| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CollationElementIterator.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/text/ChoiceFormat.html)   [**NEXT CLASS**](http://docs.google.com/java/text/CollationKey.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/CollationElementIterator.html)    [**NO FRAMES**](http://docs.google.com/CollationElementIterator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#4d34og8) |

## **java.text**

Class CollationElementIterator

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.text.CollationElementIterator**

public final class **CollationElementIterator**extends [Object](http://docs.google.com/java/lang/Object.html)

The CollationElementIterator class is used as an iterator to walk through each character of an international string. Use the iterator to return the ordering priority of the positioned character. The ordering priority of a character, which we refer to as a key, defines how a character is collated in the given collation object.

For example, consider the following in Spanish:

"ca" -> the first key is key('c') and second key is key('a').  
 "cha" -> the first key is key('ch') and second key is key('a').

And in German,

"�b"-> the first key is key('a'), the second key is key('e'), and  
 the third key is key('b').

The key of a character is an integer composed of primary order(short), secondary order(byte), and tertiary order(byte). Java strictly defines the size and signedness of its primitive data types. Therefore, the static functions primaryOrder, secondaryOrder, and tertiaryOrder return int, short, and short respectively to ensure the correctness of the key value.

Example of the iterator usage,

String testString = "This is a test";  
 RuleBasedCollator ruleBasedCollator = (RuleBasedCollator)Collator.getInstance();  
 CollationElementIterator collationElementIterator = ruleBasedCollator.getCollationElementIterator(testString);  
 int primaryOrder = CollationElementIterator.primaryOrder(collationElementIterator.next());

CollationElementIterator.next returns the collation order of the next character. A collation order consists of primary order, secondary order and tertiary order. The data type of the collation order is **int**. The first 16 bits of a collation order is its primary order; the next 8 bits is the secondary order and the last 8 bits is the tertiary order.

**See Also:**[Collator](http://docs.google.com/java/text/Collator.html), [RuleBasedCollator](http://docs.google.com/java/text/RuleBasedCollator.html)

| **Field Summary** | |
| --- | --- |
| static int | [**NULLORDER**](http://docs.google.com/java/text/CollationElementIterator.html#NULLORDER)            Null order which indicates the end of string is reached by the cursor. |

| **Method Summary** | |
| --- | --- |
| int | [**getMaxExpansion**](http://docs.google.com/java/text/CollationElementIterator.html#getMaxExpansion(int))(int order)            Return the maximum length of any expansion sequences that end with the specified comparison order. |
| int | [**getOffset**](http://docs.google.com/java/text/CollationElementIterator.html#getOffset())()            Returns the character offset in the original text corresponding to the next collation element. |
| int | [**next**](http://docs.google.com/java/text/CollationElementIterator.html#next())()            Get the next collation element in the string. |
| int | [**previous**](http://docs.google.com/java/text/CollationElementIterator.html#previous())()            Get the previous collation element in the string. |
| static int | [**primaryOrder**](http://docs.google.com/java/text/CollationElementIterator.html#primaryOrder(int))(int order)            Return the primary component of a collation element. |
| void | [**reset**](http://docs.google.com/java/text/CollationElementIterator.html#reset())()            Resets the cursor to the beginning of the string. |
| static short | [**secondaryOrder**](http://docs.google.com/java/text/CollationElementIterator.html#secondaryOrder(int))(int order)            Return the secondary component of a collation element. |
| void | [**setOffset**](http://docs.google.com/java/text/CollationElementIterator.html#setOffset(int))(int newOffset)            Sets the iterator to point to the collation element corresponding to the specified character (the parameter is a CHARACTER offset in the original string, not an offset into its corresponding sequence of collation elements). |
| void | [**setText**](http://docs.google.com/java/text/CollationElementIterator.html#setText(java.text.CharacterIterator))([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) source)            Set a new string over which to iterate. |
| void | [**setText**](http://docs.google.com/java/text/CollationElementIterator.html#setText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) source)            Set a new string over which to iterate. |
| static short | [**tertiaryOrder**](http://docs.google.com/java/text/CollationElementIterator.html#tertiaryOrder(int))(int order)            Return the tertiary component of a collation element. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### NULLORDER

public static final int **NULLORDER**

Null order which indicates the end of string is reached by the cursor.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.text.CollationElementIterator.NULLORDER)

| **Method Detail** |
| --- |

### reset

public void **reset**()

Resets the cursor to the beginning of the string. The next call to next() will return the first collation element in the string.

