

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 in 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 programarea 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 $P=U \cdot I$ ".
- 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;
}
```

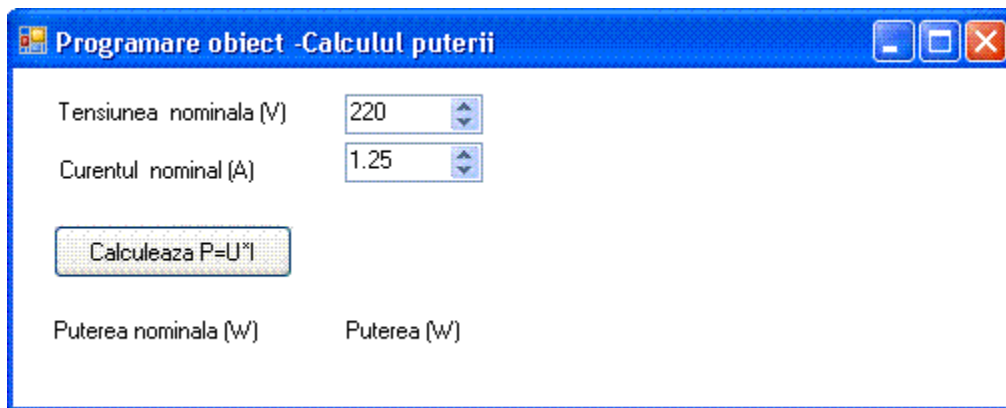
• Crearea obiectelor

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 $P=U \cdot I$ " 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 operatorul "->".

Se observa de asemenea ca "a" este un pointer negestionat(a fost definit cu **ap_electric* 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 aplicatie 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 importanta 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. In acest 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";

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

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);
    }

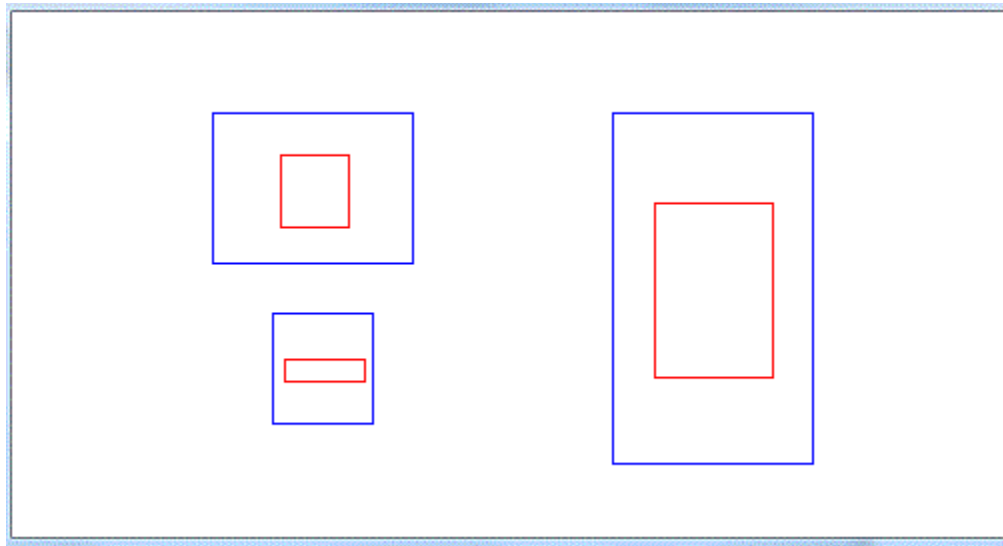
    private void timer1_Tick(object sender, EventArgs e)
    {
        drc.setdr(desen, creion_rosu, creion_albastru, radiera,
nr.Next(100 - 3), nr.Next(75 - 3));
    }
    public class drcon
    {
        int x0, y0, w0, h0;

        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



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

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 drcl;
        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();
        drcl = 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)
    {
        drcl.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);
        }
    }
}

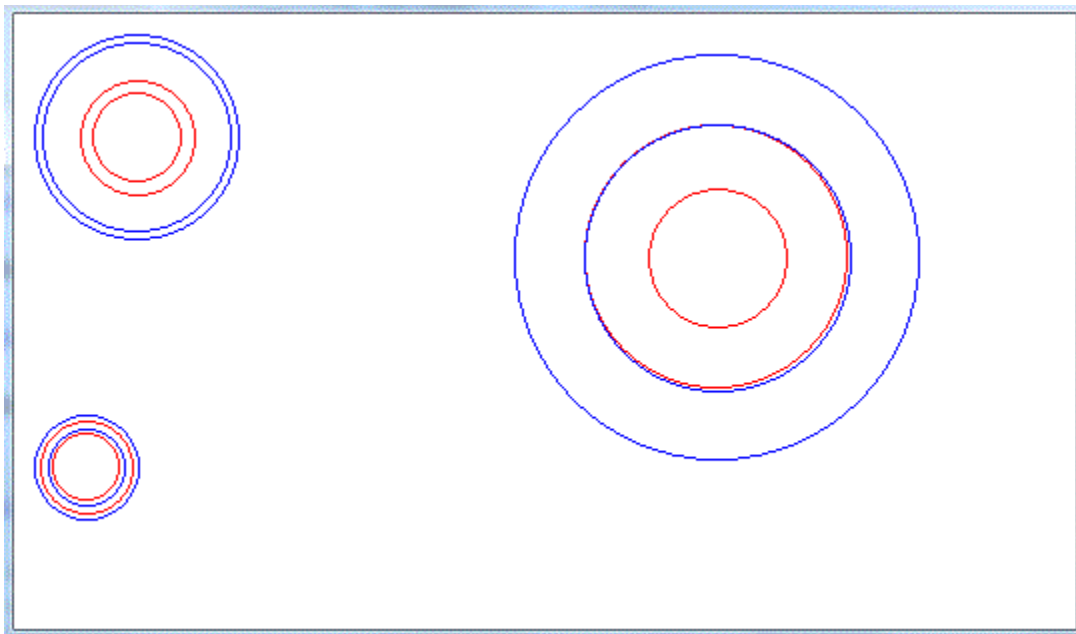
```

```

        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 realiza 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));
    }
}

