

JOE reference manual

by Marco Bertacca

Package
com.veryant.joe

com.veryant.joe

Class Block

```

java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractList
              |-- java.util.ArrayList
                    |-- com.veryant.joe.Block
  
```

All Implemented Interfaces:

InternalObject, java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable, java.util.RandomAccess, java.util.List

public class **Block**

extends java.util.ArrayList

implements java.util.List, java.util.RandomAccess, java.lang.Cloneable, java.io.Serializable, java.util.List, java.util.Collection, InternalObject

This class implements a JOE block.

Method Summary

Block	\$extends (Block b)
Block	\$new () Returns a JOE Object obtained by cloning this block.
Block	\$new (java.lang.Object[] args) same as \$new() with parameters
Block	add ()
Block	add (java.lang.Object[] args)
java.lang.Object	clone () Returns a clone of this object.
java.lang.Object	doWhileFalse (Block b) Executes this block and then executes <i>b</i> : if the result of <i>b</i> execution is false then executes itself again.
java.lang.Object	doWhileTrue (Block b) Executes this block and then executes <i>b</i> : if the result of <i>b</i> execution is true then executes itself again.
java.lang.Object	exec () Executes the chains of methods of this block.
java.lang.Object	exec (java.lang.Object[] argv) Executes the chains of methods of this block with the supplied arguments.

java.lang.Object	<u>exec</u> (java.lang.Object[] vars, java.lang.Object[] argv)
java.lang.Object[]	<u>getArgv</u> () Returns an array containing all the variables values actually passed to this block as arguments.
int	<u>getCol</u> ()
java.lang.Object	<u>getJoeClass</u> ()
int	<u>getRow</u> ()
Variable	<u>getSetLocalVariable</u> (java.lang.String name)
Variable	<u>getSetVariable</u> (java.lang.String name)
java.lang.Object	<u>getVariable</u> (Variable var)
java.lang.Object	<u>getVariable</u> (<u>WString</u> name) Returns the value of the a variable whose name is <i>name</i> .
java.lang.String[]	<u>getVariablesNames</u> () Returns an array containing all the variables names that can be accessed by this block.
java.lang.Object	<u>init</u> ()
java.lang.Object	<u>init</u> (java.lang.Object[] argv)
Variable	<u>lookForVariable</u> (java.lang.String name)
Variable	<u>lookForVariable</u> (java.lang.String name, int[] depth)
java.lang.Object	<u>multiply</u> ()
java.lang.Object	<u>multiply</u> (java.lang.Object[] argv)
java.lang.String	<u>name</u> () Returns the name of the block
<u>Block</u>	<u>setParent</u> (<u>Block</u> b)
java.lang.Object	<u>setVariable</u> (Variable var, java.lang.Object val)
java.lang.Object	<u>setVariable</u> (<u>WString</u> name, java.lang.Object val) Assigns or create a variable whose name is <i>name</i> with the value <i>val</i> .
java.lang.String	<u>toString</u> ()
java.lang.Object	<u>whileFalse</u> (<u>Block</u> b) Executes this block and if its result is <i>false</i> then executes <i>b</i> .

java.lang.Object

[whileTrue](#)([Block](#) b)Executes this block and if its result is true then executes *b*.**Methods inherited from class java.util.ArrayList**

add, add, addAll, addAll, clear, clone, contains, ensureCapacity, get, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, remove, remove, removeAll, retainAll, set, size, subList, toArray, toArray, trimToSize

Methods inherited from class java.util.AbstractList

add, add, addAll, clear, equals, get, hashCode, indexOf, iterator, lastIndexOf, listIterator, listIterator, remove, set, subList

Methods inherited from class java.util.AbstractCollection

add, addAll, clear, contains, containsAll, isEmpty, iterator, remove, removeAll, retainAll, size, toArray, toArray, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.util.Collection

add, addAll, clear, contains, containsAll, equals, hashCode, isEmpty, iterator, remove, removeAll, retainAll, size, toArray, toArray

Methods inherited from interface java.lang.Iterable

iterator

Methods inherited from interface java.util.List

add, add, addAll, addAll, clear, contains, containsAll, equals, get, hashCode, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, remove, remove, removeAll, retainAll, set, size, subList, toArray, toArray

Methods inherited from interface java.util.Collection

add, addAll, clear, contains, containsAll, equals, hashCode, isEmpty, iterator, remove, removeAll, retainAll, size, toArray, toArray

Methods inherited from interface java.lang.Iterable

iterator

Methods inherited from interface java.util.List

add, add, addAll, addAll, clear, contains, containsAll, equals, get, hashCode, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, remove, remove, removeAll, retainAll, set, size, subList, toArray, toArray

Methods inherited from interface java.util.Collection

add, addAll, clear, contains, containsAll, equals, hashCode, isEmpty, iterator, remove, removeAll, retainAll, size, toArray, toArray

Methods inherited from interface `java.lang.Iterable``iterator`

Methods

exec

```
public final java.lang.Object exec()
    throws JOEException
```

Executes the chains of methods of this block.

