```java
import java.util.ArrayList;
public class RandomNumberList
{
     private ArrayList<Long> alNumbers = new ArrayList<>();

     public void addSeries(int count, long min, long max)
     {
          Randomizer random = new Randomizer(min, max);
          for (int i = 0; i < count; i++)
          {
               alNumbers.add(random.getNext());
          }
     }

     public void clear()
     {
          alNumbers.clear();
     }

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

     public double getAverage()
     {
          if (alNumbers.size() == 0)
               return -1;

          double sum = 0;
          for (int i = 0; i < alNumbers.size(); i++)
          {
               sum = sum + alNumbers.get(i);
          }
          return sum / alNumbers.size();
     }

     public Long getMin()
     {
          if (alNumbers.isEmpty())
               return null;

          long min = alNumbers.get(0);  // calculer avec un type primitif
          for (int i = 1; i < alNumbers.size(); i++)
          {
               if (min > alNumbers.get(i))
                    min = alNumbers.get(i);
          }
          return min;
     }

     public Long getMax()
     {
          if (alNumbers.isEmpty())
               return null;

          Long max = alNumbers.get(0);   // calculer avec un objet numérique => auto-(un-)boxing
          for (int i = 1; i < alNumbers.size(); i++)
          {
               if (max < alNumbers.get(i))
                    max = alNumbers.get(i);
          }
          return max;
     }

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

```java
public class Randomizer
{
     // Les limites
     private long min;
     private long max;

     public Randomizer(long pMin, long pMax)
     {
          setLimits(pMin, pMax);
     }

     public void setLimits(long pMin, long pMax)
     {
          // Version avancée
          min = Math.min(pMin, pMax);
          max = Math.max(pMin, pMax);
     }

     public long getNext()
     {
          return (long) (Math.random() * (max - min + 1)) + min;
     }
}
```

```java
public class MainFrame extends javax.swing.JFrame
{
     private RandomNumberList numbers = new RandomNumberList();

     public MainFrame()
     {
          initComponents();
          updateView();
     }

     public void updateView()
     {
          numbersList.setListData(numbers.toArray());

          nbrLabel.setText(String.valueOf(numbers.size()));

          double avg;
          avg = numbers.getAverage();

          // Amélioration: ne pas afficher -1 ou null (qui indique uneliste vide...)
          if (avg == -1)
               averageLabel.setText("-");
          else
               averageLabel.setText(String.valueOf(avg));

          Long val;

          val = numbers.getMin();
          if (val == null)
               minLabel.setText("-");
          else
               minLabel.setText(String.valueOf(val));

          val = numbers.getMax();
          if (val == null)
               maxLabel.setText("-");
          else
               maxLabel.setText(String.valueOf(val));
     }
// Skipped: ... initComponents { ... }
    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_addButtonActionPerformed
          int nbr = Integer.valueOf(numberValuesTextField.getText());
          long min = Long.valueOf(minTextField.getText());
          long max = Long.valueOf(maxTextField.getText());

          numbers.addSeries(nbr, min, max);
          updateView();

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

    private void clearButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_clearButtonActionPerformed
          numbers.clear();
          updateView();
    }//GEN-LAST:event_clearButtonActionPerformed
// Skipped: ... Look & Feel
// Skipped: ... graphic attributes
}
```