```

```

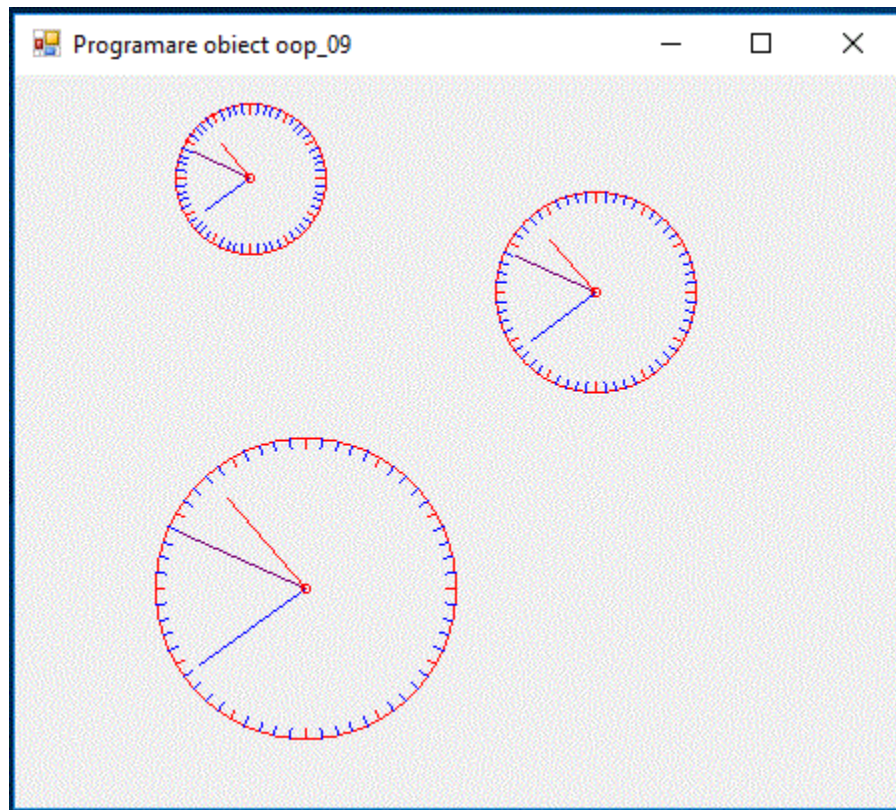
        zona_des.FillEllipse(radiera, x + 5, y + 5, d - 10, d - 10);
        zona_des.DrawLine(c_sec, x1, y1, x3, y3);
        //minutar
        i = 90 - 6 * System.DateTime.Now.Minute;
        r = (2 * System.Math.PI * i) / 360;
        x3 = Convert.ToInt16(x1 + (d / 2 - 10) *
System.Math.Cos(r));
        y3 = 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);
    }

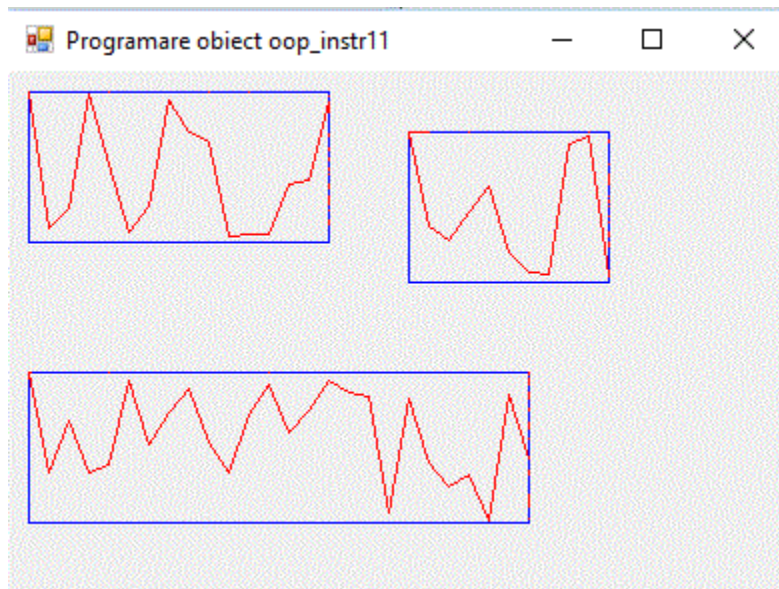
    public void setval(System.Drawing.Graphics zona_des,
System.Drawing.Pen creion, float[] vals, int nrv)
    {
        float val_v, val;
        val_v = 0;
        for (int i = 1; i <= nrv; i++)
        {

