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  
019/\*\*  
020 \* Defines a functor interface implemented by classes that perform a predicate  
021 \* test on an object.  
022 \* <p>  
023 \* A <code>Predicate</code> is the object equivalent of an <code>if</code> statement.  
024 \* It uses the input object to return a true or false value, and is often used in  
025 \* validation or filtering.  
026 \* </p>  
027 \* <p>  
028 \* Standard implementations of common predicates are provided by  
029 \* {@link PredicateUtils}. These include true, false, instanceof, equals, and,  
030 \* or, not, method invokation and null testing.  
031 \* </p>  
032 \*  
033 \* @param <T> the type that the predicate queries  
034 \*  
035 \* @since 1.0  
036 \*/  
037@FunctionalInterface  
038public interface Predicate<T> {  
039  
040 /\*\*  
041 \* Use the specified parameter to perform a test that returns true or false.  
042 \*  
043 \* @param object the object to evaluate, should not be changed  
044 \* @return true or false  
045 \* @throws ClassCastException (runtime) if the input is the wrong class  
046 \* @throws IllegalArgumentException (runtime) if the input is invalid  
047 \* @throws FunctorException (runtime) if the predicate encounters a problem  
048 \*/  
049 boolean evaluate(T object);  
050  
051}