Instrumente virtuale utilizate in domeniul ingineriei electrice

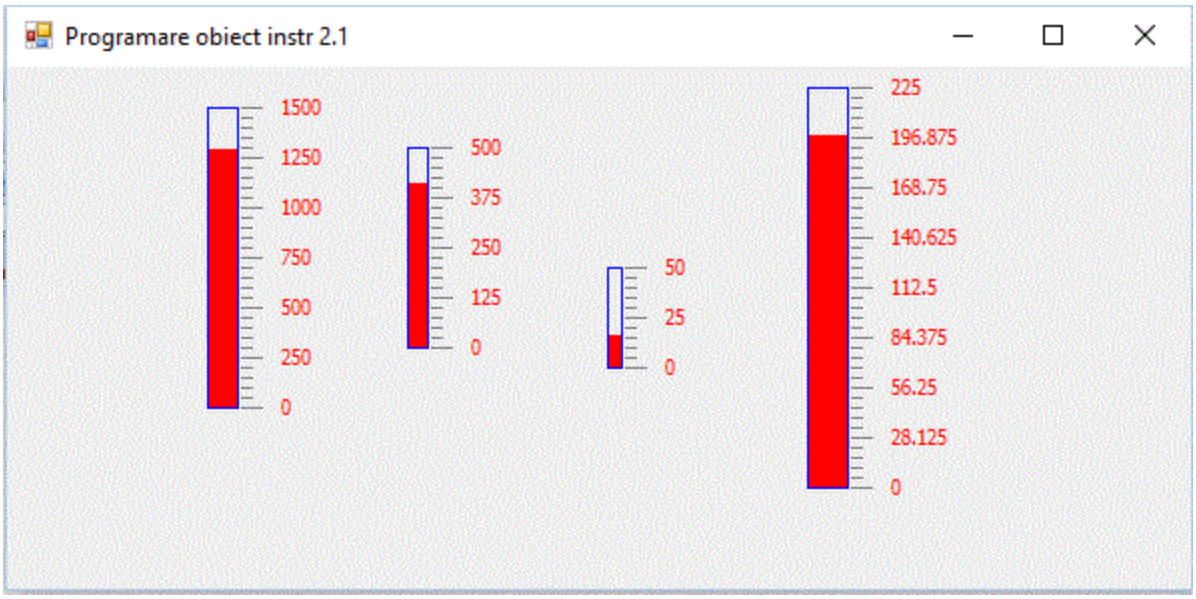
Instrumente virtuale pentru valori instantanee

- 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);
```

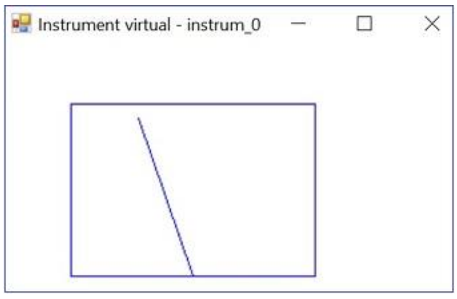
```
                    zona_des.DrawString(System.Convert.ToString(val_max - j
* val_max / h), font_ni, pens_r, x0 + w + 20, y0 + j - 7);
                }
                else
                {
                    zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0 + w
+ 7, y0 + j);
                }
            }

        }
        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.

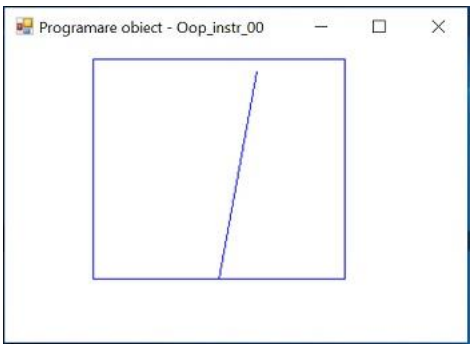
```
namespace instrum_0
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public System.Drawing.Pen creion_albastru;
        public System.Drawing.SolidBrush radiera;
        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;
```

