001/\*  
002 \* Licensed to the Apache Software Foundation (ASF) under one or more contributor license  
003 \* agreements. See the NOTICE file distributed with this work for additional information regarding  
004 \* copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the  
005 \* "License"); you may not use this file except in compliance with the License. You may obtain a  
006 \* copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable  
007 \* law or agreed to in writing, software distributed under the License is distributed on an "AS IS"  
008 \* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License  
009 \* for the specific language governing permissions and limitations under the License.  
010 \*/  
011package org.apache.commons.collections4;  
012  
013/\*\*  
014 \* An equation function, which determines equality between objects of type T.  
015 \* <p>  
016 \* It is the functional sibling of {@link java.util.Comparator}; {@link Equator} is to  
017 \* {@link Object} as {@link java.util.Comparator} is to {@link java.lang.Comparable}.  
018 \* </p>  
019 \*  
020 \* @param <T> the types of object this {@link Equator} can evaluate.  
021 \* @since 4.0  
022 \*/  
023public interface Equator<T> {  
024 /\*\*  
025 \* Evaluates the two arguments for their equality.  
026 \*  
027 \* @param o1 the first object to be equated.  
028 \* @param o2 the second object to be equated.  
029 \* @return whether the two objects are equal.  
030 \*/  
031 boolean equate(T o1, T o2);  
032  
033 /\*\*  
034 \* Calculates the hash for the object, based on the method of equality used in the equate  
035 \* method. This is used for classes that delegate their {@link Object#equals(Object) equals(Object)} method to an  
036 \* Equator (and so must also delegate their {@link Object#hashCode() hashCode()} method), or for implementations  
037 \* of {@link org.apache.commons.collections4.map.HashedMap} that use an Equator for the key objects.  
038 \*  
039 \* @param o the object to calculate the hash for.  
040 \* @return the hash of the object.  
041 \*/  
042 int hash(T o);  
043}