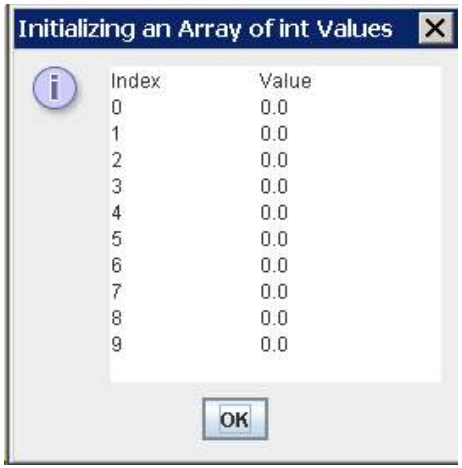
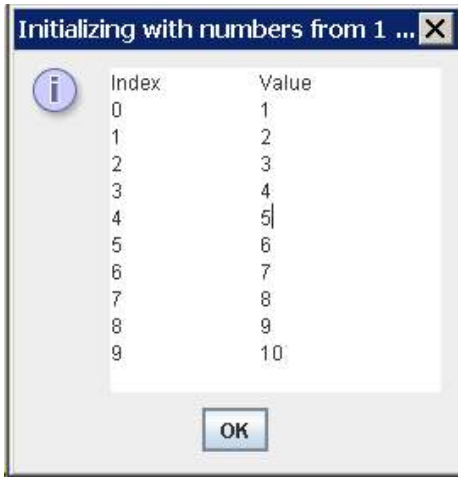
AppArray > PrArr1 > class InitArray



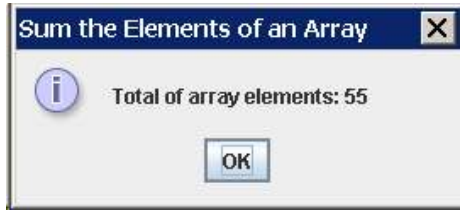

```
import javax.swing.*;
public class InitArray {
   public static void main(String[] args) {
      double tab[]; // declare reference to an array
      tab = new double[10]; // create array

      String output = "Index\tValue\n";
      // append each array element's value to String output
      for (int counter = 0; counter < tab.length; counter++)
         output += counter + "\t" + tab[counter] + "\n";

      JTextArea outputArea = new JTextArea();
      outputArea.setText(output);

      JOptionPane.showMessageDialog(null, outputArea,
                              "Initializing an Array of int Values",
                              JOptionPane.INFORMATION_MESSAGE);
      System.exit(0);
   } // end main
} // end class InitArray
```

AppArray > PrArr2 > class InitArray



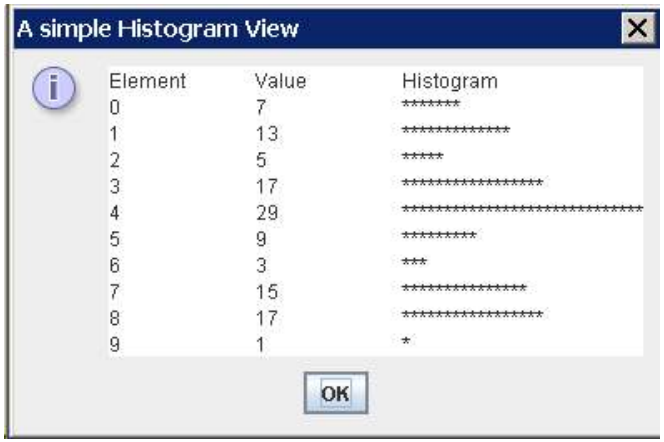

```
import javax.swing.*;

public class InitArray {
   public static void main( String args[] )
   {
      final int ARRAY_LENGTH = 10;    // constant
      int a[];                        // reference to int array
      a = new int[ ARRAY_LENGTH ];  // create array

      // calculate value for each array element
      for ( int i = 0; i < a.length; i++ )
         a[ i ] = 1 +  i;

      String output = "Index\tValue\n";

      for ( int i = 0; i < a.length; i++ )
         output += i + "\t" + a[ i ] + "\n";

      JTextArea outputArea = new JTextArea();
      outputArea.setText( output );

      JOptionPane.showMessageDialog( null, outputArea,
         "Initializing with numbers from 1 to 10",
         JOptionPane.INFORMATION_MESSAGE );

      System.exit( 0 );
   } // end main
} // end class InitArray
```