```

```

        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);
    }

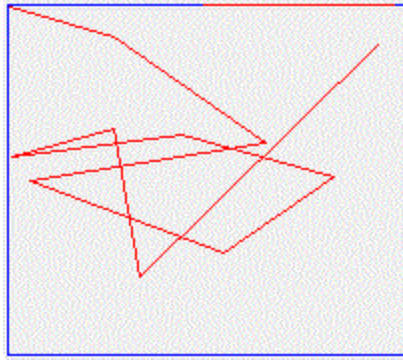
    public void setval(System.Drawing.Graphics zona_des,
System.Drawing.Pen creion, float[] valsx, float[] valsy, int nrv)
    {
        float val_vx, valx, val_vy, valy;
        val_vx = 1;
        val_vy = 1;
        for (int i = 1; i <= nrv; i++)
        {
            valx =
System.Convert.ToInt16(System.Convert.ToDouble(valsx[i]) *
(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 $\sin(i)$ $\cos(k \cdot i)$

```
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++)
        {

```

```

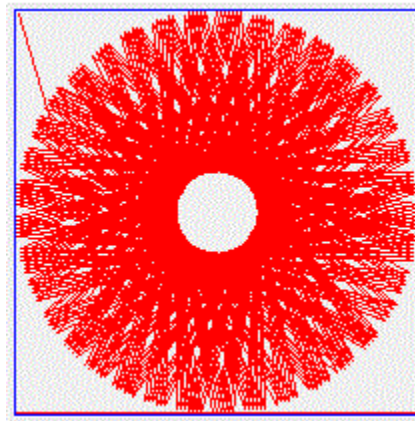
        valx =
System.Convert.ToInt16(System.Convert.ToDouble(valsx[i]) *
(System.Convert.ToDouble(w) / System.Convert.ToDouble(val_maxx))); //scalare
        valy =
System.Convert.ToInt16(System.Convert.ToDouble(valsy[i]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_maxy))); //scalare

        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

```

```

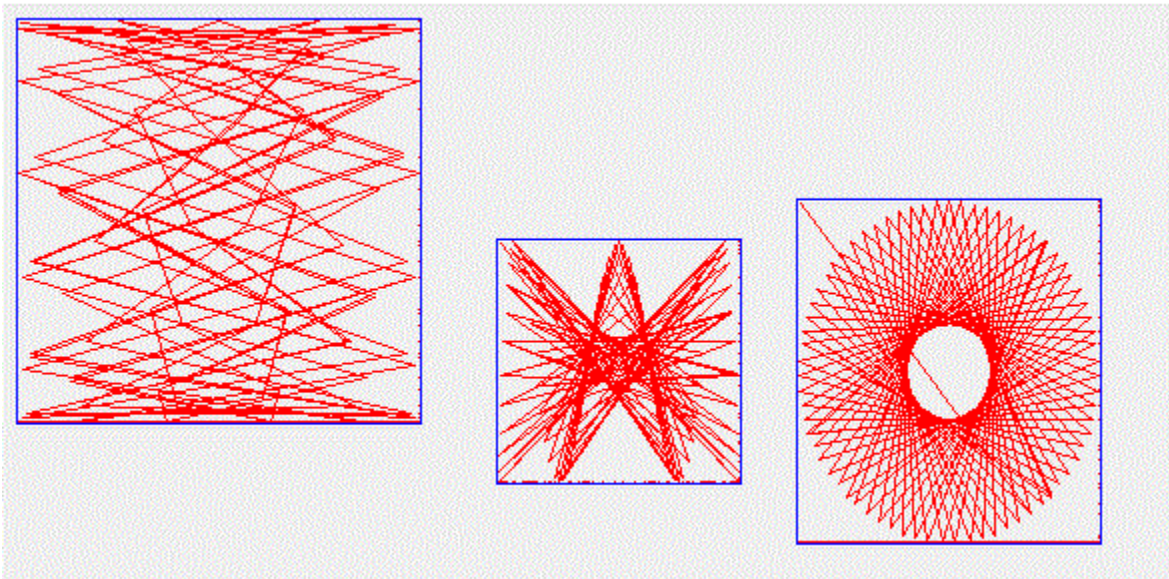
        valy =
System.Convert.ToInt16(System.Convert.ToDouble(valsy[i]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_maxy))); //scalare

        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;

}
}
}

```



Tablouri de obiecte

Claselor create anterior se instantia de cate ori dorim obtinandu-se unul sau mai multe obiecte.

Se pot crea chiar tablouri de obiecte in vederea gestionarii mai usoare a acestora. In continuare vom crea noi obiecte si vom crea tablouri cu aceste obiecte.

- **Realizarea unui obiect grafic-dreptunghi "dr"**

Vom realiza in continuare o aplicatie grafica in care este utilizat un obiect pe care il vom denumi "dr" obiect care este reprezentat grafic sub forma unui dreptunghi, are numeroase metode printre care cea mai importanta este metoda "deplasez" metoda cu care dreptunghiul se afiseaza in alta pozitie. Folosind aceasta metoda vom putea realiza o aplicatie in care vom simula miscarea unui dreptunghi.

Generam un nou proiect de tipul Windows Forms Application numit **"oop_v3"**.

Metoda "deplasez" va fi invocata la intervale regulate de timp mentru a putea simula miscarea dreptunghiului. Vom plasa deci un obiect de tip timer numit timer1 si setam proprietatea interval la 10 ms si Enabled la "true".

Vom scrie pentru inceput clasa "dr", de tip "ref" pe care o plasam in fisierul "stdafx.h". Clasa fiind declarata de tip "ref", vom putea crea instante gestionate, folosind "gcnew".

