

## AppBisect > PrBisect > class Punct

```
public class Punct {
    double x,y;

    public Punct(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getX() {
        return x;
    }

    public void setY(double y) {
        this.y = y;
    }

    public double getY() {
        return y;
    }
}
```

## AppBisect > PrBisect > class Functie

```
public class Functie {
    double x(double t) {
        return t;
    }

    double y(double t) {
        return Math.exp(t)*Math.sin(t-1) - t+0.5;
        //return t*Math.sin(t-1) - t+0.5;
        //return t*t*t-2*t*t-5*t-1;
    }
}
```

## AppBisect > PrBisect > class Bisect

```
public class Bisect {
    double x, a, b;
    double eps, dx;
    int k;
    Functie f;

    public Bisect(double a, double b, Functie f) {
        eps = 1.0E-12;
        k = 0;
        dx = 0;
        this.a = a;
        this.b = b;
        this.f = f;
    }

    public void solve() {
        dx = b - a;
        while (Math.abs(dx) > eps) {
            x = (a + b) / 2.;
            if (Math.signum(f.y(a)) * Math.signum(f.y(x)) < 0.F)
                b = x;
            else
                a = x;
            dx = b - a;
            k++;
            //System.out.printf("%3d] [%15.10f, %15.10f] >> %15.12f\n", k, a, b,
dx);
        }
    }

    public static void main(String[] argv) {
        Bisect b = new Bisect(0.2, 2., new Functie());
        b.solve();
        System.out.println("Iteration number: " + b.k);
        System.out.println("Root obtained: " + b.x);
        System.out.println("Estimated error: " + b.dx);
    }
}
```

## AppBisect > PrBisect > class Desenare

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;

import javax.swing.JPanel;

public class Desenare extends JPanel {
    //atribute publice
    //variabile de instanta
    public Color culoareScriis;

    Punct p[];
    Functie f;
    int csol, nrsolmax = 10;
    Punct sol[] = new Punct[nrsolmax];
    double xmin, xmax;
    double ymin, ymax;

    //constructori ...
    public Desenare() {
        culoareScriis = Color.BLACK;
    }

    public void addsol(double x) {
        sol[csol] = new Punct(x, 0.);
        if (csol < nrsolmax)
            ++csol;
    }

    public void setPuncte(Punct[] puncte) {
```

```
        p = puncte;
        MinMax();
    }

    void MinMax() {
        xmin = xmax = p[0].getX();
        ymin = ymax = p[0].getY();
        for (int i = 1; i < p.length; ++i) {
            if (p[i].getX() < xmin)
                xmin = p[i].getX();
            if (p[i].getY() < ymin)
                ymin = p[i].getY();
            if (p[i].getX() > xmax)
                xmax = p[i].getX();
            if (p[i].getY() > ymax)
                ymax = p[i].getY();
        }
    }

    int toScale(double x, double min, double max, double L) {
        return (int)((x - min) / (max - min) * L);
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;

        //variabile locale
        //latimea JPanel-ului
        double LX = this.getWidth();
        //inaltimea JPanel-ului
        double LY = this.getHeight();

        //Originea in sistemul de axe al graficului
        double x0 = 0.5;
```

```

double y0 = LY - 0.5;
double x1, y1, x2, y2;

g2.setPaint(culoareScris);
try {
    //0 e cuprins intre x minim si x maxim
    //avem de trasat axa OY
    if (xmin * xmax < 0) {
        g2.setPaint(Color.RED);
        int ox0 = (int)(x0 + toScale(0, xmin, xmax, LX));
        g2.draw(new Line2D.Double(ox0, 0, ox0, LY));
        g2.setPaint(culoareScris);
    }

    if (ymin * ymax < 0) {
        //0 e cuprins intre y minim si y maxim
        //avem de trasat axa OX
        g2.setPaint(Color.RED);
        int oy0 = (int)(y0 - toScale(0, ymin, ymax, LY));
        g2.draw(new Line2D.Double(0, oy0, LX, oy0));
        g2.setPaint(culoareScris);
    }

