (System.Convert.ToDouble(w) /
System.Convert.ToDouble(w))); //scalare
           g.DrawString(val.ToString(), font_ni, pens_blu, i, h - 15);
        }

        //valori axa y

        double vg = v_min;
        double pas = System.Convert.ToDouble(v_max-v_min) / System.Convert.ToDouble(h) * 50;

        for (i = 50; i < h; i += 50)
```

```
        {
            vg = vg + pas;
            val = System.Convert.ToInt16(System.Convert.ToDouble(i) * (System.Convert.ToDouble(v_max-v_min) /
System.Convert.ToDouble(h))); //scalare
            g.DrawString(Math.Round(vg, 2).ToString(), font_ni, pens_blu, 2, h - i - 10);
        }
        zona_des.DrawImage(img, x0, y0);
    }
    public osciloscop(System.Drawing.Graphics desen, int pozx, int pozy, int n_maxx, int n_maxy, double
val_max, double val_min)
    {
        x0 = pozx;
        y0 = pozy;
        w = n_maxx;
        h = n_maxy;
        v_max = val_max;
        v_min = val_min;
        zona_des = desen;
        img = new Bitmap(w, h, zona_des);
        ims = new Bitmap(w, h, zona_des);
        int i, j;

        // sterg imaginea

        for (j = 0; j < h; j++)
        {
            for (i = 0; i < w; i++)
            {
                ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
            }
        }
        // grid
        for (j = 0; j < h; j++)
        {
            // grid orizontal

            if (j % 10 == 0)
            {
                for (i = 0; i < w; i++)
                {
                    if (j % 50 == 0)
                        ims.SetPixel(i, j, System.Drawing.Color.Gray);
                    else
                        ims.SetPixel(i, j, System.Drawing.Color.LightGray);
                }
            }
            else
            {

                // grid orizontal vertical

                for (i = 0; i < w; i++)
```

```
            {
               if (i % 10 == 0)
               {
                  if (i % 50 == 0)
                     ims.SetPixel(i, j, System.Drawing.Color.Gray);
                  else
                     ims.SetPixel(i, j, System.Drawing.Color.LightGray);
               }
            }
         }
      }

      //chenar

      for (i = 0; i < w; i++)
      {
         ims.SetPixel(i, 0, System.Drawing.Color.Blue);
         ims.SetPixel(i, h - 1, System.Drawing.Color.Blue);
      }
      for (j = 0; j < h; j++)
      {
         ims.SetPixel(0, j, System.Drawing.Color.Blue);
         ims.SetPixel(w - 1, j, System.Drawing.Color.Blue);
      }
   }
}
// --------------Sfarsit clasa osciloscop ------------------
}
```

Dupa cum se observa utilizatorul nu trebuie decat sa creeze obiectul **oscil**
oscil = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
dupa care inscrie in vectorul valori[] valorile ce urmeaza a fi afisate si sa invoce metoda **setval** .

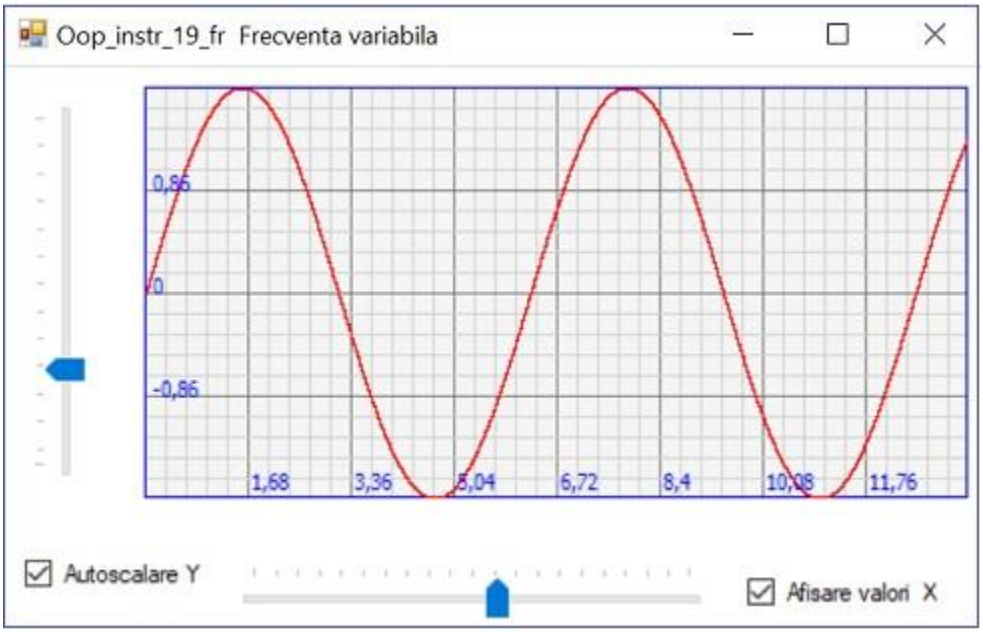
```
   private void timer1_Tick(object sender, EventArgs e)
   {
      double fi = this.trackBar1.Value / 100.0;
      double alfa = 0;
      for (int i = 1; i <= n_maxx; i++)
      {
         alfa += 0.05;
         valori[i] = Math.Sin(alfa - fi);
      }
      oscil.setval(valori, n_maxx, Color.Red);
   }
```

Autoscalare

In exemplul anterior, utilizatorul trebuia sa precizeze **val_max** si **val_min**. Avand valorile pentru afisat inscrise in vector, teoretic clasa nu ar mai avea nevoie de **val_max** si **val_min**, acestea putand fi calculate, insa daca dorim sa perfectionam in continuare clasa si sa afisam mai multe canale, utilizatorul trebuie sa decida cm vrea sa arate imaginea de ansamblu si deci stabileste **val_max** si **val_min**. Pentru a intruni aeste cerinte vom adauga o metoda pentru autoscalare, care va putea fi apelata numai daca utilizatorul doreste acest lucru.

In exemplul anterior pe coordonata x nu se afiseaza valoarea lui x, se afiseaza numarul curent al pozitiei pixelului. Pentru afisarea efectiva a valorii lui x va trebui sa mai introducem o metoda, astfel putem afisa numarul de ordine sau x.

In aplicatia urmatoare vom afisa un sinus de amplitudine si frecventa variabila si vom introduce metodele amintite.




```
namespace Oop_instr_19_fr
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public osciloscop oscil;
    int pozx = 70, pozy = 10, n_maxx = 400, n_maxy = 200;
    double val_max = 1.5, val_min = -1.5;
    static double[] valori = new double[0];
    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      Array.Resize(ref valori, n_maxx + 1);
      oscil = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
```

```
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        double a = this.trackBar2.Value / 100.0;
        double f = this.trackBar1.Value / 100.0;
        double alfa = 0, min, max;
        double pas=0.03;
        for (int i = 1; i <= n_maxx; i++)
        {
            alfa += pas;
            valori[i] = a*Math.Sin(f*alfa);

        }
        min = valori[0];
        max = valori[0];
        for (int i = 1; i <= n_maxx; i++)
        {
          if(valori[i] > max)
              max = valori[i];
          if (valori[i] < min)
              min = valori[i];
        }
        if(this.checkBox1.Checked)
            oscil.auto_sy( max,min);
        else
            oscil.auto_sy(val_max, val_min);
        if (this.checkBox2.Checked)
            oscil.auto_sx(f*pas*n_maxx, 0);
        else
            oscil.auto_sx(n_maxx, 0);
        oscil.setval(valori, n_maxx, Color.Red);
    }
}
// --------------Clasa osciloscop ------------------
public class osciloscop
{
    int x0, y0, w, h, val, val_v;
    double v_max, v_min, x_max, x_min;
    System.Drawing.Graphics zona_des;
    System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
    System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
    System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
    System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

    System.Drawing.Bitmap img;
    System.Drawing.Bitmap ims;

    public void auto_sy(double val_max, double val_min)
    {
        if (val_max - val_min != 0)
        {
```

```
            v_max = val_max;
            v_min = val_min;
        }
    }
    public void auto_sx(double x_maxim, double x_minim)
    {
        if (x_max - x_min != 0)
        {
            x_max = x_maxim;
            x_min = x_minim;
        }
        else
        {
            x_max = w;
            x_min = 0;
        }
    }
    public void setval(double[] vals, int nrv, System.Drawing.Color cul)
    {
        int i, j;
        if (w > 0 && h > 0)
        {
            img = new Bitmap(ims);

            // afisare grafic sub forma de puncte
            double amplif;
            if ((v_max - v_min) != 0)
                amplif = (System.Convert.ToDouble(h) / System.Convert.ToDouble(v_max - v_min));
            else
                amplif = 1;
            double transl = v_min * amplif;
            val_v = System.Convert.ToInt16(h + transl - amplif * System.Convert.ToDouble(vals[0])); //scalare
            if (val_v >= h)
                val_v = h - 1;
            if (val_v <= 0)
                val_v = 1;
            for (i = 0; i < w; i++)
            {
                val = System.Convert.ToInt16(h + transl - amplif * System.Convert.ToDouble(vals[i]));  //scalare
                if (val >= h)
                    val = h - 1;
                if (val <= 0)
                    val = 1;
                if (val_v < val)
                {
                    // unesc doua puncte cu o linie crescatoare
                    for (j = val_v; j <= val; j++)
                    {
                        img.SetPixel(i, j, cul);

                    }
                }
```

```
            else
            {
               // unesc doua puncte cu o linie descrescatoare
               for (j = val; j <= val_v; j++)
               {
                  img.SetPixel(i, j, cul);
               }
            }
            val_v = val;
         }
         Graphics g = Graphics.FromImage(img);

         //valori axa x

         double vx = x_min;
         double pasx = System.Convert.ToDouble(x_max - x_min) / System.Convert.ToDouble(w) * 50;
         for (i = 50; i < w; i += 50)
         {
            vx = vx + pasx;
            g.DrawString(Math.Round(vx, 2).ToString(), font_ni, pens_blu, i, h - 15);

         }

         //valori axa y

         double vy = v_min;
         double pasy = System.Convert.ToDouble(v_max - v_min) / System.Convert.ToDouble(h) * 50;
         for (i = 50; i < h; i += 50)
         {
            vy = vy + pasy;
            g.DrawString(Math.Round(vy, 2).ToString(), font_ni, pens_blu, 2, h - i - 10);
         }
         zona_des.DrawImage(img, x0, y0);
      }
   }
   public osciloscop(System.Drawing.Graphics desen, int pozx, int pozy, int n_maxx, int n_maxy, double
val_max, double val_min)
   {
      x0 = pozx;
      y0 = pozy;
      w = n_maxx;
      h = n_maxy;
      v_max = val_max;
      v_min = val_min;
      x_max = n_maxx;
      x_min = 0;
      zona_des = desen;
      if (w > 0 && h > 0)
      {
         img = new Bitmap(w, h, zona_des);
         ims = new Bitmap(w, h, zona_des);
         int i, j;
```

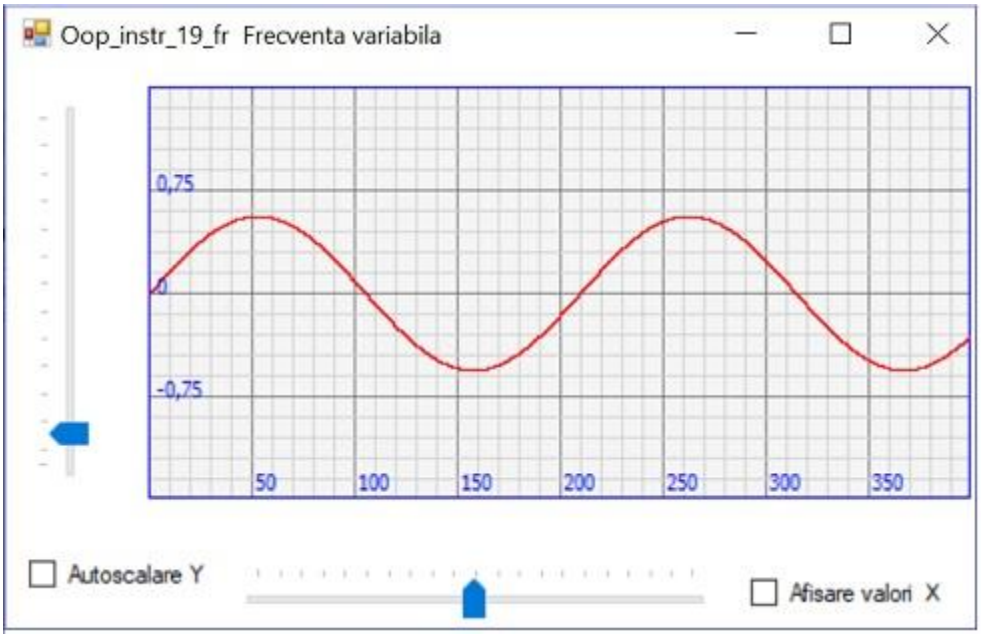
```
// sterg imaginea

for (j = 0; j < h; j++)
{
   for (i = 0; i < w; i++)
   {
      ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
   }
}
// grid
for (j = 0; j < h; j++)
{
   // grid orizontal

   if (j % 10 == 0)
   {
      for (i = 0; i < w; i++)
      {
         if (j % 50 == 0)
            ims.SetPixel(i, j, System.Drawing.Color.Gray);
         else
            ims.SetPixel(i, j, System.Drawing.Color.LightGray);
      }
   }
   else
   {

      // grid orizontal vertical

      for (i = 0; i < w; i++)
      {
         if (i % 10 == 0)
         {
            if (i % 50 == 0)
               ims.SetPixel(i, j, System.Drawing.Color.Gray);
            else
               ims.SetPixel(i, j, System.Drawing.Color.LightGray);
         }
      }
   }
}

//chenar

for (i = 0; i < w; i++)
{
   ims.SetPixel(i, 0, System.Drawing.Color.Blue);
   ims.SetPixel(i, h - 1, System.Drawing.Color.Blue);
}
for (j = 0; j < h; j++)
{
```

```
            ims.SetPixel(0, j, System.Drawing.Color.Blue);
            ims.SetPixel(w - 1, j, System.Drawing.Color.Blue);
          }
        }
      }
    }
    // ----------Sfarsit clasa osciloscop ----------
}
```

S-au adaugat deci cele doua metode:

```
    public void auto_sy(double val_max, double val_min)
    {
      if (val_max - val_min != 0)
      {
        v_max = val_max;
        v_min = val_min;
      }
    }
    public void auto_sx(double x_maxim, double x_minim)
    {
      if (x_max - x_min != 0)
      {
        x_max = x_maxim;
        x_min = x_minim;
      }
      else
      {
        x_max = w;
        x_min = 0;
      }
    }
```

In cazul in care nu se doreste autoscalare si afisare x, se obtine:



Afisare in coordonate x-t pe mai multe canale

In principiu putem folosi clasa osciloscop cu mici modificari. Dupa ce am trasat graficul unei functii, prin metoda **set_val** se face implicit stergere continut, creare imagine grafic, si afisare grafic. Pentru a putea afisa mai multe canale, trebuie sa despartim metoda **set_val** in metodele: **sterg()** , **set_val** , **display()** . Astfel vom apela metoda **sterg()** , dupa care metoda **set_val** pentru fiecare canal in parte si la sfarsit metoda **display()** care afiseaza toate canalele simultan.

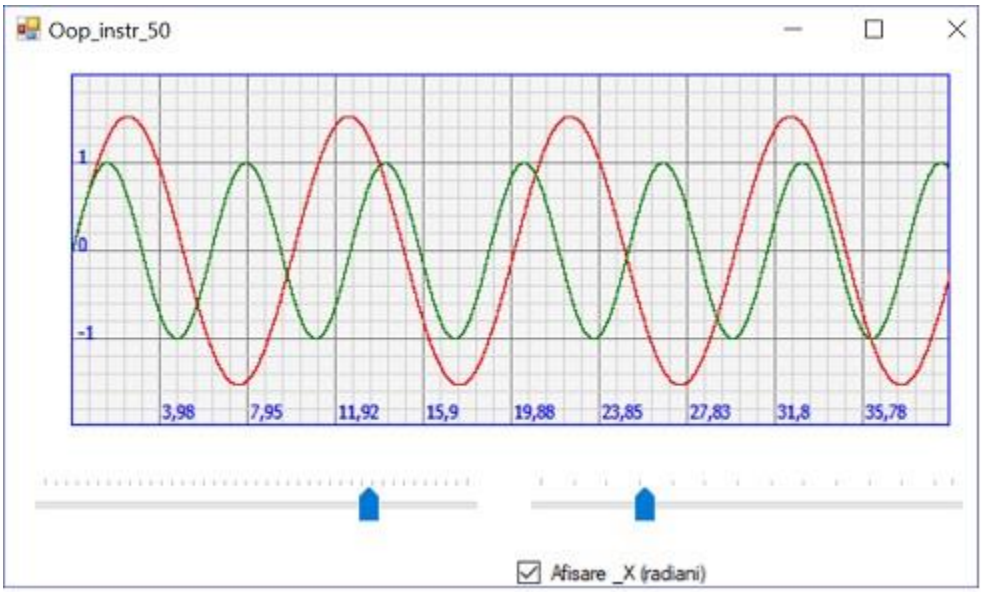
```
namespace Oop_instr_50
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    public osciloscop oscil_m;
    int pozx = 40, pozy = 10, n_maxx = 500, n_maxy = 200;
    double val_max = 2, val_min=-2;
    static double[] valori1 = new double[0];
    static double[] valori2 = new double[0];

    private void Form1_Load(object sender, EventArgs e)
    {
      desen = this.CreateGraphics();
      Array.Resize(ref valori1, n_maxx + 1);
      Array.Resize(ref valori2, n_maxx + 1);
      oscil_m = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
```

