001/\*  
002 \* Licensed to the Apache Software Foundation (ASF) under one or more  
003 \* contributor license agreements. See the NOTICE file distributed with  
004 \* this work for additional information regarding copyright ownership.  
005 \* The ASF licenses this file to You under the Apache License, Version 2.0  
006 \* (the "License"); you may not use this file except in compliance with  
007 \* the License. You may obtain a copy of the License at  
008 \*  
009 \* http://www.apache.org/licenses/LICENSE-2.0  
010 \*  
011 \* Unless required by applicable law or agreed to in writing, software  
012 \* distributed under the License is distributed on an "AS IS" BASIS,  
013 \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
014 \* See the License for the specific language governing permissions and  
015 \* limitations under the License.  
016 \*/  
017package org.apache.commons.collections4.bag;  
018  
019import java.util.Comparator;  
020  
021import org.apache.commons.collections4.SortedBag;  
022  
023/\*\*  
024 \* Decorates another <code>SortedBag</code> to provide additional behaviour.  
025 \* <p>  
026 \* Methods are forwarded directly to the decorated bag.  
027 \* </p>  
028 \*  
029 \* @param <E> the type of elements in this bag  
030 \* @since 3.0  
031 \*/  
032public abstract class AbstractSortedBagDecorator<E>  
033 extends AbstractBagDecorator<E> implements SortedBag<E> {  
034  
035 /\*\* Serialization version \*/  
036 private static final long serialVersionUID = -8223473624050467718L;  
037  
038 /\*\*  
039 \* Constructor only used in deserialization, do not use otherwise.  
040 \* @since 3.1  
041 \*/  
042 protected AbstractSortedBagDecorator() {  
043 super();  
044 }  
045  
046 /\*\*  
047 \* Constructor that wraps (not copies).  
048 \*  
049 \* @param bag the bag to decorate, must not be null  
050 \* @throws NullPointerException if bag is null  
051 \*/  
052 protected AbstractSortedBagDecorator(final SortedBag<E> bag) {  
053 super(bag);  
054 }  
055  
056 /\*\*  
057 \* Gets the bag being decorated.  
058 \*  
059 \* @return the decorated bag  
060 \*/  
061 @Override  
062 protected SortedBag<E> decorated() {  
063 return (SortedBag<E>) super.decorated();  
064 }  
065  
066 //-----------------------------------------------------------------------  
067  
068 @Override  
069 public E first() {  
070 return decorated().first();  
071 }  
072  
073 @Override  
074 public E last() {  
075 return decorated().last();  
076 }  
077  
078 @Override  
079 public Comparator<? super E> comparator() {  
080 return decorated().comparator();  
081 }  
082  
083}