multiply

```
public final java.lang.Object multiply()
    throws JOEException
```

See Also:

[exec\(\)](#)

exec

```
public final java.lang.Object exec(java.lang.Object[] argv)
    throws JOEException
```

Executes the chains of methods of this block with the supplied arguments.

multiply

```
public final java.lang.Object multiply(java.lang.Object[] argv)
    throws JOEException
```

See Also:

[\(Object...argv\).](#)

init

```
public final java.lang.Object init()
    throws JOEException
```

init

```
public final java.lang.Object init(java.lang.Object[] argv)
    throws JOEException
```

(continued on next page)

(continued from last page)

exec

```
public final java.lang.Object exec(java.lang.Object[] vars,  
    java.lang.Object[] argv)  
    throws JOEException
```

whileTrue

```
public final java.lang.Object whileTrue(Block b)  
    throws JOEException
```

Executes this block and if its result is `true` then executes *b*. Repeats until this block returns an object not equal to `true`. Returns the result of *b* last execution.

whileFalse

```
public final java.lang.Object whileFalse(Block b)  
    throws JOEException
```

Executes this block and if its result is `false` then executes *b*. Repeats until this block returns an object not equal to `false`. Returns the result of *b* last execution.

doWhileTrue

```
public final java.lang.Object doWhileTrue(Block b)  
    throws JOEException
```

Executes this block and then executes *b*; if the result of *b* execution is `true` then executes itself again. Repeats until *b* returns an object not equal to `true`

doWhileFalse

```
public final java.lang.Object doWhileFalse(Block b)  
    throws JOEException
```

Executes this block and then executes *b*; if the result of *b* execution is `false` then executes itself again. Repeats until *b* returns an object not equal to `false`

setVariable

```
public java.lang.Object setVariable(WString name,  
    java.lang.Object val)  
    throws JOEException
```

Assigns or create a variable whose name is *name* with the value *val*.

setVariable

```
public java.lang.Object setVariable(Variable var,  
    java.lang.Object val)
```

getVariable

```
public java.lang.Object getVariable(WString name)
```

Returns the value of the a variable whose name is *name*.

getVariable

```
public java.lang.Object getVariable(Variable var)
```

getVariablesNames

```
public java.lang.String[] getVariablesNames()
```

Returns an array containing all the variables names that can be accessed by this block.

getArgv

```
public java.lang.Object[] getArgv()
```

Returns an array containing all the variables values actually passed to this block as arguments.

clone

```
public java.lang.Object clone()
```

Returns a clone of this object.

\$new

```
public Block $new()  
    throws JOEException
```

Returns a JOE Object obtained by cloning this block. The block is executed and every variable assigned to a block will be treated as a method of the JOE Object object. This method can be invoked on a JOE Object too.

add

```
public Block add()  
    throws JOEException
```

See Also:

[\\$new\(\)](#)

\$new

```
public Block $new(java.lang.Object[] args)  
    throws JOEException
```

same as \$new() with parameters

See Also:

[\\$new\(\)](#)

add

```
public Block add(java.lang.Object[] args)  
    throws JOEException
```

(continued on next page)

(continued from last page)

See Also:[\(Object...args\)](#)

name

```
public java.lang.String name()
```

Returns the name of the block

getRow

```
public int getRow()
```

getCol

```
public int getCol()
```

setParent

```
public Block setParent(Block b)
```

\$extends

```
public Block $extends(Block b)
```

getJoeClass

```
public java.lang.Object getJoeClass()
```

toString

```
public java.lang.String toString()
```

lookForVariable

```
public Variable lookForVariable(java.lang.String name,  
                                int[] depth)
```

lookForVariable

```
public Variable lookForVariable(java.lang.String name)
```

getSetLocalVariable

```
public Variable getSetLocalVariable(java.lang.String name)
```

getSetVariable

```
public Variable getSetVariable(java.lang.String name)
```

com.veryant.joe Class CommandBase

java.lang.Object

└-com.veryant.joe.CommandBase

Direct Known Subclasses:

[DefaultCommand](#)

```
public class CommandBase
extends java.lang.Object
```

This class contains some methods that are useful for any flavor of JOE implementation.

Constructor Summary

public	CommandBase()
--------	-------------------------------

Method Summary

void	\$extends (Block child, Block parent) Makes the child inheriting from parent.
boolean	\$instanceof (java.lang.Object obj, java.lang.Class clazz) Similar to the java operator instanceof.
boolean	\$instanceof (java.lang.Object obj, java.lang.String clazz) Convenience method for instanceof (obj, getClass(clazz)).
java.lang.Object	\$new (java.lang.String fname) Executes the specified JOE script and returns it as an object.
java.lang.Object	\$new (java.lang.String fname, java.lang.Object[] argv) Executes the specified JOE script and returns it as an object.
void	\$throw (java.lang.String msg) Throws an Exception whose message is the argument.
void	\$throw (java.lang.Throwable ex) Throws a throwable passed as argument.
java.lang.Object	addPath (java.lang.String fname) Adds one path from which to load new object.
boolean	areSameObject (java.lang.Object o1, java.lang.Object o2) Returns true if the passed objects are the same object, false otherwise.
java.lang.Object[]	array (java.lang.Object[] items) Returns an Object array whose items are the parameters
int	asc (java.lang.String a) Returns a char representing the codepoint corresponding to the first letter in a string.

