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;  
022  
023/\*\*  
024 \* A restricted implementation of {@link java.util.Map.Entry Map.Entry} that prevents  
025 \* the {@link java.util.Map.Entry Map.Entry} contract from being broken.  
026 \*  
027 \* @param <K> the type of keys  
028 \* @param <V> the type of mapped values  
029 \* @since 3.0  
030 \*/  
031public final class DefaultMapEntry<K, V> extends AbstractMapEntry<K, V> {  
032  
033 /\*\*  
034 \* Constructs a new entry with the specified key and given value.  
035 \*  
036 \* @param key the key for the entry, may be null  
037 \* @param value the value for the entry, may be null  
038 \*/  
039 public DefaultMapEntry(final K key, final V value) {  
040 super(key, value);  
041 }  
042  
043 /\*\*  
044 \* Constructs a new entry from the specified <code>KeyValue</code>.  
045 \*  
046 \* @param pair the pair to copy, must not be null  
047 \* @throws NullPointerException if the entry is null  
048 \*/  
049 public DefaultMapEntry(final KeyValue<? extends K, ? extends V> pair) {  
050 super(pair.getKey(), pair.getValue());  
051 }  
052  
053 /\*\*  
054 \* Constructs a new entry from the specified <code>Map.Entry</code>.  
055 \*  
056 \* @param entry the entry to copy, must not be null  
057 \* @throws NullPointerException if the entry is null  
058 \*/  
059 public DefaultMapEntry(final Map.Entry<? extends K, ? extends V> entry) {  
060 super(entry.getKey(), entry.getValue());  
061 }  
062  
063}