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.iterators;  
018  
019import org.apache.commons.collections4.MapIterator;  
020import org.apache.commons.collections4.Unmodifiable;  
021  
022/\*\*  
023 \* Decorates a map iterator such that it cannot be modified.  
024 \* <p>  
025 \* Attempts to modify it will result in an UnsupportedOperationException.  
026 \* </p>  
027 \*  
028 \* @param <K> the type of keys  
029 \* @param <V> the type of mapped values  
030 \* @since 3.0  
031 \*/  
032public final class UnmodifiableMapIterator<K, V> implements MapIterator<K, V>, Unmodifiable {  
033  
034 /\*\* The iterator being decorated \*/  
035 private final MapIterator<? extends K, ? extends V> iterator;  
036  
037 //-----------------------------------------------------------------------  
038 /\*\*  
039 \* Decorates the specified iterator such that it cannot be modified.  
040 \*  
041 \* @param <K> the key type  
042 \* @param <V> the value type  
043 \* @param iterator the iterator to decorate  
044 \* @return a new unmodifiable map iterator  
045 \* @throws NullPointerException if the iterator is null  
046 \*/  
047 public static <K, V> MapIterator<K, V> unmodifiableMapIterator(  
048 final MapIterator<? extends K, ? extends V> iterator) {  
049 if (iterator == null) {  
050 throw new NullPointerException("MapIterator must not be null");  
051 }  
052 if (iterator instanceof Unmodifiable) {  
053 @SuppressWarnings("unchecked") // safe to upcast  
054 final MapIterator<K, V> tmpIterator = (MapIterator<K, V>) iterator;  
055 return tmpIterator;  
056 }  
057 return new UnmodifiableMapIterator<>(iterator);  
058 }  
059  
060 //-----------------------------------------------------------------------  
061 /\*\*  
062 \* Constructor.  
063 \*  
064 \* @param iterator the iterator to decorate  
065 \*/  
066 private UnmodifiableMapIterator(final MapIterator<? extends K, ? extends V> iterator) {  
067 super();  
068 this.iterator = iterator;  
069 }  
070  
071 //-----------------------------------------------------------------------  
072 @Override  
073 public boolean hasNext() {  
074 return iterator.hasNext();  
075 }  
076  
077 @Override  
078 public K next() {  
079 return iterator.next();  
080 }  
081  
082 @Override  
083 public K getKey() {  
084 return iterator.getKey();  
085 }  
086  
087 @Override  
088 public V getValue() {  
089 return iterator.getValue();  
090 }  
091  
092 @Override  
093 public V setValue(final V value) {  
094 throw new UnsupportedOperationException("setValue() is not supported");  
095 }  
096  
097 @Override  
098 public void remove() {  
099 throw new UnsupportedOperationException("remove() is not supported");  
100 }  
101  
102}