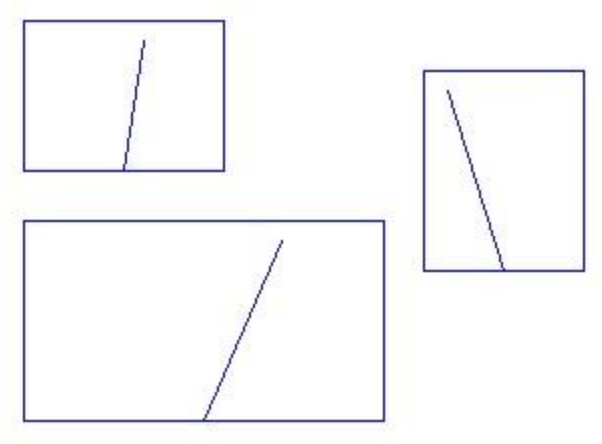
```
        System.Random nr;
        private void Form1_Load(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            creion_albastru = new
System.Drawing.Pen(System.Drawing.Color.Blue);
            radiera = new System.Drawing.SolidBrush(this.BackColor);
       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

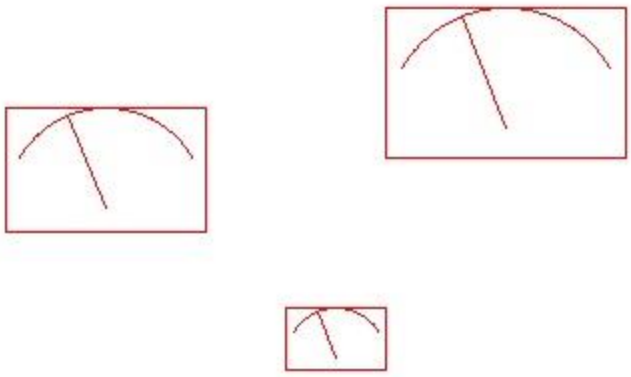
```
            instr_2.setval(nr.Next(20), desen, creion_albastru);
            instr_3.sterg(desen, radiera);
            instr_3.setval(nr.Next(400), desen, creion_albastru);
        }
    }
    public class voltm
    {
        float x0;
        float y0;
        float w;
        float h;
        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;
        }
    }
}
```

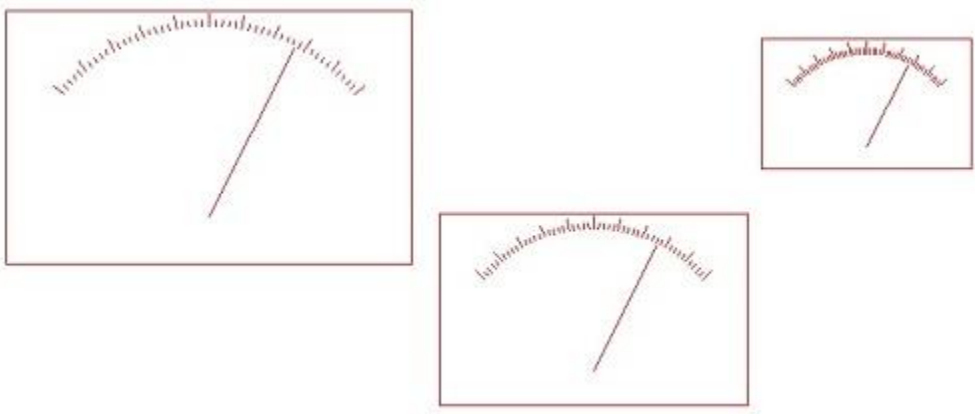


Modificam din nou clasa instrument de masura analogic denumita "voltm" dupa care realizam mai multe obiecte prin instantierea clasei voltm.

