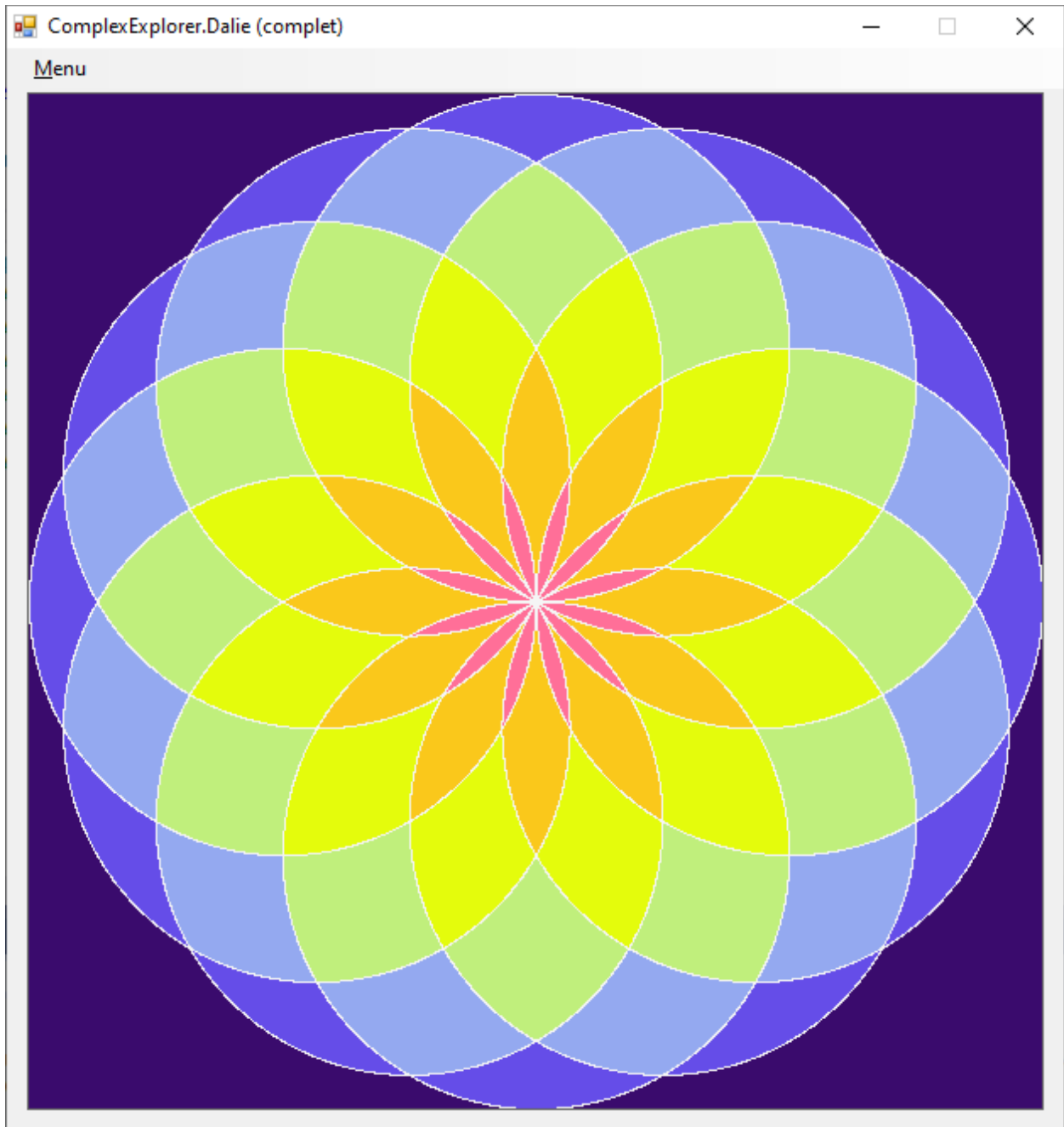


Curs 03 (plan de curs)

