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.beanutils.converters;  
018  
019import java.math.BigInteger;  
020  
021/\*\*  
022 \* {@link NumberConverter} implementation that handles conversion to  
023 \* and from <b>java.math.BigInteger</b> objects.  
024 \* <p>  
025 \* This implementation can be configured to handle conversion either  
026 \* by using BigInteger's default String conversion, or by using a Locale's pattern  
027 \* or by specifying a format pattern. See the {@link NumberConverter}  
028 \* documentation for further details.  
029 \* <p>  
030 \* Can be configured to either return a <i>default value</i> or throw a  
031 \* <code>ConversionException</code> if a conversion error occurs.  
032 \*  
033 \* @version $Id$  
034 \* @since 1.3  
035 \*/  
036public final class BigIntegerConverter extends NumberConverter {  
037  
038 /\*\*  
039 \* Construct a <b>java.math.BigInteger</b> <i>Converter</i> that throws  
040 \* a <code>ConversionException</code> if an error occurs.  
041 \*/  
042 public BigIntegerConverter() {  
043 super(false);  
044 }  
045  
046 /\*\*  
047 \* Construct a <b>java.math.BigInteger</b> <i>Converter</i> that returns  
048 \* a default value if an error occurs.  
049 \*  
050 \* @param defaultValue The default value to be returned  
051 \* if the value to be converted is missing or an error  
052 \* occurs converting the value.  
053 \*/  
054 public BigIntegerConverter(final Object defaultValue) {  
055 super(false, defaultValue);  
056 }  
057  
058 /\*\*  
059 \* Return the default type this <code>Converter</code> handles.  
060 \*  
061 \* @return The default type this <code>Converter</code> handles.  
062 \* @since 1.8.0  
063 \*/  
064 @Override  
065 protected Class<BigInteger> getDefaultType() {  
066 return BigInteger.class;  
067 }  
068  
069}