```
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;
        }
    }
}
```

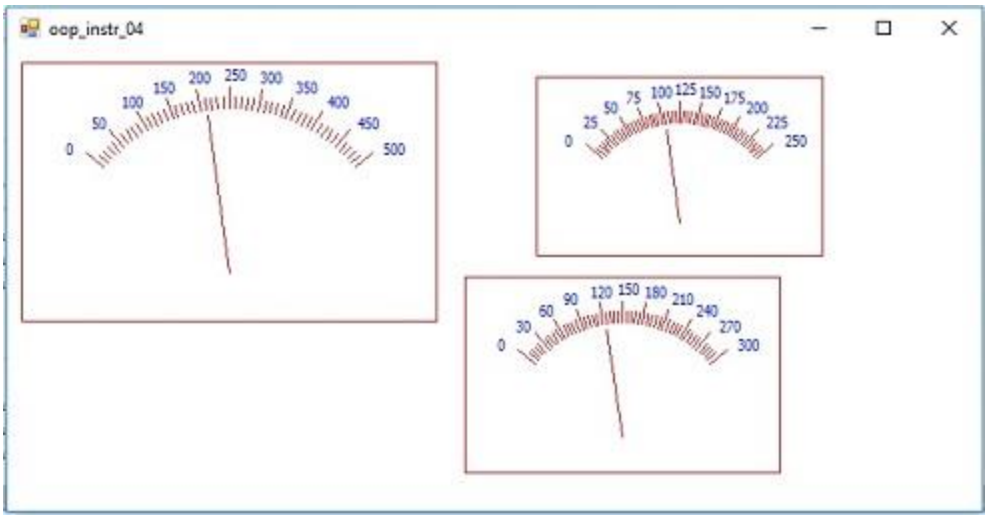


Adugam clasei voltm facilitati pentru afisarea valorilor numerice si totodata vom adauga un constructor, ne mai fiind necesara metoda init_voltm. Pentru a usura folosirea clasei, s-au scos facilitatile pentru schimbarea culorilor acestea fiind fixate in cadrul clasei. S-a mai introdus parametrul val_max 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);

            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;
                System.Drawing.SolidBrush radiera = new
System.Drawing.SolidBrush(System.Drawing.Color.White);
                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;
        }
    }
}
```



Instrumente virtuale pentru afisarea evolutiei marimilor electrice

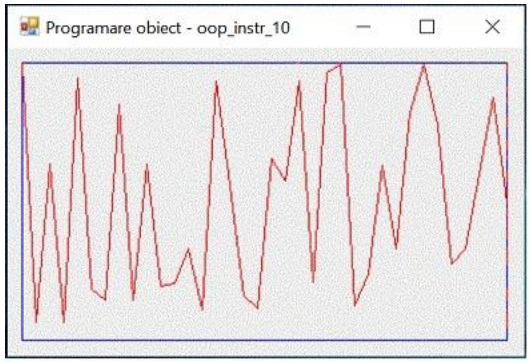
Instrumente virtuale pentru afisarea in coordonate x-t a marimilor electrice

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;
        }
    }
}
```



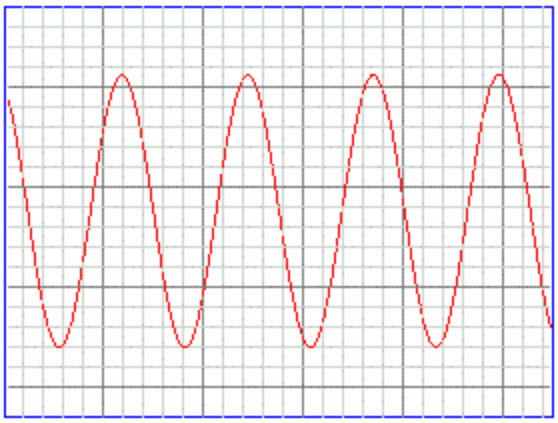
C# Aplicatia **"Oop_instr_12"**

- Utilizarea obiectului afisor_xt pentru afisare grafica in coordonate x-t pentru afisarea functiei sin

```
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;
        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_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;
            int y0;
            int w;
            int h;
            int val_max;
            int 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 obiectului afisor_xt pentru afisare grafica in coordonate x-t pentru afisarea functiei sin

```
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;
            }
        }
    }
}
```

