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.functors;  
018  
019import java.io.Serializable;  
020  
021import org.apache.commons.collections4.Equator;  
022  
023/\*\*  
024 \* Default {@link Equator} implementation.  
025 \*  
026 \* @param <T> the types of object this {@link Equator} can evaluate.  
027 \* @since 4.0  
028 \*/  
029public class DefaultEquator<T> implements Equator<T>, Serializable {  
030  
031 /\*\* Serial version UID \*/  
032 private static final long serialVersionUID = 825802648423525485L;  
033  
034 /\*\* Static instance \*/  
035 @SuppressWarnings("rawtypes") // the static instance works for all types  
036 public static final DefaultEquator INSTANCE = new DefaultEquator<>();  
037  
038 /\*\*  
039 \* Hashcode used for <code>null</code> objects.  
040 \*/  
041 public static final int HASHCODE\_NULL = -1;  
042  
043 /\*\*  
044 \* Factory returning the typed singleton instance.  
045 \*  
046 \* @param <T> the object type  
047 \* @return the singleton instance  
048 \*/  
049 public static <T> DefaultEquator<T> defaultEquator() {  
050 return DefaultEquator.INSTANCE;  
051 }  
052  
053 /\*\*  
054 \* Restricted constructor.  
055 \*/  
056 private DefaultEquator() {  
057 super();  
058 }  
059  
060 /\*\*  
061 \* {@inheritDoc} Delegates to {@link Object#equals(Object)}.  
062 \*/  
063 @Override  
064 public boolean equate(final T o1, final T o2) {  
065 return o1 == o2 || o1 != null && o1.equals(o2);  
066 }  
067  
068 /\*\*  
069 \* {@inheritDoc}  
070 \*  
071 \* @return <code>o.hashCode()</code> if <code>o</code> is non-  
072 \* <code>null</code>, else {@link #HASHCODE\_NULL}.  
073 \*/  
074 @Override  
075 public int hash(final T o) {  
076 return o == null ? HASHCODE\_NULL : o.hashCode();  
077 }  
078  
079 private Object readResolve() {  
080 return INSTANCE;  
081 }  
082  
083}