```
ref class dr {
public:
    dr(int);
    System::Double pozX;
    System::Double pozY;
    System::Double vX;
    System::Int16 w;
    System::Drawing::Pen^ creion_a;
    System::Random^ nr;
    int desenez (System::Drawing::Graphics^ zona_des)
    {
        zona_des->
>DrawRectangle(creion_a, System::Math::Round(pozX), System::Math::Round(pozY), w, w);
        return 1;
    }
    void deplasez ()
    {
        pozX += vX;

        if(pozX > 440-w) {
            vX = -vX;
            w = nr->Next(40)+2;
            pozX =440-w;
        }
        if(pozXNext(40)+2;
        pozX =0;
    }

    void init_dr (int r)
    {
        nr = gcnew System::Random(r);
        pozX = nr->Next(400);
    }
}
```

```

        pozY = nr->Next(300);
        vX = nr->Next(10)+1;
        w = nr->Next(40);
        creion_a =gcnew
System::Drawing::Pen(System::Drawing::Color::Blue);
    }
};

```

Dupa cum se obbserva clasa "dr" are o serie de functii membru cum ar fi:

- init_dr - functie care initializeaza parametrii unui dreptungi (pozitia , dimensiunea, culoarea, viteza de deplasare)
- desenez - functie care deseneaza efectiv dreptungiul cu caracteristicile setate de functia init_dr
- deplasez - functie care deseneaza un nou dreptungi pe o noua pozitie deplasata cu valoarea vX fata de pozitia anterioara

Pentru simplitate, zona de desenare a fost fixata la 400 X 300 si latimea dr maxim 40. Aplicatia poate fi generalizata pentru dimensiunile curente ale form-ului curent.

Toate functiile membru au fost incluse in clasa "dr", mai putin constructorul "dr", constructor plasat in fisierul stadfx.cpp al carui continut se poate vedea mai jos:

```

#include "stdafx.h"
dr::dr(int i)
{
    init_dr(i);
}

```

Pentru a putea utiliza un obiect din clasa "dr" vom defini un obiect d: **dr^ d**; si un desen pe care sa se afiseze definit cu **System::Drawing::Graphics^ desen**; . Vom plasa aceste declaratii la sfarsitul zonei #pragma region, deci dupa declaratia #pragma endregion:

```

System::Drawing::Graphics^ desen;
dr^ d;

```

Dupa activarea form-ului curent trebuie sa creem obiectele d si desen. Vom completa procedura deschisa pe eveniment "activated" al form-ului cu:

```

desen = this->CreateGraphics();
d = gcnew dr(1);

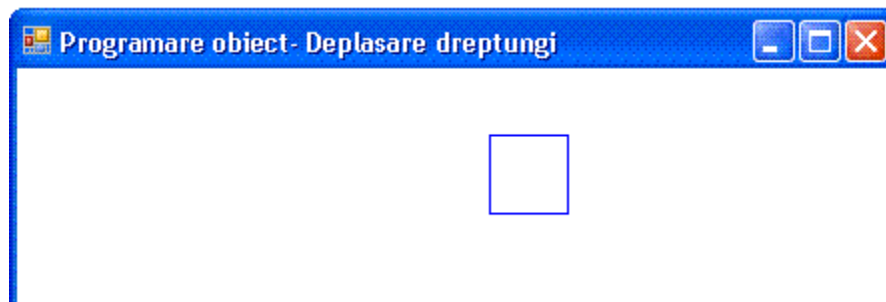
```

Miscarea dreptunghiului (desenarea pe o noua pozitie se va face pe fiecare eveniment tick al obiectului timer1.

Vom completa deci procedura deschisa pe eveniment tick al obiectului timer1 cu:

```
desen->Clear(System::Drawing::Color::White);  
d->desenez(desen);  
d->deplasez();
```

Rulam aplicatia si obtinem un dreptunghi care se deplaseaza. La atingerea pozitiei 0 sau 400 dreptunghiul isi schimba dimensiunile si directia de deplasare conform functiei membru deplasez.



C# Aplicatia "oop_v30"

- deplasarea dreptunghiului la apasarea unui buton

```
namespace oop_v30  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
        public System.Drawing.Graphics desen;  
        public dr d;  
        private void button1_Click(object sender, EventArgs e)  
        {  
            d.deplasez();  
            d.desenez(desen);  
        }  
  
        private void Form1_Load(object sender, EventArgs e)  
        {  
            desen = this.CreateGraphics();  
        }  
    }  
}
```

```

        d = new dr();
        d.init_dr();
    }

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

public class dr
{
    float pozX;
    float pozY;
    float vX;
    float w;
    System.Drawing.Pen creion_a;
    System.Random nr;
    public void desenez(System.Drawing.Graphics zona_des)
    {
        zona_des.Clear(System.Drawing.Color.White);
        zona_des.DrawRectangle(creion_a, pozX, pozY, w, w);
    }

    public void deplasez()
    {
        pozX += vX;
        if (pozX > 440 - w)
        {
            vX = -vX;
            w = nr.Next(40) + 2;
            pozX = 440 - w;
        }
        if (pozX < 0)
        {
            vX = -vX;
            w = nr.Next(40) + 2;
            pozX = 0;
        }
    }

    public void init_dr()
    {
        nr = new System.Random();
        //pozX = nr.Next(400);
        //pozY = nr.Next(300);
        //vX = nr.Next(10) + 1;
        pozX = 10;
        pozY = 100;
        vX = 27;
        w = nr.Next(30)+10;
        creion_a = new System.Drawing.Pen(System.Drawing.Color.Blue);
    }
}
}

```

C# Aplicatia "oop_v31"

- deplasarea dreptunghiului la intervale prestabilite