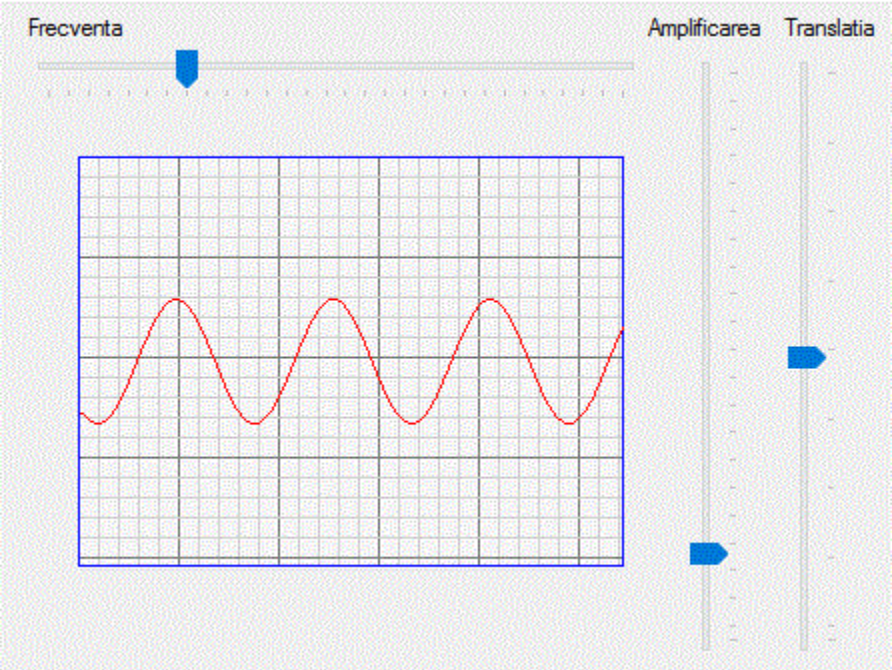
- Vom utiliza in continuare obiectul "afisor_xt" pentru afisare grafica in coordonate x-t si il vom folosi pentru afisarea functiei sin si vom utiliza controale pentru modificarea amplitudinii, frecventei, deplasarii etc.

```
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 afisor_xt1;
        int pozx = 40, pozy = 80, n_maxx = 273, n_maxy = 205;
        double alfa = 0;
        double fi = 0;
        static int[] valori = new int[0];
        public class afisor_xt
        {
            int x0;
            int y0;
            int w;
            int h;
            int f,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;
            }
        }
        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 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
```



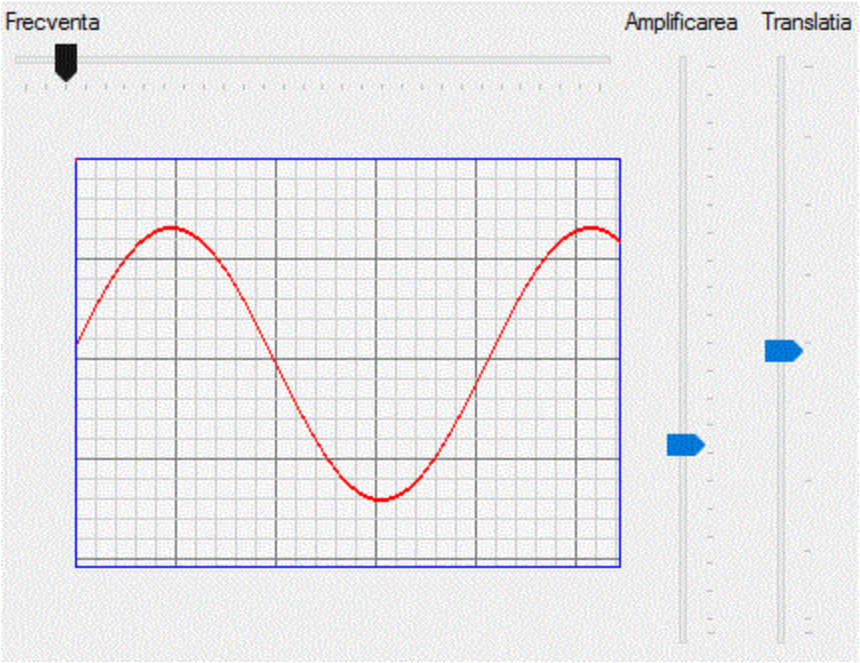
- Vom modifica in continuare obiectul "afisor_xt" pentru afisare grafica in coordonate x-t, pentru a obtine o viteza mai mare de afisare. Pentru aceasta vom crea o imagine de findal care contine grid-ul. Dupa fiecare afisare a imaginii, nu vom mai sterge imaginea si nu vom mai realiza din nou grid-ul. 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 afisor_xt1;
        int pozx = 40, pozy = 80, n_maxx = 273, n_maxy = 205;
        double alfa = 0;
        double fi = 0;
        static int[] valori = new int[0];
        public class afisor_xt
        {
            int x0;
            int y0;
            int w;
            int h;
            int val_max,val,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);

                // 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;
            }
        }
        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);

