public class SorterOlympicCountry {  
 public static void selectionSort( int [ ] array, int num )  
 {  
 int temp; // temporary location for swap  
 int max; // index of maximum value in subarray  
 for ( int i = 0; i < num; i++ )  
 {  
 max = indexOfLargestElement( array, num - i );  
 temp = array [max];  
 array [max] = array [num-1-i];  
 array [num-1-i] = temp;  
 }  
 } //end of method selectionSort  
 private static int indexOfLargestElement( int [ ] array, int size )  
 {  
 int index = 0;  
 for( int i = 1; i < size; i++ )  
 if ( array[i] > array[index] )  
 index = i;  
 return index;  
 } // end of method  
 public static void insertionSort( int [ ] array, int num)  
 {  
 int temp;  
 for ( int i = 1; i < num; i++ )  
 {  
 temp = array[i];  
 int j = i;  
 while ( j != 0 && array[j - 1] > temp ) {  
 array[j] = array[j - 1]; //shift  
 j--;  
 }  
 array[j] = temp; //stuff  
 }  
 }  
}   
 public static void bubbleSort( int [ ] array, int num ) // a “smarter” version  
 {  
 int temp = 0;  
 boolean arraySorted = false;  
 int loc = num - 1;  
 while ( loc > 0 && ! arraySorted)   
 {   
 arraySorted = true; // start a new iteration--maybe the array is sorted  
 for ( int j = 0; j < loc ; j++ ) // bubble down  
 if ( array [j] > array [j + 1] )  
 {  
 temp = array [j + 1]; // swap the adjacent elements  
 array [j + 1] = array [j];  
 array [j] = temp;  
 arraySorted = false; // and set arraySorted to false b/c  
 } //end of if  
 loc--;  
 } // end of while loop  
 } //end of method