java.lang.String	<u>chr</u> (java.lang.Integer n) Returns a string one character long containing the character whose codepoint is specified as argument.
java.lang.String	<u>debug</u> () Turns on the debugger.
java.lang.Class	<u>getClass</u> (java.lang.String clazz) Returns the Class instance of clazz.
ClassReference	<u>getClassRef</u> (java.lang.String clazz) Returns the ClassReference instance of clazz.
java.lang.String	<u>getCwd</u> () Returns the current directory.
java.lang.Object	<u>getField</u> (java.lang.Object obj, java.lang.String name) Returns a field whose name is name of the object obj.
java.lang.String[]	<u>getPath</u> () Returns the path set.
java.lang.Object	<u>getStaticField</u> (java.lang.String clazz, java.lang.String name) Returns a static field whose name is name of the class clazz.
java.net.URL	<u>getURL</u> (java.lang.String spec) Returns an URL based on the base URL of the first script run.
boolean	<u>isNull</u> (java.lang.Object obj) Returns true if the passed object is null, false otherwise.
boolean	<u>isThrowable</u> (java.lang.Object obj) Returns the result of obj instanceof Throwable
int	<u>joe</u> (java.lang.Object[] argv) Executes the specified JOE script and returns its return code.
java.lang.Object	<u>newArray</u> (java.lang.Class clazz, int n) Returns an Object array of the specified class whose length is n.
java.lang.Object[]	<u>newArray</u> (int n) Returns an Object array whose length is n.
java.lang.Object	<u>newArray</u> (java.lang.String clazz, int n) Returns an Object array of the specified class whose length is n.
java.lang.Object	<u>newInstance</u> (java.lang.Class clazz) Returns a new instance of the specified class.
java.lang.Object	<u>newInstance</u> (java.lang.Class clazz, java.lang.Object[] args) Returns a new instance of the specified class passing the arguments args to its constructor.
java.lang.Object	<u>newInstance</u> (java.lang.String className) This is a convenience method for newInstance (getClass (className)).
java.lang.Object	<u>newInstance</u> (java.lang.String clsName, java.lang.Object[] args) This is a convenience method for newInstance (getClass (clsName), args).

java.lang.Object	<code>newInterface</code> (java.lang.String intf, <code>Block</code> blk) Returns the specified interface whose method must be implemented in the supplied block in a way similar to java.lang.reflect.Proxy.
java.lang.String	<code>nl</code> () Returns a string containing the line terminator.
void	<code>systemExit</code> () Invoke System.exit (0).
void	<code>systemExit</code> (int ec) Invoke System.exit (ec).
java.lang.String	<code>toString</code> ()
java.lang.String	<code>typename</code> (java.lang.Object obj) Returns the JOE name of the object.
java.lang.String	<code>version</code> () Returns the version.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

CommandBase

```
public CommandBase()
```

Methods

newInstance

```
public java.lang.Object newInstance(java.lang.Class clazz)
    throws java.lang.Exception
```

Returns a new instance of the specified class.

newInstance

```
public java.lang.Object newInstance(java.lang.String className)
    throws java.lang.Exception
```

This is a convenience method for `newInstance (getClass (className))`.

newInstance

```
public java.lang.Object newInstance(java.lang.Class clazz,
    java.lang.Object[] args)
    throws java.lang.Exception
```

Returns a new instance of the specified class passing the arguments args to its constructor.

newInstance

```
public java.lang.Object newInstance(java.lang.String clsName,  
    java.lang.Object[] args)  
    throws java.lang.Exception
```

This is a convenience method for `newInstance (getClass (clsName), args)`.

newArray

```
public java.lang.Object[] newArray(int n)  
    throws java.lang.Exception
```

Returns an Object array whose length is n.

newArray

```
public java.lang.Object newArray(java.lang.Class clazz,  
    int n)  
    throws java.lang.Exception
```

Returns an Object array of the specified class whose length is n.

newArray

```
public java.lang.Object newArray(java.lang.String clazz,  
    int n)  
    throws java.lang.Exception
```

Returns an Object array of the specified class whose length is n.

array

```
public java.lang.Object[] array(java.lang.Object[] items)  
    throws java.lang.Exception
```

Returns an Object array whose items are the parameters

getField

```
public java.lang.Object getField(java.lang.Object obj,  
    java.lang.String name)  
    throws java.lang.Exception
```

Returns a field whose name is name of the object obj.

getStaticField

```
public java.lang.Object getStaticField(java.lang.String clazz,  
    java.lang.String name)  
    throws java.lang.Exception
```

Returns a static field whose name is name of the class clazz.