    //
    x1 = x0 + toScale(p[0].getX(), xmin, xmax, LX);
    y1 = y0 - toScale(p[0].getY(), ymin, ymax, LY);
    for (int i = 1; i < p.length; ++i) {
        x2 = x0 + toScale(p[i].getX(), xmin, xmax, LX);
        y2 = y0 - toScale(p[i].getY(), ymin, ymax, LY);

        g2.draw(new Line2D.Double(x1, y1, x2, y2));
        x1 = x2;
        y1 = y2;
        //System.out.printf("%d >>> %15.12f %15.12f\n", i,
p[i].getX(),p[i].getY());

```

```

        //System.out.printf("%15d %15d\n", toScale(p[i].getX(), xmin,
xmax, LX), toScale(p[i].getY(), ymin, ymax, LY));
    }

    //trasare solutii
    for (int i = 0; i < sol.length; ++i) {
        double xc, yc;
        xc = x0 + toScale(sol[i].getX(), xmin, xmax, LX);
        yc = y0 - toScale(sol[i].getY(), ymin, ymax, LY);
        g2.draw(new Ellipse2D.Double(xc - 5, yc - 5, 10, 10));
        //System.out.printf("%d >>> %15.12f %15.12f\n", i,
sol[i].getX(),sol[i].getY());
    }

    //desenare text
    //g2.drawString("Functia ... ", (int)x0, (int)y0);
} catch (Exception e) {
    //System.out.println(e);
}
}
}

```

## AppBisect > PrBisect > class frDesen

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Rectangle;
import javax.swing.JFrame;

public class frDesen extends JFrame {
    private Desenare jPanel1 = new Desenare();
    private Functie fdes;
    private Punct p[];

    public frDesen() {
        try {
            jbInit();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void jbInit() throws Exception {
        this.getContentPane().setLayout(null);
        this.setSize(new Dimension(400, 300));
        this.setTitle("Functie & solutie");
        jPanel1.setBounds(new Rectangle(5, 5, 370, 255));
        jPanel1.setBackground(Color.white);
        jPanel1.setOpaque(true);
        jPanel1.setLayout(null);
        this.getContentPane().add(jPanel1, null);

        double t, pas, tmin, tmax;
        int n;
        tmin = -3;
        tmax = 4;
        n = 300;
```

```
        pas = (tmax - tmin) / n;

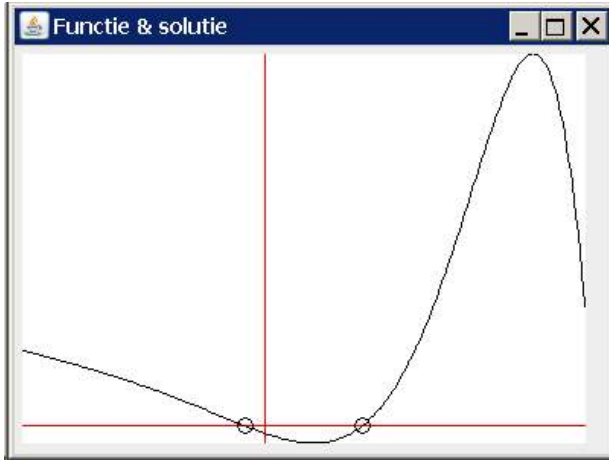
        //functia care se deseneaza
        fdes = new Functie();
        p = new Punct[n];

        for (int i = 0; i < p.length; ++i) {
            t = tmin + i * pas;
            p[i] = new Punct(fdes.x(t), fdes.y(t));
        }
        jPanel1.setPuncte(p);

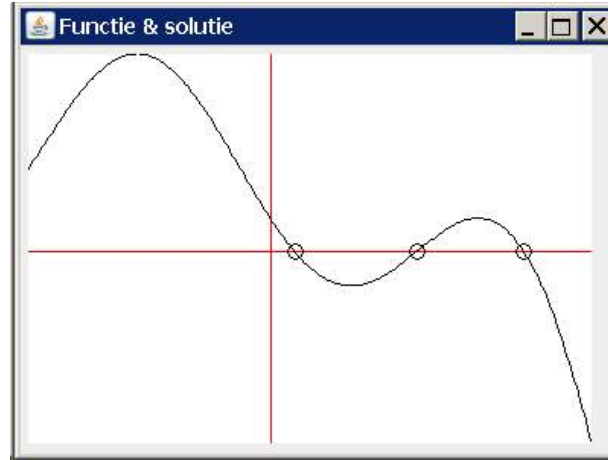
        //cauta solutii
        for (int i = 0; i < p.length - 1; ++i) {
            if (Math.signum(p[i].getY()) * Math.signum(p[i + 1].getY()) < 0) {
                Bisect b = new Bisect(p[i].getX(), p[i + 1].getX(), fdes);
                b.solve();
                //System.out.println("Iteration number: " + b.k);
                System.out.println("Root obtained: " + b.x);
                //System.out.println("Estimated error: " + b.dx);
                jPanel1.addsol(b.x);
            }
        }
    }

    public static void main(String[] args) {
        frDesen fd = new frDesen();
        fd.setVisible(true);
    }
}
```