```
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        double a = this.trackBar1.Value / 100.0;
        double k = this.trackBar2.Value/100.0;
        double alfa = 0;
        double pas = 0.05;
        for (int i = 1; i <= n_maxx; i++)
        {
            alfa += pas;
            valori1[i] = a*Math.Sin(alfa);
            valori2[i] = Math.Sin(k*alfa);
        }
        oscil_m.sterg();
        if (this.checkBox1.Checked)
            oscil_m.auto_sx(pas*k*n_maxx, 0);
        else
            oscil_m.auto_sx(n_maxx, 0);
        oscil_m.setval(valori1, n_maxx, Color.Red);
        oscil_m.setval(valori2, n_maxx, Color.Green);
        oscil_m.display();
    }
}
// ----------Clasa osciloscop ------------------
public class osciloscop
{
    int x0, y0, w, h, val, val_v;
    double v_max, v_min, x_max, x_min;
    System.Drawing.Graphics zona_des;
    System.Drawing.Pen creion_r = new System.Drawing.Pen(System.Drawing.Color.Red);
    System.Drawing.Font font_ni = new System.Drawing.Font("Nina", 8);
    System.Drawing.SolidBrush pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
    System.Drawing.SolidBrush radiera = new System.Drawing.SolidBrush(System.Drawing.Color.White);

    System.Drawing.Bitmap img;
    System.Drawing.Bitmap ims;
    public void sterg()
    {
        img = new Bitmap(ims);
    }
    public void display()
    {
        zona_des.DrawImage(img, x0, y0);
    }
    public void auto_sy(double val_max, double val_min)
    {
        if (val_max - val_min != 0)
        {
            v_max = val_max;
            v_min = val_min;
        }
    }
```

```
public void auto_sx(double x_maxim, double x_minim)
{
   if (x_max - x_min != 0)
   {
      x_max = x_maxim;
      x_min = x_minim;
   }
   else
   {
      x_max = w;
      x_min = 0;
   }
}
public void setval(double[] vals, int nrv, System.Drawing.Color cul)
{
   int i, j;
   if (w > 0 && h > 0)
   {
      // afisare grafic sub forma de puncte
      double amplif;
      if ((v_max - v_min) != 0)
         amplif = (System.Convert.ToDouble(h) / System.Convert.ToDouble(v_max - v_min));
      else
         amplif = 1;
      double transl = v_min * amplif;
      val_v = System.Convert.ToInt16(h + transl - amplif * System.Convert.ToDouble(vals[0])); //scalare
      if (val_v >= h)
         val_v = h - 1;
      if (val_v <= 0)
         val_v = 1;
      for (i = 0; i < w; i++)
      {
         val = System.Convert.ToInt16(h + transl - amplif * System.Convert.ToDouble(vals[i]));  //scalare
         if (val >= h)
            val = h - 1;
         if (val <= 0)
            val = 1;
         if (val_v < val)
         {
            // unesc doua puncte cu o linie crescatoare
            for (j = val_v; j <= val; j++)
            {
               img.SetPixel(i, j, cul);

            }
         }
         else
         {
            // unesc doua puncte cu o linie descrescatoare
            for (j = val; j <= val_v; j++)
            {
               img.SetPixel(i, j, cul);
```

```
                }
            }
            val_v = val;
        }
        Graphics g = Graphics.FromImage(img);

        //valori axa x

        double vx = x_min;
        double pasx = System.Convert.ToDouble(x_max - x_min) / System.Convert.ToDouble(w) * 50;
        for (i = 50; i < w; i += 50)
        {
            vx = vx + pasx;
            g.DrawString(Math.Round(vx, 2).ToString(), font_ni, pens_blu, i, h - 15);

        }

        //valori axa y

        double vy = v_min;
        double pasy = System.Convert.ToDouble(v_max - v_min) / System.Convert.ToDouble(h) * 50;
        for (i = 50; i < h; i += 50)
        {
            vy = vy + pasy;
            g.DrawString(Math.Round(vy, 2).ToString(), font_ni, pens_blu, 2, h - i - 10);
        }
    }
}
public osciloscop(System.Drawing.Graphics desen, int pozx, int pozy, int n_maxx, int n_maxy, double
val_max, double val_min)
{
    x0 = pozx;
    y0 = pozy;
    w = n_maxx;
    h = n_maxy;
    v_max = val_max;
    v_min = val_min;
    x_max = n_maxx;
    x_min = 0;
    zona_des = desen;
    if (w > 0 && h > 0)
    {
        img = new Bitmap(w, h, zona_des);
        ims = new Bitmap(w, h, zona_des);
        int i, j;

        // sterg imaginea

        for (j = 0; j < h; j++)
        {
            for (i = 0; i < w; i++)
            {
```

```
                ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
            }
        }
        // grid
        for (j = 0; j < h; j++)
        {
            // grid orizontal

            if (j % 10 == 0)
            {
                for (i = 0; i < w; i++)
                {
                    if (j % 50 == 0)
                        ims.SetPixel(i, j, System.Drawing.Color.Gray);
                    else
                        ims.SetPixel(i, j, System.Drawing.Color.LightGray);
                }
            }
            else
            {

                // grid orizontal vertical

                for (i = 0; i < w; i++)
                {
                    if (i % 10 == 0)
                    {
                        if (i % 50 == 0)
                            ims.SetPixel(i, j, System.Drawing.Color.Gray);
                        else
                            ims.SetPixel(i, j, System.Drawing.Color.LightGray);
                    }
                }
            }
        }
        //chenar
        for (i = 0; i < w; i++)
        {
            ims.SetPixel(i, 0, System.Drawing.Color.Blue);
            ims.SetPixel(i, h - 1, System.Drawing.Color.Blue);
        }
        for (j = 0; j < h; j++)
        {
            ims.SetPixel(0, j, System.Drawing.Color.Blue);
            ims.SetPixel(w - 1, j, System.Drawing.Color.Blue);
        }
      }
    }
  }
  // ----------Sfarsit clasa osciloscop ----------
}
```



Utilizarea clasei "osciloscop" la afisarea diverselor functii matematice

Pentru a folosi clasa osciloscop la afisarea unei functii matematice oarecare, trebuie sa mai ajustam un pic clasa pentru a evita cazurile in care valoarea functiei depaseste dimensiunea unui Int16 si la conversia vlorii in Int16 in vederea afisarii grafice sa avem depasire. Vom folosi "System.OverflowException" pentru a determina daca val depaseste un Int16.

Sa incercam sa afisam grafic functiile matematice f(x)=tg(x)*sin(x) si f(x)= sin(x) pe acelasi grafic.

Daca stabilim domeniul in care apartine x ca fiind: [-6,6], pentru anumite valori functi tg(x) are valori infinite insa captand eroarea "System.OverflowException" vom evita mesaje de eroare prin setarea valorii la depasire, la valoarea maxima de exemplu.

Tot ce trebuie sa facem pentru afisarea functiei e sa tabilim domeniul in care apartine x si valoarea maxima, respectiv valoarea minima ce urmeaza a fi afisata pe grafic.

definim deci

```
osciloscop oscil_m;
int n_maxx ; //numarul de puncte pe x
int n_maxy ; //numarul de puncte pe y
double val_max = 5; //valoarea maxima
double val_min = -5; //valoarea minima
double x0 = -6, x1 = 6;
double[] valori = new double[0];
```

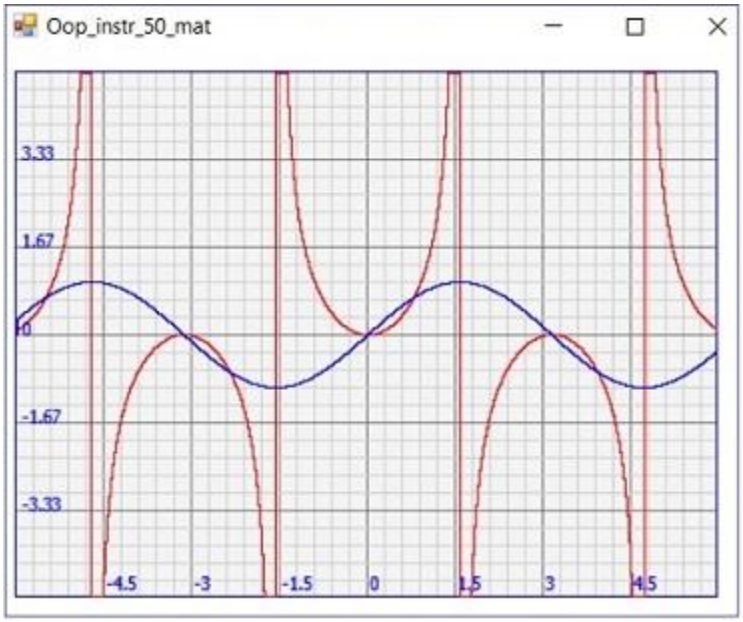
Inaintea afisarii vom inscrie in vectorul valori, valorile ce urmeaza a fi afisate grafic cu o secventa de instructiuni de genul:

```
double pas = (x1 - x0) / n_maxx;
double x = x0;
for (int i = 1; i <= n_maxx; i++)
{
   x = x + pas;
   valori[i] = Math.Tan(x) * Math.Sin(x);
}
```

Dupa care sa afisam grafic valorile:

```
oscil_m.sterg();
oscil_m.auto_sx(x1, x0);
oscil_m.setval(valori, n_maxx, Color.Red);
oscil_m.display();
```

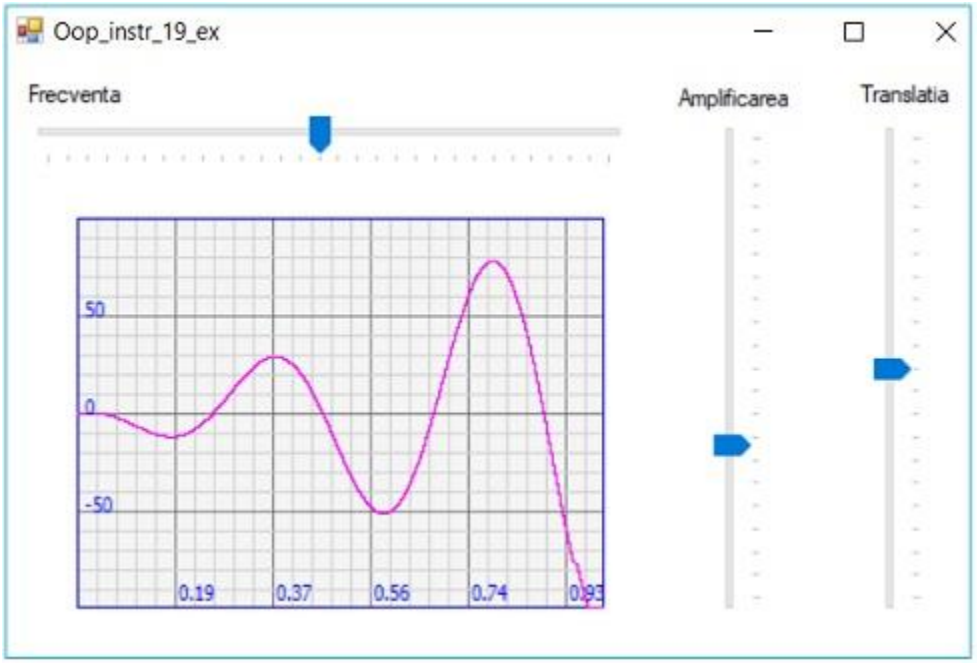
```
namespace Oop_instr_50_mat
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    System.Drawing.Graphics desen;
    public osciloscop oscil_m;
    int pozx = 10, pozy = 10, n_maxx = 400, n_maxy = 300;
    double val_max = 5; //valoarea maxima
    double val_min = -5; //valoarea minima
    double x0 = -6, x1 = 6;
    double[] valori = new double[0];
    double[] valoris = new double[0];
    private void Form1_Load(object sender, EventArgs e)
    {
      Array.Resize(ref valori, n_maxx + 1);
      Array.Resize(ref valoris, n_maxx + 1);
      desen = this.CreateGraphics();
      oscil_m = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
      double pas = (x1 - x0) / n_maxx;
      double x = x0;
      for (int i = 1; i <= n_maxx; i++)
      {
        x = x + pas;
        valori[i] = Math.Tan(x) * Math.Sin(x);
        valoris[i] = Math.Sin(x);
      }
      oscil_m.sterg();
      oscil_m.auto_sx(x1, x0);
      oscil_m.setval(valori, n_maxx, Color.Red);
      oscil_m.setval(valoris, n_maxx, Color.Blue);
      oscil_m.display();
    }
  }
}
```



Utilizarea clasei osciloscop

Utilizarea clasei "osciloscop" pentru afisarea functiei (1-e^x)*sin(k*x)

Folosim acum clasa pentru a afisa o functie mai complexa cum ar fi: graficul functiei (1-e^x)*sin(k*x) pentru k apartine intervalului [0-15] si x apartine intervalului [0-1]

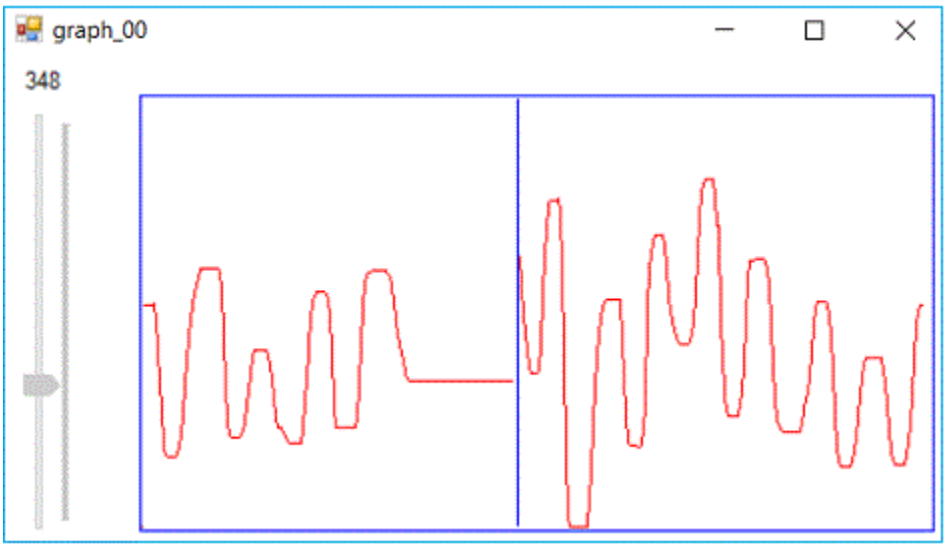


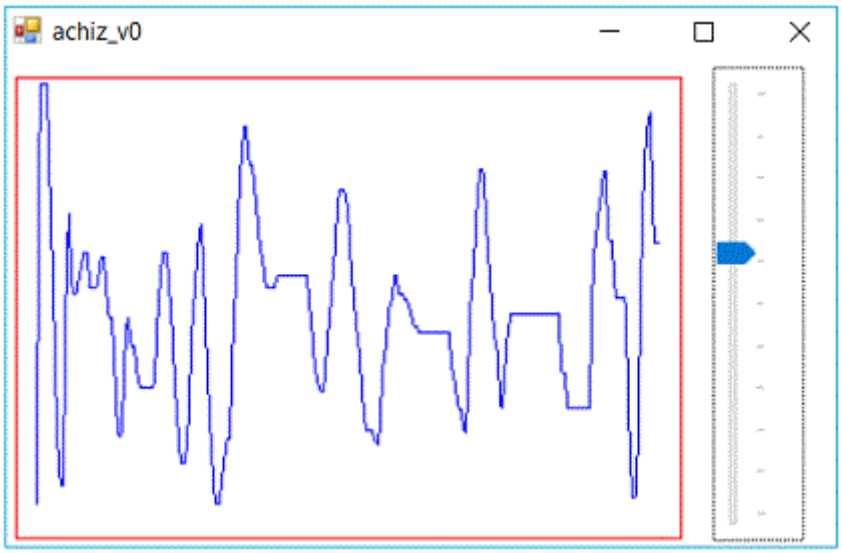
```
namespace Oop_instr_19_ex
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics desen;
      osciloscop instrum1;
      int pozx = 40, pozy = 80, n_maxx = 270, n_maxy = 200;
      double val_min = -100;
      double val_max = 100;
      double x0 = 0;
      double x1 = 1;
      double alfa = 0;
      double fi = 0;
      double[] valori = new double[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         Array.Resize(ref valori, n_maxx + 1);
         desen = this.CreateGraphics();
         instrum1 = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
      }

      private void Form1_Paint(object sender, PaintEventArgs e)
      {
         this.trackBar1.Maximum = System.Convert.ToInt16(val_max);
         this.trackBar1.Minimum = System.Convert.ToInt16(val_min);
         this.trackBar1.Value = 0;
         this.trackBar2.Maximum = n_maxy;
         this.trackBar2.Value = n_maxy / 3;
         this.trackBar3.Minimum = 1;
         this.trackBar3.Value = 15;
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         fi = fi + 0.1;
         alfa = fi;
         int transl = this.trackBar1.Value;
         int amplif = this.trackBar2.Value;
         double pas = (x1 - x0) / n_maxx;
         double x = x0;
         for (int i = 1; i <= n_maxx; i++)
         {
            x = x + pas;
            if (x > 1)
```

```
            x = 0;
          int k = System.Convert.ToInt32(this.trackBar3.Value);
          valori[i] = transl + amplif * (Math.Sin(k * x + fi) * (1 - Math.Pow(Math.E, x)));
        }
        instrum1.sterg();
        instrum1.auto_sx(x1, x0);
        instrum1.setval(valori, n_maxx, Color.Magenta);
        instrum1.display();
      }
    }
}
```

Utilizarea clasei "osciloscop" pentru a simula un ekg

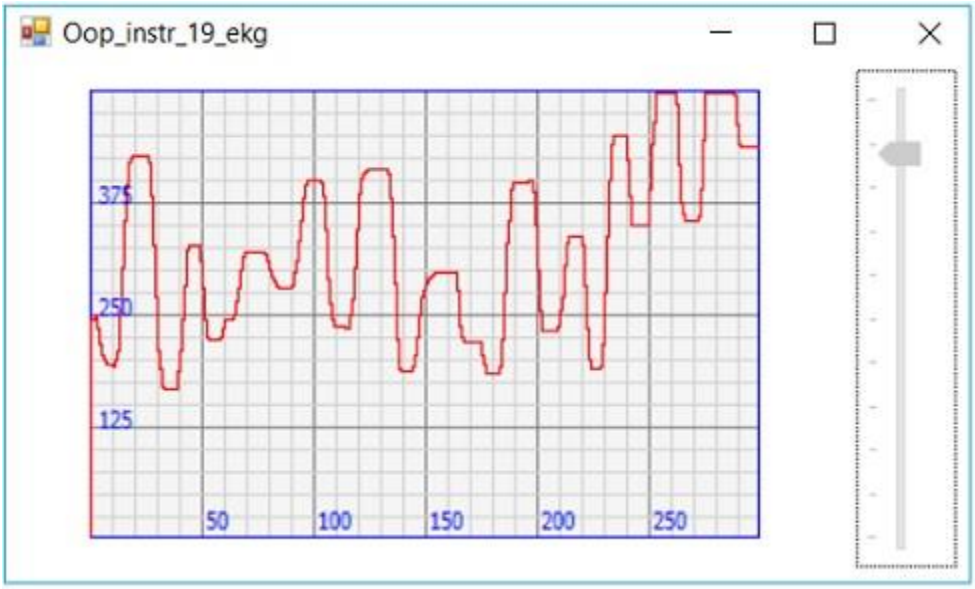
Revenim la aplicatia "achiz_v0" in care se afisa evolutia in timp a unui parametru





Folosind clasa osciloscop, vom crea aplicatia "Oop_instr_19_ekg" pentru a simula un ekg:

```
namespace Oop_instr_19_ekg
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public osciloscop ekg;
      int pozx = 40, pozy = 10, n_maxx = 300, n_maxy = 200;
      double val_max = 500, val_min=0 ;
      static double[] valori = new double[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         Array.Resize(ref valori, n_maxx + 1);
         ekg = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         double f = System.Convert.ToDouble(this.trackBar1.Value);
         for (int i = 0; i < n_maxx - 1; i++)
         {
            valori[i] = valori[i + 1];
         }
         valori[n_maxx - 1] = f;
         ekg.setval(valori, n_maxx, Color.Red);
      }
   }
}
```



Putem simula forma de unda a unui ekg folosind functii matematice. Vom realiza astfel aplicatia: **Oop_instr_19_ekg_s** in care vom simula un ekg.

