Exercise 4.6. Write a method

public static String maximum(ArrayList<String> a, Comparator<String> c)

that computes the largest string in the array list, using the ordering relationship that is defined by the given comparator. Supply a test program that uses this method to find the longest string in the list.

Exercise 4.7. Define an interface type Measurer as follows:

```
public interface Measurer {
      double measure (Object x) ;
}
```

Then supply a method

public static Object maximum(Object [] a, Measurer m)

that computes the object in the array with the largest measure. Test your method by populating an array list with rectangles and finding the one with the largest area.

Exercise 4.8. Define an interface type Filter as follows:

```
public interface Filter {
      boolean accept (String x) ;
}
```

Then supply a method

public static String[] filter(String [] a, Filter f)

that returns an array containing all elements of a that are accepted by the filter. Test your method by filtering an array of strings and accepting all strings that contain at most three characters.

Exercise 4.12. Add two methods

public static Comparator<Country> createComparatorByName(final boolean increasing)
public static Comparator<Country> createComparatorByArea(final boolean increasing)

to the Country class. The methods should return instances of anonymous classes that implement the Comparator interface type. The boolean parameters indicate whether the comparison should be in increasing or decreasing order. The parameters are declared fina1 so that you can access them in your compare methods.