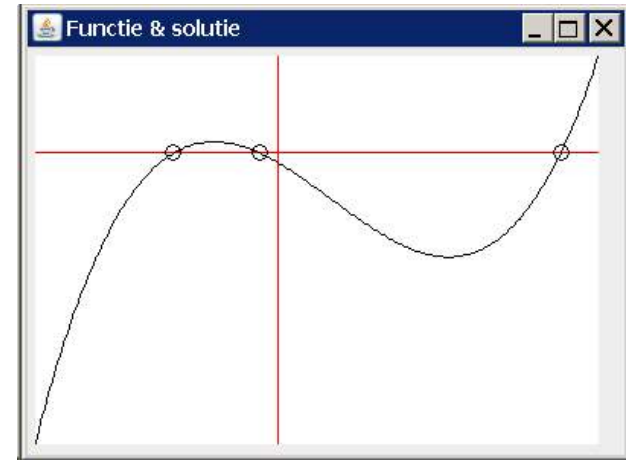
$$\text{Math.exp}(t) * \text{Math.sin}(t-1) - t + 0.5;$$

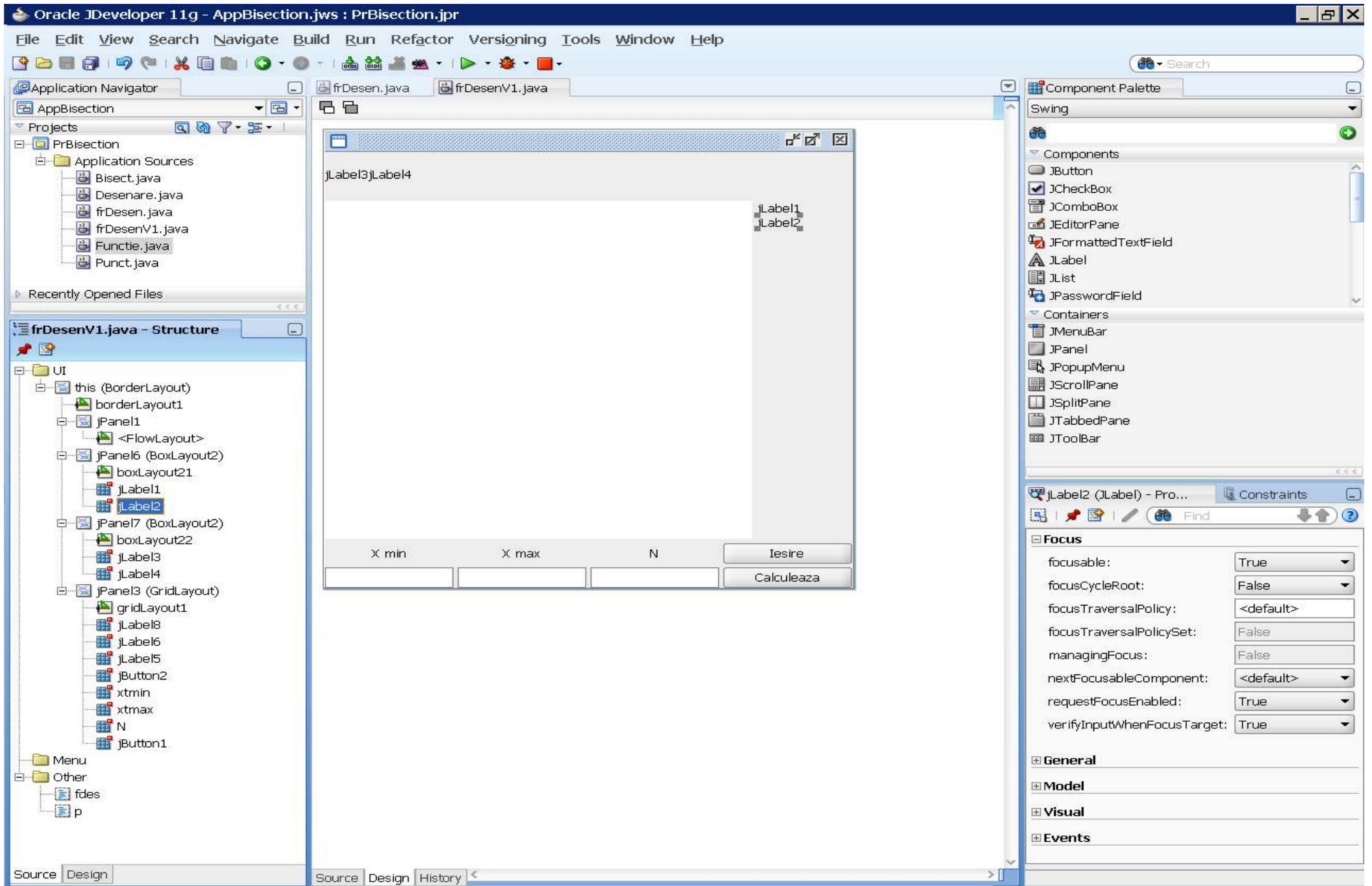


$$t * \text{Math.sin}(t-1) - t + 0.5;$$



$$t * t * t - 2 * t * t - 5 * t - 1;$$







## AppBisect > PrBisect > class frDesenV1

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import oracle.jdeveloper.layout.BoxLayout2;

public class frDesenV1 extends JFrame {
    private Functie fdes;
    private Punct p[];
    private BorderLayout BorderLayout1 = new BorderLayout();
    double t, pas, tmin=-4, tmax=4;
    int n=300;
    private Desenare jPanel1 = new Desenare();
    private JPanel jPanel3 = new JPanel();
    private JPanel jPanel6 = new JPanel();
    private JPanel jPanel7 = new JPanel();
    private GridLayout GridLayout1 = new GridLayout();
    private JTextField N = new JTextField();
    private JTextField xtmax = new JTextField();
    private JTextField xtmin = new JTextField();
    private JButton jButton1 = new JButton();
    private JLabel jLabel1 = new JLabel();
```

```
private JLabel jLabel2 = new JLabel();