```
namespace Oop_instr_19_ekg_s
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public osciloscop ekg;
      int pozx = 100, pozy = 10, n_maxx = 500, n_maxy = 200;
      double val_max = 4, val_min = 0;
      static double[] valori = new double[0];
      double a_pwav = 0.25, d_pwav = 0.09, t_pwav = 0.16;
      double a_qwav = 0.025, d_qwav = 0.066, t_qwav = 0.166;
      double li = 0.5, a_qrswav = 1.6, d_qrswav = 0.09;
      double a_swav = 0.25, d_swav = 0.066, t_swav = 0.09;
      double a_twav = 0.35, d_twav = 0.142, t_twav = 0.2;
      double a_uwav = 0.035, d_uwav = 0.0476, t_uwav = 0.433;
      double p_wav(double x, double a_pwav, double d_pwav, double t_pwav, double li)
      {
         double l, a, b, p1, p2, harm1, pwav1, pwav;
         l = li;
         a = a_pwav;
         x = x + t_pwav;
         b = (2 * l) / d_pwav;
         int n = 100;
         p1 = 1 / l;
```

```
        p2 = 0;
        harm1 = 0.0;
        for (int i = 1; i < n; i++)
        {
           harm1 = (((Math.Sin((Math.PI / (2 * b)) * (b - (2 * i)))) / (b - (2 * i)) + (Math.Sin((Math.PI / (2 * b)) * (b + (2
* i)))) / (b + (2 * i))) * (2 / Math.PI)) * Math.Cos((i * Math.PI * x) / l);
           p2 = p2 + harm1;
        }
        pwav1 = p1 + p2;
        pwav = a * pwav1;
        return pwav;
     }
     double qrs_wav(double x, double a_qrswav, double d_qrswav, double li)
     {
        double l, a, b, harm, qrs1, qrs2, qrswav;
        l = li;
        a = a_qrswav;
        b = (2 * l) / d_qrswav;
        int n = 100;
        qrs1 = (a / (2 * b)) * (2 - b);
        qrs2 = 0;
        for (int i = 1; i < n; i++)
        {
           harm = (((2 * b * a) / (i * i * Math.PI * Math.PI)) * (1 - Math.Cos((i * Math.PI) / b))) * Math.Cos((i *
Math.PI * x) / l);
           qrs2 = qrs2 + harm;
        }
        qrswav = qrs1 + qrs2;
        return qrswav;
     }
     double q_wav(double x, double a_qwav, double d_qwav, double t_qwav, double li)
     {
        double l, a, b, q1, q2, harm5, qwav;
        l = li;
        x = x + t_qwav;
        a = a_qwav;
        b = (2 * l) / d_qwav;
        int n = 100;
        q1 = (a / (2 * b)) * (2 - b);
        q2 = 0;
        for (int i = 1; i < n; i++)
        {
           harm5 = (((2 * b * a) / (i * i * Math.PI * Math.PI)) * (1 - Math.Cos((i * Math.PI) / b))) * Math.Cos((i *
Math.PI * x) / l);
           q2 = q2 + harm5;
        }
        qwav = -1 * (q1 + q2);
        return qwav;
     }
     double s_wav(double x, double a_swav, double d_swav, double t_swav, double li)
     {
        double l, a, b, s1, s2, harm3, swav;
```

```
        l = li;
        x = x - t_swav;
        a = a_swav;
        b = (2 * l) / d_swav;
        int n = 100;
        s1 = (a / (2 * b)) * (2 - b);
        s2 = 0;
        for (int i = 1; i < n; i++)
        {
           harm3 = (((2 * b * a) / (i * i * Math.PI * Math.PI)) * (1 - Math.Cos((i * Math.PI) / b))) * Math.Cos((i *
Math.PI * x) / l);
           s2 = s2 + harm3;
        }
        swav = -1 * (s1 + s2);
        return swav;
     }
     double t_wav(double x, double a_twav, double d_twav, double t_twav, double li)
     {
        double l, a, b, t1, t2, harm2, twav1, twav;
        l = li;
        a = a_twav;
        x = x - t_twav - 0.045;
        b = (2 * l) / d_twav;
        int n = 100;
        t1 = 1 / l;
        t2 = 0;
        for (int i = 1; i < n; i++)
        {
           harm2 = (((Math.Sin((Math.PI / (2 * b)) * (b - (2 * i)))) / (b - (2 * i)) + (Math.Sin((Math.PI / (2 * b)) * (b + (2
* i)))) / (b + (2 * i))) * (2 / Math.PI)) * Math.Cos((i * Math.PI * x) / l);
           t2 = t2 + harm2;
        }
        twav1 = t1 + t2;
        twav = a * twav1;
        return twav;
     }
     double u_wav(double x, double a_uwav, double d_uwav, double t_uwav, double li)
     {
        double l, a, b, u1, u2, harm4, uwav1, uwav;
        l = li;
        a = a_uwav;
        x = x - t_uwav;
        b = (2 * l) / d_uwav;
        int n = 100;
        u1 = 1 / l;
        u2 = 0;
        for (int i = 1; i < n; i++)
        {
           harm4 = (((Math.Sin((Math.PI / (2 * b)) * (b - (2 * i)))) / (b - (2 * i)) + (Math.Sin((Math.PI / (2 * b)) * (b + (2
* i)))) / (b + (2 * i))) * (2 / Math.PI)) * Math.Cos((i * Math.PI * x) / l);
           u2 = u2 + harm4;
        }
```

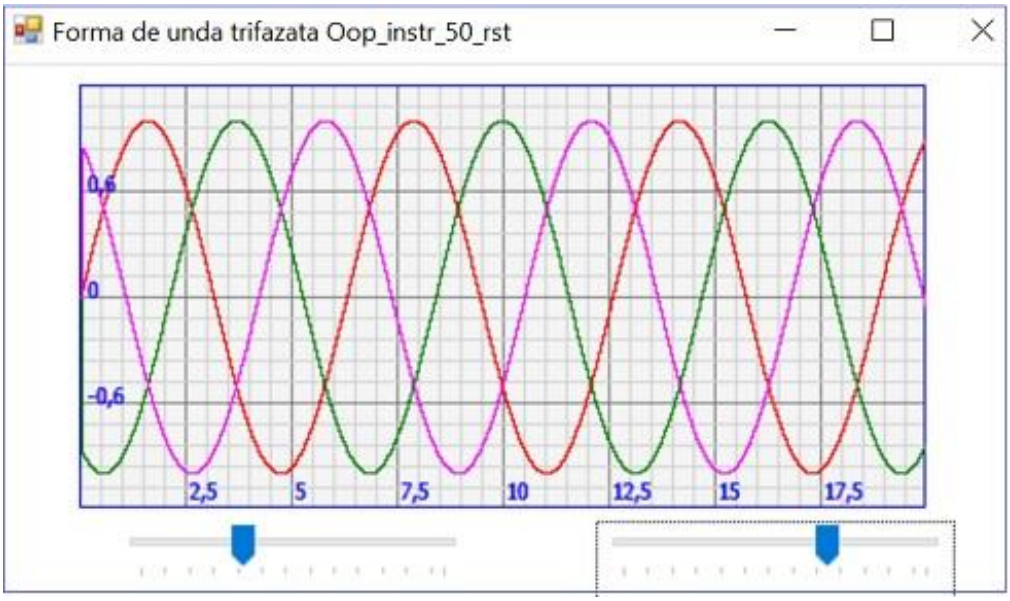
```
        uwav1 = u1 + u2;
        uwav = a * uwav1;
        return uwav;
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        desen = this.CreateGraphics();
        Array.Resize(ref valori, n_maxx + 1);
        ekg = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        int p;
        a_qrswav = System.Convert.ToDouble(this.trackBar2.Value) / 10.0;
        double val_m = System.Convert.ToDouble(this.trackBar1.Value);
        li = 30.0 / val_m;
        this.label1.Text = Math.Round(val_m).ToString();
        double pas = 0.004;
        double x = pas;
        double pwav, qwav, qrswav, swav, twav, uwav, ecg;
        for (p = 0; p < n_maxx; p++)
        {
            pwav = p_wav(x, a_pwav, d_pwav, t_pwav, li);
            qwav = q_wav(x, a_qwav, d_qwav, t_qwav, li);
            qrswav = qrs_wav(x, a_qrswav, d_qrswav, li);
            swav = s_wav(x, a_swav, d_swav, t_swav, li);
            twav = t_wav(x, a_twav, d_twav, t_twav, li);
            uwav = u_wav(x, a_uwav, d_uwav, t_uwav, li);
            ecg = pwav + qrswav + twav + swav + qwav + uwav;
            valori[p] = ecg;
            x = x + pas;
        }
        double min = valori[0];
        for (p = 0; p < n_maxx; p++)
        {
            if (valori[p] < min)
                min = valori[p];
        }
        for (p = 0; p < n_maxx; p++)
        {
            valori[p] = valori[p] - min + 0.5; // 0.5 translatie pe y
        }
        ekg.sterg();
        if (this.checkBox1.Checked)
            ekg.auto_sx(pas * n_maxx, 0);
        else
            ekg.auto_sx(n_maxx, 0);

        ekg.setval(valori, n_maxx, Color.Red);
        ekg.display();
}}}
```

Utilizarea clasei "osciloscop" pentru a afisa forma de unda a tensiunii trifazate

Vom folosi acum clasa **osciloscop** nou creata pentru a afisa forma de unda a tensiunii trifazate.

```
namespace Oop_instr_50_rst
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public System.Drawing.Graphics desen;
      public osciloscop oscil_m;
      int pozx = 40, pozy = 10, n_maxx = 400, n_maxy = 200;
      double val_max = 1.2, val_min = -1.2;
      static double[] valori1 = new double[0];
      static double[] valori2 = new double[0];
      static double[] valori3 = new double[0];
      private void Form1_Load(object sender, EventArgs e)
      {
         desen = this.CreateGraphics();
         Array.Resize(ref valori1, n_maxx + 1);
         Array.Resize(ref valori2, n_maxx + 1);
         Array.Resize(ref valori3, n_maxx + 1);
         oscil_m = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         double fi1 = this.trackBar1.Value / 100.0;
         double fi2 = this.trackBar2.Value / 100.0;
         double alfa = 0;
         double pas = 0.05;
         for (int i = 1; i <= n_maxx; i++)
         {
            alfa += pas;
```

```
            valori1[i] = Math.Sin(alfa);
            valori2[i] = Math.Sin(alfa - fi1);
            valori3[i] = Math.Sin(alfa - fi2);
        }
        oscil_m.sterg();
        oscil_m.auto_sx(pas * n_maxx, 0);
        oscil_m.setval(valori1, n_maxx, Color.Red);
        oscil_m.setval(valori2, n_maxx, Color.Green);
        oscil_m.setval(valori3, n_maxx, Color.Magenta);
        oscil_m.display();
    }
  }
}
```



Vom folosi aceeai clasa **osciloscop** pentru a afisa formele de unda ale unui regultor PI.

```
namespace Oop_instr_50_pi
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    public System.Drawing.Graphics desen;
    //public System.Drawing.Pen creion_rosu;
    public osciloscop oscil_m;
    int pozx = 180, pozy = 10, n_maxx = 400, n_maxy = 300;
    double r, er, ir, e_v, u, u_v, y, y_v, kp, ikp, Ti, iTi, Te;
    double val_max = 800; //valoarea maxima
    double val_min = -100; //valoarea minima
```

```
    static double[] valori_0 = new double[0];
    static double[] valori_1 = new double[0];
    private void Form1_Load(object sender, EventArgs e)
    {
      Array.Resize(ref valori_0, n_maxx + 1);
      Array.Resize(ref valori_1, n_maxx + 1);
      desen = this.CreateGraphics();
      //creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
      oscil_m = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy, val_max, val_min);
      kp = 1.1;
      Ti = 2.5;
      Te = 0.1;
      r = 175;
      u = 50;
      y = 0;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      y = 0;
      u = 50;
      e_v = 0;
      y_v = 0;
      u_v = 0;
      ir = System.Convert.ToInt32(this.trackBar1.Value);
      ikp = System.Convert.ToDouble(this.trackBar2.Value);
      iTi = System.Convert.ToInt32(this.trackBar3.Value);
      r = System.Convert.ToDouble(ir);
      kp = System.Convert.ToDouble(ikp / 10);
      Ti = System.Convert.ToDouble(iTi / 10);
      this.label4.Text = r.ToString();
      this.label5.Text = kp.ToString();
      this.label6.Text = Ti.ToString();
      for (int i = 1; i <= n_maxx; i++)
      {
        er = r - y;
        u = u_v + er * (kp + (Ti * (Te / 2))) + e_v * ((Ti * (Te / 2)) - kp);
        y = (u * Te + 5 * y_v) / (5 + Te);
        valori_0[i] = y;
        valori_1[i] = u;
        e_v = er;
        y_v = y;
        u_v = u;
      }
      oscil_m.sterg();
      oscil_m.setval(valori_0, n_maxx, Color.Red);
      oscil_m.setval(valori_1, n_maxx, Color.Green);
      oscil_m.display();
    }
  }
}
```

Instrumente virtuale pentru valori binare

- Instrument virtual - binar

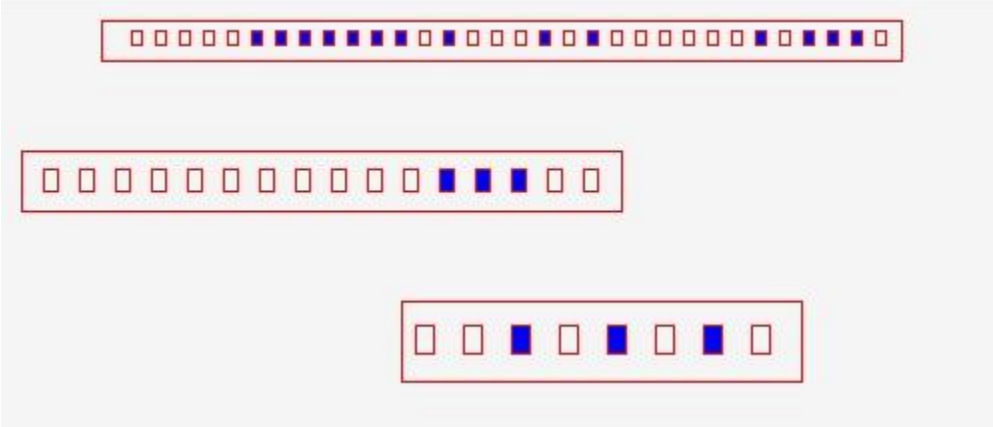
Aplicatia C# **"oop_08"** foloseste clasa **binar** pentru a afisa grafic valori binare.

Vom creea o clasa denumita "binar" dupa care vom realiza trei obiecte prin instantierea clasei binar.

