Java Class #1

```
import gnu.io.*;import java.io.IOException;import java.io.PrintStream;import java.util.Enumeration;
import java.util.logging.Level;import java.util.logging.Logger;
public class MultiServo {
Enumeration listaPuertos = CommPortIdentifier.getPortIdentifiers();
CommPortIdentifier puertoId ;
PrintStream arduinoOut ;
SerialPort puerto;
MultiServo(String pto){
  while(listaPuertos.hasMoreElements()){
    puertoId = (CommPortIdentifier)listaPuertos.nextElement();
    if(puertoId.getName().equals(pto)){
      break;
    }
  }
  try {
    puerto = (SerialPort)puertoId.open("Serial", 1000);
    puerto.setSerialPortParams(28800, SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
    arduinoOut = new PrintStream(puerto.getOutputStream());
  } catch (PortInUseException | IOException | UnsupportedCommOperationException ex) {
    Logger.getLogger(MultiServo.class.getName()).log(Level.SEVERE, null, ex);
  }
}

public void enviar(int num){
  arduinoOut.write(num);
}
  public static void main(String[] args) {
    GUI app = new GUI();
  }
}
```

Java Class #2

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;

public class GUI extends KeyAdapter implements ActionListener, ChangeListener {

  JFrame vt = new JFrame("Servos");
  JLabel txt1 = new JLabel("Selecciona el puerto de tu Arduino");
  String[] puertosOp = {"COM1", "COM2", "COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9"};
  JComboBox puertosSelec = new JComboBox(puertosOp);
  JPanel panel1 = new JPanel();

  JSlider motor1 = new JSlider(JSlider.VERTICAL, 0, 180, 0);
  JTextField manual1 = new JTextField(3);
  JPanel m1 = new JPanel();
  JSlider motor2 = new JSlider(JSlider.VERTICAL, 0, 180, 0);
  JTextField manual2 = new JTextField(3);
  JPanel m2 = new JPanel();
  JSlider motor3 = new JSlider(JSlider.VERTICAL, 0, 180, 0);
```

```
JTextField manual3 = new JTextField(3);
JPanel m3 = new JPanel();

JPanel panel2 = new JPanel();

JButton enviar = new JButton("Enviar");
JPanel panel3 = new JPanel();

MultiServo conexion;

GUI() {
  vt.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  vt.setResizable(false);
  vt.setLayout(new BorderLayout());
  vt.add(panel1, BorderLayout.NORTH);
  vt.add(panel2, BorderLayout.CENTER);

  panel1.add(txt1);
  panel1.add(puertosSelec);
  puertosSelec.addActionListener(this);

  panel2.add(m1);
  panel2.add(new JLabel("   "));
  panel2.add(m2);
  panel2.add(new JLabel("   "));
  panel2.add(m3);

  m1.setLayout(new BorderLayout());
  m1.add(motor1, BorderLayout.CENTER);
  m1.add(manual1, BorderLayout.SOUTH);
  motor1.addChangeListener(this);
  manual1.addKeyListener(this);

  m2.setLayout(new BorderLayout());
  m2.add(motor2, BorderLayout.CENTER);
  m2.add(manual2, BorderLayout.SOUTH);
  motor2.addChangeListener(this);
  manual2.addKeyListener(this);

  m3.setLayout(new BorderLayout());
  m3.add(motor3, BorderLayout.CENTER);
  m3.add(manual3, BorderLayout.SOUTH);
  motor3.addChangeListener(this);
  manual3.addKeyListener(this);

  vt.add(panel3, BorderLayout.SOUTH);
  panel3.add(enviar);
  enviar.addActionListener(this);
  motor1.setEnabled(false);
  motor2.setEnabled(false);
  motor3.setEnabled(false);
  manual1.setEditable(false);
  manual2.setEditable(false);
  manual3.setEditable(false);
  vt.pack();
  vt.setVisible(true);
}
```

```
@Override
public void actionPerformed(ActionEvent e) {
  Object f = e.getSource();
  if (f == puertosSelec) {
    motor1.setEnabled(true);
    motor2.setEnabled(true);
    motor3.setEnabled(true);
    String op = (String) puertosSelec.getSelectedItem();
    conexion = new MultiServo(op);

    manual1.setEditable(true);
    manual2.setEditable(true);
    manual3.setEditable(true);

    puertosSelec.setEnabled(false);
  }
  if (f == enviar) {
    String val;
    val = manual1.getText();
    motor1.setValue(Integer.parseInt(val));

    val = manual2.getText();
    motor2.setValue(Integer.parseInt(val));

    val = manual3.getText();
    motor3.setValue(Integer.parseInt(val));
  }
}

@Override
public void stateChanged(ChangeEvent e) {
  Object f = e.getSource();
  String val;
  if (f == motor1) {
    val = String.valueOf(motor1.getValue());
    conexion.enviar(Map.rTres(motor1.getValue()));
    manual1.setText(val);
  }
  if (f == motor2) {
    val = String.valueOf(motor2.getValue());
    conexion.enviar(Map.rTres(motor2.getValue()) + 85);
    manual2.setText(val);
  }
  if (f == motor3) {
    val = String.valueOf(motor3.getValue());
    conexion.enviar(Map.rTres(motor3.getValue()) + 170);
    manual3.setText(val);
  }
}

@Override
public void keyReleased(KeyEvent e) {
  Object f = e.getSource();
  int tecla = e.getKeyCode();
  String val;
  if (f == manual1 && tecla == KeyEvent.VK_ENTER) {
    val = manual1.getText();
    motor1.setValue(Integer.parseInt(val));
```

```
      }
      if (f == manual2 && tecla == KeyEvent.VK_ENTER) {
        val = manual2.getText();
        motor2.setValue(Integer.parseInt(val));
      }
      if (f == manual3 && tecla == KeyEvent.VK_ENTER) {
        val = manual3.getText();
        motor3.setValue(Integer.parseInt(val));
      }
    }
}
```

Java Class #3

```
public class Map {
   static int rTres(int n){
      //85 x n/180
      int resultado = (85*n)/180;
      return resultado;
   }
}
```

Arduino Program

```
#include <Servo.h>
Servo motor1;
Servo motor2;
Servo motor3;
int num;
int mov;

void setup(){
 Serial.begin(28800);
 motor1.attach(9);
 motor1.write(8);
 //motor1.write(169);

 motor2.attach(10);
 motor2.write(5);
 //motor2.write(163);

 motor3.attach(11);
 motor3.write(8);
 //motor3.write(171);
}

void loop(){
 if(Serial.available()>0){
   num = Serial.read();
```

```
  if(num >=0 && num <85){
    mov = map(num, 0, 85, 8, 169);
    motor1.write(mov);
  }
  if(num >=85 && num <170){
    mov = map(num, 85, 169, 5, 163);
    motor2.write(mov);
  }
  if(num >=170 && num <= 255){
    mov = map(num, 170, 255, 8, 171);
    motor3.write(mov);
  }
 }
}
```

Interface

The interface capable of running the three servo motors is very simple but interesting.