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;  
018  
019import java.util.Comparator;  
020import java.util.SortedMap;  
021  
022/\*\*  
023 \* Defines a map that allows bidirectional lookup between key and values  
024 \* and retains both keys and values in sorted order.  
025 \* <p>  
026 \* Implementations should allow a value to be looked up from a key and  
027 \* a key to be looked up from a value with equal performance.  
028 \* </p>  
029 \*  
030 \* @param <K> the type of the keys in the map  
031 \* @param <V> the type of the values in the map  
032 \* @since 3.0  
033 \*/  
034public interface SortedBidiMap<K, V> extends OrderedBidiMap<K, V>, SortedMap<K, V> {  
035  
036 /\*\*  
037 \* Gets a view of this map where the keys and values are reversed.  
038 \* <p>  
039 \* Changes to one map will be visible in the other and vice versa.  
040 \* This enables both directions of the map to be accessed equally.  
041 \* <p>  
042 \* Implementations should seek to avoid creating a new object every time this  
043 \* method is called. See <code>AbstractMap.values()</code> etc. Calling this  
044 \* method on the inverse map should return the original.  
045 \* <p>  
046 \* Implementations must return a <code>SortedBidiMap</code> instance,  
047 \* usually by forwarding to <code>inverseSortedBidiMap()</code>.  
048 \*  
049 \* @return an inverted bidirectional map  
050 \*/  
051 @Override  
052 SortedBidiMap<V, K> inverseBidiMap();  
053  
054 /\*\*  
055 \* Get the comparator used for the values in the value-to-key map aspect.  
056 \* @return Comparator<? super V>  
057 \*/  
058 Comparator<? super V> valueComparator();  
059}