```
namespace oop_08
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      System.Drawing.Graphics Desen;
      System.Drawing.Pen Creion_rosu;
      System.Drawing.SolidBrush Pens_blu;
      System.Drawing.SolidBrush Pens_back;
      public binar binar1;
      public binar binar2;
      public binar binar3;
      System.Random nr;
      UInt64 num;

      public class binar
      {
```

```
        int x0;
        int y0;
        int w;
        int h;


        public void setval(int nrb, UInt64 n, System.Drawing.Graphics zona_des, System.Drawing.Pen creion,
System.Drawing.SolidBrush pens_albastra, System.Drawing.SolidBrush radiera)
        {

           int wb = w / (3 * nrb);
           int hb = h / 3;
           int x = x0 + w - 3 * wb;
           int y = y0 + hb;
           int i;
           zona_des.DrawRectangle(creion, x0, y0, w, h);
           for (i = nrb - 1; i >= 0; i--)
           {
              System.UInt64 bit = ((n >> (nrb - i - 1)) & 1);
              zona_des.DrawRectangle(creion, x - 1, y - 1, wb + 1, hb + 1);
              if (bit == 1)
                 zona_des.FillRectangle(pens_albastra, x, y, wb, hb);
              else
                 zona_des.FillRectangle(radiera, x, y, wb, hb);

              x -= 3 * wb;

           }
        }
        public void init_binar(int pozx, int pozy, int lat, int inalt)
        {
           x0 = pozx;
           y0 = pozy;
           w = lat;
           h = inalt;
        }
     }
     private void Form1_Load(object sender, EventArgs e)
     {
        Creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
        Pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
        Pens_back = new System.Drawing.SolidBrush(this.BackColor);
        Desen = this.CreateGraphics();
        nr = new System.Random();
        binar1 = new binar();
        binar1.init_binar(50, 10, 400, 20);
        binar2 = new binar();
        binar2.init_binar(10, 75, 300, 30);
        binar3 = new binar();
        binar3.init_binar(200, 150, 200, 40);
     }
```

```
    private void timer1_Tick(object sender, EventArgs e)
    {
      binar1.setval(32, System.Convert.ToUInt64(nr.Next(1999999999)), Desen, Creion_rosu, Pens_blu,
Pens_back);
      binar2.setval(16, num, Desen, Creion_rosu, Pens_blu, Pens_back);
      binar3.setval(8, System.Convert.ToUInt64(nr.Next(255)), Desen, Creion_rosu, Pens_blu, Pens_back);
      num += 1;
      if (num > 256 * 256)
        num = 0;
    }
  }
}
```



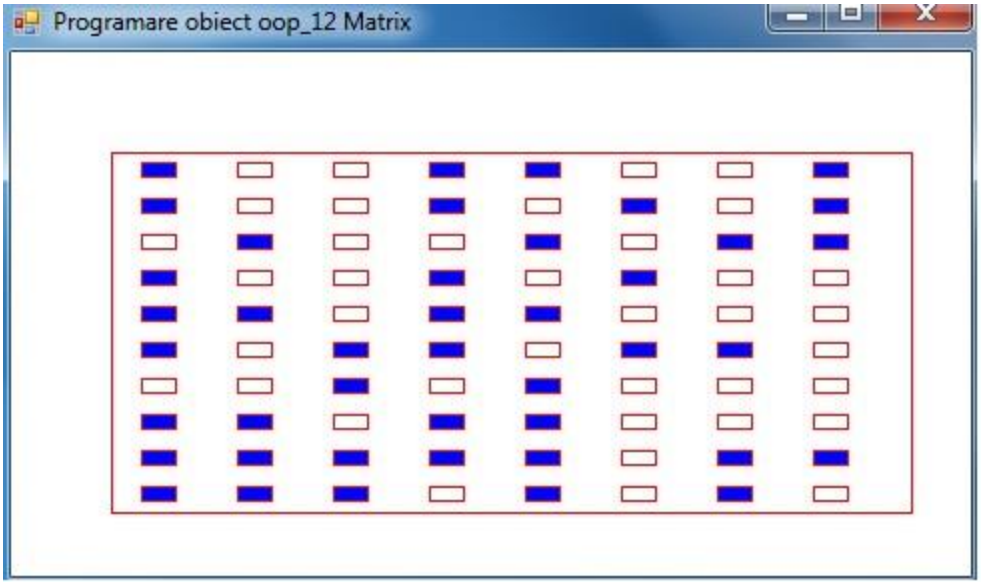
- Instrument virtual - matrice binara

Aplicatia C# **"oop_08"** foloseste clasa **matrix** pentru a afisa matricea binara.

```
namespace oop_12
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    System.Drawing.Graphics Desen;
    System.Drawing.Pen Creion_rosu;
    System.Drawing.SolidBrush Pens_blu;
    System.Drawing.SolidBrush Pens_back;
    System.Random nr;
    public matrix matrix1;
    static Int64[] num = new Int64[0];
    int biti = 8, rnd=10;
```

```
    private void Form1_Load(object sender, EventArgs e)
    {
       Creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
       Pens_blu = new System.Drawing.SolidBrush(System.Drawing.Color.Blue);
       Pens_back = new System.Drawing.SolidBrush(this.BackColor);
       Desen = this.CreateGraphics();
       nr = new System.Random();
       Array.Resize(ref num, rnd + 1);
       matrix1 = new matrix(50, 50, 400, 20);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
       matrix1.setval(biti, rnd,num, Desen, Creion_rosu, Pens_blu, Pens_back);
       this.timer1.Interval = 50;
       /*
       for(int j=0; j < rnd; j++){
          num[j]++;
          if (num[j] > 256)
             num[j] = 0;
          }
        */
        for(int j=0; j < rnd; j++){
          num[j]=System.Convert.ToInt64(nr.Next(255));
        }
    }
  }
  public class matrix
  {
    int x0;
    int y0;
    int w;
    int h;
    public void setval(int nrb, int nrr, Int64[] n, System.Drawing.Graphics zona_des, System.Drawing.Pen creion,
System.Drawing.SolidBrush pens_albastra, System.Drawing.SolidBrush radiera)
    {
       int wb = w / (3 * nrb);
       int hb = h / 3;
       int x = x0 + w - 3 * wb;
       int y = y0 + hb;
       int i,j;
       zona_des.DrawRectangle(creion, x0, y0, w, h*(nrr-1));
       for (j = 0; j < nrr; j++)
       {
          for (i = nrb - 1; i >= 0; i--)
          {
             System.Int64 bit = ((n[j] >> (nrb - i - 1)) & 1);
             zona_des.DrawRectangle(creion, x - 1, y - 1, wb + 1, hb + 1);
             if (bit == 1)
                zona_des.FillRectangle(pens_albastra, x, y, wb, hb);
             else
                zona_des.FillRectangle(radiera, x, y, wb, hb);
```

```
            x -= 3 * wb;
          }
          x = x0 + w - 3 * wb;
          y += 3 * hb;
        }
      }
      public matrix(int pozx, int pozy, int lat, int inalt)
      {
        x0 = pozx;
        y0 = pozy;
        w = lat;
        h = inalt;
      }
    }
}
```



Functii pentru caractere, siruri C si clase de sir C++

Citirea caracerelor si a sirurilor de caractere

Majoritatea calculatoarelor au instalat cel putin un editor de text. Functiile referitoare la siruri de caractere, respectiv la texte au deci o importanta capitala in implementarea diferitelor aplicatii de procesare texte. Sirurile de caractere sunt realizate din "insiruirea" mai multor caractere.

Vom incerca sa utilizam cateva functii referitoare la sirurile de caractere, disponibile in limbajul C++. Limbajul de programare C++, dispune de tipul de date *string*, cunoscut si sub numele de clasa sir C++.

- **Citirea unui caracter de la tastatura**

In capitolele anterioare am realizat o serie de aplicatii in care se astepta introducerea unui caracter sau a unui text.

Am utilizat astfel, la sfarsitul unui program, urmatoarea secventa de mai jos pentru a intreba utilizatorul daca vrea sa continue aplicatia sau renunta:

```
// Citirea unui caracter de la tastatura
#include "stdafx.h"
#include < iostream >
using namespace std;
int main(void)
{
	system("TITLE Citirea unui caracter de la tastatura");// Titlul ferestrei consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	do{
		cout << "\n\n\tApasati tasta E pentru iesire sau C pentru continuare :";
		cin >> car;
		if (car !='E' && car !='e')
			cout << "\n\tAti ales -Continuare !";
		else
			cout << "\n\tAti ales -Iesire";
	}while (car !='E' && car !='e');
   return 0;
}
```

Dupa cum se observa, continuarea programului se face prin apasarea oricarei taste. Numai la apasara tastei E sau e se face iesirea din program. In cazul in care se apasa tasta Enter sau spatiu, nu se afiseaza nimic, functia cin << ignora caracterele spatiu si Enter de la inceput.

Pentru a lua in considerare si tasa Enter, vom folosi metoda **get** a obiectului **cin** . Metoda unui obiect se mai numeste si functia membru. In general, se utilizeaza expresia: "se invoca metoda **get** a obiectului **cin** .

```
// Citirea caracterului Enter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Citirea caracterului 'Enter'");// Titlul ferestrei consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	do{
		cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru continuare :";
		cin.get(car);
		if (car !='E' && car !='e')
			cout << "\n\tAti ales -Continuare !";
		else
			cout << "\n\tAti ales -Iesire";
	}while (car !='E' && car !='e');
  return 0;
}
```

In cazul de sus continuarea aplicatiei se face apasand Enter. In cazul in care se introduce un caracter, urmat de Enter, programul se repeta de doua ori, adica de numarul de caractere introdus. Pentru a corecta aceasta problema, va trebui sa stergem bufferul de caractere dupa executia instructiunii **cin.get(car);** prin invocarea metodei **ignore** a obiectului **cin**. Trebuie avut insa grija sa nu stergem bufferul daca in el se afla numai caracterul Enter. Vom conditiona deci executia instructiunii : **cin.ignore();** cu o instructiune conditionala de genul **if (car !='\n')**

```
// Citirea caracterului Enter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Citirea caracterului 'Enter'");// Titlul ferestrei consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	do{
		cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru continuare :";
```

```
                cin.get(car);
                if (car !='\n')
                cin.ignore();
                if (car !='E' && car !='e')
                        cout << "\n\tAti ales -Continuare !";
                else
                        cout << "\n\tAti ales -Iesire";
        }while (car !='E' && car !='e');
  return 0;
}
```

- **Citirea unui sir de caractere**

Din aplicatiile anterioare s-a observat ca utilizand instructiunile:

string nume;

cin >> nume;

nu putem citi numele si prenumele unei persoane pentru ca **cin** , va atribui variabilei nume, caracterele introduse pana la primul caracter spatiu, ignorand restul caracterelor. In cadrul tablourilor am rezolvat aceasta problema, invocand metoda **getline** a obiectului **cin** astfel:

```
// Citirea unui sir de caractere ce contine caractere "spatiu"
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Citirea unui sir de caracter de la tastatura");// Titlul ferestrei consola
        system("COLOR F9"); // Fundal alb caractere albastre
        char car;
        char nume[80];
        do{
                cout << "\n\n\tIntroduceti numele d-voastra :";
                cin.getline(nume,80);
                cout << "\n\tNumele d-voastra este :" << nume ;
                cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru continuare :";
                cin.get(car);
                if (car !='\n')
                cin.ignore();
        }while (car !='E' && car !='e');
  return 0;
}
```

Functii pentru caractere si siruri de caractere

In spatiul de nume std:: functiile pentru siruri sunt grupate in biblioteca "cstring" . Bibloteca "string" furnizeaza functii pentru calsa sir din C++.

• Functii pentru caractere

Exista situatii in care trebuie efectuam operatii asupra unui carater cum ar fi: sa testam valoarea unui caracter sau sa-l convertim din minuscula in majuscula etc. Exista functii special definite pentru operatii pe caractere.

Functii pentru conversia tipului de caracter

Sa reluam aplicatia de sus si sa folosim functia **touper** prin care sa convertim caracterul citit in litera mare astfel nu va mai fi necesara dubla testare (litera mare sau mica)

```
// Utilizrea functiei "toupper"
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Utilizrea functiei 'toupper'");// Titlul ferestrei consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	char nume[80];
	do{
		cout << "\n\n\tIntroduceti numele d-voastra :";
		cin.getline(nume,80);
		cout << "\n\tNumele d-voastra este :" << nume ;
		cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru continuare :";
		cin.get(car);
		car=toupper(car);
		if (car !='\n')
		cin.ignore();
	}while (car !='E');
   return 0;
}
```

Functii pentru verificarea valorii unui caracter

De multe ori trebuie sa verificam valoarea unui caracter de exemplu: sa testam daca un caracter este o litera sau

o cifra, sa testam o litera este majuscula sau minscula etc.

Vom realiza in continuare un program care cere o litera, o analizeaza si afiseaza tipul caracterului.

```
//  Functii pentru verificarea valorii unui caracter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Utilizrea functiilor pe caracter");// Titlul ferestrei consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	char car_t;
	do{
		cout << "\n\n\tIntroduceti un caracter :";
		cin.get(car_t);
		if (isalnum(car_t)){
			if (isalpha(car_t)){
				if (isupper(car_t))
					cout << "\n\n\tAti introdus un caracter majuscula :";
				else
					cout << "\n\n\tAti introdus un caracter minuscula :";
			}else{
				cout << "\n\n\tAti introdus o cifra :";
			}
		}else{
		cout << "\n\n\tNu ati introdus nici cifra nici caracter :";
		}
		cin.ignore();
		cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru continuare :";
		cin.get(car);
		car=toupper(car);
		if (car !='\n')
		cin.ignore();
	}while (car !='E');
   return 0;
}
```

- **Functii pentru siruri de caractere**

Determinarea lungimii unui sir

Functia **strlen**

Forma generala: **strlen (nume_sir)**

Functia determina lungimea unui sir. Functia returneaza un numar întreg ce reprezinta lungimea unui sir de caractere, fara a numara terminatorul de sir.

Urmatoarea aplicatie determina lungimea a trei siruri declarate in moduri diferite.

```
// Programul afiseaza lungimea unui sir de caractere
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
 system("TITLE Determinarea lungimii unui sir ");
 system("COLOR F9");
 cout << "\n\t- Programul afiseaza lungimea diverselor siruri de caractere.\n\n\n";
 char* fc ="Facultatea de Inginerie";
 cout <<"\n\n\tLungimea sirului: " << fc << " este: " << strlen(fc);
 char ctd[55]="Inginerie Electrica";
 cout <<"\n\n\tLungimea sirului: " << ctd << " este: " << strlen(ctd);
 cout <<"\n\n\tLungimea sirului: " << "Sectia Calculatoare" << " este: " << strlen("Sectia Calculatoare");
 cin.get();
 return 0;
}
```

Functia **strlen** poate fi folosita de asemenea pentru a determina lungimea unui sir preluat de la utilizator.

```
// Programul afiseaza lungimea unui sir de caractere introdus de la tastatura.
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
        system("TITLE Determinarea lungimii unui sir preluat ");
        system("COLOR F9");
        cout << "\n\t- Programul afiseaza lungimea unui sir preluat de la tastatura.\n\n\n";
        char n_pr[80];
        cout << "\n\t- Introduceti numele d-voastra:";
```

```
	cin.getline(n_pr,80);
	cout <<"\n\n\tLungimea sirului: " << n_pr << " este: " << strlen(n_pr);
	cin.ignore();
	cin.get();
	return 0;
}
```

Sa combinam functiile pe caractere cu functiile pe siruri pentru a verifica daca un sir este introdus corect. Sa luam de exemplu o adresa de mail. Vom verifica daca are cel putin 5 caractere si contine caracterele . @.

```
// Programul cere adresa de mail si verifica daca:
// - adresa are cel putin 5 caractere
// - adresa contine caracterul .
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
	system("TITLE Validarea unui sir preluat ");
	system("COLOR F9");
	bool dim=false, at=false, punct=false;
	cout << "\n\tProgramul cere adresa de mail si verifica daca:.\n";
	cout << "\n\t - adresa are cel putin 5 caractere";
	cout << "\n\t - adresa contine caracterul @";
	cout << "\n\t - adresa contine caracterul .";
	char adresa[80];
	cout << "\n\n\n\t- Introduceti adresa de mail:";
	cin.getline(adresa,80);
	if (strlen(adresa) >5){
		dim=true;
	}
	for (int i=0; i <= strlen(adresa); i++){
		if (adresa[i]=='@')
			at=true;
		if (adresa[i]=='.')
			punct=true;
	}
	if(((at)&(punct)&(dim))==0){
		if (dim==0)
			cout <<"\n\n\tLungimea adresei: " << adresa << " este este mai mica de 5 caractere!";
		if (at==0)
			cout <<"\n\n\tAdresa : "<< adresa << "nu contine @ ";
		if (punct==0)
			cout <<"\n\n\tAdresa : "<< adresa << " nu contine .";
		}else{
			cout <<"\n\n\tAdresa : "<< adresa << " este corecta !";
		}
	cin.get();
	return 0;
}
```

Operatia efectuata anterior se numeste operatia de validare camp.

Copierea unui sir intr-un alt sir

Forma generala: **strcpy (sir_destinatie, sir_sursa)**

Functia copiaza sirul sursa în sirul destinatie. Pentru a fi posibila copierea, lungimea sirului destinatie trebuie sa fie mai mare sau egala cu cea a sirului sursa, altfel pot apare erori.

Sa luam o aplicatie de geniul:

```
//Copierea unui sir intr-un alt sir

#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
	system("TITLE Copierea unui sir intr-un alt sir  ");
	system("COLOR F9");
	char* sir1="Alfa";
	char* sir2="Beta";
	sir1=sir2;
	cout << "\n\n\tSirul 1 este: " << sir1;
	cin.get();
	return 0;
}
```

Aplicatia nu cpiaza continutul sir 2 peste sir 1 ci atribuie pointerului sir1 valoarea pointerului sir2.

Pentru a demonstra acest lucru, vom mai introduce un pointer numit sir_1 care va indica tot spre sir1

```
//Copierea unui sir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
char* sir_1;
int main(void){
	system("TITLE Copierea unui sir intr-un alt sir  ");
	system("COLOR F9");
	char* sir1="Alfa";
```

```
	char* sir2="Beta";
	sir_1=&sir1[0];// pointerul sir_1 indica spre sir1
	sir1=sir2;
	cout << "\n\n\tSirul 1 este: " << sir1;
	// daca in urma operatiei sir1=sir2 s-ar copia sir2 peste sir1, instructiunea de jos
	// ar trebui sa afiseze tot "Beta" insa afiseaza "Alfa"
	cout << "\n\n\tSirul 1 este: " << sir_1;
	cin.get();
	return 0;
}
```

Pentru a copia efectiv sirul, vom folosi functia **strcpy** :

```
//Copierea unui sir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
char* sir_1;
int main(void){
	system("TITLE Copierea unui sir intr-un alt sir  ");
	system("COLOR F9");
	char sir1[5]="Alfa";
	char sir2[5]="Beta";
	sir_1=&sir1[0];// pointerul sir_1 indica spre sir1
	strcpy(sir1,sir2);
	cout << "\n\n\tSirul 1 este: " << sir1;
	cout << "\n\n\tSirul 1 este: " << sir_1;
	cin.get();
	return 0;
}
```

De data aceasta se afiseaza acelasi lucru deci copierea s-a facut efectiv.
In cazul ca vrem sa copiem un numar limitat de caractere, folosim functia **strncpy** astfel:

```
//Copierea unui subsir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
```

```
using namespace std;
int main(void){
	system("TITLE Copierea unui sir intr-un alt sir  ");
	system("COLOR F9");
	char sir1[10]="Alfa-Soft";
	char sir2[5]="Beta";
	strncpy(sir1,sir2,3);
	cout << "\n\n\tSirul 1 este: " << sir1;
	cin.get();
	return 0;
}
```

Raspunsul aplicatiei va fi de data aceasta: "Beta-Soft"

Concatenarea a doua siruri

Forma generala: **strcat (sir_destinatie, sir_sursa)**

Operatia de adunare a doua siruri adica operatia de a copia un sir la sfarsitul altui sir se numeste concatenare.

Functia **strcat** concateneaza cele doua siruri: sirul sursa este adaugat la sfârsitul sirului destinatie. Tabloul care contine sirul destinatie trebuie sa aiba cel putin dimensiunea celor doua siruri.

Pentru a concatena doua siruri, procedam astfel:

```
//Adaugarea unui sir la un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void){
	system("TITLE Adaugarea unui sir la un alt sir  ");
	system("COLOR F9");
	char sir1[15]="Alfa-Soft-";
	char sir2[5]="Beta";
	strcat(sir1,sir2);
	cout << "\n\n\tSirul 1 este: " << sir1;
	cin.get();
	return 0;
}
```

Raspunsul aplicatiei va fi de data aceasta: "Alfa-Soft-Beta". Trebuie sa avem grija ca dimensiunea sirului sir1 sa fie suficienta pentru a adauga sir2.

Compararea a doua siruri

Forma generala: **strcmp (sir_1, sir_2)**

Functia compara cele doua siruri date ca argument si returneaza o valoare întreaga egala diferenta dintre codurile ASCII ale primelor caractere care nu coincid.

Pentru a compara doua siruri, vom folosi functia **strcmp** . Functia returneaza 0 daca sirurile sunt egale, un numar negativ daca primul sir< al doilea sir respectiv pozitiv daca primul sir > al doilea sir.

