

```

import java.util.ArrayList;
public class Equ2Deg
{
    private double a, b, c;
    private ArrayList<Double> alSolutions = new ArrayList<>();

    public Equ2Deg(double pA, double pB, double pC)
    {
        a = pA;
        b = pB;
        c = pC;
    }

    public int getDegree()
    {
        if (a != 0)
            return 2;
        else if (b != 0)
            return 1;
        else
            return 0;
    }

    public boolean isQuadratic()
    {
        return (getDegree() == 2);
    }

    public boolean isLinear()
    {
        return (getDegree() == 1);
    }

    public boolean isConstant()
    {
        return (getDegree() == 0);
    }

    public boolean isIndeterminate()
    {
        return (isConstant() && (c == 0));
    }

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

    public Double getNumberOfSolutions()
    {
        if (isIndeterminate())
            return Double.POSITIVE_INFINITY;
        else
            return Double.valueOf(alSolutions.size());
    }

    public void solve()
    {
        // réinitialiser la liste des solutions
        alSolutions.clear();
        if (isQuadratic()) // équation du second degré
        {
            double delta = b * b - 4 * a * c;
            if (delta == 0)
            {
                alSolutions.add(-b / (2 * a));
            }
            else if (delta > 0)
            {
                alSolutions.add((-b + Math.sqrt(delta)) / (2 * a));
                alSolutions.add((-b - Math.sqrt(delta)) / (2 * a));
            }
            else
            {
                // pas de solutions (réelles)
            }
        }
        else // a=0 -> équation du premier degré
        {
            if (isLinear())
            {
                alSolutions.add(-c / b);
            }
            else // a=0 et b=0: constante
            {
                if (isIndeterminate())
                {
                    alSolutions.add(Double.NEGATIVE_INFINITY); //  $0x=0 \rightarrow S=\mathbb{R}$ 
                    alSolutions.add(0.0);
                    alSolutions.add(Double.POSITIVE_INFINITY);
                }
                else
                {
                    //  $c=0 / c \neq 0 \rightarrow S=\text{vide}$ 
                }
            }
        }
    }
}

```

```

public String toString()
{
    // version simple:
    //
    // return a+"x^2 + "+b+"x + "+c+" = 0";

    // Version améliorée:
    //
    String res;

    // on traite le bloc pour x^2
    if (a == 0)
        res = "";
    else if (a == 1)
        res = "x^2";
    else if (a == -1)
        res = "-x^2";
    else
        res = a + "x^2";

    // on traite le bloc pour x
    if (b == 1)
    {
        if (res.equals(""))
            res = "x";
        else
            res = res + " + x";
    }
    else if (b == -1)
    {
        if (res.equals(""))
            res = "-x";
        else
            res = res + " - x";
    }
    else if (b < 0)
    {
        if (res.equals(""))
            res = "-";
        else
            res = res + " - ";
        res = res + Math.abs(b) + "x";
    }
    else if (b > 0)
    {
        if (res.equals(""))
            res = b + "x";
        else
            res = res + " + " + b + "x";
    }
    // pour b == 0 il n'y a pas de x

    // on traite le bloc pour la constante
    if (c < 0)
    {
        if (res.equals(""))
            res = "-";
        else
            res = res + " - ";
        res = res + Math.abs(c);
    }
    else if (c > 0)
    {
        if (res.equals(""))
            res = String.valueOf(c);
        else
            res = res + " + " + c;
    }
    // pour c == 0 il n'y a rien

    // !!!!
    // Cas spécial si tous les facteurs sont nuls (ég. indéterminée)
    // !!!!
    if (res.equals(""))
        res = "0";

    return res + " = 0";
}
}

```

```

public class MainFrame extends javax.swing.JFrame
{
    private Equ2Deg equation = null;

    public MainFrame()
    {
        initComponents();
    }
// Skipped: ... initComponents { ... }
    private void solveButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_solveButtonActionPerformed
    {//GEN-HEADEREND:event_solveButtonActionPerformed
        double a = Double.valueOf(aTextField.getText());
        double b = Double.valueOf(bTextField.getText());
        double c = Double.valueOf(cTextField.getText());

        // création, résolution de l'équation et affichage des résultats
        equation = new Equ2Deg(a, b, c);
        equation.solve();
        resultsList.setListData(equation.solutionsToArray());

        // afficher l'équation sous forme  $ax^2+bx+c=0$  (présentée correctement)
        equationLabel.setText(equation.toString());

        // présenter le texte du résultat proprement construit...
        Double nbr = equation.getNumberOfSolutions();
        String nbrSolutions;
        if (equation.isIndeterminate())
        {
            nbrSolutions = "une infinité de";
        }
        else
        {
            // pour éviter le 0.0, 1.0 ou 2.0 -> conversions: Double -> double -> int
            int n = (int) ((double) nbr);
            nbrSolutions = String.valueOf(n);
        }

        int degree = equation.getDegree();
        String degreeVal;
        if (degree == 0)
            degreeVal = "constante";
        else if (degree == 1)
            degreeVal = "du 1er degré";
        else
            degreeVal = "du 2d degré";

        String result = "Cette équation " + degreeVal + " admet " + nbrSolutions;

        if (nbr == 1)
            result = result + " solution réelle";
        else
            result = result + " solutions réelles";
        resultLabel.setText(result);
    }//GEN-LAST:event_solveButtonActionPerformed
// Skipped: ... Look & Feel
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JTextField aTextField;
    private javax.swing.JTextField bTextField;
    private javax.swing.JTextField cTextField;
    private javax.swing.JLabel equationLabel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JLabel resultLabel;
    private javax.swing.JList resultsList;
    private javax.swing.JButton solveButton;
    // End of variables declaration//GEN-END:variables
}

```