```
namespace oop_v31
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public System.Drawing.Pen creion_a;
        public dr d;
        private void Form1_Load(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            creion_a = new System.Drawing.Pen(System.Drawing.Color.Blue);
            d = new dr();
            d.init_dr();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            desen.Clear(System.Drawing.Color.White);
            d.deplasez();
            d.desenez(desen, creion_a);
        }
    }
    public class dr
    {
        int pozX, pozY, vX, w;
        System.Random nr;
        public void desenez(System.Drawing.Graphics zona_des,
            System.Drawing.Pen creion)
        {
            zona_des.DrawRectangle(creion, pozX, pozY, w, w);
        }

        public void desplasez()
        {
            pozX += vX;
            if (pozX > 440 - w)
            {
                vX = -vX;
                w = nr.Next(40) + 2;
                pozX = 440 - w;
            }
            if (pozX < 0)
            {

```

```

        vX = -vX;
        w = nr.Next(40) + 2;
        pozX = 0;
    }
}

public void init_dr()
{
    nr = new System.Random();
    pozX = nr.Next(400);
    pozY = nr.Next(200);
    vX = nr.Next(50) + 1;
    w = nr.Next(40);
}
}
}

```

- **Utilizarea clasei grafice "dr" pentru a afisa mai multe dreptunghiuri**

Vom modifica aplicatia anterioara "oop_v3" realizand o noua aplicatie Windows Forms Application numit **"oop_v4"**.

Noua aplicatie difera de prima prin faptul ca vom genera mai multe dreptunghiuri care se vor deplasa pe form-ul curent.

Vom utiliza aceeasi clasa "dr" deci continutul fisierelor "stdafx.h" si "stdafx.cpp" va fi acelasi cu cel din aplicatia "oop_v3". Dupa declaratia #pragma endregion: vom plasa urmatorul continut:

```

System::Drawing::Graphics^ desen;
array < dr^>^ drr;
System::Int16 nr_d;
System::Void sterg_desen () {
    desen->Clear(System::Drawing::Color::White);
}
System::Void creare_drr () {
    nr_d = 20;
    drr = gcnew array < dr^ >(nr_d);
    System::Int16 i;
    for(i=0; i < nr_d; i++)
        drr[i] = gcnew dr(i);
}
System::Void trasez_drr() {
    System::Int16 i;
    for(i=0; i < nr_d; i++)
        drr[i]->desenez(desen);
}
System::Void depl_drr () {
    System::Int16 i;
    for(i=0; i < nr_d; i++)
        drr[i]->deplasez();
}

```

```
}
```

Se observa ca de data aceasta am creat un tablou de obiecte cu declaratia "array ^ drr;" si a trebuit sa scriem functii noi care sa creeze, deplaseze si initializeze un tablou de obiecte de tip "dr". Sigur, functiile scrise invoca metode ale clase "dr".

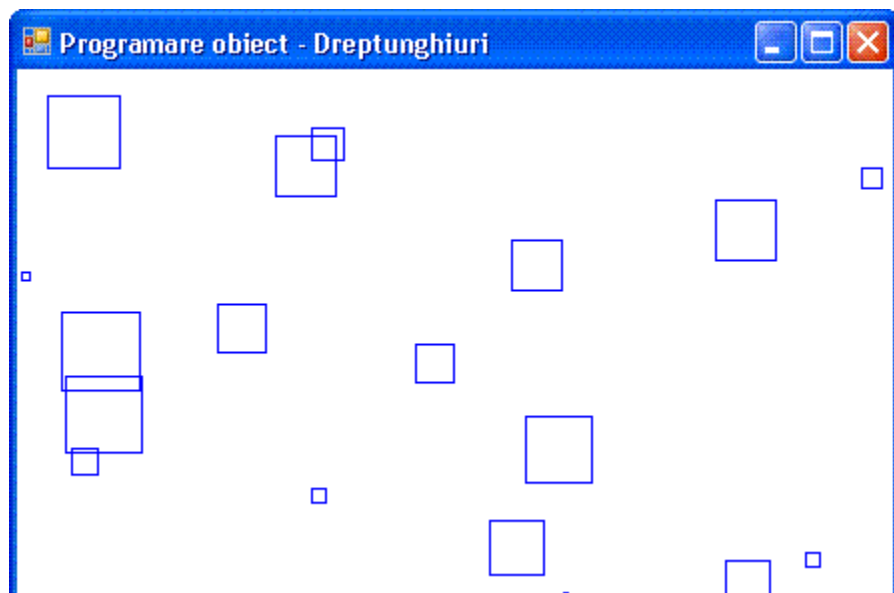
Dupa activarea form-ului curent trebuie sa creem tabloul de obiectele "drr" si obiectul "desen". Vom completa procedura deschisa pe eveniment "activated" al form-ului cu:

```
desen = this->CreateGraphics();  
creare_drr();
```

Completam procedura deschisa pe eveniment tick al obiectului timer1 cu:

```
this->Form1::sterg_desen();  
this->trasez_drr();  
this->depl_drr();  
// Functiile se mai pot apela si:  
//Form1::sterg_desen();  
//Form1::trasez_drr();  
//Form1::depl_drr();
```

Rulam aplicatia si obtinem o serie de dreptunghiuri care se deplaseaza cu diverse viteze, schimbandu-si directia la marginile form-ului.



C# Aplicatia "oop_v4"

- deplasarea mai multor dreptunghiului

