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.SortedMap;  
020  
021/\*\*  
022 \* Defines the interface for a prefix tree, an ordered tree data structure. For  
023 \* more information, see <a href="http://en.wikipedia.org/wiki/Trie">Tries</a>.  
024 \*  
025 \* @param <K> the type of the keys in this map  
026 \* @param <V> the type of the values in this map  
027 \* @since 4.0  
028 \*/  
029public interface Trie<K, V> extends IterableSortedMap<K, V> {  
030  
031 /\*\*  
032 \* Returns a view of this {@link Trie} of all elements that are prefixed  
033 \* by the given key.  
034 \* <p>  
035 \* In a {@link Trie} with fixed size keys, this is essentially a  
036 \* {@link #get(Object)} operation.  
037 \* <p>  
038 \* For example, if the {@link Trie} contains 'Anna', 'Anael',  
039 \* 'Analu', 'Andreas', 'Andrea', 'Andres', and 'Anatole', then  
040 \* a lookup of 'And' would return 'Andreas', 'Andrea', and 'Andres'.  
041 \*  
042 \* @param key the key used in the search  
043 \* @return a {@link SortedMap} view of this {@link Trie} with all elements whose  
044 \* key is prefixed by the search key  
045 \*/  
046 SortedMap<K, V> prefixMap(K key);  
047  
048}