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 an iterator that operates over an ordered <code>Map</code>.  
021 \* <p>  
022 \* This iterator allows both forward and reverse iteration through the map.  
023 \* </p>  
024 \*  
025 \* @param <K> the type of the keys in the map  
026 \* @param <V> the type of the values in the map  
027 \* @since 3.0  
028 \*/  
029public interface OrderedMapIterator<K, V> extends MapIterator<K, V>, OrderedIterator<K> {  
030  
031 /\*\*  
032 \* Checks to see if there is a previous entry that can be iterated to.  
033 \*  
034 \* @return <code>true</code> if the iterator has a previous element  
035 \*/  
036 @Override  
037 boolean hasPrevious();  
038  
039 /\*\*  
040 \* Gets the previous <em>key</em> from the <code>Map</code>.  
041 \*  
042 \* @return the previous key in the iteration  
043 \* @throws java.util.NoSuchElementException if the iteration is finished  
044 \*/  
045 @Override  
046 K previous();  
047  
048}