private BoxLayout2 boxLayout21 = new BoxLayout2();
private JLabel jLabel3 = new JLabel();
private JLabel jLabel4 = new JLabel();
private BoxLayout2 boxLayout22 = new BoxLayout2();
private JLabel jLabel5 = new JLabel();
private JLabel jLabel6 = new JLabel();
private JButton jButton2 = new JButton();
private JLabel jLabel8 = new JLabel();
```

```
public frDesenV1() {
    try {
        jbInit();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
void calcul() {
    pas = (tmax - tmin) / n;

    //functia care se deseneaza
    fdes = new Functie();
    p = new Punct[n];

    for (int i = 0; i < p.length; ++i) {
        t = tmin + i * pas;
        p[i] = new Punct(fdes.x(t), fdes.y(t));
    }
    jPanel1.setPuncte(p);

    //actualizare valori minime si maxime
    jLabel3.setText(String.valueOf(jPanel1.xmin));
    jLabel4.setText(String.valueOf(jPanel1.xmax));
    jLabel1.setText(String.valueOf(jPanel1.ymax));
```

```

jLabel2.setText(String.valueOf(jPanel1.ymin));

//actualizare valori pe frame
xtmin.setText(String.valueOf(tmin));
xtmax.setText(String.valueOf(tmax));
N.setText(String.valueOf(n));