```
namespace oop_v4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        System.Drawing.Graphics desen;
        dr[] drr;
        int nr_c, i;
        void sterg_desen()
        {
            desen.Clear(this.BackColor);
        }
        void creare_drr()
        {
            nr_c = 20;
            drr = new dr[nr_c];
            for (i = 0; i < nr_c; i++)
            {
                drr[i] = new dr();
                drr[i].init_dr(i, this.Width - 50, this.Height - 80);
            }
        }
        void trasez_drr()
        {
            for (i = 0; i < nr_c; i++)
                drr[i].desenez(desen, this.Width - 20, this.Height - 45);
        }
        void depl_drr()
        {
            for (i = 0; i < nr_c; i++)
                drr[i].deplasez(this.Width - 20, this.Height - 45);
        }
        private void Form1_Activated(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            creare_drr();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            sterg_desen();
            trasez_drr();
            depl_drr();
        }
    }
}
```



```

private void button1_Click(object sender, EventArgs e)
{
    creare_drr();
}

private void trackBar1_Scroll(object sender, EventArgs e)
{
    this.timer1.Interval = this.trackBar1.Value;
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    desen = this.CreateGraphics();
    creare_drr();
}
}
public class dr
{
    public Int32 pozX, pozY, vX, vY, accX, accY, raza;
    public System.Drawing.Pen creion_a;

    public int desenez(System.Drawing.Graphics zona_des, int lung, int
lat)
    {
        zona_des.DrawRectangle(creion_a, pozX, pozY, raza, raza);
        return 1;
    }
    public void deplasez(int lung, int lat)
    {
        System.Random nr = new System.Random();
        pozX += vX;
        if ((pozX > lung - raza - 40) || (pozX < 0))
        {
            raza = 10 + nr.Next(40);
            vX = -vX;
        }
    }

    public void init_dr(int r, int lung, int lat)
    {
        System.Random nr = new System.Random(r);
        pozX = 5 + nr.Next(lung - 50);
        pozY = 50 + nr.Next(lat - 70);
        vX = 10 + nr.Next(20);
        raza = 10 + nr.Next(40);
        creion_a = new System.Drawing.Pen(System.Drawing.Color.Blue);
    }
}
}

```

- **Realizarea unui obiect grafic-"cerc"**

Asemanator cu aplicatiile anterioare, vom realiza in continuare o aplicatie grafica in care este utilizat un obiect pe care il vom denumi "cerc" obiect care este reprezentat grafic sub forma unui cerc. Metoda "deplasez", este mai complexa de data aceasta, fiind scrisa in vederea simularii deplasarii unei mingi care cade intr-o camera si se reflecta din pardoseala sau din peretii laterali.

Generam un nou proiect de tipul Windows Forms Application numit **"oop_v1"** .

Metoda "deplasez" va fi invocata la intervale regulate de timp mentru a putea simula miscarea unei mingi. Vom plasa deci un obiect de tip timer numit timer1 si setam proprietatea interval la 10 ms si Enabled la "true".

Sariturile mingii vor fi amortizate. Pentru a "arunca" din nou mingea vom plasa un obiect de tip button numit button1. Actionand acest buton, mingea va porni iar din starea initiala.

Vom scrie pentru inceput clasa "cerc" pe care o plasam in fisierul "stdafx.h".

```
ref class cerc {
public:
    cerc(int);
    System::Double pozX;
    System::Double pozY;
    System::Double vX;
    System::Double vY;
    System::Double accX;
    System::Double accY;
    System::Int16 raza;
    System::Drawing::Pen^ creion_a;
    int desenez (System::Drawing::Graphics^ zona_des)
    {
        zona_des->
        >DrawEllipse(creion_a, System::Math::Round(pozX), System::Math::Round(pozY), raza, raza);
        return 1;
    }
    void deplasez ()
    {
        pozX += vX;
        pozY += vY;
        vX = vX + accX;
        vY = vY + accY;
        vY -= vY/100;
        if ((pozX>290-raza) || (pozX<70-raza) || (pozY>400) || (pozY<0))
        {
            pozY = nr->Next(300);
            vX = nr->Next(5);
            vY = nr->Next(1);
            accX = 0;
            accY = 0.5;
            raza = nr->Next(40);
            creion_a =gcnew System::Drawing::Pen(System::Drawing::Color::Blue);
        }
    }
};
```

Dupa cum se obbserva clasa "cerc" are o serie de functii membru cum ar fi:

- `init_cerc` - functie care initializeaza parametrii unui cerc (pozitia , raza, culoarea, viteza de deplasare `px` si `py` , acceleratia pe `x` si pe `y`)
- `desenez` - functie care deseneaza efectiv cercul cu caracteristicile setate de functia `init_cerc`
- `deplasez` - functie care deseneaza un nou cerc pe o noua pozitie deplasata cu valoarea `vX` ,`vy` fata de pozitia anterioara

Pentru simplitate, zona de desenare a fost fixata la 400 X 300 si raza maxima 40. Aplicatia poate fi generalizata pentru dimensiunile curente ale form-ului curent.

Toate functiile membru au fost incluse in clasa "cerc", mai putin constructorul "cerc", constructor plasat in fisierul `stdafx.cpp` al carui continut se poate vedea mai jos:

```
#include "stdafx.h"
cerc::cerc(int i)
{
    init_cerc(i);
}
```

Pentru a putea crea un obiect din clasa "cerc" vom defini un obiect `c: cerc^ d`; si un desen pe care sa se afiseze definit cu `System::Drawing::Graphics^ desen`; . Vom plasa aceste declaratii la sfarsitul zonei `#pragma region`, deci dupa declaratia `#pragma endregion`:

```
System::Drawing::Graphics^ desen;
cerc^ c;
```

Dupa activarea form-ului curent trebuie sa creem obiectele `c` si `desen`. Vom completa procedura deschisa pe eveniment "activated" al form-ului cu:

