

```
import java.awt.Color;
import java.awt.Graphics;
public class DrawPanel extends javax.swing.JPanel
{
    private NumberInfos numberInfos = null;

    public void setNumberInfos(NumberInfos pNumberInfos)
    {
        numberInfos = pNumberInfos;
    }

    public DrawPanel()
    {
        initComponents();
    }

    public void paintComponent(Graphics g)
    {
        int h = getHeight();
        int w = getWidth();

        g.setColor(Color.white);
        g.fillRect(0, 0, w, h);

        if (numberInfos != null)
            numberInfos.draw(g, w, h);
    }
    // Skipped: ... initComponents { ... }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    // End of variables declaration//GEN-END:variables
}
```

```
public class NumberInfo
{
    private double value;
    private String label;

    public NumberInfo(double value, String label)
    {
        this.value = value;
        this.label = label;
    }

    public String getLabel()
    {
        return label;
    }

    public double getValue()
    {
        return value;
    }

    public String toString()
    {
        return label + " : " + value;
    }
}
```

```

import java.awt.Color;
import java.awt.Graphics;
import java.util.ArrayList;
public class NumberInfos
{
    private ArrayList<NumberInfo> alNumberInfos = new ArrayList<>();

    private String title;

    public NumberInfos(String title)
    {
        this.title = title;
    }

    public String getTitle()
    {
        return title;
    }

    public void add(double value, String label)
    {
        alNumberInfos.add(new NumberInfo(value, label));
    }

    public double getTotal()
    {
        double result = 0;
        for (int i = 0; i < alNumberInfos.size(); i++)
            result = result + alNumberInfos.get(i).getValue();
        return result;
    }

    public double getMaximum()
    {
        double result = 0;
        for (int i = 0; i < alNumberInfos.size(); i++)
            if (result < alNumberInfos.get(i).getValue())
                result = alNumberInfos.get(i).getValue();
        return result;
    }

    public Object[] toArray()
    {
        return alNumberInfos.toArray();
    }

    public int size()
    {
        return alNumberInfos.size();
    }

    public NumberInfo get(int i)
    {
        return alNumberInfos.get(i);
    }

    public void draw(Graphics g, int pWidth, int pHeight)
    {
        // ne dessiner que s'il y au moins 1 barre
        if (alNumberInfos.size() > 0)
        {
            NumberInfo info;
            double barHeight;
            double max = getMaximum();
            double barWidth = (double) (pWidth - 1) / alNumberInfos.size();
            for (int i = 0; i < alNumberInfos.size(); i++)
            {
                info = alNumberInfos.get(i);

                barHeight = info.getValue() / max * pHeight;

                // alterner les couleurs pour dessiner la barre (sans contour)
                if (i % 2 == 0)
                    g.setColor(Color.YELLOW);
                else
                    g.setColor(Color.ORANGE);
                g.fillRect((int) (i * barWidth), pHeight - (int) barHeight, (int) barWidth, (int) barHeight);

                // dessiner le contour
                g.setColor(Color.RED);
                g.drawRect((int) (i * barWidth), pHeight - (int) barHeight, (int) barWidth, (int) barHeight);

                // dessiner le teste et la valeur en bas de la barre
                g.setColor(Color.BLUE);
                g.drawString(info.getLabel(), (int) (i * barWidth + 2), pHeight - 4);

                g.drawString(String.valueOf(info.getValue()), (int) (i * barWidth + 2), pHeight - 18);
            }

            // dessiner le titre de l'histogramme (avec un effet)
            g.setColor(Color.DARK_GRAY);
            g.drawString(title, 3, 11);
            g.setColor(Color.BLACK);
            g.drawString(title, 2, 10);
        }
    }
}

```

```

public class MainFrame extends javax.swing.JFrame
{
    private NumberInfos numberInfos = null;

    public MainFrame()
    {
        initComponents();
    }

    private void updateView()
    {
        numberInfoJList.setListData(numberInfos.toArray());
        repaint();
    }
// Skipped: ... initComponents { ... }
    private void newButtonActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_newButtonActionPerformed
        numberInfos = new NumberInfos(titleTextField.getText());
        drawPanel.setNumberInfos(numberInfos);
        addButton.setEnabled(true);

        // TODO: TEST - pour tester... à commenter/enlever quand c'est OK!
        numberInfos.add(63, "JAN");
        numberInfos.add(65, "FEB");
        numberInfos.add(58, "MAR");
        numberInfos.add(77, "APR");
        numberInfos.add(111, "MAY");
        numberInfos.add(124, "JUN");
        numberInfos.add(104, "JUL");
        numberInfos.add(99, "AUG");
        numberInfos.add(80, "SEP");
        numberInfos.add(66, "OCT");
        numberInfos.add(103, "NOV");
        numberInfos.add(80, "DEC");
        // TODO: TEST

        updateView();
    }//GEN-LAST:event_newButtonActionPerformed

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_addButtonActionPerformed
        numberInfos.add(Double.valueOf(valueTextField.getText()), labelTextField.getText());
        updateView();
    }//GEN-LAST:event_addButtonActionPerformed
// Skipped: ... Look & Feel
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton addButton;
    private DrawPanel drawPanel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField labelTextField;
    private javax.swing.JButton newButton;
    private javax.swing.JList numberInfoJList;
    private javax.swing.JTextField titleTextField;
    private javax.swing.JTextField valueTextField;
    // End of variables declaration//GEN-END:variables
}

```