//cauta solutii cu injumatatirea intervalului
for (int i = 0; i < p.length - 1; ++i) {
    if (Math.signum(p[i].getY()) * Math.signum(p[i + 1].getY()) < 0) {
        Bisect b = new Bisect(p[i].getX(), p[i + 1].getX(), fdes);
        b.solve();
        //System.out.println("Iteration number: " + b.k);
        System.out.println("Root obtained: " + b.x);
        //System.out.println("Estimated error: " + b.dx);
        jPanel1.addsol(b.x);
    }
}

repaint();
}

private void jbInit() throws Exception {
    this.getContentPane().setLayout(borderLayout1);
    this.setSize(new Dimension(502, 518));
    borderLayout1.setHgap(5);
    borderLayout1.setVgap(5);

    jPanel1.setBorder(BorderFactory.createEmptyBorder(1, 1, 1, 1));
    jPanel1.setBackground(Color.white);
    jPanel3.setPreferredSize(new Dimension(30, 50));
    jPanel3.setLayout(gridLayout1);

    jPanel3.setSize(new Dimension(330, 83));

    jPanel6.setPreferredSize(new Dimension(90, 50));
    jPanel6.setLayout(boxLayout21);
    jPanel7.setPreferredSize(new Dimension(50, 50));

    jPanel7.setLayout(boxLayout22);
    gridLayout1.setHgap(4);
    gridLayout1.setVgap(4);
    gridLayout1.setColumns(4);
    gridLayout1.setRows(2);
    N.setMinimumSize(new Dimension(7, 40));
    xtmin.setMinimumSize(new Dimension(7, 40));
    jButton1.setText("Calculeaza");
    jButton1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            jButton1_actionPerformed(e);
        }
    });
    jLabel1.setText("jLabel1");
    jLabel2.setText("jLabel2");
    boxLayout21.setAxis(BoxLayout.Y_AXIS);
    jLabel3.setText("jLabel3");
    jLabel4.setText("jLabel4");
    jLabel5.setText("N");
    jLabel5.setToolTipText("Numarul de puncte in [X min. X max], pasul
este constant.");
    jLabel5.setHorizontalAlignment(SwingConstants.CENTER);
    jLabel6.setText("X max");
    jLabel6.setToolTipText("Valoarea de sfarsit a lui X");
    jLabel6.setHorizontalAlignment(SwingConstants.CENTER);
    jButton2.setText("Iesire");
    jButton2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            jButton2_actionPerformed(e);
        }
    });
}

```

```

jLabel8.setText("X min");
jLabel8.setToolTipText("Valoarea de inceput a lui X");
jLabel8.setHorizontalAlignment(SwingConstants.CENTER);
jPanel3.add(jLabel8, null);
jPanel3.add(jLabel6, null);
jPanel3.add(jLabel5, null);
jPanel3.add(jButton2, null);
jPanel3.add(xtmin, null);
jPanel3.add(xtmax, null);
jPanel3.add(N, null);
jPanel3.add(jButton1, null);

jPanel6.add(jLabel1, null);
jPanel6.add(Box.createGlue());
jPanel6.add(jLabel2, null);

jPanel7.add(jLabel3, null);
jPanel7.add(Box.createGlue());
jPanel7.add(jLabel4, null);

this.getContentPane().add(jPanel1, BorderLayout.CENTER);
this.getContentPane().add(jPanel6, BorderLayout.EAST);
this.getContentPane().add(jPanel7, BorderLayout.NORTH);
this.getContentPane().add(jPanel3, BorderLayout.SOUTH);

calcul();
}

public static void main(String[] args) {
    frDesenV1 fd = new frDesenV1();
    fd.setVisible(true);
}

private void jButton2_actionPerformed(ActionEvent e) {

    System.exit(0);
}

private void jButton1_actionPerformed(ActionEvent e) {
    tmin = Double.parseDouble(xtmin.getText());
    tmax = Double.parseDouble(xtmax.getText());
    n = Integer.parseInt(N.getText());
    jPanel1.resetSolcount();
    calcul();
}
}

```