getClass

```
public java.lang.Class getClass(java.lang.String clazz)  
    throws java.lang.Exception
```

(continued from last page)

Returns the Class instance of clazz.

getClassRef

```
public ClassReference getClassRef(java.lang.String clazz)
    throws java.lang.Exception
```

Returns the ClassReference instance of clazz. The ClassReference object can be used to invoke a static method of a class, e.g. the ClassReference of java.lang.System can be used to invoke System.getProperty(String prop).

newInterface

```
public java.lang.Object newInterface(java.lang.String intf,
    Block blk)
    throws java.lang.Exception
```

Returns the specified interface whose method must be implemented in the supplied block in a way similar to java.lang.reflect.Proxy.

systemExit

```
public void systemExit(int ec)
```

Invoke System.exit (ec).

systemExit

```
public void systemExit()
```

Invoke System.exit (0).

\$throw

```
public void $throw(java.lang.String msg)
    throws java.lang.Exception
```

Throws an Exception whose message is the argument.

\$throw

```
public void $throw(java.lang.Throwable ex)
    throws java.lang.Throwable
```

Throws a throwable passed as argument.

isNull

```
public boolean isNull(java.lang.Object obj)
```

Returns true if the passed object is null, false otherwise.

areSameObject

```
public boolean areSameObject(java.lang.Object o1,
    java.lang.Object o2)
```

Returns true if the passed objects are the same object, false otherwise.

(continued from last page)

isThrowable

```
public boolean isThrowable(java.lang.Object obj)
```

Returns the result of obj instanceof Throwable

\$instanceof

```
public boolean $instanceof(java.lang.Object obj,  
    java.lang.Class clazz)
```

Similar to the java operator instanceof.

typename

```
public java.lang.String typename(java.lang.Object obj)
```

Returns the JOE name of the object.

addPath

```
public java.lang.Object addPath(java.lang.String fname)  
    throws java.lang.Exception
```

Adds one path from which to load new object.

getPath

```
public java.lang.String[] getPath()
```

Returns the path set.

\$new

```
public java.lang.Object $new(java.lang.String fname)  
    throws java.lang.Exception
```

Executes the specified JOE script and returns it as an object.

\$new

```
public java.lang.Object $new(java.lang.String fname,  
    java.lang.Object[] argv)  
    throws java.lang.Exception
```

Executes the specified JOE script and returns it as an object.

joe

```
public int joe(java.lang.Object[] argv)  
    throws java.lang.Exception
```

Executes the specified JOE script and returns its return code.

\$instanceof

```
public boolean $instanceof(java.lang.Object obj,  
    java.lang.String clazz)
```

(continued from last page)

Convenience method for instanceof (obj, getClass(clazz)).

nl

```
public java.lang.String nl()
```

Returns a string containing the line terminator.

chr

```
public java.lang.String chr(java.lang.Integer n)
```

Returns a string one character long containing the character whose codepoint is specified as argument.

asc

```
public int asc(java.lang.String a)
```

Returns a char representing the codepoint corresponding to the first letter in a string.

\$extends

```
public void $extends(Block child,  
                    Block parent)
```

Makes the child inheriting from parent.

getURL

```
public java.net.URL getURL(java.lang.String spec)  
    throws java.net.MalformedURLException
```

Returns an URL based on the base URL of the first script run.

version

```
public java.lang.String version()
```

Returns the version.

getcwd

```
public java.lang.String getcwd()
```

Returns the current directory.

debug

```
public java.lang.String debug()  
    throws JOEException
```

Turns on the debugger.

toString

```
public java.lang.String toString()
```

com.veryant.joe Class DefaultCommand

```

java.lang.Object
|
+-com.veryant.joe.CommandBase
|
+-com.veryant.joe.DefaultCommand

```

public class **DefaultCommand**
extends [CommandBase](#)

This class contains methods of ! standard.

Nested Class Summary

class	DefaultCommand.Switch DefaultCommand.Switch
-------	--

Constructor Summary

public	DefaultCommand()
--------	----------------------------------

Method Summary

java.lang.Object	\$break() Causes the exit from the current block.
java.lang.Object	\$break(java.lang.String blkName) Causes the exit from the block whose name is blkName.
java.lang.Object	\$for(int start, int end, Block code) This is a convenience method for for (start, end, 1).
java.lang.Object	\$for(int start, int end, int step, Block code) This method implements a 'for' cycle.
java.lang.Object	\$if(Block cond, Block ifTrue) if implementation without 'else'.
java.lang.Object	\$if(Block cond, Block ifTrue, Block ifFalse) if implementation with 'else'.
java.lang.Object	\$if(java.lang.Boolean cond, Block ifTrue) If cond is true then the block ifTrue is executed and its return value returned, otherwise cond is returned.
java.lang.Object	\$if(boolean cond, Block ifTrue, Block ifFalse) If cond is true then the block ifTrue is executed and its return value returned, otherwise ifFalse is executed and its return value returned.
java.lang.Object	\$switch(java.lang.Object cftr) Returns an instance of class Switch created using cftr as parameter.