```
//Compararea a doua siruri;
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void){
            system("TITLE Compararea a doua siruri  ");
            system("COLOR F9");
            char sir1[10]="Alfa";
            char sir2[5]="Alfa";
            if (strcmp(sir1,sir2)==0)
            cout << "\n\n\tCele doua siruri sunt egale !" ;
            else
            cout << "\n\n\tCele doua siruri sunt diferite !" ;
            cin.get();
            return 0;
}
```

Raspunsul este desigur "Cele doua siruri sunt egale". Daca am folosi conditia :"if (sir1==sir2) raspunsul ar fi tot timpul fals deoarece se compara defapt doua adrese ale celor doua siruri.

Conversia unui intreg intr-un sir

```
//  Conversia unui intreg intr-un sir
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
            system("TITLE Conversia unui intreg intr-un sir");
```

```
            system("COLOR F9");
            int n;
            char buffer [33];
            cout << "\n\n\tIntroduceti un numar intreg: " ;
            cin >> n;
            itoa(n,buffer,10);//functia pentru conversia unui intreg intr-un sir
            cout << "\n\n\tSirul rezultat din convertirea numarului este: " << buffer;
            cin.ignore();
            cin.get();
 return 0;
}
```

Pozitia unde apare subsirul s2 in sirul s1

Forma generala: **strstr(s1, s2)**

Functia returneaza pozitia unde apare subsirul s2 in sirul s1 sau altfel spus, functia returneaza un pointer la începutul subsirului s2, sau NULL daca subsirul nu este gasit.

Urmatoarea aplicatie foloseste functia *strstr(s1, s2)* pentru a gasi pozitia unde se gaseste un subsir si afiseaza sirul initial incepand din aceasta pozitie.

```
// Programul cere un sir si un subsir de caractere .
// Programul afiseaza sirul din pozitia unde incepe subsirul in cadrul sirului .
// Se foloseste functia
// Functia strstr(s1, s2) – returneaza pozitia unde apare subsirul s2 in sirul s1
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;

int main(void)
{
system("TITLE Pozitia unui subsir in cadrul unui sir ");
system("COLOR F9");
char s1[100],s2[25],*p;
cout << "\n\n\tIntroduceti sirul = ";
cin.getline(s1,80);
cout << "\n\tIntroduceti subsirul = ";
cin.getline(s2,80);
p=strstr(s1,s2);
if (p!=NULL)
cout << "\n\n\n\tSirul este: = " << p;
else
cout << "\n\n\n\tSubsirul:  " << s2 << " nu s-a gasit";
cin.ignore();
cin.get();
return 0;
}
```

Conversia unui sir intr-un intreg

Pentru a converti un text ce reprezinta un intreg folosim functia **atoi**

```
//  Conversia unui sir intr-un intreg
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE  Conversia unui sir intr-un intreg ");
	system("COLOR F9");
	char nr_ascii [10]="123";
	int nr=atoi(nr_ascii);
	cout << "\n\n\tValoarea zecimala a sirului este: " << nr ;
	cout << "\n\n\tPatratul valoarii zecimale a sirului este: " << nr*nr;
	cin.ignore();
	cin.get();
 return 0;
}
```

Siruri de caracere in spatiul System

Diverse conversii

In spatiul de nume System se foloseste System::Convert::... pentru diverse conversii.

```
// Conversia unui sir intr-o valoare numerica in System
// Functia ReadLine citeste tot timpul un sir de caractere
// Se foloseste System::Convert::... pentru diverse conversii
#include "stdafx.h"
#include < iostream >
using namespace std;
using namespace System;

int main(void)
{
	std::system("TITLE Conversia unui sir intr-o valoare numerica");
	std::system("COLOR F9");
	double raza; // valoarea numerica a razei
	String^ raza_s; // sirul ce contine valoarea citita
```

```
        double pi= System::Math::PI;
        Console::Write( L"\n\n\tIntroduceti raza cercului:" );
        raza_s= Console::ReadLine();
        raza = System::Convert::ToDouble( raza_s );
        Console::WriteLine( "\n\n\tAria cercului de raza: "+ raza +" este: "+pi*raza*raza );
        Console::ReadLine();
  return 0;
}
```

La fel procedam si pemtru aplicatii CLR Windows Form Applications.

Pentru a realiza o aplicatie care permite sa zicem introducerea unei valori numerice care reprezinta un unghi in radiani dupa care afiseaza sinusul acelei valori, avem nevoie de o functie ce converteste valoarea in Double, pentru a calcula sinusul apoi avem nevoie de conversia valorii sinusului intr-un text in vederea afisarii intr-un obiect de tip label.

Deschidem un nou proiect Windows Forms Application intitulat "convers_01" pe care plasam un obiect de tip numericUpDown numit numericUpDown1 si doua obiecte de tip label numite label1,label 2.

Completam procedura deschisa pe evenimentul "ValueChanged" cu:

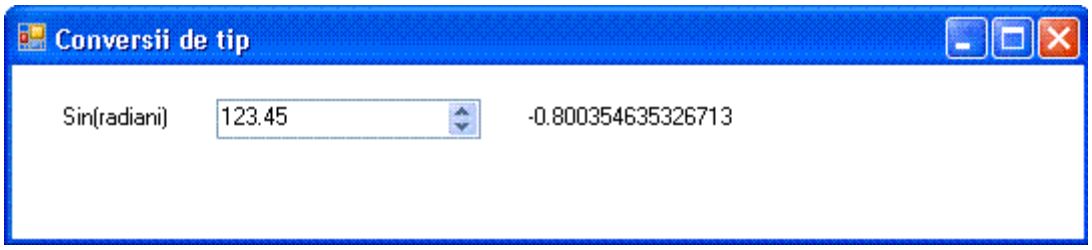
```
        double rad2;
        rad2=System::Convert::ToDouble(this->numericUpDown1->Value);
        this->label2->Text =System::Convert::ToString(System::Math::Sin(rad2));
```

C#

```
        double rad2;
        rad2=System.Convert.ToDouble(this.numericUpDown1.Value);
        this.label2.Text =System.Convert.ToString(System.Math.Sin(rad2));
```

Metoda: **System.Convert.ToDouble(this.numericUpDown1.Value)**- converteste valoarea introdusa in Double iar **System.Convert.ToString(System.Math.Sin(rad2))** converteste vaoarea double in text.

Dupa rularea aplicatiei, tastarea unei valori, apoi tasta Enter, se obtine:



Obiectul numericUpDown poate fi inlocuit cu un obiect de tipul textBox numit textBox1. Avantajul este ca dupa fiecare tastare a unei cifre se afiseaza o noua valoare a sinusului. Dezavantajul este ca daca se tasteaza alte caractere inafara cifrelor, programul da o eroare de exceptie care trebuie tratata. In acest caz procedura deschisa pe evenimentul "ValueChanged" a obiectului textBox1 devine:

```
double rad2;
try {
        rad2=System::Convert::ToDouble(this->textBox1->Text );
        this->label2->Text =System::Convert::ToString(System::Math::Sin(rad2));
}
catch(System::FormatException^){
                this->label2->Text="Eroare format";
}
```

C#

```
double rad2;
try {
        rad2=System.Convert.ToDouble(this.textBox1.Text );
        this.label2.Text=System.Convert.ToString(System.Math.Sin(rad2));
}
catch(System.FormatException){
                this.label2.Text="Eroare format";
}
```

Metode pentru siruri utilizate in WFA

Vom utiliza in continuare cateva metode referitoare la siruri de caractere. Vom realiza o aplicatie WFA care afiseaza in mod continuu numarul de caractere introduse de la tastatura. In cadrul acestei aplicatii vom utiliza un obiect label care ne va permite sa afisam numarul de caractere introdus de la tastatura prin intermediul unui obiect de tip TextBox. Vom utiliza un obiect ProgressBar care ne va permite sa afisam grafic procentul de caractere

introdus. Afisarea se va face procentual relativ la numarul maxim de 300 de caractere setat pentru aceasta aplicatie. Se va afisa de asemenea si procentul sub forma numerica, atat pe 15 caractere cat si pe maxim 5 caractere.

Vor fi folosite printre altele metodele :

- **Length** -pentru determinarea lungimii unui sir de caractere
- **Trim** -pentru eliminarea spatiilor din cadrul unui sir de caractere
- **System::String::Substring** - pentru a substrage un subsir dintr-un sir de caractere
- **System::Convert::ToString** - pentru a converti diverse tipuri in string

Deschidem un nou proiect Windows Forms Application intitulat "text_bar" pe care plasam:

- 7 obiecte de tip **label**
- un obiect de tip **textBox** caruia ii setam proprietatea "multiline" la "True".
- un obiect de tip **progressBar**
- un obiect de tip **timer** caruia ii setam proprietatea "enabled" la "True" si interval la 100.

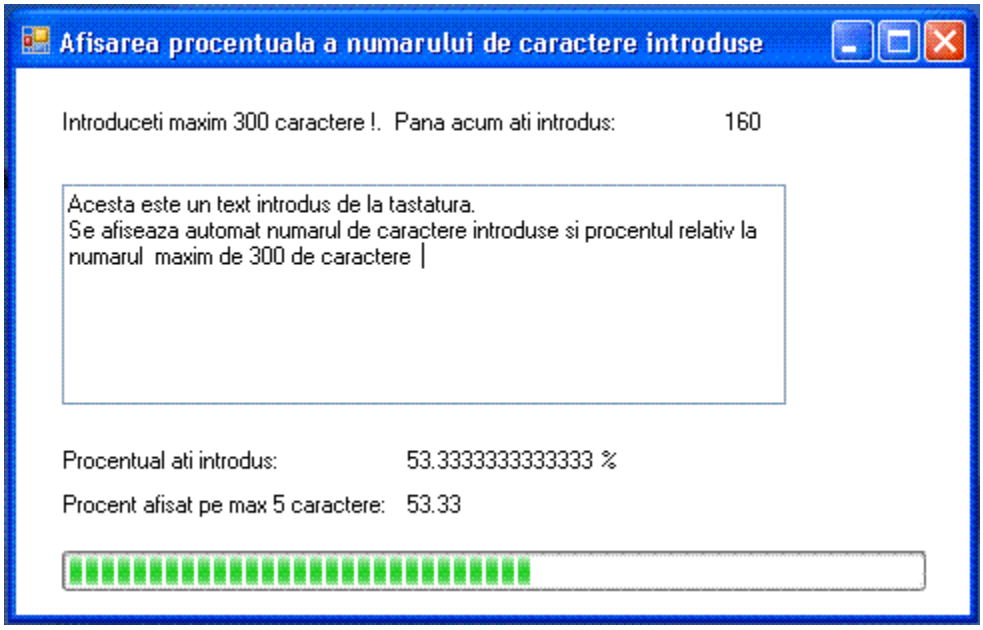
Completam procedura deschisa pe evenimentul Tick al obiectului timer1 cu :

```
double nr_car=(this->textBox1->Text)->Length;
this->label1->Text=System::Convert::ToString(nr_car);
double proc=System::Convert::ToDouble(100*nr_car/300);
String^ proc_s=System::Convert::ToString(proc);
this->label2->Text=proc_s;
String^ proc_st=proc_s->Trim();
String^ proc_s5;
if (proc_st->Length > 5)
        proc_s5=proc_s->System::String::Substring(0,5);
else
        proc_s5=proc_s;

if (proc>100)
        proc=100;
this->label3->Text=proc_s5;
this->progressBar1->Value =proc;
```

C#:

```
namespace text_bar
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void timer1_Tick(object sender, EventArgs e)
      {
      String proc_s, proc_st, proc_s5;
      double nr_car=(this.textBox1.Text).Length;
      double proc;
                        this.label2.Text=System.Convert.ToString(nr_car);
                        proc=System.Convert.ToDouble(100*nr_car/300);
                        proc_s=System.Convert.ToString(proc);
                        this.label4.Text=proc_s;
                        proc_st=proc_s.Trim();

                        if (proc_st.Length > 5)
                                 proc_s5=proc_s.Substring(0,5);
                        else
                                 proc_s5=proc_s;

                        if (proc>100)
                                 proc=100;
                        this.label6.Text=proc_s5;
      this.progressBar1.Value = System.Convert.ToInt16(proc);
      }
   }
}
```

Rulam aplicatia si obtinem:

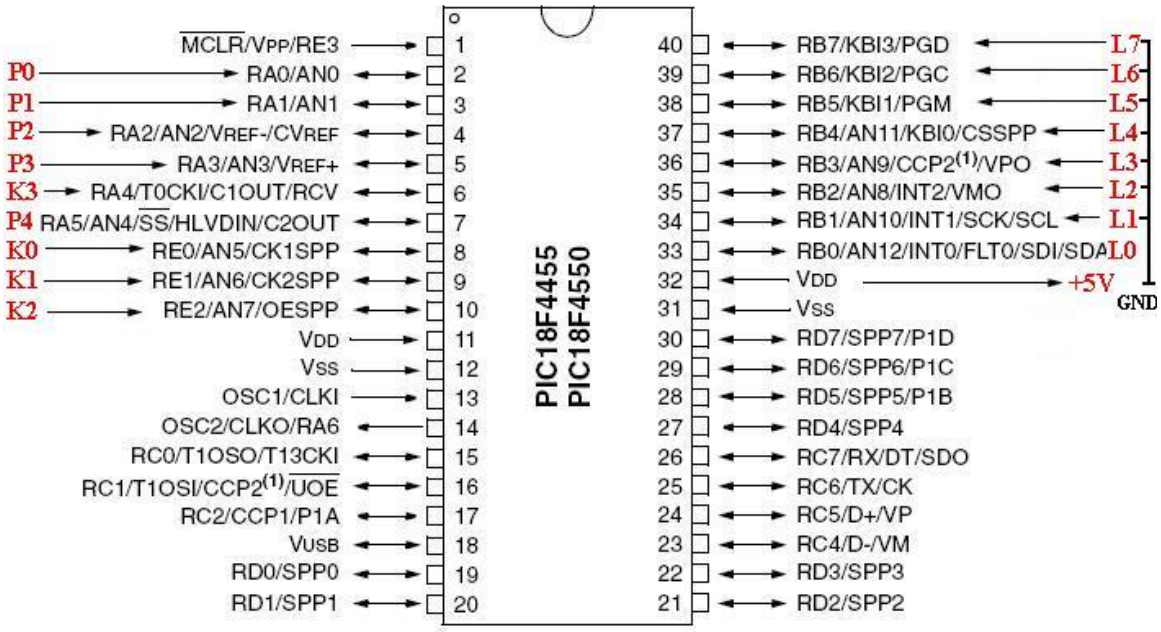


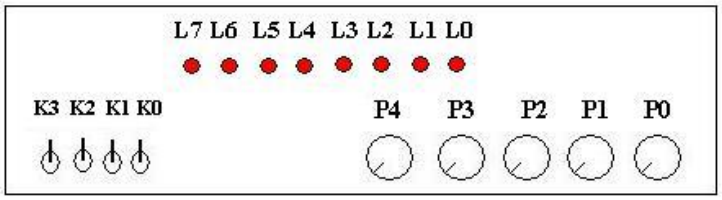
Vehicularea sirurilor de caractere prin intermediul portului serial

Portul serial a fost utilizat pentru a realiza prima legatura intre doua calculatoare. Portul serial este des utilizat si in prezent pentru a conecta diverse dispozitive la calculator. Chiar daca majoritatea calculatoarelor nu mai dispun fizic de un port serial, se utilizeaza des porturi seriale virtuale, realizate prin intermediul portului USB (Universal Serial Bus). Cu alte cuvinte exista o serie de dispozitive conectate la calculator prin intermediul USB insa din punct de vedere logic ele sunt conectate printr-un port serial virtual.

Vom lucra cu un sistem de achizitie date realizat pe baza unui sistem de dezvoltare PIC

Sistemul de achizitie dispune de 5 intrari analogice 4 digitale si 8 iesiri digitale, conectate astfel:





- Comanda pentru citirea simultana a celor cinci canale analogice este:**Ai**
- Comanda pentru citirea unei intrari analogice este:**Ai** unde i=0..4,
- Comanda pentru inscrierea celor 9 iesiri digitale este **On** unde n este numarul in zecimal care va fi afisat in binar pe cele 8 iesiri digitale
- Comanda D1 pentru citirea unui numar zecimal care codifica cele 4 intrari digitale.

Pentru programator, conectarea prin USB este identica cu o conectare prin intermediul unui port serial fizic. Conectarea prin intermediul portului USB fiind deci "transparenta" din punctul de vedere al scrierii aplicatiilor, acestea fiind scrise la fel cu aplicatiile pentru portul serial.

- **Configurarea si initializarea portului serial**

C# - selectarea portului dorit

```
int i, j;
// Listez porturile seriale
Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
this.Porturi_s.Items.Clear();

