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;  
020import java.util.HashSet;  
021import java.util.Set;  
022  
023import org.apache.commons.collections4.Predicate;  
024  
025/\*\*  
026 \* Predicate implementation that returns true the first time an object is  
027 \* passed into the predicate.  
028 \*  
029 \* @since 3.0  
030 \*/  
031public final class UniquePredicate<T> implements Predicate<T>, Serializable {  
032  
033 /\*\* Serial version UID \*/  
034 private static final long serialVersionUID = -3319417438027438040L;  
035  
036 /\*\* The set of previously seen objects \*/  
037 private final Set<T> iSet = new HashSet<>();  
038  
039 /\*\*  
040 \* Factory to create the predicate.  
041 \*  
042 \* @param <T> the type that the predicate queries  
043 \* @return the predicate  
044 \* @throws IllegalArgumentException if the predicate is null  
045 \*/  
046 public static <T> Predicate<T> uniquePredicate() {  
047 return new UniquePredicate<>();  
048 }  
049  
050 /\*\*  
051 \* Constructor that performs no validation.  
052 \* Use <code>uniquePredicate</code> if you want that.  
053 \*/  
054 public UniquePredicate() {  
055 super();  
056 }  
057  
058 /\*\*  
059 \* Evaluates the predicate returning true if the input object hasn't been  
060 \* received yet.  
061 \*  
062 \* @param object the input object  
063 \* @return true if this is the first time the object is seen  
064 \*/  
065 @Override  
066 public boolean evaluate(final T object) {  
067 return iSet.add(object);  
068 }  
069  
070}