java.lang.Object	<u>\$try</u> (<u>Block</u> blk) Executes the block passed as argument: if the execution throws an exception then the exception is returned instead of the expected return object.
java.lang.Object	<u>\$try</u> (<u>Block</u> blk, <u>Block</u> excpt) Executes the block passed as 1st argument: if the execution throws an exception then the block passed as 2nd argument is executed with the exception as argument.
java.lang.Object	<u>\$while</u> (<u>Block</u> cond, <u>Block</u> code) This method implements a loop.
java.lang.Object	<u>arraySort</u> (java.lang.Object[] array, <u>Block</u> blk) Sort an array according to the block.
java.lang.Object	<u>binarySearch</u> (java.lang.Object[] array, java.lang.Object key, <u>Block</u> blk) Binary search on an array according to the block.
void	<u>breakLoop</u> () Causes the exit from a loop.
java.lang.Object	<u>doWhile</u> (<u>Block</u> code, <u>Block</u> cond) This method implements a loop.
java.lang.Object	<u>eprint</u> (java.lang.Object[] b) Prints the specified objects on stderr.
java.lang.Object	<u>eprintln</u> () Prints a newline on stderr.
java.lang.Object	<u>eprintln</u> (java.lang.Object[] b) Prints the specified objects followed by a newline on stderr.
int	<u>exec</u> (java.lang.Object[] cmds) Executes the specified command and returns its return code.
int	<u>exec</u> (java.lang.ProcessBuilder pb) Executes the specified ProcessBuilder
int	<u>execFromDir</u> (java.io.File dir, java.lang.Object[] cmds) Executes the specified command from the specified directory and returns its return code.
int	<u>execFromDir</u> (java.lang.String dir, java.lang.Object[] cmds) Executes the specified command from the specified directory and returns its return code.
java.lang.String	<u>execGetOut</u> (java.lang.Object[] cmds) Executes the specified command and returns its standard output as a string.
int	<u>execJoe</u> (java.lang.Object[] cmds) Stops the execution of the current script and executes the script specified as argument.
java.lang.Object	<u>foreach</u> (java.util.Collection list, <u>Block</u> code) This is a convenience method for foreach (list, 0, code).
java.lang.Object	<u>foreach</u> (java.util.Collection list, int n, <u>Block</u> code) This method implements a 'for each'.
java.lang.Object	<u>foreach</u> (java.util Enumeration it, <u>Block</u> code) This is a convenience method for foreach (list, 0, code).

java.lang.Object	foreach (java.util Enumeration it, int n, Block code) This method implements a 'for each'.
java.lang.Object	foreach (java.lang.Iterable it, Block code) This is a convenience method for foreach (list, 0, code).
java.lang.Object	foreach (java.lang.Iterable it, int n, Block code) This method implements a 'for each'.
java.lang.Object	foreach (java.util.Iterator it, Block code) This is a convenience method for foreach (list, 0, code).
java.lang.Object	foreach (java.util.Iterator it, int n, Block code) This method implements a 'for each'.
java.lang.Object	foreach (java.lang.Object[] list, Block code) This is a convenience method for foreach (list, 0, code).
java.lang.Object	foreach (java.lang.Object[] list, int n, Block code) This method implements a 'for each'.
Glob	getGlob (java.lang.String glob, boolean caseInsensitive) Returns a Glob object that performs match operations on paths thru the method 'matches'.
boolean	isConsole ()
Pipe	pipe () Returns an object of type Pipe that allows the execution of programs in a pipeline.
java.lang.Object	print (java.lang.Object[] b) Prints the specified objects.
java.lang.Object	println () Prints a newline.
java.lang.Object	println (java.lang.Object[] b) Prints the specified objects followed by a newline.
java.lang.Object	random () Returns a random number in the range 0.0 - 1.0.
java.lang.String	readLine () Read a line from the console.
java.lang.Object	runAsBlock (Block blk, java.lang.String name, java.lang.Object[] cmds) Executes the specified JOE script as it were an inner block of the given argument.
int	runJoe (java.lang.Object[] cmds) Executes the specified JOE script and returns its return code.
java.lang.String	showInputDialog (java.lang.Object[] message) Shows the specified objects in a graphical window and reads an input that returns.
void	showMessageDialog (java.lang.Object[] message) edit Shows the specified objects in a graphical window.
void	sleep (int millis) Suspends execution for the specified number of milliseconds

java.lang.Object	switchTrue() Returns an instance of class Switch created using Boolean.TRUE as parameter.
int	system() (java.lang.String[] cmd)
java.lang.String	systemGetenv() (java.lang.String name) Reads the value of the specified environment variable.