//Adaug porturile exixtente

for (i = 0; i < Nume_porturi.Length; i++)
{
   this.Porturi_s.Items.Add(Nume_porturi[i]);

}

//Pozitionarea listei pe primul element
this.Porturi_s.SelectedIndex = 0;
```

Unde "Porturi_s" este un obiect de tip ListBox.

C# - Conectarea la portul dorit

```
if (!this.serialPort1.IsOpen)
{
   this.serialPort1.PortName = System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
   this.serialPort1.Open();
   this.label1.Text = "Portul a fost deschis";
}
else
{
   this.label1.Text = "Portul este deja deschis deschis";
}
this.textBox1.Text = "AA";
this.serialPort1.Write("AA");
```

C# - Deconectarea de la portul serial

```
this.serialPort1.Close();
this.label1.Text = "Portul este deconectat!";
```

C# - scrierea si citirea portului serial

```
cda = this.textBox1.Text;

if (this.serialPort1.IsOpen)
{
  this.serialPort1.Write(cda);
  txt = "";
  txt = this.serialPort1.ReadExisting();
  if (txt.Length > 0)
  {
    this.textBox2.Text = txt;
  }
  else
  {
    this.label1.Text = "Nu vin date!";
  }
}
```

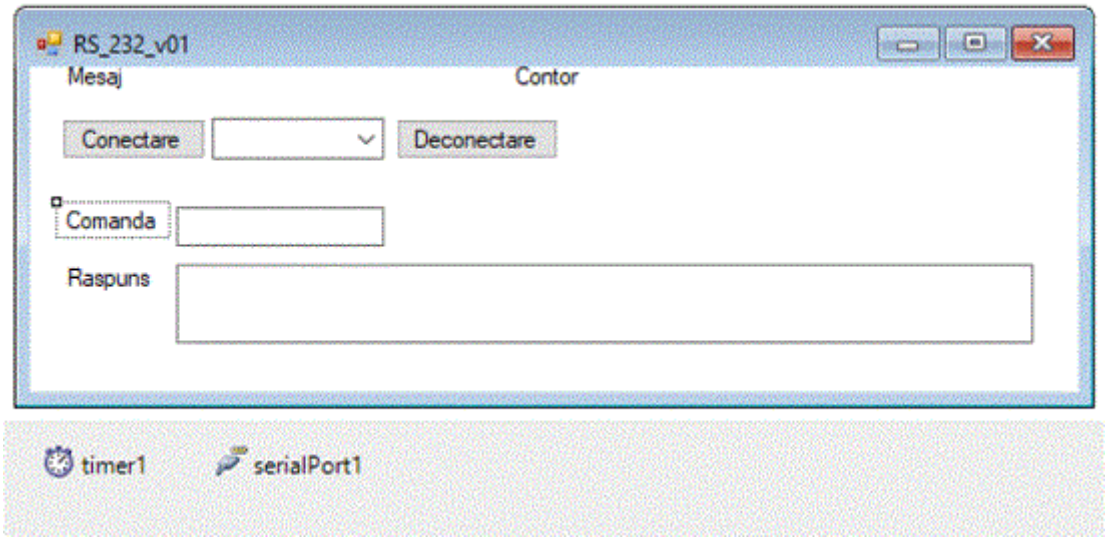
Utilizare sistem de achizitie

Pentru a putea fi utilizat, portul serial trebuie configurat, cu alte cuvinte trebuiesc stabilite valorile pentru diversi parametri, cum ar fi: viteza de lucru, lungimea cuvantului, paritate etc.

In vederea utilizarii portului serial, deschidem un nou proiect C# Windows Forms Application intitulat "RS_232_v1" . In cadrul acestei aplicatii, e nevoie de o interfata care sa permita selectarea portului serial, deschiderea si inchiderea acestuia, stabilirea unui sir de caractere pentru a fi trimis prin portul serial, trimiterea acestuia, citirea unui sir de caractere si afisarea sirului citit. Vom plasa deci

- 2 obiecte de tip **button** 2 pentru conectarea si deconectarea la portul serial(cu numele "but_con" respectiv "but_dec"),
- un obiect de tip **ListBox** cu numele "Porturi_s" pentru alegerea portului serial
- 2 obiecte de tip **textBox** pentru comanda respectiv pentru afisarea raspunsului
- un obiect de tip **timer**

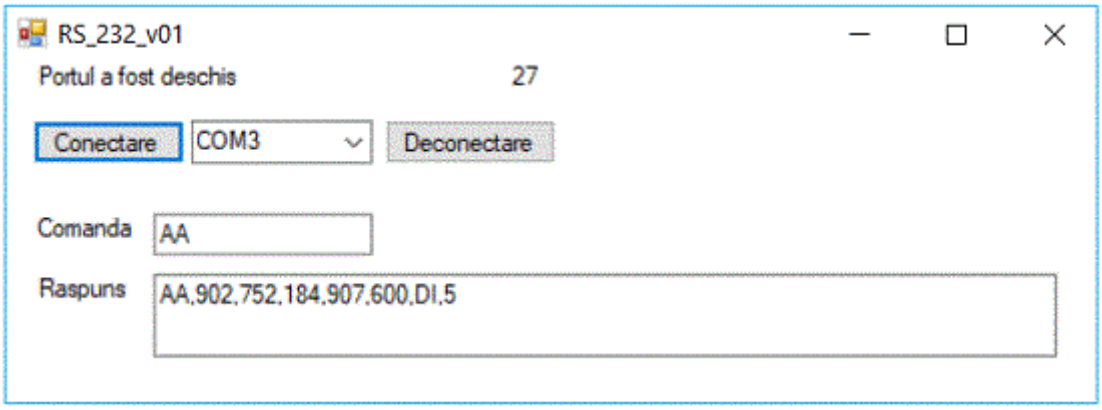
- un obiect de tip **serialPort**



Obiectul de tip "ListBox" este folosit pentru selectarea unui port serial. La un moment dat pot fi deschise mai multe porturi seriale, deci trebuie sa dispunem de o metoda prin care sa selectam portul serial dorit.

Obiectul de tip "ListBox" este cel mai potrivit pentru a selecta unul din porturile seriale cu conditia ca in momentul lansarii aplicatiei sa fie insctrise elementele listei cu numele porturilor deschise in acel moment.

- **Citire intrari analogice**

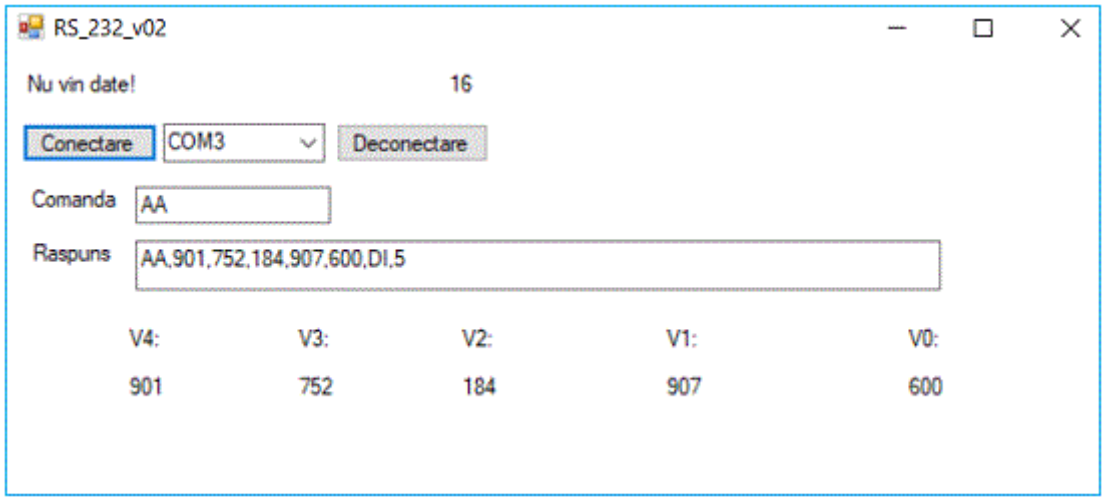


Codul sursa fiind:

```
namespace RS_232_v01
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      string txt, cda;
      int k;
      static String[] Nume_porturi = new String[11];
      private void Form1_Load(object sender, EventArgs e)
      {
         int i, j;
         // Listez porturile seriale
         Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
         this.Porturi_s.Items.Clear();

         //Adaug porturile exixtente

         for (i = 0; i < Nume_porturi.Length; i++)
         {
            this.Porturi_s.Items.Add(Nume_porturi[i]);
         }

         //Pozitionarea listei pe primul element
         this.Porturi_s.SelectedIndex = 0;
         this.label1.Text = "Conectati-va la portul serial!";

      }

      private void but_con_Click(object sender, EventArgs e)
      {
         if (!this.serialPort1.IsOpen)
         {
            this.serialPort1.PortName = System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
            this.serialPort1.Open();
            this.label1.Text = "Portul a fost deschis";
         }
         else
         {
            this.label1.Text = "Portul este deja deschis deschis";
         }
         this.textBox1.Text = "AA";
         this.serialPort1.Write("AA");
      }
```

```
    private void but_dec_Click(object sender, EventArgs e)
    {
      this.serialPort1.Close();
      this.label1.Text = "Portul este deconectat!";

    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      k++;
      if (k > 123456789)
         k = 0;
      this.label2.Text = k.ToString();

      cda = this.textBox1.Text;

      if (this.serialPort1.IsOpen)
      {
        this.serialPort1.Write(cda);
        txt = "";
        txt = this.serialPort1.ReadExisting();
        if (txt.Length > 0)
        {
          this.textBox2.Text = txt;
        }
        else
        {
          this.label1.Text = "Nu vin date!";
        }
      }
    }
  }
}
```

Datele primite trebiesc despachetate pentru a putea fi utilizate. Urmatoarea aplicatie "RS_232_v02" Extrage valorie V4-V0 din sirul primit.




```
namespace RS232_v02
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
    string cda, rec;
    int k;
    static String[] Nume_porturi = new String[11];
    private void Form1_Load(object sender, EventArgs e)
    {
      int i, j;
      // Listez porturile seriale
      Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
      this.Porturi_s.Items.Clear();

      //Adaug porturile exixtente

      for (i = 0; i < Nume_porturi.Length; i++)
      {
        this.Porturi_s.Items.Add(Nume_porturi[i]);
      }

      //Pozitionarea listei pe primul element
```

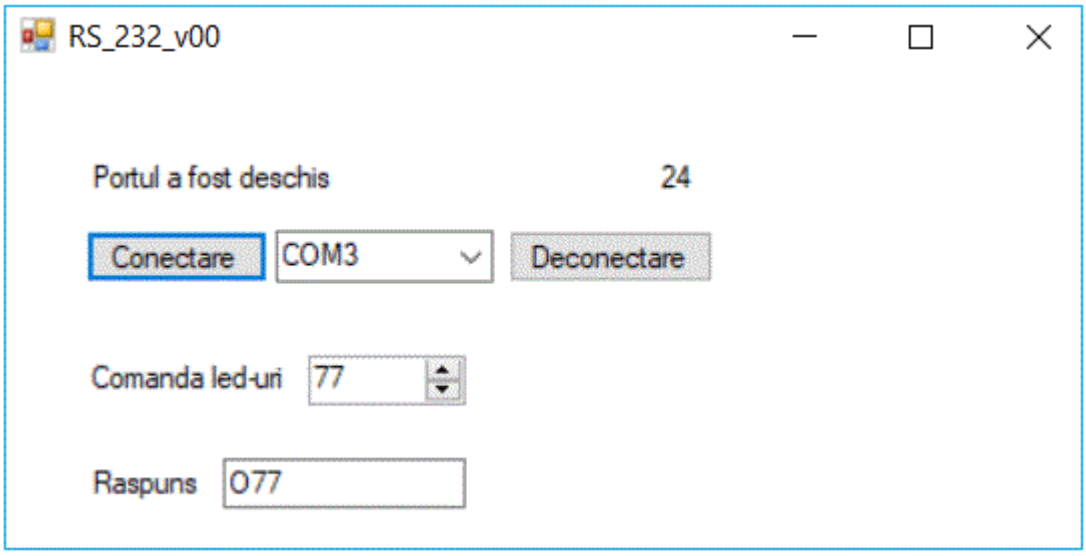
```
    this.Porturi_s.SelectedIndex = 0;
    this.label1.Text = "Conectati-va la portul serial!";

}

private void but_con_Click(object sender, EventArgs e)
{
    if (!this.serialPort1.IsOpen)
    {
        this.serialPort1.PortName = System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
        this.serialPort1.Open();
        this.label1.Text = "Portul a fost deschis";
    }
    else
    {
        this.label1.Text = "Portul este deja deschis deschis";
    }
    this.textBox1.Text = "AA";
    this.serialPort1.Write("AA");
}

private void but_con_Click_1(object sender, EventArgs e)
{
    if (!this.serialPort1.IsOpen)
    {
        this.serialPort1.PortName = System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
        this.serialPort1.Open();
        this.label1.Text = "Portul a fost deschis";
    }
    else
    {
        this.label1.Text = "Portul este deja deschis deschis";
    }
    this.textBox1.Text = "AA";
    this.serialPort1.Write("AA");
}

private void but_dec_Click(object sender, EventArgs e)
{
    this.serialPort1.Close();
    this.label1.Text = "Portul este deconectat!";
}

private void timer1_Tick(object sender, EventArgs e)
{
    k++;
    if (k > 123456789)
        k = 0;
    this.label2.Text = k.ToString();

    cda = this.textBox1.Text;
```

```
        if (this.serialPort1.IsOpen)
        {
    this.serialPort1.Write(cda);
    rec = "";
          rec = this.serialPort1.ReadExisting();
    if (rec.Length > 0)
    {
    this.textBox2.Text = rec;
    int p1 = rec.IndexOf(",");
    int p2 = rec.IndexOf(",", p1 + 1);
    int p3 = rec.IndexOf(",", p2 + 1);
    int p4 = rec.IndexOf(",", p3 + 1);
    int p5 = rec.IndexOf(",", p4 + 1);
    int p6 = rec.IndexOf(",", p5 + 1);
    this.label7.Text = "V4: \n\n" + rec.Substring(p1 + 1, p2 - p1 - 1);
    his.label8.Text = "V3: \n\n" + rec.Substring(p2 + 1, p3 - p2 - 1);
    this.label9.Text = "V2: \n\n" + rec.Substring(p3 + 1, p4 - p3 - 1);
    this.label10.Text = "V1: \n\n" + rec.Substring(p4 + 1, p5 - p4 - 1);
    this.label11.Text = "V0: \n\n" + rec.Substring(p5 + 1, p6 - p5 - 1);
    }
          else
          {
            this.label1.Text = "Nu vin date!";
            this.label10.Text = " ";
            this.label9.Text = " ";
            this.label8.Text = " ";
            this.label7.Text = " ";
          }
        }
    }
  }
}
```

Se observa ca p1-p6 sunt pozitiile in care se gaseste separatorul "," iar p6 este pozitia in care se termina sirul, respectiv lungimea acestuia. Pozitia este determinata folosind metoda "IndexOf". Pentru a extrage parametrul curent, stiindu-se pozitia celor doi separatori "," adica inceputul si sfarsitul subsirului ce reprezinta parametrul respectiv, s-a folosit metoda "Substring".

- **Comanda iesiri digitale**

Urmatoarea aplicatie utilizeaza iesirile digitale:




```
namespace RS_232_v00
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      string txt, cda;
      int k;
      static String[] Nume_porturi = new String[11];
      private void Form1_Load(object sender, EventArgs e)
      {
         int i, j;
         // Listez porturile seriale
         Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
         this.Porturi_s.Items.Clear();

         //Adaug porturile exixtente

         for (i = 0; i < Nume_porturi.Length; i++)
         {
            this.Porturi_s.Items.Add(Nume_porturi[i]);
         }

         //Pozitionarea listei pe primul element
```

```
        this.Porturi_s.SelectedIndex = 0;
        this.label1.Text = "Conectati-va la portul serial!";
    }

    private void but_con_Click(object sender, EventArgs e)
    {
        if (!this.serialPort1.IsOpen)
        {
            this.serialPort1.PortName = System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
            this.serialPort1.Open();
            this.label1.Text = "Portul a fost deschis";
        }
        else
        {
            this.label1.Text = "Portul este deja deschis deschis";
        }
        this.serialPort1.Write("O77");
    }

    private void but_dec_Click(object sender, EventArgs e)
    {
        this.serialPort1.Close();
        this.label1.Text = "Portul este deconectat!";
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        k++;
        if (k > 123456789)
            k = 0;
        this.label2.Text = k.ToString();

        cda = "O" + System.Convert.ToString(this.numericUpDown1.Value);

        if (this.serialPort1.IsOpen)
        {
            this.serialPort1.Write(cda);
            txt = "";
            txt = this.serialPort1.ReadExisting();
            if (txt.Length > 0)
            {
                this.textBox2.Text = txt;
            }
            else
            {
                this.label1.Text = "Nu vin date!";
            }
        }
    }
  }
}
```

Fisiere

Un fisier, este o colectie de date stocata pe un mediu permanent de stocare (hard disk, memory stik, CD etc)
Datele inscrise in fisiere sunt persistente, pastrandu-se si dupa inchiderea calculatorului.

Tipuri de fisiere

Un fisier este identificat prin nume si extensie. Numele poate fi dat arbitrar iar extensia reflecta de obicei tipul de date si aplicatia ce foloseste datele din fisier. In functie de tipul de date, se pot defini doua tipuri de fisiere: fisiere text sau binare.

- **Fisiere text**

Un fisier text contine informatii sub forma text. Textele contin litere codificate ASCII

- **Fisiere binare**

Un fisier binar poate stoca, pe langa informatii de tip text, orice alt fel de informatii, cum ar fi imagini, baze de date programe executabile etc. Fisierele binare sunt accesate in mod diferit in functie de aplicatia care le deschide. Pentru programatori, cele mai importante fisiere sunt fisierele text, acestea , ofera o metoda standard de accesare fiind simplu de utilizat.

- **Biblioteca standard fstream**

Functiile pentru citirea si scrierea intr-un fisier sunt grupate in biblioteca "fstream" . Bibliotece "fstream" se include in aplicatii folosind :
#include < fstream >
Limbajul nu defineste acest tip de data special pentru ca fisiere. Fisierele reprezinta mai degraba niste fluxuri de intrare / iesire standard puse la dispozitia utilizatorului. Din acest motiv biblioteca se numeste "fstream".
In biblioteca "fstream" se definesc trei tipuri de date:
ofstream - fluxul de iesire
ifstream - fluxul de intrare
fstream - fluxul de intrare/iesire

- **Accesul la fisiere**

Cand se acceseaza un fisire (fie pentru citire fie pentru scriere) se parcurg urmatorii pasi:
se deschide fisierul
se scrie sau se citeste din fisier
se inchide fisierul

- **Deschiderea unui fisier**

Inainte de a se putea scrie sau citi intr-un fisier, el trebuie deschis. Deschiderea unui fisier stabileste o tegatura intre fisier si un obiect flux de date din program. Obiectul flux de date poate fi unul din obiectele "istream" "ostream" sau "stream"

- **Deschiderea unui fisier pentru citire**

Obiectul flux de date folosite pentru deschiderea unui fisier pentru citire, poate fi unul din obiectele "istream" sau "stream".
Deschiderea unui fisier pentru scriere se face invocand metoda open astfel:

```
ifstream fis_scr;
fis_scr.open("nume fisier");
```

S-a deschis deci obiectul de flux fis_scr si s-a invocat metoda open asupra fisierului "nume fis"
Instructiunile de sus pot fi comasate intr-o singura istructiune astfel:

```
ifstream fis_r("nume fisier");
```

- **Deschiderea unui fisier pentru scriere**

Obiectul flux de date folosite pentru deschiderea unui fisier pentru scriere, poate fi unul din obiectele "ostream" sau "stream".
Deschiderea unui fisier pentru scriere se face invocand metoda open astfel:

```
ofstream fis_scr;
fis_scr.open("nume fisier");
```

S-a deschis deci obiectul de flux fis_scr si s-a invocat metoda open asupra fisierului "nume fis"
Deschiderea unui fisier pentru scriere poate fi realizata utilizand o singura instructiune de forma:

```
ofstream fis_w("nume fisier");
```

- **Inchiderea unui fisier**

Obiectul flux de date folosite pentru deschiderea unui fisier pentru scriere, poate fi unul din obiectele "ostream" sau "stream".
Inchiderea unui fisier se face invocand metoda close astfel:

```
ifstrean fis_scr;
fis_scr.open("nume fisier");// s-a deschis fisierul "nume fisier" pentru citire
fis_scr.close();//s-a inchis fisierul
```

- **Citirea unui fisier**

Citirea caracterelor dintr-un fisier se pot face caracter cu caracter si afisarea acestora pe masura ce au fost citite.