AppArray > PrArr3 > class SumArray




```
import javax.swing.*;

public class SumArray {

   public static void main(String[] args) {
      int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
      int suma = 0;

      // add each element's value to total
      for (int i = 0; i < a.length; i++)
         suma += a[i];

      JOptionPane.showMessageDialog(null, "Total of array elements: " + suma,
                                    "Sum the Elements of an Array",
                                    JOptionPane.INFORMATION_MESSAGE);
      System.exit(0);
   } // end main
} // end class SumArray
```

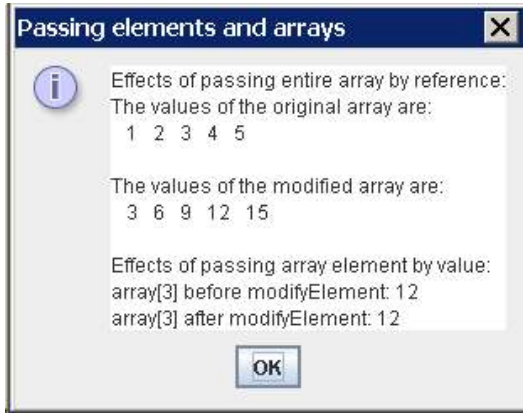
AppArray > PrArr4 > class Histogram




```
import javax.swing.*;

public class Histogram {
  public static void main( String args[] )
  {
    int a[] = { 7, 13, 5, 17, 29, 9, 3, 15, 17, 1 };

    String iesire = "Element\tValue\tHistogram";

    // for each array element, output a bar in histogram
    for ( int i = 0; i < a.length; i++ ) {
      iesire += "\n" + i + "\t" + a[ i ] + "\t";

      // print bar of asterisks
      for ( int stars = 0; stars < a[ i ]; stars++ )
        iesire += "*";

    } // end outer for

    JTextArea outputArea = new JTextArea();
    outputArea.setText( iesire );

    JOptionPane.showMessageDialog( null, outputArea,
      "A simple Histogram View", JOptionPane.INFORMATION_MESSAGE );

    System.exit( 0 );
  } // end main
} // end class Histogram
```

AppArray > PrArr5 > class PassArray

```
import javax.swing.*;

public class PassArray {

   public static void main(String[] args) {

      String iesire = "Effects of passing entire array by reference:\n" +
         "The values of the original array are:\n";

      int a[] = { 1, 2, 3, 4, 5 };

      // append original array elements to String output
      for (int i = 0; i < a.length; i++)
         iesire += "   " + a[i];

      modifyArray(a); // array passed by reference
      iesire += "\n\nThe values of the modified array are:\n";

      // append modified array elements to String output
      for (int i = 0; i < a.length; i++)
         iesire += "   " + a[i];

      iesire += "\n\nEffects of passing array element by value:\n" +
            "array[3] before modifyElement: " + a[3];

      modifyElement(a[3]); // attempt to modify array[ 3 ]

      iesire += "\narray[3] after modifyElement: " + a[3];

      //show the results
      JTextArea outputArea = new JTextArea();
      outputArea.setText(iesire);
      JOptionPane.showMessageDialog(null, outputArea,
                                    "Passing elements and arrays",
                                    JOptionPane.INFORMATION_MESSAGE);

   } // end method main

   // multiply each element of an array by 3
   public static void modifyArray(int[] array2) {
      for (int i = 0; i < array2.length; i++)
         array2[i] *= 3;
   }

   // multiply argument by 2
   public static void modifyElement(int element) {
      element *= 2;
   }
} // end class PassArray
```

