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