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.Map;  
020  
021/\*\*  
022 \* The "write" subset of the {@link Map} interface.  
023 \* <p>  
024 \* NOTE: in the original {@link Map} interface, {@link Map#put(Object, Object)} is known  
025 \* to have the same return type as {@link Map#get(Object)}, namely {@code V}. {@link Put}  
026 \* makes no assumptions in this regard (there is no association with, nor even knowledge  
027 \* of, a "reading" interface) and thus defines {@link #put(Object, Object)} as returning  
028 \* {@link Object}.  
029 \* </p>  
030 \*  
031 \* @param <K> the type of the keys in this map  
032 \* @param <V> the type of the values in this map  
033 \*  
034 \* @since 4.0  
035 \* @see Get  
036 \*/  
037public interface Put<K, V> {  
038  
039 /\*\*  
040 \* @see Map#clear()  
041 \*/  
042 void clear();  
043  
044 /\*\*  
045 \* Note that the return type is Object, rather than V as in the Map interface.  
046 \* See the class Javadoc for further info.  
047 \*  
048 \* @param key key with which the specified value is to be associated  
049 \* @param value value to be associated with the specified key  
050 \* @return the previous value associated with <code>key</code>, or  
051 \* <code>null</code> if there was no mapping for <code>key</code>.  
052 \* (A <code>null</code> return can also indicate that the map  
053 \* previously associated <code>null</code> with <code>key</code>,  
054 \* if the implementation supports <code>null</code> values.)  
055 \* @see Map#put(Object, Object)  
056 \*/  
057 Object put(K key, V value);  
058  
059 /\*\*  
060 \* @param t mappings to be stored in this map  
061 \* @see Map#putAll(Map)  
062 \*/  
063 void putAll(Map<? extends K, ? extends V> t);  
064  
065}