Methods inherited from class [com.veryant.joe.CommandBase](#)

[\\$extends](#), [\\$instanceof](#), [\\$instanceof](#), [\\$new](#), [\\$new](#), [\\$throw](#), [\\$throw](#), [addPath](#), [areSameObject](#), [array](#), [asc](#), [chr](#), [debug](#), [getClass](#), [getClassRef](#), [getcwd](#), [getField](#), [getPath](#), [getStaticField](#), [getURL](#), [isNull](#), [isThrowable](#), [joe](#), [newArray](#), [newArray](#), [newArray](#), [newInstance](#), [newInstance](#), [newInstance](#), [newInstance](#), [newInterface](#), [nl](#), [systemExit](#), [systemExit](#), [toString](#), [typename](#), [version](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructors

DefaultCommand

```
public DefaultCommand()
```

Methods

isConsole

```
public boolean isConsole()
```

readLine

```
public java.lang.String readLine()
    throws java.io.IOException
```

Read a line from the console.

println

```
public java.lang.Object println()
```

Prints a newline.

eprintln

```
public java.lang.Object eprintln()
```

Prints a newline on stderr.

print

```
public java.lang.Object print(java.lang.Object[] b)
```

Prints the specified objects.

eprint

```
public java.lang.Object eprint(java.lang.Object[] b)
```

Prints the specified objects on stderr.

println

```
public java.lang.Object println(java.lang.Object[] b)
```

Prints the specified objects followed by a newline.

eprintln

```
public java.lang.Object eprintln(java.lang.Object[] b)
```

Prints the specified objects followed by a newline on stderr.

\$if

```
public java.lang.Object $if(java.lang.Boolean cond,  
    Block ifTrue)  
    throws java.lang.Exception
```

If cond is true then the block ifTrue is executed and its return value returned, otherwise cond is returned.

\$if

```
public java.lang.Object $if(boolean cond,  
    Block ifTrue,  
    Block ifFalse)  
    throws java.lang.Exception
```

If cond is true then the block ifTrue is executed and its return value returned, otherwise ifFalse is executed and its return value returned.

\$if

```
public java.lang.Object $if(Block cond,  
    Block ifTrue)  
    throws java.lang.Exception
```

if implementation without 'else'.

\$if

```
public java.lang.Object $if(Block cond,  
    Block ifTrue,  
    Block ifFalse)  
    throws java.lang.Exception
```

if implementation with 'else'.

\$while

```
public java.lang.Object $while(Block cond,  
    Block code)  
    throws JOEException
```

This method implements a loop. The code block is executed while the cond block is executed and it returns a boolean value true or until the breakLoop () invocation is encountered.

doWhile

```
public java.lang.Object doWhile(Block code,  
    Block cond)  
    throws JOEException
```

This method implements a loop. This loop will execute the code block once, before checking if the cond block returns the boolean value true, then it will repeat the loop as long as the condition is true or until the breakLoop () invocation is encountered..

\$for

```
public java.lang.Object $for(int start,  
    int end,  
    int step,  
    Block code)  
    throws JOEException
```

This method implements a 'for' cycle. The code block is executed for passing the index as argument.

\$for

```
public java.lang.Object $for(int start,  
    int end,  
    Block code)  
    throws JOEException
```

This is a convenience method for for (start, end, 1).

foreach

```
public java.lang.Object foreach(java.lang.Object[] list,  
    int n,  
    Block code)  
    throws JOEException
```

This method implements a 'for each'. The code block is executed for each object passed in the array startint from the n-th, each object is passed to the block as argument.

foreach

```
public java.lang.Object foreach(java.lang.Object[] list,  
    Block code)  
    throws JOEException
```

This is a convenience method for foreach (list, 0, code).

(continued from last page)

foreach

```
public java.lang.Object foreach(java.util.Collection list,  
    int n,  
    Block code)  
    throws JOEException
```

This method implements a 'for each'. The code block is executed for each object passed in the Collection startint from the n -th, each object is passed to the block as argument.

foreach

```
public java.lang.Object foreach(java.util.Collection list,  
    Block code)  
    throws JOEException
```

This is a convenience method for foreach (list, 0, code).

foreach

```
public java.lang.Object foreach(java.util.Iterator it,  
    int n,  
    Block code)  
    throws JOEException
```

This method implements a 'for each'. The code block is executed for each object passed in the Iterator starting from n, each object is passed to the block as argument.

foreach

```
public java.lang.Object foreach(java.util.Iterator it,  
    Block code)  
    throws JOEException
```

This is a convenience method for foreach (list, 0, code).

foreach

```
public java.lang.Object foreach(java.util.Enumeration it,  
    int n,  
    Block code)  
    throws JOEException
```

This method implements a 'for each'. The code block is executed for each object passed in the Enumeration starting from n, each object is passed to the block as argument.

foreach

```
public java.lang.Object foreach(java.util.Enumeration it,  
    Block code)  
    throws JOEException
```

This is a convenience method for foreach (list, 0, code).

foreach

```
public java.lang.Object foreach(java.lang.Iterable it,  
    int n,  
    Block code)  
    throws JOEException
```


(continued from last page)

This method implements a 'for each'. The code block is executed for each object passed in the Iterable.iterator starting from n, each object is passed to the block as argument.

foreach

```
public java.lang.Object foreach(java.lang.Iterable it,  
    Block code)  
    throws JOEException
```

This is a convenience method for foreach (list, 0, code).

breakLoop

```
public void breakLoop()  
    throws BreakLoopException
```

Causes the exit from a loop.

