| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Short.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/lang/SecurityManager.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/StackOverflowError.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Short.html)    [**NO FRAMES**](http://docs.google.com/Short.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

## **java.lang**

Class Short

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Short](http://docs.google.com/java/lang/Short.html)>

public final class **Short**extends [Number](http://docs.google.com/java/lang/Number.html)implements [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Short](http://docs.google.com/java/lang/Short.html)>

The Short class wraps a value of primitive type short in an object. An object of type Short contains a single field whose type is short.

In addition, this class provides several methods for converting a short to a String and a String to a short, as well as other constants and methods useful when dealing with a short.

**Since:** JDK1.1 **See Also:**[Number](http://docs.google.com/java/lang/Number.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.lang.Short)

| **Field Summary** | |
| --- | --- |
| static short | [**MAX\_VALUE**](http://docs.google.com/java/lang/Short.html#MAX_VALUE)            A constant holding the maximum value a short can have, 215-1. |
| static short | [**MIN\_VALUE**](http://docs.google.com/java/lang/Short.html#MIN_VALUE)            A constant holding the minimum value a short can have, -215. |
| static int | [**SIZE**](http://docs.google.com/java/lang/Short.html#SIZE)            The number of bits used to represent a short value in two's complement binary form. |
| static [Class](http://docs.google.com/java/lang/Class.html)<[Short](http://docs.google.com/java/lang/Short.html)> | [**TYPE**](http://docs.google.com/java/lang/Short.html#TYPE)            The Class instance representing the primitive type short. |

| **Constructor Summary** | |
| --- | --- |
| [**Short**](http://docs.google.com/java/lang/Short.html#Short(short))(short value)            Constructs a newly allocated Short object that represents the specified short value. |
| [**Short**](http://docs.google.com/java/lang/Short.html#Short(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Constructs a newly allocated Short object that represents the short value indicated by the String parameter. |

| **Method Summary** | |
| --- | --- |
| byte | [**byteValue**](http://docs.google.com/java/lang/Short.html#byteValue())()            Returns the value of this Short as a byte. |
| int | [**compareTo**](http://docs.google.com/java/lang/Short.html#compareTo(java.lang.Short))([Short](http://docs.google.com/java/lang/Short.html) anotherShort)            Compares two Short objects numerically. |
| static [Short](http://docs.google.com/java/lang/Short.html) | [**decode**](http://docs.google.com/java/lang/Short.html#decode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Decodes a String into a Short. |
| double | [**doubleValue**](http://docs.google.com/java/lang/Short.html#doubleValue())()            Returns the value of this Short as a double. |
| boolean | [**equals**](http://docs.google.com/java/lang/Short.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this object to the specified object. |
| float | [**floatValue**](http://docs.google.com/java/lang/Short.html#floatValue())()            Returns the value of this Short as a float. |
| int | [**hashCode**](http://docs.google.com/java/lang/Short.html#hashCode())()            Returns a hash code for this Short. |
| int | [**intValue**](http://docs.google.com/java/lang/Short.html#intValue())()            Returns the value of this Short as an int. |
| long | [**longValue**](http://docs.google.com/java/lang/Short.html#longValue())()            Returns the value of this Short as a long. |
| static short | [**parseShort**](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Parses the string argument as a signed decimal short. |
| static short | [**parseShort**](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) s, int radix)            Parses the string argument as a signed short in the radix specified by the second argument. |
| static short | [**reverseBytes**](http://docs.google.com/java/lang/Short.html#reverseBytes(short))(short i)            Returns the value obtained by reversing the order of the bytes in the two's complement representation of the specified short value. |
| short | [**shortValue**](http://docs.google.com/java/lang/Short.html#shortValue())()            Returns the value of this Short as a short. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/Short.html#toString())()            Returns a String object representing this Short's value. |
| static [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/Short.html#toString(short))(short s)            Returns a new String object representing the specified short. |
| static [Short](http://docs.google.com/java/lang/Short.html) | [**valueOf**](http://docs.google.com/java/lang/Short.html#valueOf(short))(short s)            Returns a Short instance representing the specified short value. |
| static [Short](http://docs.google.com/java/lang/Short.html) | [**valueOf**](http://docs.google.com/java/lang/Short.html#valueOf(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Returns a Short object holding the value given by the specified String. |
| static [Short](http://docs.google.com/java/lang/Short.html) | [**valueOf**](http://docs.google.com/java/lang/Short.html#valueOf(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) s, int radix)            Returns a Short object holding the value extracted from the specified String when parsed with the radix given by the second argument. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [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** |
| --- |

### MIN\_VALUE

public static final short **MIN\_VALUE**

A constant holding the minimum value a short can have, -215.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.lang.Short.MIN_VALUE)

### MAX\_VALUE

public static final short **MAX\_VALUE**

A constant holding the maximum value a short can have, 215-1.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.lang.Short.MAX_VALUE)

### TYPE

public static final [Class](http://docs.google.com/java/lang/Class.html)<[Short](http://docs.google.com/java/lang/Short.html)> **TYPE**

The Class instance representing the primitive type short.