            //-------- 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);
            afisor_xt1.setval(desen, creion_rosu, valori, n_maxx);
        }
    }
}
```



Imbogatim in continuare clasa afisor x-t adaugand afisarea valorilor numerice si totodata vom adauga un constructor. Vom numi clasa osciloscop

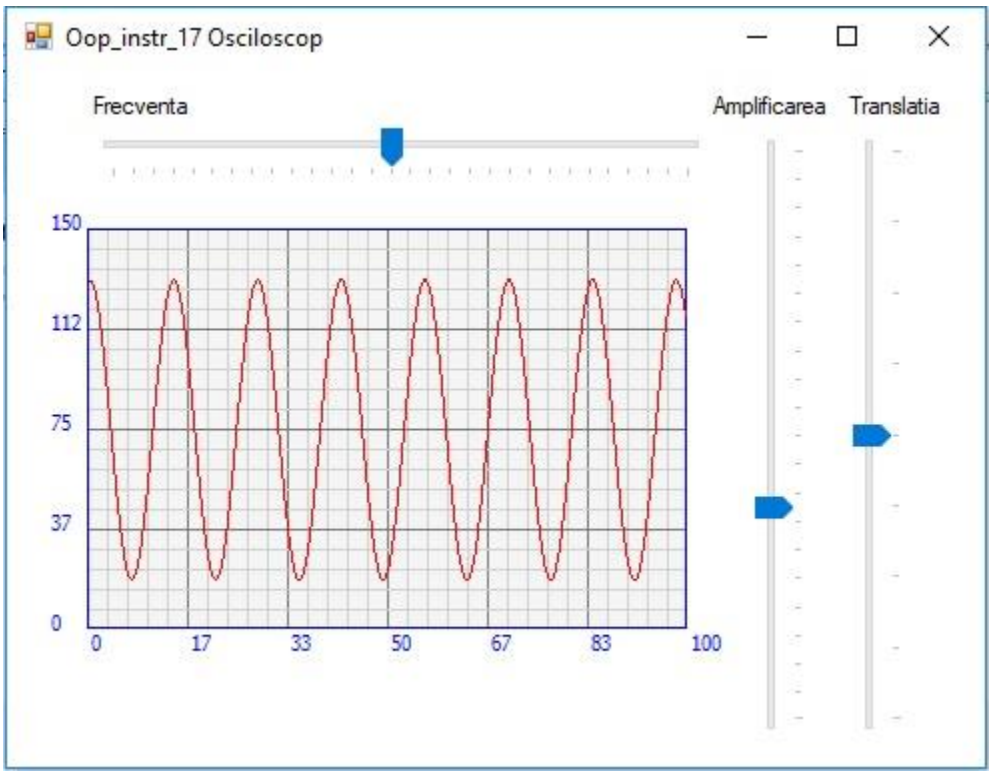
```
namespace Oop_instr_17
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public osciloscop osciloscop1;
        int pozx = 40, pozy = 80, n_maxx = 300, n_maxy = 200;
        double alfa = 0;
        double fi = 0;
        static int[] valori = new int[0];
        public class osciloscop
        {
            int x0;
            int y0;
            int w;
            int h;
            int val_max, val, val_v;
            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(nr_max, val_max, zona_des);
                int val, i, j;

                // 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(ims, x0, y0);
                zona_des.DrawImage(img, x0, y0);