\$switch

```
public java.lang.Object $switch(java.lang.Object cfrt)
```

Returns an instance of class Switch created using cfrt as parameter.

switchTrue

```
public java.lang.Object switchTrue()
```

Returns an instance of class Switch created using Boolean.TRUE as parameter.

\$break

```
public java.lang.Object $break()  
    throws java.lang.Exception
```

Causes the exit from the current block.

\$break

```
public java.lang.Object $break(java.lang.String blkName)  
    throws java.lang.Exception
```

Causes the exit from the block whose name is blkName.

\$try

```
public java.lang.Object $try(Block blk)  
    throws java.lang.Exception
```

Executes the block passed as argument: if the execution throws an exception then the exception is returned instead of the expected return object.

\$try

```
public java.lang.Object $try(Block blk,  
    Block excpt)  
    throws java.lang.Exception
```

Executes the block passed as 1st argument: if the execution throws an exception then the block passed as 2nd argument is executed with the exception as argument.

exec

```
public int exec(java.lang.ProcessBuilder pb)
    throws java.lang.Exception
```

Executes the specified ProcessBuilder

exec

```
public int exec(java.lang.Object[] cmds)
    throws java.lang.Exception
```

Executes the specified command and returns its return code.

execFromDir

```
public int execFromDir(java.io.File dir,
    java.lang.Object[] cmds)
    throws java.lang.Exception
```

Executes the specified command from the specified directory and returns its return code.

execFromDir

```
public int execFromDir(java.lang.String dir,
    java.lang.Object[] cmds)
    throws java.lang.Exception
```

Executes the specified command from the specified directory and returns its return code.

system

```
public int system(java.lang.String[] cmd)
    throws java.lang.Exception
```

execGetOut

```
public java.lang.String execGetOut(java.lang.Object[] cmds)
    throws java.lang.Exception
```

Executes the specified command and returns its standard output as a string.

pipe

```
public Pipe pipe()
```

Returns an object of type Pipe that allows the execution of programs in a pipeline.

runJoe

```
public int runJoe(java.lang.Object[] cmds)
    throws java.lang.Exception
```

Executes the specified JOE script and returns its return code.

(continued from last page)

runAsBlock

```
public java.lang.Object runAsBlock(Block blk,  
    java.lang.String name,  
    java.lang.Object[] cmds)  
    throws java.lang.Exception
```

Executes the specified JOE script as it were an inner block of the given argument.

execJoe

```
public int execJoe(java.lang.Object[] cmds)  
    throws ExecException
```

Stops the execution of the current script and executes the script specified as argument.

systemGetenv

```
public java.lang.String systemGetenv(java.lang.String name)
```

Reads the value of the specified environment variable.

arraySort

```
public java.lang.Object arraySort(java.lang.Object[] array,  
    Block blk)
```

Sort an array according to the block.

binarySearch

```
public java.lang.Object binarySearch(java.lang.Object[] array,  
    java.lang.Object key,  
    Block blk)
```

Binary search on an array according to the block.

random

```
public java.lang.Object random()
```

Returns a random number in the range 0.0 - 1.0.

sleep

```
public void sleep(int millis)  
    throws java.lang.Exception
```

Suspends execution for the specified number of milliseconds

getGlob

```
public Glob getGlob(java.lang.String glob,  
    boolean caseInsensitive)
```

Returns a Glob object that performs match operations on paths thru the method 'matches'.

(continued from last page)

showMessageDialog

```
public void showMessageDialog( java.lang.Object[] message)
```

edit Shows the specified objects in a graphical window.

showInputDialog

```
public java.lang.String showInputDialog( java.lang.Object[] message)
```

Shows the specified objects in a graphical window and reads an input that returns.

com.veryant.joe Class DefaultCommand.Switch

java.lang.Object

└─com.veryant.joe.DefaultCommand.Switch

public static class **DefaultCommand.Switch**
extends java.lang.Object

This class is used in order to implement a behaviour similar to the Java switch.

Method Summary

java.lang.Object	<code>\$case</code> (<code>Block</code> blk2)
java.lang.Object	<code>\$case</code> (<code>Block</code> blk2, <code>Block</code> block) This method is similar to <code>\$case</code> (Object cfrt2, Block block) The <i>block</i> is supposed to returns a boolean and it is executed only if no block in the switch has been executed yet.
java.lang.Object	<code>\$case</code> (java.lang.Object cfrt2) This method compares cfrt2 with the object specified in the constructor and if they are equal then the following <code>\$case</code> (cfrt2, block), if any, will be executed independently from its argument.
java.lang.Object	<code>\$case</code> (java.lang.Object cfrt2, <code>Block</code> block) This method executes block under the following conditions: - any previous invocation of this method on this object hasn't been executed yet; - the argument of any previous <code>\$case</code> (cfrt2) invocation, or the argument of this invocation is equal to the object specified in the constructor.
java.lang.Object	<code>\$default</code> (<code>Block</code> block) This method executes block if any previous invocation of <code>\$case</code> (cfrt2, block) method on this object hasn't be executed yet.
java.lang.Object	<code>endSwitch</code> () This method returns the return code of the block executed or null if no block has been executed by this object.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods

\$case

public java.lang.Object **\$case**(java.lang.Object cfrt2)

This method compares cfrt2 with the object specified in the constructor and if they are equal then the following `$case` (cfrt2, block), if any, will be executed independently from its argument. It returns this.

