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.Predicate;  
022  
023/\*\*  
024 \* Predicate implementation that returns true if the input is the same object  
025 \* as the one stored in this predicate.  
026 \*  
027 \* @since 3.0  
028 \*/  
029public final class IdentityPredicate<T> implements Predicate<T>, Serializable {  
030  
031 /\*\* Serial version UID \*/  
032 private static final long serialVersionUID = -89901658494523293L;  
033  
034 /\*\* The value to compare to \*/  
035 private final T iValue;  
036  
037 /\*\*  
038 \* Factory to create the identity predicate.  
039 \*  
040 \* @param <T> the type that the predicate queries  
041 \* @param object the object to compare to  
042 \* @return the predicate  
043 \*/  
044 public static <T> Predicate<T> identityPredicate(final T object) {  
045 if (object == null) {  
046 return NullPredicate.<T>nullPredicate();  
047 }  
048 return new IdentityPredicate<>(object);  
049 }  
050  
051 /\*\*  
052 \* Constructor that performs no validation.  
053 \* Use <code>identityPredicate</code> if you want that.  
054 \*  
055 \* @param object the object to compare to  
056 \*/  
057 public IdentityPredicate(final T object) {  
058 super();  
059 iValue = object;  
060 }  
061  
062 /\*\*  
063 \* Evaluates the predicate returning true if the input object is identical to  
064 \* the stored object.  
065 \*  
066 \* @param object the input object  
067 \* @return true if input is the same object as the stored value  
068 \*/  
069 @Override  
070 public boolean evaluate(final T object) {  
071 return iValue == object;  
072 }  
073  
074 /\*\*  
075 \* Gets the value.  
076 \*  
077 \* @return the value  
078 \* @since 3.1  
079 \*/  
080 public T getValue() {  
081 return iValue;  
082 }  
083  
084}