```
desen = this->CreateGraphics();
c = gcnew cerc(1);
```

Miscarea mingii (desenarea cercului pe o noua pozitie se va face pe fiecare eveniment tick al obiectului `timer1`.

Vom completa deci procedura deschisa pe eveniment tick al obiectului `timer1` cu:

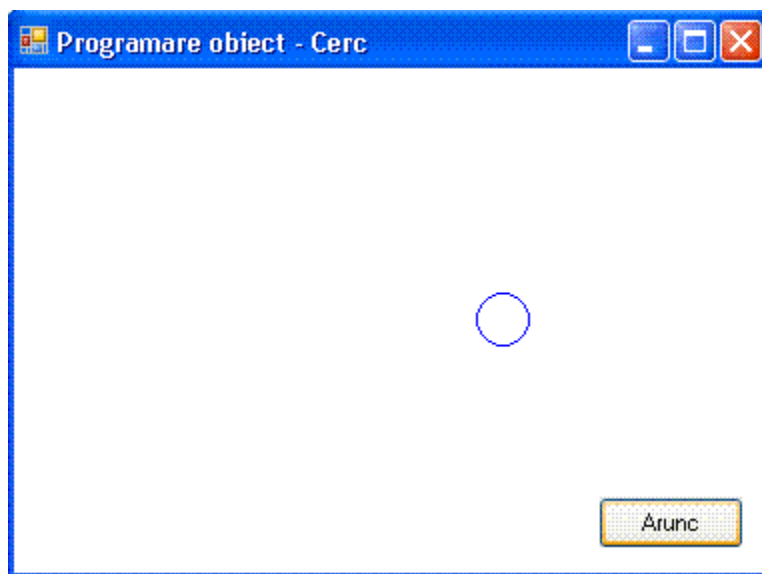
```
desen->Clear(System::Drawing::Color::White);
c->desenez(desen);
c->deplasez();
```

Sariturile mingii vor fi amortizate. Pentru a "arunca" din nou mingea vom actiona butonul "Arunca", astfel mingea va porni iar din starea initiala.

Vom completa deci procedura deschisa pe eveniment click al obiectului butto1 cu:

```
c = gnew cerc(1);
```

Rulam aplicatia si obtinem un cerc care se deplaseaza imitand sariturile unei mingi intr-o camera.



C# Aplicatia "oop_v1"

```
namespace Oop_v1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        System.Drawing.Graphics desen;
        cerc c;
        private void Form1_Activated(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            c = new cerc();
        }
    }
}
```

```

        c.init_cerc(1);
    }
    private void button1_Click(object sender, EventArgs e)
    {
        c.init_cerc(1);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        desen.Clear(this.BackColor);
        c.desenez(desen);
        c.deplasez();
    }

}

public class cerc {
    public float pozX, pozY, vX, vY, accX, accY, raza;
    public System.Drawing.Pen creion_a;
    public int desenez (System.Drawing.Graphics zona_des)
    {
        zona_des.DrawEllipse(creion_a, pozX, pozY, raza, raza);
        return 1;
    }
    public void desplasez ()
    {
        pozX += vX;
        pozY += vY;
        vX = vX + accX;
        vY = vY + accY;
        vY -= vY/100;
        if ((pozX > 290 - raza) || (pozX < 270 - raza) || (pozY > lung - raza) ||
(pozX < 0))
        {
            vX = -vX;
        }
        if ((pozY > lat - raza) || (pozY < 0))
        {
            vY = -vY;
        }
    }

    public void init_cerc(int r, int lung, int lat)
    {
        System.Random nr = new System.Random(r);
        pozX = nr.Next(lung);
        pozY = nr.Next(lat);
        //vX = nr.Next(5);
        //vY = nr.Next(1);
        vX = 3;
        vY = 1;
        accX = 1 / 2;
        accY = 1;
        raza = 10 + nr.Next(40);
        creion_a = new System.Drawing.Pen(System.Drawing.Color.Blue);
    }
}

```

```
}  
}
```

- **Utilizarea clasei grafice "cerc" pentru a afisa mai multe cercuri care se deplaseaza**

Vom modifica aplicatia anterioara "oop_v1" realizand o noua aplicatie Windows Forms Application numit **"oop_v2"**.

Noua aplicatie difera de prima prin faptul ca vom genera mai multe cercuri care se vor deplasa pe form-ul curent imitand sariturile mai multor mingi aruncate intr-o camera.

Vom utiliza aceeasi clasa "cerc" deci continutul fisierelor "stdafx.h" si "stdafx.cpp" va fi acelasi cu cel din aplicatia "oop_v1". Dupa declaratia #pragma endregion: vom plasa urmatorul continut:

```
System::Drawing::Graphics^ desen;  
array < cerc^ >^ cercuri;  
System::Int16 nr_c;  
System::Void sterg_desen () {  
    desen->Clear(System::Drawing::Color::White);  
}  
System::Void creare_cercuri () {  
    nr_c = 20;  
    cercuri = gcnew array < cerc^ >(nr_c);  
    System::Int16 i;  
    for(i=0; i < nr_c; i++)  
        cercuri[i] = gcnew cerc(i);  
}  
System::Void trasez_cercuri() {  
    System::Int16 i;  
    for(i=0; i < nr_c; i++)  
        cercuri[i]->desenez(desen);  
}  
System::Void depl_cercuri () {  
    System::Int16 i;  
    for(i=0; i < nr_c; i++)  
        cercuri[i]->deplasez();  
}
```

Se observa ca de data aceasta am creat un tablou de obiecte cu declaratia "array ^ cercuri;" si a trebuit sa scriem functii noi care sa creeze, deplaseze si initializeze un tablou de obiecte de tip "cerc". Sigur, functiile scrise invoca metode ale clasei "cercuri".

Dupa activarea form-ului curent trebuie sa creem tabloul de obiecte "cercuri" si obiectul "desen". Vom completa procedura deschisa pe eveniment "activated" al form-ului cu:

```
desen = this->CreateGraphics();  
creare_cercuri();
```

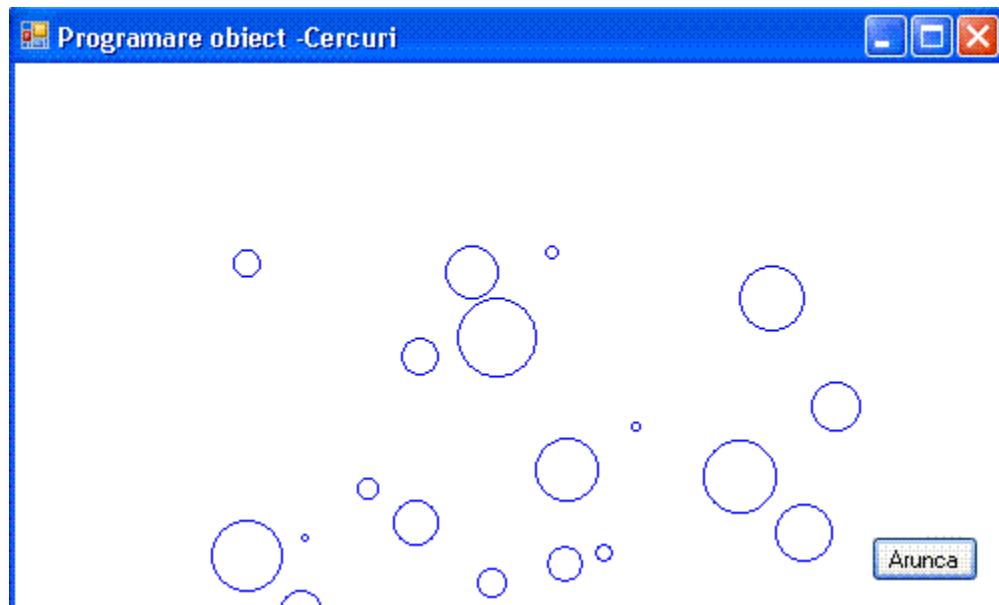
Completam procedura deschisa pe eveniment tick al obiectului timer1 cu:

```
this->sterg_desen();  
this->trasez_cercuri();  
this->depl_cercuri();  
//Functiile se mai pot apela:  
//Form1::sterg_desen();  
//Form1::trasez_cercuri();  
//Form1::depl_cercuri();
```

Completam procedura deschisa pe eveniment click al obiectului button1 cu:

```
creare_cercuri();
```

Rulam aplicatia si obtinem o serie de cercuri care se deplaseaza si imita sariturile mai multor mingi aruncate intr-o camera.



C# Aplicatia "**oop_v2**"

```

namespace oop_v2
{
    public partial class Form1 : Form
    {
        System.Drawing.Graphics desen;
        cerc[] cercuri;
        int nr_c, i;
        void sterg_desen()
        {
            desen.Clear(this.BackColor);
        }
        void creare_cercuri()
        {
            nr_c = 20;
            cercuri = new cerc[nr_c];
            for (i = 0; i < nr_c; i++)
            {
                cercuri[i] = new cerc();
                cercuri[i].init_cerc(i, this.Width-50, this.Height-80);
            }
        }
        void trasez_cercuri()
        {
            for (i = 0; i < nr_c; i++)
                cercuri[i].desenez(desen, this.Width-20, this.Height-45);
        }
        void depl_cercuri()
        {
            for (i = 0; i < nr_c; i++)
                cercuri[i].deplasez(this.Width-20, this.Height-45);
        }
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Activated(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            creare_cercuri();
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            //desen = this.CreateGraphics();
            //creare_cercuri();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            creare_cercuri();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {

```



```

        sterg_desen();
        trasez_cercuri();
        depl_cercuri();
    }

    private void trackBar1_Scroll(object sender, EventArgs e)
    {
        this.timer1.Interval = this.trackBar1.Value;
    }
}

public class cerc
{
    public float pozX, pozY, vX, vY, accX, accY, raza;
    public System.Drawing.Pen creion_a;

    public int desenez(System.Drawing.Graphics zona_des, int lung, int
lat)
    {
        zona_des.DrawEllipse(creion_a, pozX, pozY, raza, raza);
        return 1;
    }
    public void deplasez(int lung, int lat)
    {
        pozX += vX;
        pozY += vY;
        vX = vX + accX;
        vY = vY + accY;
        vY -= vY / 100;
        if ((pozX > lung - raza) || (pozX < 0))
        {
            vX = -vX;
        }
        if ((pozY > lat - raza) || (pozY < 0))
        {
            vY = -vY;
        }
    }

    public void init_cerc(int r, int lung, int lat)
    {
        System.Random nr = new System.Random(r);
        pozX = nr.Next(lung);
        pozY = nr.Next(lat);
        //vX = nr.Next(5);
        //vY = nr.Next(1);
        vX = 3;
        vY = 1;
        accX = 1 / 2;
        accY = 1;
        raza = 10 + nr.Next(40);
        creion_a = new System.Drawing.Pen(System.Drawing.Color.Blue);
    }
}
}

```

