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.keyvalue;  
018  
019import java.util.Map;  
020  
021import org.apache.commons.collections4.KeyValue;  
022import org.apache.commons.collections4.Unmodifiable;  
023  
024/\*\*  
025 \* A {@link java.util.Map.Entry Map.Entry} that throws  
026 \* UnsupportedOperationException when <code>setValue</code> is called.  
027 \*  
028 \* @param <K> the type of keys  
029 \* @param <V> the type of mapped values  
030 \* @since 3.0  
031 \*/  
032public final class UnmodifiableMapEntry<K, V> extends AbstractMapEntry<K, V> implements Unmodifiable {  
033  
034 /\*\*  
035 \* Constructs a new entry with the specified key and given value.  
036 \*  
037 \* @param key the key for the entry, may be null  
038 \* @param value the value for the entry, may be null  
039 \*/  
040 public UnmodifiableMapEntry(final K key, final V value) {  
041 super(key, value);  
042 }  
043  
044 /\*\*  
045 \* Constructs a new entry from the specified <code>KeyValue</code>.  
046 \*  
047 \* @param pair the pair to copy, must not be null  
048 \* @throws NullPointerException if the entry is null  
049 \*/  
050 public UnmodifiableMapEntry(final KeyValue<? extends K, ? extends V> pair) {  
051 super(pair.getKey(), pair.getValue());  
052 }  
053  
054 /\*\*  
055 \* Constructs a new entry from the specified <code>Map.Entry</code>.  
056 \*  
057 \* @param entry the entry to copy, must not be null  
058 \* @throws NullPointerException if the entry is null  
059 \*/  
060 public UnmodifiableMapEntry(final Map.Entry<? extends K, ? extends V> entry) {  
061 super(entry.getKey(), entry.getValue());  
062 }  
063  
064 /\*\*  
065 \* Throws UnsupportedOperationException.  
066 \*  
067 \* @param value the new value  
068 \* @return the previous value  
069 \* @throws UnsupportedOperationException always  
070 \*/  
071 @Override  
072 public V setValue(final V value) {  
073 throw new UnsupportedOperationException("setValue() is not supported");  
074 }  
075  
076}