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.iterators;  
018  
019import java.util.Iterator;  
020  
021/\*\*  
022 \* Provides basic behaviour for decorating an iterator with extra functionality.  
023 \* <p>  
024 \* All methods are forwarded to the decorated iterator.  
025 \*  
026 \* @since 3.0  
027 \*/  
028public abstract class AbstractIteratorDecorator<E> extends AbstractUntypedIteratorDecorator<E, E> {  
029  
030 //-----------------------------------------------------------------------  
031 /\*\*  
032 \* Constructor that decorates the specified iterator.  
033 \*  
034 \* @param iterator the iterator to decorate, must not be null  
035 \* @throws NullPointerException if the iterator is null  
036 \*/  
037 protected AbstractIteratorDecorator(final Iterator<E> iterator) {  
038 super(iterator);  
039 }  
040  
041 /\*\* {@inheritDoc} \*/  
042 @Override  
043 public E next() {  
044 return getIterator().next();  
045 }  
046  
047}