```java
public class Car
{
     private String id;
     private String brand;
     private String model;
     private double consumption;
     private int year;
     private int power;

     public Car(String id, String brand, String model, double consumption, int year, int power)
     {
          this.id = id;
          this.brand = brand;
          this.model = model;
          this.consumption = consumption;
          this.year = year;
          this.power = power;
     }

     public String getBrand()
     {
          return brand;
     }

     public String getModel()
     {
          return model;
     }

     public double getConsumption()
     {
          return consumption;
     }

     public String getID()
     {
          return id;
     }

     public int getYear()
     {
          return year;
     }

     public int getPower()
     {
          return power;
     }

     public String toString()
     {
          return "[" + id + "] " + getBrand() + " - " + getModel() + "/" + getYear() + " - " + getPower() + "kW (" + getConsumption() + " l/100km)";
     }
}
```

```java
import java.util.ArrayList;
public class Garage
{
    private ArrayList<Car> alCars = new ArrayList<>();

    public void addCar(Car pCar)
    {
        alCars.add(pCar);
    }

    public int countCars()
    {
        return alCars.size();
    }

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

    public Double computeHighestConsumption()
    {
        Car c;

        if (alCars.isEmpty())
            return null;

        double max = alCars.get(0).getConsumption();
        for (int i = 1; i < alCars.size(); i++)
        {
            c = alCars.get(i);
            if (c.getConsumption() > max)
                max = c.getConsumption();
        }

        return max;
    }

    public Double computeAverageConsumption()
    {
        Car c;

        if (alCars.isEmpty())
            return null;

        double sum = 0;
        for (int i = 0; i < alCars.size(); i++)
        {
            c = alCars.get(i);
            sum += c.getConsumption();
        }

        return sum / alCars.size();
    }

    public String getSmallestID()
    {
        Car c;

        if (alCars.isEmpty())
            return null;

        String min = alCars.get(0).getID();
        for (int i = 1; i < alCars.size(); i++)
        {
            c = alCars.get(i);
            if (c.getID().compareTo(min) < 0)
                min = c.getID();
        }

        return min;
    }

    // Note: il y a 6 variantes de cette methode

    public String getIDOfOlderstCar()
    {
        if (alCars.isEmpty())
            return null;

        Car min = alCars.get(0);
        for (int i = 1; i < alCars.size(); i++)
        {
            if (min.getYear() > alCars.get(i).getYear())
                min = alCars.get(i);
        }

        return min.getID();
    }
```

```
public String getIDOfOlderstCar1()
{
     if (alCars.isEmpty())     // moins de duplication de code
          return null;

     Car min = alCars.get(0);
     for (int i = 1; i < alCars.size(); i++)
     {
          Car c = alCars.get(i);
          if (min.getYear() > c.getYear())
               min = c;
     }

     return min.getID();
}

public String getIDOfOlderstCar2()
{
     if (alCars.isEmpty())
          return null;

     int min = alCars.get(0).getYear();
     String id = alCars.get(0).getID();
     for (int i = 1; i < alCars.size(); i++)
     {
          if (min > alCars.get(i).getYear())
          {
               min = alCars.get(i).getYear();
               id = alCars.get(i).getID();
          }
     }

     return id;
}

public String getIDOfOlderstCar3()
{
     Car c;     // version avec moins de duplication de code

     if (alCars.isEmpty())
          return null;

     int min = alCars.get(0).getYear();
     String id = alCars.get(0).getID();
     for (int i = 1; i < alCars.size(); i++)
     {
          c = alCars.get(i);
          if (c.getYear() < min)
          {
               min = c.getYear();
               id = c.getID();
          }
     }

     return id;
}

public String getIDOfOlderstCar4()
{
     if (alCars.isEmpty())
          return null;

     Car minCar = alCars.get(0);
     int min = minCar.getYear();
     for (int i = 1; i < alCars.size(); i++)
     {
          if (min > alCars.get(i).getYear())
          {
               minCar = alCars.get(i);
               min = minCar.getYear();
          }
     }

     return minCar.getID();
}

public String getIDOfOlderstCar5()
{
     if (alCars.isEmpty())
          return null;

     int min = alCars.get(0).getYear();
     int posMin = 0;
     for (int i = 1; i < alCars.size(); i++)
     {
          if (min > alCars.get(i).getYear())
          {
               min = alCars.get(i).getYear();
               posMin = i;
          }
     }

     return alCars.get(posMin).getID();
}
```

```
    public int getPosOfLowestPower()
    {
        if (alCars.isEmpty())
            return -1;

        double min = alCars.get(0).getPower();
        int pos = 0;
        for (int i = 1; i < alCars.size(); i++)
        {
            if (min > alCars.get(i).getPower())
            {
                min = alCars.get(i).getPower();
                pos = i;
            }
        }

        return pos;
    }
}
```