1. `class Dalie: ComplexForm`



```

using System;
using System.Drawing;
using System.Collections.Generic;
using static System.Math;

namespace ComplexExplorer
{
    public class Dalie : ComplexForm
    {
        public override void makeImage()
        {
            double R = 30;
            setXminXmaxYminYmax(-2 * R, 2 * R, -2 * R, 2 * R);

            //memoram centrele
            int N = 12;
            Complex[] q = new Complex[N];
            double delta = 2 * Math.PI / N;
            for (int k = 0; k < N; k++)
            {
                q[k] = Complex.setRoTheta(R, k * delta);
            }

            //coloram regiunile
            for (int ii = 0; ii <= imax; ii++)
            {
                for (int jj = 0; jj <= jmax; jj++)
                {
                    Complex z = getZ(ii, jj);
                    int niv = 0;
                    for (int k = 0; k < N; k++)
                    {
                        if ((z - q[k]).Ro <= R) niv++;
                    }
                    setPixel(ii, jj, getColor(1000 + 60 * niv));
                    //if (niv == 0) setPixel(ii, jj, Color.WhiteSmoke);
                }
                resetScreen();
            }

            //trasam contururile
            int nQ = 2000;
            double deltaQ = 2 * Math.PI / nQ;
            for (int k = 0; k < N; k++)
            {
                for (int h = 0; h < nQ; h++)
                {
                    setPixel(q[k] + Complex.setRoTheta(R, h * deltaQ),
Color.WhiteSmoke);
                }
            }
            resetScreen();
        }
    }

    /*****

```

```

public class Adunarea : ComplexForm
{
    Complex i = new Complex(0, 1);
    public override void makeImage()
    {
        double r = 10;
        setXminXmaxYminYmax(-r / 5, r, -r / 5, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        Complex u = 5 + 2 * i;
        Complex v = 1 + 2 * i;
        Complex w = u + v;
        setLine(0, u, PenColor);
        setLine(0, v, PenColor);
        setLine(0, w, Color.Red);
        setLine(w, u, PenColor);
        setLine(w, v, PenColor);
        setTextAt(u, "u");
        setTextAt(v + i / 2, "v");
        setTextAt(w + i / 2, "u+v");
    }
}
/*****

```

```

public class Inmultirea : ComplexForm
{
    Complex i = new Complex(0, 1);
    void traseazaCerculUnitate(Color color)
    {
        int nr = 1000;
        double delta = 2 * Math.PI / nr;
        for (int i = 0; i <= nr; i++)
        {
            setPixel(Complex.setRoTheta(1, i * delta), color);
        }
    }
    public override void makeImage()
    {
        double r = 3;
        setXminXmaxYminYmax(-r, r, -r, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        traseazaCerculUnitate(Color.DarkMagenta);
        //Complex u = 2.5 + 1 * i;
        Complex u = Complex.setRoTheta(1.2, PI / 10);
        Complex v = -1 + 2 * i;
        Complex w = u * v;
        setLine(0, u, Color.Green);
        setLine(1, u, Color.Green);
        setLine(0, 1, Color.Green);
        setLine(0, v, Color.Red);
        setLine(0, w, Color.Red);
        setLine(w, v, Color.Red);
        setTextAt(u, "u");
        setTextAt(v, "v");
    }
}

```

```

        setTextAt(w, "u*v");
        setTextAt(0, "0");
        setTextAt(1, "1");
    }
}
/*****

public class RidicareaLaPutere : ComplexForm
{
    Complex i = new Complex(0, 1);
    void traseazaCerculUnitate(Color color)
    {
        int nr = 1000;
        double delta = 2 * Math.PI / nr;
        for (int i = 0; i <= nr; i++)
        {
            setPixel(Complex.setRoTheta(1, i * delta), color);
        }
    }

    public override void makeImage()
    {
        double r = 3;
        setXminXmaxYminYmax(-r, r, -r, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        traseazaCerculUnitate(Color.DarkMagenta);

        Complex u = Complex.setRoTheta(1.051, 0.222);
        setTextAt(1, "1");
        setTextAt(u, "u");
        Complex pv = 1, pn = u;
        Color color = Color.Red;

        for (int k = 1; k <= 30; k++)
        {
            setLine(0, pv, color);
            setLine(0, pn, color);
            setLine(pn, pv, color);
            pv = pn;
            pn *= u;
            setTextAt(pv, "u");
            setTextAt(pv + i / 10, " " + k);

            resetScreen();
            delaySec(0.1);
        }
        setAxis();
        resetScreen();
    }
}
/*****

