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 transform one  
021 \* object into another.  
022 \* <p>  
023 \* A <code>Transformer</code> converts the input object to the output object.  
024 \* The input object should be left unchanged.  
025 \* Transformers are typically used for type conversions, or extracting data  
026 \* from an object.  
027 \* </p>  
028 \* <p>  
029 \* Standard implementations of common transformers are provided by  
030 \* {@link TransformerUtils}. These include method invocation, returning a constant,  
031 \* cloning and returning the string value.  
032 \* </p>  
033 \*  
034 \* @param <I> the input type to the transformer  
035 \* @param <O> the output type from the transformer  
036 \*  
037 \* @since 1.0  
038 \*/  
039@FunctionalInterface  
040public interface Transformer<I, O> {  
041  
042 /\*\*  
043 \* Transforms the input object (leaving it unchanged) into some output object.  
044 \*  
045 \* @param input the object to be transformed, should be left unchanged  
046 \* @return a transformed object  
047 \* @throws ClassCastException (runtime) if the input is the wrong class  
048 \* @throws IllegalArgumentException (runtime) if the input is invalid  
049 \* @throws FunctorException (runtime) if the transform cannot be completed  
050 \*/  
051 O transform(I input);  
052  
053}