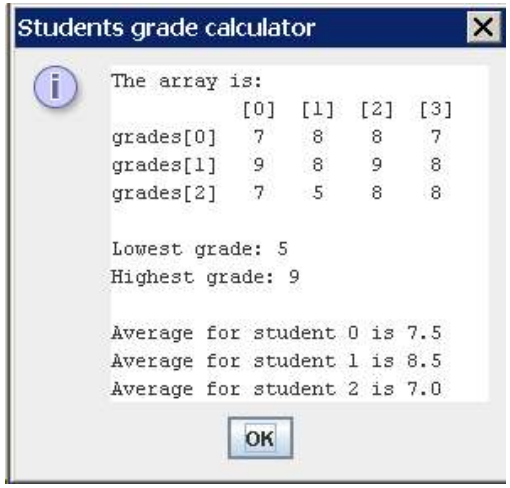


AppArray > PrArr6

class DoubleArray

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

public class DoubleArray {
   int grades[][] = { { 7, 8, 8, 7 },
                      { 9, 8, 9, 8 },
                      { 7, 5, 8, 8 } };

   int students, exams;
   String output;
   JTextArea outputArea;

   // constructor
   public DoubleArray() {

      students = grades.length;     // number of students
      exams = grades[ 0 ].length;   // number of exams

      // create JTextArea to shwo the rezults
      outputArea = new JTextArea();

      // build output string
      output = "The array is:\n";
      buildString();

      // call methods minimum and maximum
      output += "\n\nLowest grade: " + minimum() +
         "\nHighest grade: " + maximum() + "\n";

      // call method average to calculate each student's average
      for ( int counter = 0; counter < students; counter++ )
         output += "\nAverage for student " + counter + " is " +
            average( grades[ counter ] );  // pass one row of array grades

      // change outputArea's display font
      outputArea.setFont( new Font( "Monospaced", Font.PLAIN, 12 ) );

      // place output string in outputArea
      outputArea.setText( output );
       JOptionPane.showMessageDialog( null, outputArea,
         "Students grade calculator",
JOptionPane.INFORMATION_MESSAGE );

   } // end constructor

   // find minimum grade
   public int minimum()
   {
      // assume first element of grades array is smallest
      int min = grades[ 0 ][ 0 ];

      // loop through rows of grades array
      for ( int row = 0; row < students; row++ )
         // loop through columns of current row
         for ( int column = 0; column < exams; column++ )
            // if grade is less than lowGrade, assign it to lowGrade
            if ( grades[ row ][ column ] < min )
               min = grades[ row ][ column ];
      return min;  // return lowest grade
   } // end method minimum

   // find maximum grade
```

```
public int maximum()
{
  // assume first element of grades array is largest
  int max = grades[ 0 ][ 0 ];
  // loop through rows of grades array
  for ( int row = 0; row < students; row++ )
    // loop through columns of current row
    for ( int column = 0; column < exams; column++ )
      // if grade is greater than highGrade, assign it to highGrade
      if ( grades[ row ][ column ] > max )
        max = grades[ row ][ column ];
  return max;  // return highest grade
} // end method maximum

// determine average grade for particular student (or set of grades)
public double average( int setOfGrades[] )
{
  int suma = 0;  // initialize total
  // sum grades for one student
  for ( int i = 0; i < setOfGrades.length; i++ )
    suma += setOfGrades[ i ];
  // return average of grades
  return ( double ) suma / setOfGrades.length;
} // end method average

// build output string
public void buildString()
{
  output += "          ";  // used to align column heads
  // create column heads
  for ( int i = 0; i < exams; i++ )
    output += "[" + i + "]  ";
  // create rows/columns of text representing array grades
  for ( int row = 0; row < students; row++ ) {
    output += "\ngrades[" + row + "]   ";
    for ( int column = 0; column < exams; column++ )
      output += grades[ row ][ column ] + "    "; //4 spaces here
    }
  } // end method buildString
} // end class DoubleArray
```

class TestMat

```
public class TestMat {
   public static void main(String[] args) {
      DoubleArray testMat = new DoubleArray();
   }
}
```