Java

Java's Collections.sort() and Arrays.sort() methods are guaranteed stable.

The following sample demonstrates Java's sort stability:

```
import java.util.Arrays;
import java.util.Comparator;

public class RJSortStability {

  public static void main(String[] args) {
    String[] cityList = { "UK  London", "US  New York", "US  Birmingham", "UK  Birmingham", };

    String[] cn = cityList.clone();
    System.out.println("\nBefore sort:");
    for (String city : cn) {
      System.out.println(city);
    }

    // sort by city
    Arrays.sort(cn, new Comparator<String>() {
      public int compare(String lft, String rgt) {
        return lft.substring(4).compareTo(rgt.substring(4));
      }
    });

    System.out.println("\nAfter sort on city:");
    for (String city : cn) {
      System.out.println(city);
    }

    cn = cityList.clone();
    System.out.println("\nBefore sort:");
    for (String city : cn) {
      System.out.println(city);
    }

    // sort by country
    Arrays.sort(cn, new Comparator<String>() {
      public int compare(String lft, String rgt) {
        return lft.substring(0, 2).compareTo(rgt.substring(0, 2));
      }
    });

    System.out.println("\nAfter sort on country:");
    for (String city : cn) {
      System.out.println(city);
    }

    System.out.println();
  }
}
```

Output

```
Before sort:
UK  London
US  New York
US  Birmingham
UK  Birmingham

After sort on city:
US  Birmingham
UK  Birmingham
UK  London
US  New York

Before sort:
UK  London
US  New York
US  Birmingham
UK  Birmingham

After sort on country:
UK  London
UK  Birmingham
US  New York
US  Birmingham
```