

```
public class Cistern
{
    private double maximumVolume;
    private double currentVolume;

    public Cistern(double pRadius, double pHeight)
    {
        maximumVolume = Math.PI * pRadius * pRadius * pHeight * 1000; // Attention à la conversion m^3 -> litres
        currentVolume = 0;
    }

    public void add(double pVolume)
    {
        if (currentVolume + pVolume <= maximumVolume)
        {
            currentVolume = currentVolume + pVolume;
        }
        else
        {
            currentVolume = maximumVolume;
        }
    }

    public void drain(double pVolume)
    {
        if (pVolume <= currentVolume)
        {
            currentVolume = currentVolume - pVolume;
        }
        else
        {
            currentVolume = 0;
        }
    }

    public double getCurrentVolume()
    {
        return currentVolume;
    }

    public double getPercentage()
    {
        return currentVolume / maximumVolume * 100;
    }

    public double getMaximumVolume()
    {
        return maximumVolume;
    }

    public String toString()
    {
        return "Fill level : " + currentVolume + "l / " + maximumVolume + "l (" + getPercentage() + "%)";
    }
}
```

```

public class MainFrame extends javax.swing.JFrame
{
    private Cistern cistern = new Cistern(1, 2);

    public MainFrame()
    {
        initComponents();
        int max = (int) cistern.getMaximumVolume();
        volumeSlider.setMaximum(max);
        volumeSlider.setMajorTickSpacing(max / 10); // Afficher uniquement 10 ticks (lignes)
        updateView();
    }

    public double round2Decimals(double v)
    {
        // pour expliquer le principe
        return ((int) (v * 100)) / 100.0;
    }

    public double roundDecimals(double v, int nbrDecimals)
    {
        double p = Math.pow(10, nbrDecimals);
        return ((int) (v * p)) / p;
    }

    public void updateView()
    {
        currentVolumeLabel.setText(roundDecimals(cistern.getCurrentVolume(), 2) + " litres");
        currentRateLabel.setText(roundDecimals(cistern.getPercentage(), 2) + "%");
        currentRateProgressBar.setValue((int) cistern.getPercentage());
    }
}

// Skipped: ... initComponents { ... }
private void addButtonActionPerformed(java.awt.event.ActionEvent evt) //GEN-FIRST:event_addButtonActionPerformed
{ //GEN-HEADEREND:event_addButtonActionPerformed
    double v = Double.valueOf(volumeTextField.getText());
    cistern.add(v);
    updateView();
} //GEN-LAST:event_addButtonActionPerformed

private void drainButtonActionPerformed(java.awt.event.ActionEvent evt) //GEN-FIRST:event_drainButtonActionPerformed
{ //GEN-HEADEREND:event_drainButtonActionPerformed
    cistern.drain(Double.valueOf(volumeTextField.getText()));
    updateView();
} //GEN-LAST:event_drainButtonActionPerformed

private void volumeSliderStateChanged(javax.swing.event.ChangeEvent evt) //GEN-FIRST:event_volumeSliderStateChanged
{ //GEN-HEADEREND:event_volumeSliderStateChanged
    volumeTextField.setText(String.valueOf(volumeSlider.getValue()));
} //GEN-LAST:event_volumeSliderStateChanged

// Skipped: ... Look & Feel
// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton addButton;
private javax.swing.JLabel currentRateLabel;
private javax.swing.JProgressBar currentRateProgressBar;
private javax.swing.JLabel currentVolumeLabel;
private javax.swing.JButton drainButton;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JSlider volumeSlider;
private javax.swing.JTextField volumeTextField;
// End of variables declaration //GEN-END:variables
}

```