```java
package sort;

public abstract class InsertionSort {

    protected abstract int compare(Object o1, Object o2);

    private void insert(int i, Object[] data) {
        if (i < data.length - 1) {
            if (compare(data[i], data[i + 1]) > 0) {
                Object tmp = data[i];
                data[i] = data[i + 1];
                data[i + 1] = tmp;
                insert(i + 1, data);
            }
        }
    }

    public void doSort(Object[] a) {
        for (int i = a.length - 2; i >= 0; --i) {
            insert(i, a);
        }
    }
}
```

```java
import sort.InsertionSort;

public class TestSort {

    public static void printArray(Object[] a) {
        for (Object o : a) {
            System.out.print(o + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        InsertionSort sort = new InsertionSort() {
            public int compare(Object o1, Object o2) {
                return ((String) o1).compareTo((String)o2);
            }
        };
        sort.doSort(args);
        printArray(args);
    }
}
```

```java
package sort;

public interface SortableData<T> {

    public int size();

    public void swap(int i, int j);

    public T get(int i);

    public int compare(int i, int j);
}
```

SortableComparableData.java

```
1 package sort;
2
3 public abstract class SortableComparableData<T extends Comparable<?
  super T>>
4         implements SortableData<T> {
5
6     public final int compare(int i, int j) {
7         return get(i).compareTo(get(j));
8     }
9 }
```

```java
package sort;

public class SortableArray<T extends Comparable<? super T>> extends SortableComparableData<T> {
    private T[] array;

    public SortableArray(T[] a) {
        this.array = a;
    }

    public int size() {
        return array.length;
    }

    public void swap(int i, int j) {
        T tmp = array[i];
        array[i] = array[j];
        array[j] = tmp;
    }

    public T get(int i) {
        return array[i];
    }
}
```

```java
package sort;

import java.util.List;

public class SortableList<T extends Comparable<? super T>> extends
        SortableComparableData<T> {

    List<T> list;

    public SortableList(List<T> list) {
        this.list = list;
    }

    public int size() {
        return list.size();
    }

    public void swap(int i, int j) {
        T tmp = list.get(i);
        list.set(i, list.get(j));
        list.set(j, tmp);
    }

    public T get(int i) {
        return list.get(i);
    }
}
```

```java
package sort;

public interface Sort {
    public void doSort(SortableData<?> data);
}
```

```java
package sort;

public class QuickSort implements Sort {

    private SortableData<?> data;

    public QuickSort() {
    }

    private void quickSort(int left, int right) {
        if (left < right) {
            int i = left;
            int j = right + 1;
            while (i < j) {
                while (data.compare(++i, left) < 0 && i < right)
                    ;
                while (data.compare(--j,left) > 0)
                    ;
                if (i < j) {
                    data.swap(i, j);
                }
            }
            data.swap(left, j);
            quickSort(left, j - 1);
            quickSort(j + 1, right);
        }
    }

    public void doSort(SortableData<?> data) {
        this.data = data;
        quickSort(0, data.size() - 1);
    }
}
```

```java
package sort;

public class InsertionSort implements Sort  {

    private SortableData<?> data;

    public InsertionSort() {
    }

    private void insert(int i) {
        if (i < data.size() - 1) {
            if (data.compare(i, i + 1) > 0) {
                data.swap(i, i+1);
                insert(i + 1);
            }
        }
    }

    public void doSort(SortableData<?> data) {
        this.data = data;
        for (int i = data.size() - 2; i >= 0; --i) {
            insert(i);
        }
    }
}
```