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.List;  
020  
021/\*\*  
022 \* Defines a map that holds a list of values against each key.  
023 \* <p>  
024 \* A {@code ListValuedMap} is a Map with slightly different semantics:  
025 \* </p>  
026 \* <ul>  
027 \* <li>Putting a value into the map will add the value to a {@link List} at that key.</li>  
028 \* <li>Getting a value will return a {@link List}, holding all the values put to that key.</li>  
029 \* </ul>  
030 \*  
031 \* @param <K> the type of the keys in this map  
032 \* @param <V> the type of the values in this map  
033 \* @since 4.1  
034 \*/  
035public interface ListValuedMap<K, V> extends MultiValuedMap<K, V> {  
036  
037 /\*\*  
038 \* Gets the list of values associated with the specified key.  
039 \* <p>  
040 \* This method will return an <b>empty</b> list if  
041 \* {@link #containsKey(Object)} returns {@code false}. Changes to the  
042 \* returned list will update the underlying {@code ListValuedMap} and  
043 \* vice-versa.  
044 \*  
045 \* @param key the key to retrieve  
046 \* @return the {@code List} of values, implementations should return an  
047 \* empty {@code List} for no mapping  
048 \* @throws NullPointerException if the key is null and null keys are invalid  
049 \*/  
050 @Override  
051 List<V> get(K key);  
052  
053 /\*\*  
054 \* Removes all values associated with the specified key.  
055 \* <p>  
056 \* The returned list <i>may</i> be modifiable, but updates will not be  
057 \* propagated to this list-valued map. In case no mapping was stored for the  
058 \* specified key, an empty, unmodifiable list will be returned.  
059 \*  
060 \* @param key the key to remove values from  
061 \* @return the {@code List} of values removed, implementations  
062 \* typically return an empty, unmodifiable {@code List} for no mapping found  
063 \* @throws UnsupportedOperationException if the map is unmodifiable  
064 \* @throws NullPointerException if the key is null and null keys are invalid  
065 \*/  
066 @Override  
067 List<V> remove(Object key);  
068  
069}