### next

public int **next**()

Get the next collation element in the string.

This iterator iterates over a sequence of collation elements that were built from the string. Because there isn't necessarily a one-to-one mapping from characters to collation elements, this doesn't mean the same thing as "return the collation element [or ordering priority] of the next character in the string".

This function returns the collation element that the iterator is currently pointing to and then updates the internal pointer to point to the next element. previous() updates the pointer first and then returns the element. This means that when you change direction while iterating (i.e., call next() and then call previous(), or call previous() and then call next()), you'll get back the same element twice.

### previous

public int **previous**()

Get the previous collation element in the string.

This iterator iterates over a sequence of collation elements that were built from the string. Because there isn't necessarily a one-to-one mapping from characters to collation elements, this doesn't mean the same thing as "return the collation element [or ordering priority] of the previous character in the string".

This function updates the iterator's internal pointer to point to the collation element preceding the one it's currently pointing to and then returns that element, while next() returns the current element and then updates the pointer. This means that when you change direction while iterating (i.e., call next() and then call previous(), or call previous() and then call next()), you'll get back the same element twice.

**Since:** 1.2

### primaryOrder

public static final int **primaryOrder**(int order)

Return the primary component of a collation element.

**Parameters:**order - the collation element **Returns:**the element's primary component

### secondaryOrder

public static final short **secondaryOrder**(int order)

Return the secondary component of a collation element.

**Parameters:**order - the collation element **Returns:**the element's secondary component

### tertiaryOrder

public static final short **tertiaryOrder**(int order)

Return the tertiary component of a collation element.

**Parameters:**order - the collation element **Returns:**the element's tertiary component

### setOffset

public void **setOffset**(int newOffset)

Sets the iterator to point to the collation element corresponding to the specified character (the parameter is a CHARACTER offset in the original string, not an offset into its corresponding sequence of collation elements). The value returned by the next call to next() will be the collation element corresponding to the specified position in the text. If that position is in the middle of a contracting character sequence, the result of the next call to next() is the collation element for that sequence. This means that getOffset() is not guaranteed to return the same value as was passed to a preceding call to setOffset().

**Parameters:**newOffset - The new character offset into the original text.**Since:** 1.2

### getOffset

public int **getOffset**()

Returns the character offset in the original text corresponding to the next collation element. (That is, getOffset() returns the position in the text corresponding to the collation element that will be returned by the next call to next().) This value will always be the index of the FIRST character corresponding to the collation element (a contracting character sequence is when two or more characters all correspond to the same collation element). This means if you do setOffset(x) followed immediately by getOffset(), getOffset() won't necessarily return x.

**Returns:**The character offset in the original text corresponding to the collation element that will be returned by the next call to next().**Since:** 1.2

### getMaxExpansion

public int **getMaxExpansion**(int order)

Return the maximum length of any expansion sequences that end with the specified comparison order.

**Parameters:**order - a collation order returned by previous or next. **Returns:**the maximum length of any expansion sequences ending with the specified order.**Since:** 1.2

### setText

public void **setText**([String](http://docs.google.com/java/lang/String.html) source)

Set a new string over which to iterate.

**Parameters:**source - the new source text**Since:** 1.2

### setText

public void **setText**([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) source)

Set a new string over which to iterate.

**Parameters:**source - the new source text.**Since:** 1.2

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CollationElementIterator.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/text/ChoiceFormat.html)   [**NEXT CLASS**](http://docs.google.com/java/text/CollationKey.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/CollationElementIterator.html)    [**NO FRAMES**](http://docs.google.com/CollationElementIterator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#4d34og8) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).