```

```

public class RadacinileUnitatii : ComplexForm
{
    Complex i = new Complex(0, 1);
    void traseazaCerculUnitate(Color color)
    {
        int nr = 1000;
        double delta = 2 * Math.PI / nr;
        for (int i = 0; i <= nr; i++)
        {
            setPixel(Complex.setRoTheta(1, i * delta), color);
        }
    }

    public override void makeImage()
    {
        double r = 3;
        setXminXmaxYminYmax(-r, r, -r, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        traseazaCerculUnitate(Color.DarkMagenta);

        int n = 5;
        Complex u = Complex.setRoTheta(1, 2 * PI / n);

        Color color = Color.Red;
        Complex pv = 1, pn = u;
        //for (int k = 1; k <= n; k++)
        //{
        //    setLine(0, pv, color);
        //    setLine(0, pn, color);
        //    setLine(pn, pv, color);
        //    pv = pn;
        //    pn *= u;
        //    setTextAt(pv, "u");
        //    setTextAt(pv + i / 10, " " + k);

        //    resetScreen();
        //    delaySec(0.1);
        //}

        double delta = 2 * PI / n;
        for (int k = 1; k <= n; k++)
        {
            pv = Complex.setRoTheta(1, (k - 1) * delta);
            pn = Complex.setRoTheta(1, k * delta);
            setLine(0, pv, color);
            setLine(0, pn, color);
            setLine(pn, pv, color);
            setTextAt(pn, "u");
            setTextAt(pn + i / 10, " " + k);
            resetScreen();
            delaySec(0.1);
        }
        setAxis();
        resetScreen();
    }
}

```

```

/*****
public class Conjugarea : ComplexForm
{
    Complex i = new Complex(0, 1);

    public override void makeImage()
    {
        double r = 10;
        setXminXmaxYminYmax(-r, r, -r, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        Complex a = 3 + 5 * i;
        Complex b = 1 + i;
        Complex c = 8 + 1.5 * i;
        setTextAt(a, "a");
        setTextAt(b, "b");
        setTextAt(c, "c");
        Color color1 = Color.Green;
        setLine(a, b, color1);
        setLine(b, c, color1);
        setLine(c, a, color1);

        Complex aprim = a.Conj;
        Complex bprim = b.Conj;
        Complex cprim = c.Conj;

        setTextAt(aprim, "ap");
        setTextAt(bprim, "bp");
        setTextAt(cprim, "cp");
        Color color2 = Color.Red;
        setLine(aprim, bprim, color2);
        setLine(bprim, cprim, color2);
        setLine(cprim, aprim, color2);

    }
}
*****/

```

```

public class Asemanarea : ComplexForm
{
    Complex i = new Complex(0, 1);

    public override void makeImage()
    {
        double r = 10;
        setXminXmaxYminYmax(-r, r, -r, r);
        ScreenColor = Color.AliceBlue;
        PenColor = Color.DarkBlue;
        setAxis();
        Complex a = 3 + 3 * i;
        Complex b = 1 + i;
        Complex c = 8 + 0.5 * i;
        setTextAt(a, "a");
        setTextAt(b, "b");
    }
}

```

```

    setTextAt(c, "c");
    Color color1 = Color.Green;
    setLine(a, b, color1);
    setLine(b, c, color1);
    setLine(c, a, color1);

    //Complex aprim = -1 + 10 * i;
    Complex bprim = -7 + i;
    Complex cprim = 1 + 6 * i;

    Complex omega = (a - b) / (c - b);
    //(aprim-bprim)/(cprim-bprim)=omega sau =omega.Conj

    Complex aprim = omega * (cprim - bprim) + bprim;

    setTextAt(aprim, "ap");
    setTextAt(bprim, "bp");
    setTextAt(cprim, "cp");
    Color color2 = Color.Red;
    setLine(aprim, bprim, color2);
    setLine(bprim, cprim, color2);
    setLine(cprim, aprim, color2);
}
}
}

```