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