### SIZE

public static final int **SIZE**

The number of bits used to represent a short value in two's complement binary form.

**Since:** 1.5 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.lang.Short.SIZE)

| **Constructor Detail** |
| --- |

### Short

public **Short**(short value)

Constructs a newly allocated Short object that represents the specified short value.

**Parameters:**value - the value to be represented by the Short.

### Short

public **Short**([String](http://docs.google.com/java/lang/String.html) s)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Constructs a newly allocated Short object that represents the short value indicated by the String parameter. The string is converted to a short value in exactly the manner used by the parseShort method for radix 10.

**Parameters:**s - the String to be converted to a Short **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - If the String does not contain a parsable short.**See Also:**[parseShort(java.lang.String, int)](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String,%20int))

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

### toString

public static [String](http://docs.google.com/java/lang/String.html) **toString**(short s)

Returns a new String object representing the specified short. The radix is assumed to be 10.

**Parameters:**s - the short to be converted **Returns:**the string representation of the specified short**See Also:**[Integer.toString(int)](http://docs.google.com/java/lang/Integer.html#toString(int))

### parseShort

public static short **parseShort**([String](http://docs.google.com/java/lang/String.html) s)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Parses the string argument as a signed decimal short. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value. The resulting short value is returned, exactly as if the argument and the radix 10 were given as arguments to the [parseShort(java.lang.String, int)](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String,%20int)) method.

**Parameters:**s - a String containing the short representation to be parsed **Returns:**the short value represented by the argument in decimal. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - If the string does not contain a parsable short.

### parseShort

public static short **parseShort**([String](http://docs.google.com/java/lang/String.html) s,  
 int radix)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Parses the string argument as a signed short in the radix specified by the second argument. The characters in the string must all be digits, of the specified radix (as determined by whether [Character.digit(char, int)](http://docs.google.com/java/lang/Character.html#digit(char,%20int)) returns a nonnegative value) except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value. The resulting byte value is returned.

An exception of type NumberFormatException is thrown if any of the following situations occurs:

* The first argument is null or is a string of length zero.
* The radix is either smaller than [Character.MIN\_RADIX](http://docs.google.com/java/lang/Character.html#MIN_RADIX) or larger than [Character.MAX\_RADIX](http://docs.google.com/java/lang/Character.html#MAX_RADIX).
* Any character of the string is not a digit of the specified radix, except that the first character may be a minus sign '-' ('\u002D') provided that the string is longer than length 1.
* The value represented by the string is not a value of type short.

**Parameters:**s - the String containing the short representation to be parsedradix - the radix to be used while parsing s **Returns:**the short represented by the string argument in the specified radix. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - If the String does not contain a parsable short.

### valueOf

public static [Short](http://docs.google.com/java/lang/Short.html) **valueOf**([String](http://docs.google.com/java/lang/String.html) s,  
 int radix)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Returns a Short object holding the value extracted from the specified String when parsed with the radix given by the second argument. The first argument is interpreted as representing a signed short in the radix specified by the second argument, exactly as if the argument were given to the [parseShort(java.lang.String, int)](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String,%20int)) method. The result is a Short object that represents the short value specified by the string.

In other words, this method returns a Short object equal to the value of:

new Short(Short.parseShort(s, radix))

**Parameters:**s - the string to be parsedradix - the radix to be used in interpreting s **Returns:**a Short object holding the value represented by the string argument in the specified radix. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - If the String does not contain a parsable short.

### valueOf

public static [Short](http://docs.google.com/java/lang/Short.html) **valueOf**([String](http://docs.google.com/java/lang/String.html) s)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Returns a Short object holding the value given by the specified String. The argument is interpreted as representing a signed decimal short, exactly as if the argument were given to the [parseShort(java.lang.String)](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String)) method. The result is a Short object that represents the short value specified by the string.

In other words, this method returns a Byte object equal to the value of:

new Short(Short.parseShort(s))

**Parameters:**s - the string to be parsed **Returns:**a Short object holding the value represented by the string argument **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - If the String does not contain a parsable short.