                */

                // afisare grafic sub forma de linii

                zona_des.DrawImage(ims, x0, y0);
                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;
                }
                zona_des.FillRectangle(radiera, x0 , y0 +h, w+20, 20);
                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);
                }
                zona_des.FillRectangle(radiera, x0 - 20, y0-10, 20,h+20);
```

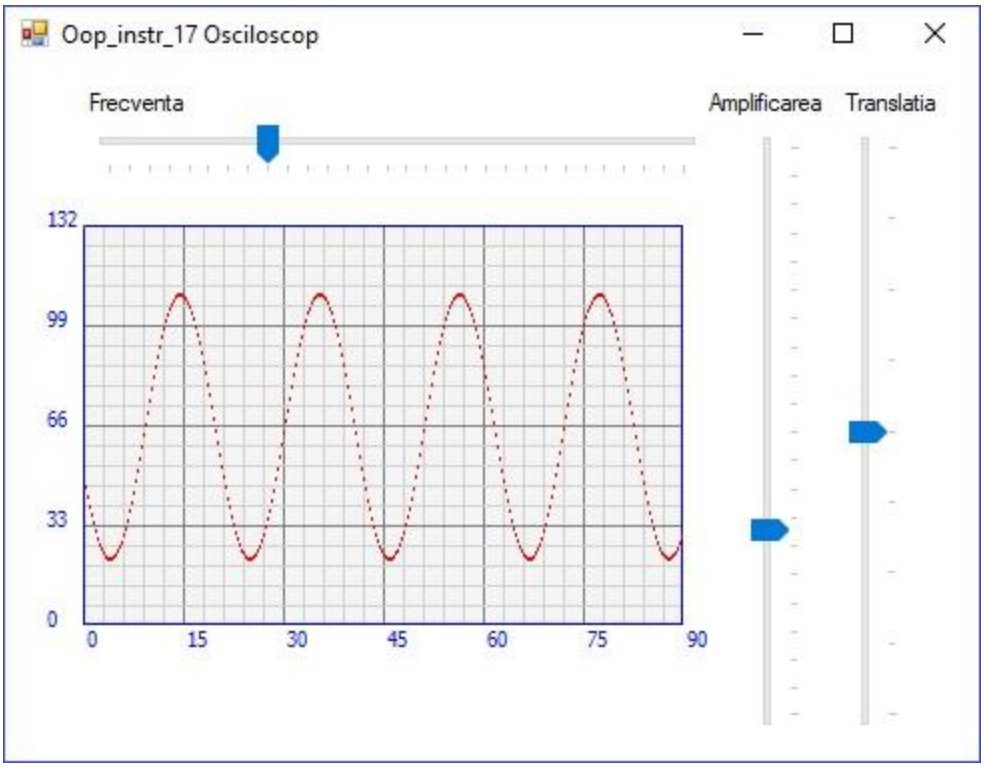
```
                for (i = 0; i <= h; i += 50)
                {
                    val =
System.Convert.ToInt16(System.Convert.ToDouble(i*ampl/100) *
(System.Convert.ToDouble(val_max) / System.Convert.ToDouble(h))); //scalare
                    zona_des.DrawString(val.ToString(), font_ni, pens_blu,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

            }
            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;
                img = new Bitmap(nr_max, n_maxy, zona_des);
                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);
                }

            }

        }

        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);
        }
    }
}
```



Daca afisarea ar fi sub forma de puncte viteza ar fi mult mai mare insa calitatea imaginii scade



Modificam afisarea astfel incat sa pastram afisarea sub forma de puncte dar adaugam puncte suplimentare astfel incat graficul sa para continuu.

```
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];
        public class osciloscop
        {
            int x0;
            int y0;
            int w;
            int h;
```

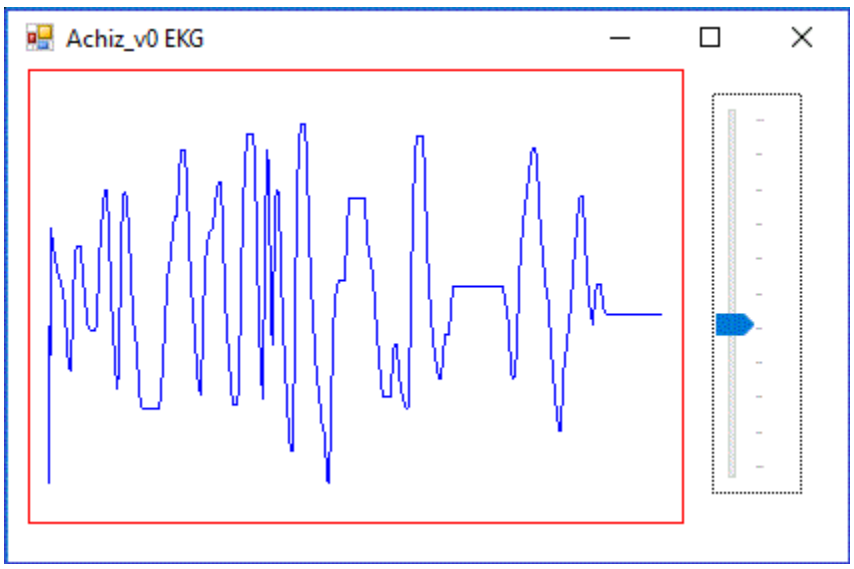
```
            int val_max, val, val_v;
            int nr_max;
            System.Drawing.Graphics zona_des;
            System.Drawing.Pen creion_r = new
System.Drawing.Pen(System.Drawing.Color.Red);
            System.Drawing.Font font_ni = new System.Drawing.Font("Nina",
8);
            System.Drawing.SolidBrush 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(nr_max, val_max, zona_des);
                int val, 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