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

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

     public void updateView()
     {
          carsList.setListData(garage.toArray());
          String msg;
          int nbr = garage.countCars();
          if (nbr == 0)
               numberLabel.setText("Pas de voitures...");
          else if (nbr == 1)
               numberLabel.setText("1 voiture");
          else
               numberLabel.setText(nbr + " voitures");

          Double average = garage.computeAverageConsumption();
          if (average == null)
               averageLabel.setText("-");
          else
               averageLabel.setText(average + "  l/100km");

          Double highest = garage.computeHighestConsumption();
          if (highest == null)
               highestLabel.setText("-");
          else
               highestLabel.setText(highest + "  l/100km");

          idTextField.setText("");
          modelTextField.setText("");
          typeTextField.setText("");
          consumptionTextField.setText("");
          yearTextField.setText("");
          powerTextField.setText("");

          oldestCarLabel.setText("-");
          idLabel.setText("-");
     }
// Skipped: ... initComponents { ... }
    private void addButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_addButtonActionPerformed
    {//GEN-HEADEREND:event_addButtonActionPerformed
          String id = idTextField.getText();
          String type = typeTextField.getText();
          String model = modelTextField.getText();
          double consumption = Double.valueOf(consumptionTextField.getText());
          int year = Integer.valueOf(yearTextField.getText());
          int power = Integer.valueOf(powerTextField.getText());

          garage.addCar(new Car(id, type, model, consumption, year, power));

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

    private void oldestCarButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_oldestCarButtonActionPerformed
    {//GEN-HEADEREND:event_oldestCarButtonActionPerformed
          String model = garage.getIDOfOlderstCar();
          if (model == null)
               oldestCarLabel.setText("Pas de voitures!");
          else
               oldestCarLabel.setText(model);
    }//GEN-LAST:event_oldestCarButtonActionPerformed

     private void lowPowerButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_lowPowerButtonActionPerformed
     {//GEN-HEADEREND:event_lowPowerButtonActionPerformed
          int pos = garage.getPosOfLowestPower();
          if (pos >= 0)
               carsList.setSelectedIndex(pos);
     }//GEN-LAST:event_lowPowerButtonActionPerformed

     private void unselectButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_unselectButtonActionPerformed
     {//GEN-HEADEREND:event_unselectButtonActionPerformed
          carsList.clearSelection();
     }//GEN-LAST:event_unselectButtonActionPerformed

     private void clearButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_clearButtonActionPerformed
     {//GEN-HEADEREND:event_clearButtonActionPerformed
          garage = new Garage();
          updateView();
     }//GEN-LAST:event_clearButtonActionPerformed

     private void idButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_idButtonActionPerformed
     {//GEN-HEADEREND:event_idButtonActionPerformed
          String id = garage.getSmallestID();
          if (id == null)
               idLabel.setText("Pas de voitures!");
          else
               idLabel.setText(id);
     }//GEN-LAST:event_idButtonActionPerformed
// Skipped: ... Look & Feel
// Skipped: ... graphic attributes
}
```