### valueOf

public static [Short](http://docs.google.com/java/lang/Short.html) **valueOf**(short s)

Returns a Short instance representing the specified short value. If a new Short instance is not required, this method should generally be used in preference to the constructor [Short(short)](http://docs.google.com/java/lang/Short.html#Short(short)), as this method is likely to yield significantly better space and time performance by caching frequently requested values.

**Parameters:**s - a short value. **Returns:**a Short instance representing s.**Since:** 1.5

### decode

public static [Short](http://docs.google.com/java/lang/Short.html) **decode**([String](http://docs.google.com/java/lang/String.html) nm)  
 throws [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html)

Decodes a String into a Short. Accepts decimal, hexadecimal, and octal numbers given by the following grammar:*DecodableString:* *Signopt DecimalNumeral* *Signopt* 0x *HexDigits* *Signopt* 0X *HexDigits* *Signopt* # *HexDigits* *Signopt* 0 *OctalDigits*

*Sign:* -*DecimalNumeral*, *HexDigits*, and *OctalDigits* are defined in [§3.10.1](http://java.sun.com/docs/books/jls/second_edition/html/lexical.doc.html#48282) of the [Java Language Specification](http://java.sun.com/docs/books/jls/html/).

The sequence of characters following an (optional) negative sign and/or radix specifier ("0x", "0X", "#", or leading zero) is parsed as by the Short.parseShort method with the indicated radix (10, 16, or 8). This sequence of characters must represent a positive value or a [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) will be thrown. The result is negated if first character of the specified String is the minus sign. No whitespace characters are permitted in the String.

**Parameters:**nm - the String to decode. **Returns:**a Short object holding the short value represented by nm **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - if the String does not contain a parsable short.**See Also:**[parseShort(java.lang.String, int)](http://docs.google.com/java/lang/Short.html#parseShort(java.lang.String,%20int))

### byteValue

public byte **byteValue**()

Returns the value of this Short as a byte.

**Overrides:**[byteValue](http://docs.google.com/java/lang/Number.html#byteValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type byte.

### shortValue

public short **shortValue**()

Returns the value of this Short as a short.

**Overrides:**[shortValue](http://docs.google.com/java/lang/Number.html#shortValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type short.

### intValue

public int **intValue**()

Returns the value of this Short as an int.

**Specified by:**[intValue](http://docs.google.com/java/lang/Number.html#intValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type int.

### longValue

public long **longValue**()

Returns the value of this Short as a long.

**Specified by:**[longValue](http://docs.google.com/java/lang/Number.html#longValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type long.

### floatValue

public float **floatValue**()

Returns the value of this Short as a float.

**Specified by:**[floatValue](http://docs.google.com/java/lang/Number.html#floatValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type float.

### doubleValue

public double **doubleValue**()

Returns the value of this Short as a double.

**Specified by:**[doubleValue](http://docs.google.com/java/lang/Number.html#doubleValue()) in class [Number](http://docs.google.com/java/lang/Number.html) **Returns:**the numeric value represented by this object after conversion to type double.

### toString

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

Returns a String object representing this Short's value. The value is converted to signed decimal representation and returned as a string, exactly as if the short value were given as an argument to the [toString(short)](http://docs.google.com/java/lang/Short.html#toString(short)) method.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the value of this object in base 10.

### hashCode

public int **hashCode**()

Returns a hash code for this Short.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Compares this object to the specified object. The result is true if and only if the argument is not null and is a Short object that contains the same short value as this object.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object to compare with **Returns:**true if the objects are the same; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### compareTo

public int **compareTo**([Short](http://docs.google.com/java/lang/Short.html) anotherShort)

Compares two Short objects numerically.

**Specified by:**[compareTo](http://docs.google.com/java/lang/Comparable.html#compareTo(T)) in interface [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Short](http://docs.google.com/java/lang/Short.html)> **Parameters:**anotherShort - the Short to be compared. **Returns:**the value 0 if this Short is equal to the argument Short; a value less than 0 if this Short is numerically less than the argument Short; and a value greater than 0 if this Short is numerically greater than the argument Short (signed comparison).**Since:** 1.2

### reverseBytes

public static short **reverseBytes**(short i)

Returns the value obtained by reversing the order of the bytes in the two's complement representation of the specified short value.

**Returns:**the value obtained by reversing (or, equivalently, swapping) the bytes in the specified short value.**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Short.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/lang/SecurityManager.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/StackOverflowError.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Short.html)    [**NO FRAMES**](http://docs.google.com/Short.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

[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).