```
```

```
// Programul cere numele unui fisier text, il deschide si afiseaza continutul.
#include "stdafx.h"
#include < iostream >
#include < fstream >
using namespace std;

int main(void)
{
system("TITLE Citirea unui fisier text ");
system("COLOR F9");
char c;
char nume_f[80];
  cout << "\n\tSe afiseaza un fisier text\n";
  cout << "\n\tIntroduceti numele fisierului : ";
  cin.getline(nume_f,80);
  ifstream fis_r(nume_f);
  if(!fis_r){
              cout << "\n\tNu se poate deschide fisierul : " << nume_f ;
    cin.get();
    return -1;
   }
  cout << "\n\tContinutul fisierului este: \n\n\n";
  while(fis_r.get(c))
            cout << c;
            fis_r.close();
            cin.get();
  return 0;
}
```

Citirea caracterelor dintr-un fisier se poate face si linie cu linie. In acest caz avem nevoie de un buffer de caractere care pastreaza o intreaga linie. Vom defini un sir de caractere "linie[100]" in care se citeste o linie din fisier, apoi se afiseaza.

```
// Programul cere numele unui fisier text, il deschide si afiseaza continutul.
#include "stdafx.h"
#include < iostream >
#include < fstream >
using namespace std;

int main(void)
{
system("TITLE Citirea unui fisier text linie cu lnie ");
system("COLOR F9");
char c;
char nume_f[80],linie[100];
  cout << "\n\tSe afiseaza un fisier text\n";
  cout << "\n\tIntroduceti numele fisierului : ";
```

```
  cin.getline(nume_f,80);
  ifstream fis_r(nume_f);
  if(!fis_r){
              cout << "\n\tNu se poate deschide fisierul : " << nume_f ;
     cin.get();
     return -1;
   }
  cout << "\n\tContinutul fisierului este: \n\n\n";
  while(!fis_r.eof()){
            fis_r.getline(linie,100);
   cout << linie << '\n';
            }
            fis_r.close();
            cin.get();
  return 0;
}
```

- **Scrierea intr-un fisier**

Scrierea in fisier se poate face caracter cu caracter, pe masura ce ele sosesc de la tastatura.

```
// Programul cere numele unui fisier text, il preia apoi scrie in el.
#include "stdafx.h"
#include < iostream >
#include < fstream >
using namespace std;

int main(void)
{
system("TITLE Crearea si scrierea intr-un fisier ");
system("COLOR F9");
            char nume_f[80];
            char c;
            cout << "\n\tProgramul creaza un fisier\n";
            cout << "\n\tIntroduceti numele fisierului ";
            cin.getline(nume_f,80);
   ofstream fis_w(nume_f);
   if(!fis_w){
                        cout<<"\n\tNu se poate crea fisierul " << nume_f <<'\n';
                        cin.get();
                        return -1;
     }
  cout<<"\n\tIntroduceti caractere.Sfarsit Enter CTRL+Z\n";
  while(cin.get(c))
   fis_w << c;
  fis_w.close();
  cin.clear();
  cout << "\n\tApasati o tasta ";
```

```
  cin.get();
  return 0;
}
```

Caracterele introduse de la tastatura pot fi preluate intr-un buffer si apoi transferate in fisier.

```
// Programul cere numele unui fisier text, il preia apoi scrie in el.
// Caracterele se preiau in bufferul txt
// txt se scrie in fisier.
#include "stdafx.h"
#include < iostream >
#include < fstream >
using namespace std;

int main(void)
{
system("TITLE Crearea si scrierea intr-un fisier ");
system("COLOR F9");
            char nume_f[80];
            char txt[80];
            cout << "\n\tProgramul creaza un fisier\n";
            cout << "\n\tIntroduceti numele fisierului ";
            cin.getline(nume_f,80);
   ofstream fis_w(nume_f);
   if(!fis_w){
                        cout<<"\n\tNu se poate crea fisierul " << nume_f <<'\n';
                        cin.get();
                        return -1;
     }
  cout<<"\n\tIntroduceti caractere.";
  cin.getline(txt,80);
  fis_w << txt;
  fis_w.close();
  cin.clear();
  cout << "\n\tApasati o tasta ";
  cin.get();
  return 0;
}
```

Fisiere in spatiul System

- **Scrierea intr-un fisier in spatiul System**

Sa realizam pentru inceput o aplicatie CLR in spatiul System, care creaza un fisier text si scrie in el.

```
// Aplicatie CLR in spatiul System
// Programul scrie un text intr-un fisier .
#include "stdafx.h"
#include < iostream >
using namespace std;
using namespace System;
using namespace System::IO;

int main()
{
            system("TITLE Crearea si scrierea intr-un fisier ");
            system("COLOR F9");
            String^ nume_f;
            Console::WriteLine("\n\tIntroduceti numele fisierului:");
            nume_f=Console::ReadLine();
            StreamWriter^ sw = gcnew StreamWriter(nume_f);
            sw->WriteLine("Text scris in fisier!");
   sw->WriteLine(DateTime::Now);
   sw->Close();
            Console::WriteLine("\n\n\tFisierul:" +nume_f+" a fost creat!" );
   cin.get();
  return 0;
}
```

In CLR, Windows Forms Application se pot realiza operatii cu fisiere folosind aceleasi instructiuni pe care le-am utilizat anterior.Vom realiza deci o aplicatie Windows Forms Application numita **fisier_v0** care permite cautarea crearea si scrierea intr-un fisier.

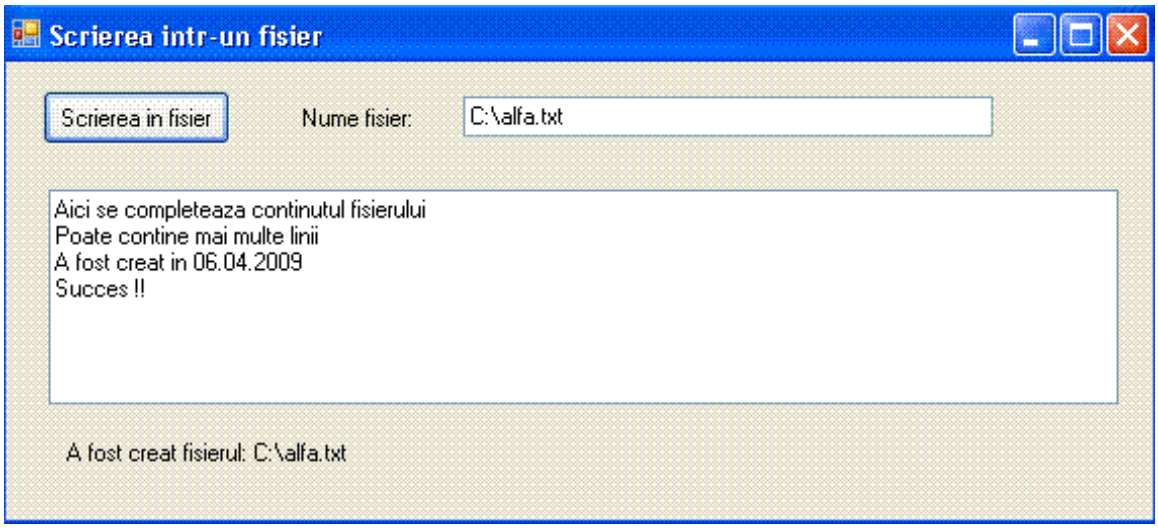
- Plasam un obiect de tip button numit button1 apoi schimbam proprietatea "Text" in "Scriere fisier"
- Plasam un obiect de tip textBox numit textBox1 in care vom scrie numele fisierului. Se poate schimba atributul text cu un nume initial de fisier care va putea fi pastrat sau schimbat in timpul rularii programului.
- Plasam un obiect de tip textBox numit textBox2 in care vom scrie continutul fisierului. Se poate schimba atributul text cu continutul initial al fisierului care va putea fi pastrat sau schimbat in timpul rularii programului. Se seteaza proprietatea "Multiline" la true.
- Plasam un obiect de tip label numit label1, in care vom afisa un mesaj de terminare operatiune de scriere..
- Deoarece vom lucra si cu spatiul de nume *System::IO* , vom completa namespace fisier_v0 cu:

using namespace System::IO;

Pe evenimentul click al obiectului button1 punem procedura :

```
                System::IO::StreamWriter^ fis_w;
                fis_w=  gcnew System::IO::StreamWriter(this->textBox1->Text);
                fis_w->WriteLine(this->textBox2->Text);
                fis_w->Close();
                this->label1->Text="A fost creat fisierul: "+this->textBox1->Text;
```

Rulam aplicatia si obtinem:



- **Citirea unui fisier in spatiul System**

Urmatoarea aplicatie deschide un fisier text si afiseaza continutul.

```
// Programul citeste un text dintr-un fisier .
// Aplicatie CLR in spatiul System
// Aplicatia nu trateaza eventuale erori la deschiderea fisierului.
#include "stdafx.h"
#include < iostream >
using namespace std;
using namespace System;
using namespace System::IO;
int main()
{
            system("TITLE Citirea unui fisier text ");
            system("COLOR F9");
            String^ nume_f;
            String^ text;
            Console::Write("\n\tIntroduceti numele fisierului:");
            nume_f=Console::ReadLine();
            StreamReader^ sr= gcnew StreamReader(nume_f);
            int nr_lin = 0;
            Console::WriteLine("\n\n\tContinutul fisierului:"+nume_f+" este:\n\n" );
            while ((text = sr->ReadLine()) != nullptr) {
      nr_lin++;
                        Console::WriteLine("\tlinia :"+nr_lin+"\t"+text );
    }
            sr->Close();
   cin.get();
  return 0;
}
```

Programul anterior, nu tine cont de eventualele erori in caz ca fisierul nu este gasit sau nu se pot citi date din fisier. Vom relua aplicatia si vom trata aceste cazuri.

```
// Programul citeste un text dintr-un fisier .
// Aplicatie CLR in spatiul System
// Aplicatia trateaza eventuale erori la deschiderea fisierului.
#include "stdafx.h"
#include < iostream >
using namespace std;
using namespace System;
using namespace System::IO;

int main()
{
            system("TITLE Citirea unui fisier text ");
            system("COLOR F9");
            String^ nume_f;
            String^ text;
            try {
            Console::Write("\n\tIntroduceti numele fisierului:");
            nume_f=Console::ReadLine();
            StreamReader^ sr = gcnew StreamReader(nume_f);
            int nr_lin = 0;
            Console::WriteLine("\n\n\tContinutul fisierului:"+nume_f+" este:\n\n" );
            while ((text = sr->ReadLine()) != nullptr) {
       nr_lin++;
                        Console::WriteLine("\tlinia :"+nr_lin+"\t"+text );
    }
            sr->Close();
            }
            catch (Exception^ e){
    if (dynamic_cast < FileNotFoundException^ > (e))
      Console::WriteLine("\n\n\tFisierul "+nume_f+" nu a fost gasit!");
    else
                        Console::WriteLine("\n\n\tNu se pot citi date din fisierul: "+nume_f);
            }
   cin.get();
  return 0;
}
```

In aplicatiile in care se doreste deschiderea unui fisier si afisarea continutului ar fi util sa completam aplicatia cu o aplicatie ce ne permite cautarea fisierului. Acest lucru este posibil numai in aplicatii de genul Windows Forms Application in care se poate crea o instanta a unei clase care realizeaza cautarea fisierelor in mod interactiv. Vom realiza deci o aplicatie Windows Forms Application numita **fisier_v1** care permite cautarea fisierului si afisarea continutului acestuia (primele 500 de caractere).

- Plasam un obiect de tip button numit button1 apoi schimbam proprietatea "Text" in "Citire fisier"
- Plasam un obiect de tip openFileDialog numit openFileDialog1.
- Plasam un obiect de tip label numit label1, in care vom afisa continutul fisierului. Completam #pragma region cu :

```
static System::String^ alfa;
static System::String^ sir_c;
static System::IO::Stream^ stream_r;
static array < unsigned char,1 >^ text = gcnew array < unsigned char,1 > (500);
static int i;
```

Pe evenimentul click al obiectului button1 punem procedura :

C++

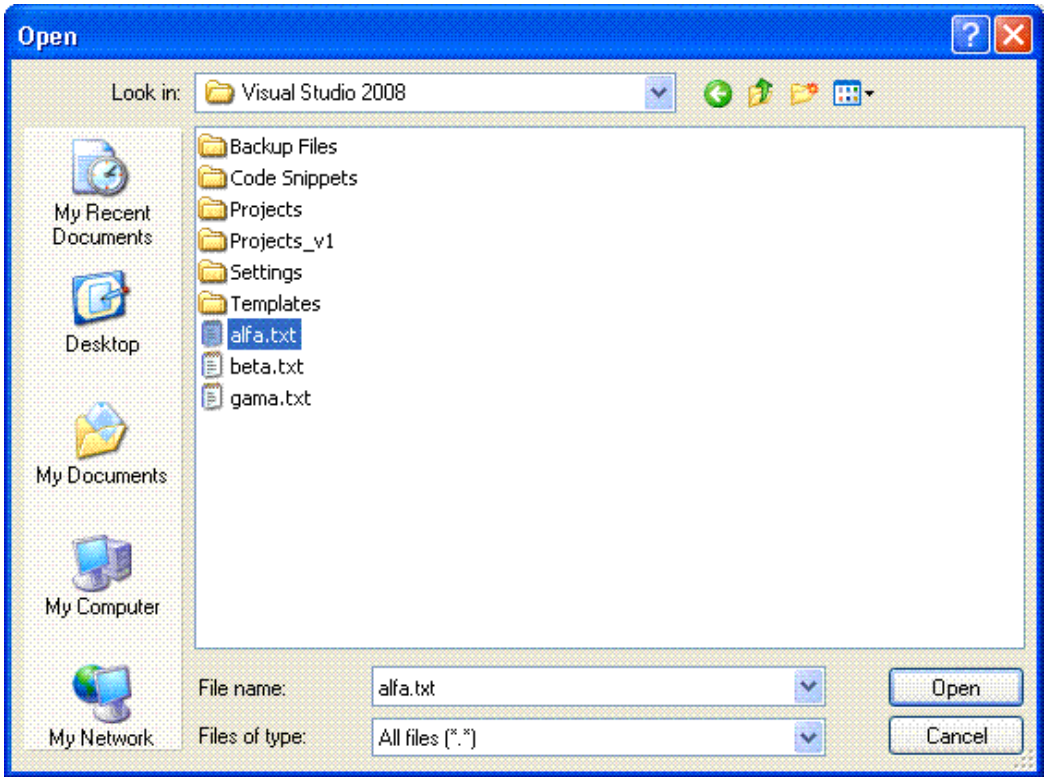
```
OpenFileDialog^ openFileDialog1 = gcnew OpenFileDialog;
openFileDialog1->InitialDirectory = "c:\\";
openFileDialog1->Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
openFileDialog1->FilterIndex = 2;
openFileDialog1->RestoreDirectory = true;

if ( openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK )
{
  if ( (stream_r = openFileDialog1->OpenFile()) != nullptr )
  {
                              stream_r->Read(text,1,499);
                              sir_c=" ";
                              for (i=1 ; i < 499; i++){
                              alfa=System::Convert::ToString(System::Convert::ToChar(text[i]));
                              sir_c+=alfa;
                              }
                              this->label1->Text=sir_c;
    stream_r->Close();
  }
}
```

C#

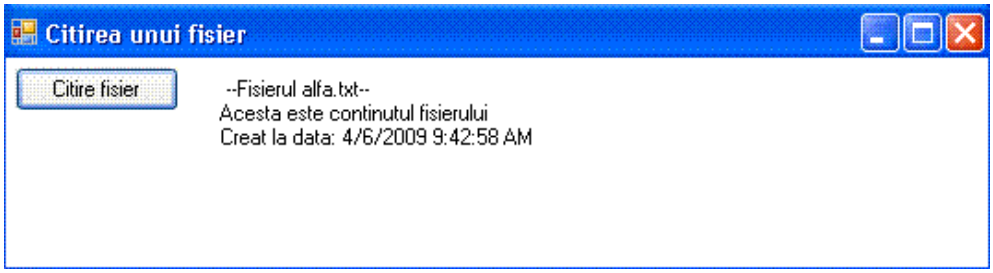
```
    String alfa;
    String sir_c;
    System.IO.Stream stream_r;
    byte[] text = new byte[500];
    int i;
    int lung=1;
    private void button1_Click(object sender, EventArgs e)
```

