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 org.apache.commons.collections4.Transformer;  
020  
021/\*\*  
022 \* Transformer implementation that returns a clone of the input object.  
023 \* <p>  
024 \* Clone is performed using <code>PrototypeFactory.prototypeFactory(input).create()</code>.  
025 \* </p>  
026 \* <p>  
027 \* <b>WARNING:</b> from v4.1 onwards this class will <b>not</b> be serializable anymore  
028 \* in order to prevent potential remote code execution exploits. Please refer to  
029 \* <a href="https://issues.apache.org/jira/browse/COLLECTIONS-580">COLLECTIONS-580</a>  
030 \* for more details.  
031 \* </p>  
032 \*  
033 \* @since 3.0  
034 \*/  
035public class CloneTransformer<T> implements Transformer<T, T> {  
036  
037 /\*\* Singleton predicate instance \*/  
038 @SuppressWarnings("rawtypes") // the singleton instance works for all types  
039 public static final Transformer INSTANCE = new CloneTransformer<>();  
040  
041 /\*\*  
042 \* Factory returning the singleton instance.  
043 \*  
044 \* @param <T> the type of the objects to be cloned  
045 \* @return the singleton instance  
046 \* @since 3.1  
047 \*/  
048 public static <T> Transformer<T, T> cloneTransformer() {  
049 return INSTANCE;  
050 }  
051  
052 /\*\*  
053 \* Constructor.  
054 \*/  
055 private CloneTransformer() {  
056 super();  
057 }  
058  
059 /\*\*  
060 \* Transforms the input to result by cloning it.  
061 \*  
062 \* @param input the input object to transform  
063 \* @return the transformed result  
064 \*/  
065 @Override  
066 public T transform(final T input) {  
067 if (input == null) {  
068 return null;  
069 }  
070 return PrototypeFactory.prototypeFactory(input).create();  
071 }  
072  
073}