

```
public class SchoolSubject
{
    // PARTIE 2: algorithmes correctement programmés
    //          avec quelques améliorations du GUI (avec bouton close)

    private String name;
    private ArrayList<Integer> alMarks = new ArrayList<Integer>();

    public SchoolSubject(String pName)
    {
        name = pName;
    }

    public String getName()
    {
        return name;
    }

    public void addMark(int score)
    {
        alMarks.add(score);
    }

    public int getBestMark()
    {
        int best;

        if (alMarks.size() > 0)
        {
            best = alMarks.get(0);
            for (int i = 1; i < alMarks.size(); i++)
            {
                if (alMarks.get(i) > best)
                    best = alMarks.get(i);
            }
            return best;
        }
        else
            return -1;
    }

    public int getWorstMark()
    {
        int worst;

        if (alMarks.size() > 0)
        {
            worst = alMarks.get(0);
            for (int i = 1; i < alMarks.size(); i++)
            {
                if (alMarks.get(i) < worst)
                    worst = alMarks.get(i);
            }
            return worst;
        }
        else
            return -1;
    }

    public double calculateAverage()
    {
        double sum = 0;

        if (alMarks.size() > 0)
        {
            for (int i = 0; i < alMarks.size(); i++)
                sum = sum + alMarks.get(i);

            return sum / alMarks.size();
        }
        else
            return -1;
    }

    public int calculateRoundedAverage()
    {
        double avg = calculateAverage();
        int res = (int) avg;
        if (avg != -1)
        {
            if ((int) avg != avg)
                res = ((int) avg) + 1;
        }
        return res;
    }

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

    public void clear()
    {
        alMarks.clear();
    }
}
```

```

public class MainFrame extends javax.swing.JFrame
{
    // PARTIE 2: algorithmes correctement programmés
    //         avec quelques améliorations du GUI (avec bouton close)

    private SchoolSubject subject = null;

    public MainFrame()
    {
        initComponents();
        subjectPanel.setVisible(false);
        closeButton.setEnabled(false);
    }

    public void updateView()
    {
        nameLabel.setText(subject.getName());
        marksList.setListData(subject.toArray());

        // Amélioration: ne pas afficher -1 (qui indique une liste vide...)
        int val;

        val = subject.calculateRoundedAverage();
        if (val == -1)
            averageLabel.setText("-");
        else
            averageLabel.setText(String.valueOf(val));

        val = subject.getWorstMark();
        if (val == -1)
            minLabel.setText("-");
        else
            minLabel.setText(String.valueOf(val));

        val = subject.getBestMark();
        if (val == -1)
            maxLabel.setText("-");
        else
            maxLabel.setText(String.valueOf(val));
    }
    // Skipped: ... initComponents { ... }
    private void createActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_createButtonActionPerformed

        String s = subjectTextField.getText();
        if (s.equals(""))
            return; // ignorer si la matière est vide

        subject = new SchoolSubject(s);
        subjectPanel.setVisible(true);
        updateView();

        closeButton.setEnabled(true);
        subjectTextField.setEnabled(false);
        createButton.setEnabled(false);
    } //GEN-LAST:event_createButtonActionPerformed

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_addButtonActionPerformed
        subject.addMark(Integer.valueOf(markTextField.getText()));
        updateView();

        markTextField.requestFocus();
    } //GEN-LAST:event_addButtonActionPerformed

    private void clearButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_clearButtonActionPerformed
        subject.clear();
        updateView();

        markTextField.requestFocus();
    } //GEN-LAST:event_clearButtonActionPerformed

    private void closeButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_closeButtonActionPerformed
        subjectPanel.setVisible(false);

        subject = null;

        closeButton.setEnabled(false);
        subjectTextField.setEnabled(true);
        createButton.setEnabled(true);
    } //GEN-LAST:event_closeButtonActionPerformed
    // Skipped: ... Look & Feel
    // Skipped: ... graphic attributes
}

```