

## Curs 02

(plan de curs)

### 1. De ce ( C, +, · ) ?

- polinoame → serii de puteri → funcții analitice:
- operatori liniari, vectori proprii → matrice, polinom caracteristic

### 2. De ce C# ?

- Java, Sun Microsystems (1995)
- operator overloading: C++, ~~Java~~, C#:

```
Complex MyExp(Complex z)
{
    Complex s = 0, p = 1;
    for (int i = 1; i < 100; i++)
    {
        s += p;
        p *= z / i;
    }
    return s;
}
```

- Pascal, Borland Software, Delphi → Anders Hejlsberg → Windows C# (1996)

### 3. Obiecte de tip valoare vs. obiecte de tip referință:

C++: numai valori, Java: numai referințe, C#: și valori și referințe

```
using System;
namespace Exemplu
{
    struct Punct
    //class Punct
    {
        public int x, y;
    }
    class Program
    {
        static void Main(string[] args)
        {
            int dim = 5;
            Punct[] tab = new Punct[dim];
            Punct p = new Punct();
            for (int i = 0; i < dim; i++)
            {
                p.x = i;
                tab[i] = p;
                Console.Write(tab[i].x + " ");
            }
            Console.WriteLine("-> ");
            for (int i = 0; i < dim; i++)
            {
                Console.Write(tab[i].x + " ");
            }
            //0 1 2 3 4 -> 0 1 2 3 4 Press any key to continue . . .
            //0 1 2 3 4 -> 4 4 4 4 4 Press any key to continue . . .
        }
    }
}
```

#### 4) Windows Forms Application:

- windows, controls, dialogs, forms
- Object → Application: `Application.Run(new MyForm());`
- event-driven application: `Application.DoEvents();`
- programare în mod vizual:
  - etapa automată (design, `MyForm.Designer.cs`),
  - etapa manuală (codul handler-ilor de evenimente, `MyForm.cs`)

Fișierul `MainForm.Desingner.cs`:

```
namespace Cercuri
{
    partial class MainForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.buttonStart = new System.Windows.Forms.Button();
            this.buttonStop = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // buttonStart
            //
            this.buttonStart.Location = new System.Drawing.Point(115, 400);
            this.buttonStart.Name = "buttonStart";
            this.buttonStart.Size = new System.Drawing.Size(75, 23);
            this.buttonStart.TabIndex = 0;
            this.buttonStart.Text = "Start";
            this.buttonStart.UseVisualStyleBackColor = true;
            this.buttonStart.Click += new System.EventHandler(this.buttonStart_Click);
            //
            // buttonStop
            //
            this.buttonStop.Location = new System.Drawing.Point(282, 400);
```

```

        this.buttonStop.Name = "buttonStop";
        this.buttonStop.Size = new System.Drawing.Size(75, 23);
        this.buttonStop.TabIndex = 1;
        this.buttonStop.Text = "Stop";
        this.buttonStop.UseVisualStyleBackColor = true;
        this.buttonStop.Click += new System.EventHandler(this.buttonStop_Click);
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(484, 461);
        this.Controls.Add(this.buttonStop);
        this.Controls.Add(this.buttonStart);
        this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
        this.MaximizeBox = false;
        this.Name = "MainForm";
        this.Text = "Cercuri";
        this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.MainForm_FormClosed);
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.ResumeLayout(false);

    }
    #endregion

    private System.Windows.Forms.Button buttonStart;
    private System.Windows.Forms.Button buttonStop;
}
}

```

Fișierul MainForm.cs:

```

using System;
using System.Drawing;
using System.Windows.Forms;
namespace Cercuri
{
    public partial class MainForm : Form
    {
        Graphics dc; //Device context
        SolidBrush redBrush, blackBrush;
        bool seDeseneaza;

        public MainForm()
        {
            InitializeComponent();
            seDeseneaza = false;
            redBrush = new SolidBrush(Color.Red);
            blackBrush = new SolidBrush(Color.Black);
        }
        private void MainForm_Load(object sender, EventArgs e)
        {
            dc = this.CreateGraphics();
        }
        public virtual void LegeaOrara(double t, ref double x, ref double y)
        {
            x = 250 + 150 * Math.Cos(t);
            y = 200 + 150 * Math.Sin(t);
        }
        private void buttonStart_Click(object sender, EventArgs e)

```

```

    {
        //MessageBox.Show("Cucu!");
        if (seDeseneaza) return;
        this.Refresh();
        this.Focus();
        double x = 0, y = 0;
        seDeseneaza = true;
        for (double t = 0; seDeseneaza; t += 0.001)
        {
            LegeaOrara(t, ref x, ref y);
            dc.FillRectangle(redBrush, (int)x, (int)y, 1, 1);
            LegeaOrara(t - 1, ref x, ref y);
            dc.FillRectangle(blackBrush, (int)x, (int)y, 1, 1);
            Application.DoEvents();
        }
        seDeseneaza = false;
    }
    private void MainForm_FormClosed(object sender, FormClosedEventArgs e)
    {
        seDeseneaza = false;
    }
    private void buttonStop_Click(object sender, EventArgs e)
    {
        //MessageBox.Show("Bau!");
        seDeseneaza = false;
    }
}
}

```

Fișierul Program.cs:

```

using System;
using System.Windows.Forms;

namespace Cercuri
{
    class MyForm : MainForm
    {
        public override void LegeaOrara(double t, ref double x, ref double y)
        {
            x = 250 + 150 * Math.Cos(t);
            y = 200 + 150 * Math.Sin(3*t);
        }
    }

    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MyForm());
        }
    }
}

```

## 5) Utilizare Complex explorer

```
public class Sablon : ComplexForm    {...}

public void setXminXmaxYminYmax(double xm, double XM, double ym, double YM)

public void setPixel(int i, int j, Color c)

public void setPixel(double x, double y, Color c)

public void setPixel(Complex z, Color c)

public Color getColor(int k)    //Atentie: k trebuie sa fie pozitiv

public void setLine(int i0, int j0, int i1, int j1, Color c)

public void setLine(double x0, double y0, double x1, double y1, Color c)

public void setLine(Complex z0, Complex z1, Color c)

public bool resetScreen()
{
    pictureBox.Refresh();    //pentru re-pictarea ferestrei
    Application.DoEvents();    //pentru procesarea mesajelor
    return doWork;    //pentru semnalizarea opririi desenarii
}

public virtual void makeImage()    //trebuie suprascrisa
{
    setText("Metoda makeImage()", "trebuie suprascris\u0033!");
}
}
```

## 6) Spirala logaritmică

Fișierul MyToolbox.cs

```
using System;
using System.Drawing;
namespace ComplexExplorer
{
    public class SpiralaLogaritmica : ComplexForm
    {
        Complex i = new Complex(0, 1);
        Complex MyExp(Complex z)
        {
            Complex s = 0, p = 1;
            for (int i = 1; i < 100; i++)
            {
                s += p;
                p *= z / i;
            }
            return s;
        }
    }
}
```