                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)
                    {
                        for (j = val_v; j <= val; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);
                    }
                    else
                    {
                        for (j = val; j <= val_v; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);

                    }
                    val_v = val;
                }

                zona_des.DrawImage(ims, x0, y0);
                zona_des.DrawImage(img, x0, y0);

                // afisare valori numerice

                zona_des.FillRectangle(radiera, x0, y0 + h, w + 20, 20);
                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);
                }
                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);
                }

            }
            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;
                img = new Bitmap(nr_max, n_maxy, zona_des);
                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);
                    }

                }

            }

            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);

        }
    }
}
```

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



Folosind clasa osciloscop vom imbunatati aceasta clasa si vom crea aplicatia "Oop_instr_19":

Pentru utilizarea mai simpla a acestei clase, s-a modificat 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.

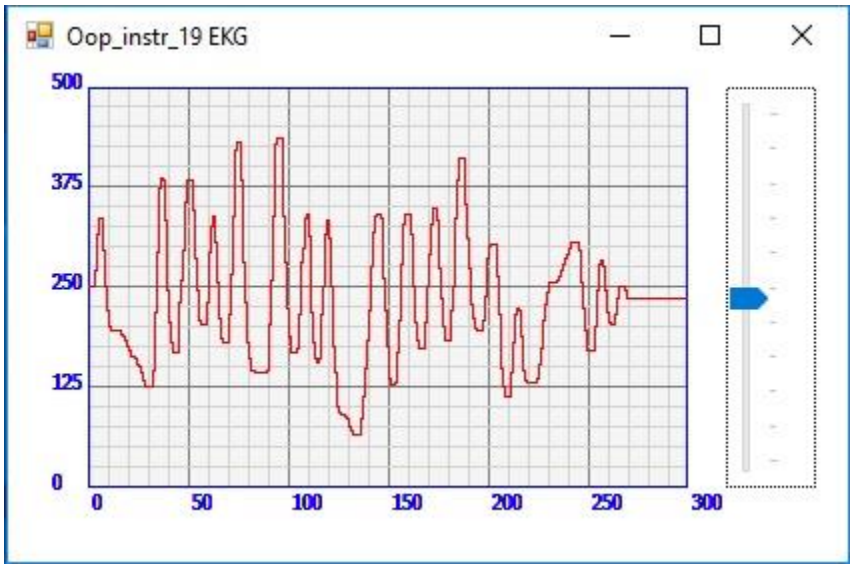
```
namespace Oop_instr_19
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public System.Drawing.Pen creion_blu;
        public System.Drawing.Pen creion_rosu;
        public System.Drawing.Pen creion_gri;
        public System.Drawing.SolidBrush pens_blu;
        public System.Drawing.SolidBrush pens_red;
        public System.Drawing.SolidBrush pens_back;
        public System.Drawing.SolidBrush radiera;
        public System.Drawing.Font font_nina;
        public osciloscop ekg;
        int pozx = 40, pozy = 10, n_maxx = 300, n_maxy = 200;
        Int32 val, val_max = 500;
        static int[] valori = new int[0];
        private void Form1_Load(object sender, EventArgs e)
        {
            desen = this.CreateGraphics();
            creion_blu = new System.Drawing.Pen(System.Drawing.Color.Blue);
            creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
            creion_gri = new
System.Drawing.Pen(System.Drawing.Color.LightGray);
            pens_blu = new
System.Drawing.SolidBrush(System.Drawing.Color.Blue);
            pens_red = new
System.Drawing.SolidBrush(System.Drawing.Color.Red);
            pens_back = new System.Drawing.SolidBrush(this.BackColor);
            font_nina = new System.Drawing.Font("Nina", 8);
            Array.Resize(ref valori, n_maxx + 1);
            ekg = new osciloscop(desen, pozx, pozy, n_maxx, n_maxy,
val_max);