```
 {
    OpenFileDialog openFileDialog1 = new OpenFileDialog();
    openFileDialog1.InitialDirectory = "c:\\";
    openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    openFileDialog1.FilterIndex = 2;
    openFileDialog1.RestoreDirectory = true;

if ( openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK )
{
  stream_r = openFileDialog1.OpenFile();
  if (stream_r!=null)
  {

    stream_r.Read(text,1,499);
                          sir_c=" ";
                          for (i=1 ; i < 499; i++){
                          alfa=System.Convert.ToString(System.Convert.ToChar(text[i]));
                          sir_c+=alfa;
                          }
    this.textBox1.Text = sir_c;
    stream_r.Close();
  }
}
```

Dupa rularea aplicatiei si apasarea butonului citire fisier, obtinem o fereastra de dialog ce ne permite sa selectam fisierul dorit.



Sa presupunem ca am selectat un fisier text "alfa.txt". Dupa selectia fisierului, se afiseaza continutul acestuia :



Utilizarea fisierelor ca suport de date

In diverse aplicatii pentru controlul si monitorizarea proceselor tehnologice se folosesc fisiere ca suport pentru datele achizitionate. Fisierele sunt utilizate pentru realizarea "istoricelor" de date fie pentru partajarea datelor intre diverse aplicatii.

Sa realizam o aplicatie Windows Forms Application numita **fisier_v2** care permite genereaza un set de date, le afiseaza si totodata le salveaza intr-un fisier text. Majoritatea protocoalelor de comunicatii intre diverse sisteme sunt protocoale de tip text. Datele si comenzile sunt codificate la nivel de caracter. Din acest motiv vom folosi fisiere de tip text pentru suportul de date. Fisierele text, contin caractere codificate ASCII in setul extins. Caracterele ASCII din setul extins sunt reprezentate pe un octet deci valoarea lor se incadreaza in domeniul 0-255. Orce valoare de orce tip poate fi codificata folosind 1 sau mai multi octeti deci poate fi codificata folosind 1 sau mai multe caractere .

In urmatorul exemplu vom genera valori intre 0 si 255 deci vom codifica datele utilizand cate un caracter pe cate o valoare.

Dupa generarea unui set de valori acestea vor fi afisate grafic si salvate intr-un fisier. Aplicatia poate deschide fisierul creat sau alte fisiere care contin date, create anterior, si sa afiseze datele din aceste fisiere.

Generam un nou proiect de tipul Windows Forms Application numita **fisier_v2** . In cadrul acestei aplicatii se vor genera valori intre 0 si 255. La afisarea datelor nu vom face scalare in functie de inaltimea form-ului asa ca vom seta proprietatea "Size-Height" la cel putin 359 px.

- Plasam un obiect de tip button numit button1 apoi schimbam proprietatea "Text" in "Generare valori"
- Plasam un obiect de tip button numit button2 apoi schimbam proprietatea "Text" in "Scriere in fisier". Atribuim proprietatii "Enabled" valoarea false.
- Plasam un obiect de tip button numit button3 apoi schimbam proprietatea "Text" in "Citire din fisier"
- Plasam un obiect de tip button numit button4 apoi schimbam proprietatea "Text" in "Sterg"
- Plasam un obiect de tip textBox numit textBox1 in care vom scrie numele fisierului. Se poate schimba atributul text cu un nume initial de fisier care va putea fi pastrat sau schimbat in timpul rularii programului.
- Plasam un obiect de tip label numit label1, in care vom afisa diverse mesaje .
- Deoarece vom lucra si cu spatiul de nume System::IO , vom completa namespace fisier_v0 cu:

using namespace System::IO;

Completam deci #pragma region cu :

```
        static int nr_max;
        static unsigned char val;
        static array < unsigned char,1 >^ valori = gcnew array < unsigned char,1 > (0);
        System::Drawing::Graphics^ Desen;
```

```
System::Drawing::Pen^ Creion_albastru;
System::Drawing::Pen^ Creion_rosu;
```

Pe evenipentul "Paint" al form-ului, punem: br>

```
nr_max=1+this->Width/20;
Desen = this->CreateGraphics();
Creion_albastru=gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
Creion_rosu=gcnew System::Drawing::Pen(System::Drawing::Color::Red);
```

Pe evenimentul click al obiectului button1 punem procedura :

```
int i=0,x,y,x_v,y_v;
valori = gcnew array < unsigned char,1 > (nr_max);
System::Random^ n = gcnew System::Random();
Desen->Clear(System::Drawing::Color(this->BackColor));
x_v=0;
val=n->Next(255);
valori[0]=val;
y_v=this->Height-50-val;
i=1;
for ( x=20; x <= this->Width; x+=20,i++){
        val=n->Next(255);
        valori[i]=val;
        y=this->Height-50-val;
        Desen->DrawLine( Creion_albastru,x_v,y_v,x,y);
        x_v=x;
        y_v=y;
}
this->button2->Enabled=true;
```

Pe evenimentul click al obiectului button2 punem procedura :

```
System::IO::StreamWriter^ fis_w;
fis_w=  gcnew System::IO::StreamWriter(this->textBox1->Text);
for (int i=0; i < nr_max; i++)
fis_w->Write(valori[i]);
fis_w->Close();
this->label1->Text="A fost creat fisierul: "+this->textBox1->Text;
```

Pe evenimentul click al obiectului button3 punem procedura :

```
        int i=0,x,y,x_v,y_v;
        System::IO::StreamReader^ fis_r;
        fis_r=  gcnew System::IO::StreamReader(this->textBox1->Text);
        val=fis_r->Read();
        val=fis_r->Read();
        y_v=Height-50-val;
        x_v=0;
        for ( x=20; x <= this->Width; x+=20){
                val=fis_r->Read();
                y=this->Height-50-val;
                Desen->DrawLine( Creion_rosu,x_v,y_v,x,y);
                x_v=x;
                y_v=y;
        }
        fis_r->Close();
        this->label1->Text="A fost citit: "+this->textBox1->Text;
```

Pe evenimentul click al obiectului button4 punem procedura :

```
        Desen->Clear(System::Drawing::Color(this->BackColor));
```

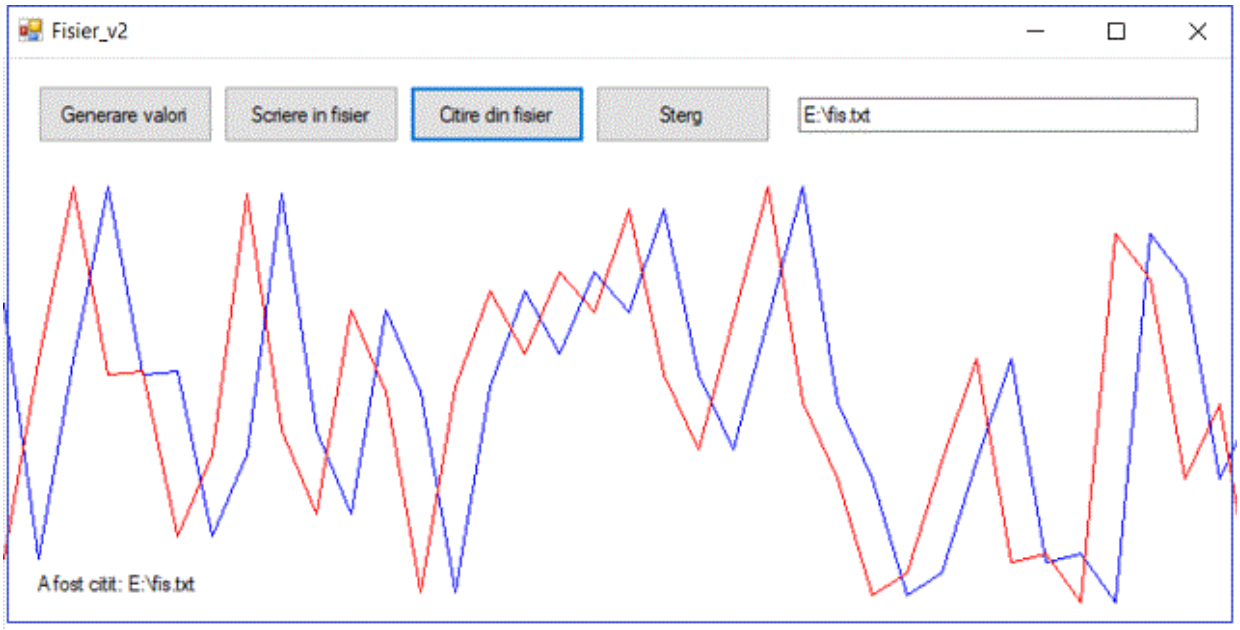
C#

```
namespace fisier_v2
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
        static int nr_max;
        Char val;
        static Char[] valori = new Char[0];
        System.Drawing.Graphics Desen;
        System.Drawing.Pen Creion_albastru;
        System.Drawing.Pen Creion_rosu;

  private void Form1_Paint(object sender, PaintEventArgs e)
  {
        nr_max=1+this.Width/20;
        Desen = this.CreateGraphics();
        Creion_albastru=new System.Drawing.Pen(System.Drawing.Color.Blue);
```

```
        Creion_rosu=new System.Drawing.Pen(System.Drawing.Color.Red);
}

private void button1_Click(object sender, EventArgs e)
{
        int i=0,x,y,x_v,y_v;
        Array.Resize(ref valori,nr_max+1);
        System.Random n = new System.Random();
        Desen.Clear(this.BackColor);
        x_v=0;
        val=System.Convert.ToChar(n.Next(255));
        valori[0]=val;
        y_v=this.Height-50-val;
        i=1;
        for ( x=20; x <= this.Width; x+=20,i++){
                val=System.Convert.ToChar(n.Next(255));
                valori[i]=val;
                y=this.Height-50-val;
                Desen.DrawLine( Creion_albastru,x_v,y_v,x,y);
                x_v=x;
                y_v=y;
        }
        this.button2.Enabled=true;
}

private void button2_Click(object sender, EventArgs e)
{
        System.IO.StreamWriter fis_w;
        fis_w= new System.IO.StreamWriter(this.textBox1.Text);
        for (int i=0; i <= nr_max; i++)
        fis_w.Write(System.Convert.ToChar(valori[i]));
        fis_w.Close();
        this.label1.Text="A fost creat fisierul: "+this.textBox1.Text;
}

private void button3_Click(object sender, EventArgs e)
{
        int i=0,x,y,x_v,y_v;
        System.IO.StreamReader fis_r;
        fis_r=new System.IO.StreamReader(this.textBox1.Text);
        val = System.Convert.ToChar(fis_r.Read());
        val = System.Convert.ToChar(fis_r.Read());
        y_v=Height-50-val;
        x_v=0;
        for ( x=20; x <= this.Width; x+=20){
                val = System.Convert.ToChar(fis_r.Read());
                y=this.Height-50-val;
                Desen.DrawLine( Creion_rosu,x_v,y_v,x,y);
                x_v=x;
                y_v=y;
        }
        fis_r.Close();
```

```
            this.label1.Text="A fost citit: "+this.textBox1.Text;
    }

    private void button4_Click(object sender, EventArgs e)
    {
       Desen.Clear(this.BackColor);
    }
  }
}
```

Rulam aplicatia, generam valori, scriem in fisier, citim din fisier si obtinem:



Dupa cum se vede in imaginea de sus, datele citite din fisier sunt reprezentate cu rosu si sunt decalate cu o pozitie.

Decalarea s-a facut intentionat in sectiunea citire din fisier prin aplalrea de doua ori a citirii din fisier la inceputul procedurii adica:

val=fis_r.Read();
val=fis_r.Read();

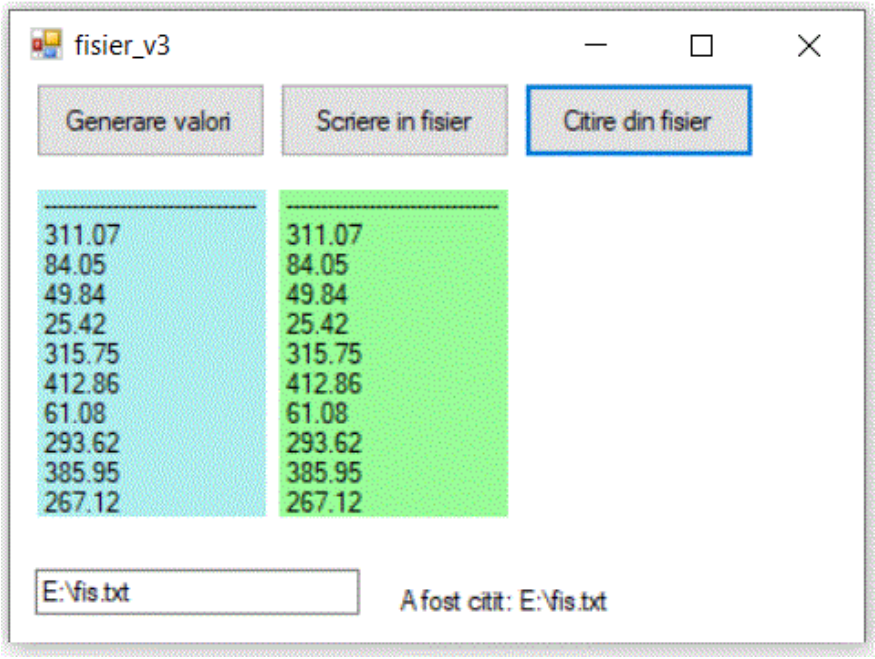
In aplicatia anterioara am creat un fisier in care am scris si am citit caracter cu caracter.

In multe aplicatii e nevoie sa salvam date de tipuri diverse. Sa realizam o aplicatie care genereaza aleator 10 numere de tip double, pe care le convertste intr-un text si le salveaza sub forma de linii intr-un fisier text, folosind: **WriteLine()**;dupa care citeste linie cu linie folosind **ReadLine();**

```
namespace fisier_v3
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      static int nr_max;
      double val;
      String txt;
      static String[] valori = new String[0];
      System.Drawing.Graphics Desen;
      System.Drawing.Pen Creion_albastru;
      System.Drawing.Pen Creion_rosu;
      private void Form1_Load(object sender, EventArgs e)
      {
         nr_max = 10;
         Desen = this.CreateGraphics();
         Creion_albastru = new System.Drawing.Pen(System.Drawing.Color.Blue,4);
         Creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red,4);
         Array.Resize(ref valori, nr_max + 1);
      }

      private void button1_Click(object sender, EventArgs e)
      {
         int i = 0;
         txt = "-------------------------------";
         System.Random n = new System.Random();
         i = 0;
         for (i = 0; i < nr_max; i++)
         {
            val = n.Next(50000)/100.0;
            valori[i] = val.ToString();
            txt = txt + "\n" + val.ToString();
         }
         this.label1.Text = txt;
         this.button2.Enabled = true;
      }
      private void button2_Click(object sender, EventArgs e)
      {
         System.IO.StreamWriter fis_w;
         fis_w = new System.IO.StreamWriter(this.textBox1.Text);
         for (int i = 0; i < nr_max; i++)
            fis_w.WriteLine(valori[i]);
         fis_w.Close();
         this.label3.Text = "A fost creat fisierul: " + this.textBox1.Text;
      }
      private void button3_Click(object sender, EventArgs e)
      {
         int i = 0;
```

```
        txt = "-------------------------------";
        System.IO.StreamReader fis_r;
        fis_r = new System.IO.StreamReader(this.textBox1.Text);
        for (i = 0; i < nr_max; i++)
        {
           val = System.Convert.ToDouble(fis_r.ReadLine());
           txt = txt + "\n" + val.ToString();
        }
        fis_r.Close();
        this.label2.Text = txt;
        this.label3.Text = "A fost citit: " + this.textBox1.Text;
     }
   }
}
```

Rulam aplicatia, generam valori, scriem in fisier, citim din fisier si obtinem:



Bibliografie

[1].Traian Turc, Elemente de programare C++ utile in ingineria electrica, Ed.Matrixrom, Bucuresti,2010.
[2].Traian Turc, Programare avansata C++ pentru ingineria electrica, Ed.Matrixrom, Bucuresti,2010.
[3].Traian Turc, Programarea calculatoarelor si limbaje de programare 1, Univ.'Petru Maior' , Tg.Mures,2009.
[4].Traian Turc, Programarea calculatoarelor si limbaje de programare 2, Univ.'Petru Maior' ,Tg.Mures, 2009.
[5]. http://www.science.upm.ro/~traian -Cursuri informatica, 2021.
[6].Catrina,O.,Cojocaru, I., Turbo C++, Ed.Teora, Bucuresti, 1993.
[7].Cozac,I., Programare în limbajul C, uz intern, Univ.'Petru Maior', Tg.Mures, 2004.
[8].Kernighan,B.W., Ritchie,D.M., The C programming language, Prentice Hall, 1988.
[9].Petrovici,V., Programarea în limbajul C, Ed.Tehnica, Bucuresti, 1993.
[10].Runceanu,A., Programarea si utilizarea calculatoarelor.Limbajul C++. Ed.Academica Brâncusi, Tg.Jiu, 2003.
[11].Stefanescu,D., Segal,C., Initiere in limbajele C/C++,Ed.Fundatiei Universitare'Dunarea de Jos' Galati, 2000.
[12].Mircea Popovici,Tehnologia orientata pe obiecte.Aplicatii, Ed.Teora, 1996.
[13].Namir C. Shammas, Curs rapid de Borland C++, Ed.Teora, 1996.
[14].Jeff Kent, C++ fara mistere,Ed.Rosetti Educational, 2004.