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            int transl = 0;
            int amplif = n_maxy;
             // Trasare grafic
            val = this.trackBar1.Value;
            int f = System.Convert.ToInt32(transl + amplif *
System.Convert.ToDouble(val) / val_max);
            for (int i = 0; i < n_maxx - 1; i++)
            {
                valori[i] = valori[i + 1];
```

```
            }
            valori[n_maxx - 1] = f;
            ekg.setval(valori, n_maxx);
        }
    }

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

        public class osciloscop
        {
            int x0;
            int y0;
            int w;
            int h;
            int val_max, val_max_af, val, val_v;
            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)
            {
                img = new Bitmap(nr_max, val_max, zona_des);
                int i, j;

                // afisare grafic sub forma de puncte

                val_v = val_max - 1 -
System.Convert.ToInt16(System.Convert.ToDouble(vals[0]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_max))); //scalare
                if (val_v >= val_max)
                    val_v = val_max - 1;
                if (val_v <= 0)
                    val_v = 1;
                for (i = 0; i < w; i++)
                {
                    val = val_max - 1 -
System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_max))); //scalare
                    if (val >= val_max)
                        val = val_max - 1;
                    if (val <= 0)
                        val = 1;
                    if (val_v < val)
                    {
                        for (j = val_v; j <= val; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);
```

```
                    }
                    else
                    {
                        for (j = val; j <= val_v; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);

                    }
                    val_v = val;
                }
                zona_des.DrawImage(ims, x0, y0);
                zona_des.DrawImage(img, x0, y0);

                //valori axa x
                //zona_des.FillRectangle(radiera, x0, y0 + h, w + 20, 20);
// pentru afisare dinamica valori axa x

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

                //valori axa y
                //zona_des.FillRectangle(radiera, x0 - 20, y0 - 10, 20, h +
20);// pentru afisare dinamica valori axa y

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

            }
            public osciloscop(System.Drawing.Graphics desen, int pozx, int
pozy, int n_maxx, int n_maxy, int vmaxa)
            {
                x0 = pozx;
                y0 = pozy;
                w = n_maxx;
                h = n_maxy;
                nr_max = n_maxx;
                val_max = n_maxy;
                val_max_af = vmaxa;
                zona_des = desen;
                int i, j;
                img = new Bitmap(nr_max, n_maxy, zona_des);
                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 ((n_maxy - j - 1) % 10 == 0)
                    {
                        for (i = 0; i < nr_max; i++)
                        {
                            if ((n_maxy - j - 1) % 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 ------------------
}
```



Instrumente virtuale pentru valori binare

- Instrument virtual - binar

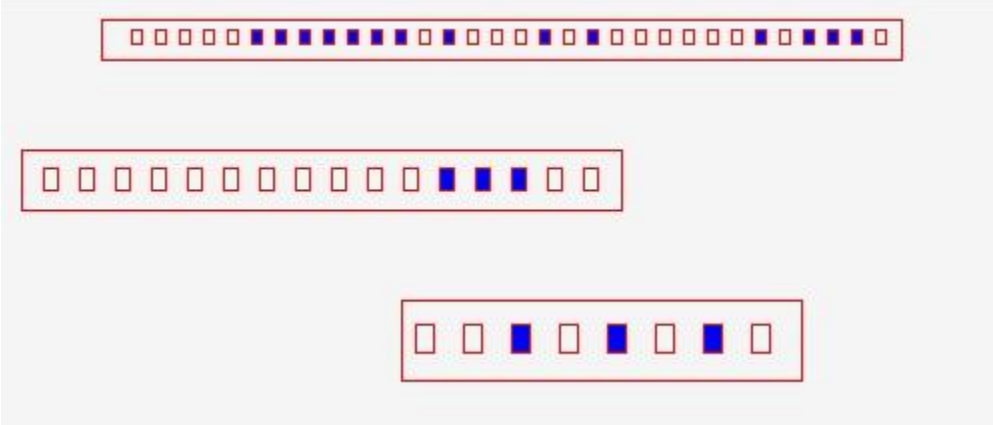
Aplicatia C# **"oop_08"** foloseste clasa **binar** pentru a afisa grafic valori binare.

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;
        }
    }
}
```