(continued from last page)

\$case

```
public java.lang.Object $case(Block blk2)
    throws JOEException
```

\$case

```
public java.lang.Object $case(java.lang.Object cfrt2,
    Block block)
    throws JOEException
```

This method executes block under the following conditions: - any previous invocation of this method on this object hasn't been executed yet; - the argument of any previous \$case(cfrt2) invocation, or the argument of this invocation is equal to the object specified in the constructor.

\$case

```
public java.lang.Object $case(Block blk2,
    Block block)
    throws JOEException
```

This method is similar to \$case (Object cfrt2, Block block) The *block* is supposed to returns a boolean and it is executed only if no block in the switch has been executed yet.

\$default

```
public java.lang.Object $default(Block block)
    throws java.lang.Exception
```

This method executes block if any previous invocation of \$case (cfrt2, block) method on this object hasn't be executed yet.

endSwitch

```
public java.lang.Object endSwitch()
```

This method returns the return code of the block executed or null if no block has been executed by this object.

com.veryant.joe

Class WArray

```

java.lang.Object
|
+-com.veryant.joe Wrapper
|
+-com.veryant.joe.WArray

```

All Implemented Interfaces:
InternalObject

public class **WArray**
extends [Wrapper](#)

This class implements a JOE array and it is a wrapper of a Java array. In order to create an array from JOE you can use the following instructions:

```

! array obj0 [, obj1 ... , objn ]
which creates an array containing all the objects specified or:
! newArray aInteger
which creates an array of aInteger elements all containing null.

```

Constructor Summary

public	WArray (java.lang.Object[] array)
public	WArray (WNumber len)
public	WArray (int len)

Method Summary

java.lang.Object	add (java.lang.Object obj) Returns a new array longer by one of this array with <i>obj</i> in the last position.
java.lang.Object	clone () Returns a copy of this array
java.lang.Object	get (WNumber idx) Returns the object at position <i>idx</i> .
java.lang.Object	getWrapped () This is for internal use only
WInteger	length () Returns the length of this array.
java.lang.Object	set (int idx, java.lang.Object obj) Sets the object <i>obj</i> at position <i>idx</i> .
java.lang.Object	set (WNumber idx, java.lang.Object obj)
java.lang.Object	shift () Returns a new array shorter by one of this array without the object in the first position.

java.lang.Object	<code>shift</code> (int <code>shft</code>) Returns a new array shorter by <i>shft</i> of this array without <i>shft</i> objects at the beginning.
java.lang.Object	<code>shift</code> (WNumber <code>shiftValue</code>)
WInteger	<code>size</code> () Returns the length of this array.
java.lang.Object	<code>slice</code> (WNumber <code>from</code> , WNumber <code>to</code>) Returns the specified range of this array as a new array.
Wrapper.Type	<code>type</code> ()
java.lang.Object	<code>unshift</code> (java.lang.Object <code>obj</code>) Returns a new array longer by one of this array with <i>obj</i> in the first position.

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

WArray

```
public WArray(java.lang.Object[] array)
```

WArray

```
public WArray(WNumber len)
```

WArray

```
public WArray(int len)
```

Methods

type

```
public Wrapper.Type type()
```


(continued from last page)

getWrapped

```
public java.lang.Object getWrapped()
```

This is for internal use only

length

```
public WInteger length()
```

Returns the length of this array.

size

```
public WInteger size()
```

Returns the length of this array.

See Also:

[length\(\)](#)

get

```
public java.lang.Object get(WNumber idx)
```

Returns the object at position *idx*.

set

```
public java.lang.Object set(int idx,  
    java.lang.Object obj)
```

Sets the object *obj* at position *idx*.

set

```
public java.lang.Object set(WNumber idx,  
    java.lang.Object obj)
```

See Also:

[set\(int, Object\)](#)

add

```
public java.lang.Object add(java.lang.Object obj)
```

Returns a new array longer by one of this array with *obj* in the last position.

shift

```
public java.lang.Object shift(int shft)
```

Returns a new array shorter by *shft* of this array without *shft* objects at the beginning.

(continued from last page)

shift

```
public java.lang.Object shift(WNumber shiftValue)
```

See Also:

[shift\(int\)](#)

shift

```
public java.lang.Object shift()
```

Returns a new array shorter by one of this array without the object in the first position.

unshift

```
public java.lang.Object unshift(java.lang.Object obj)
```

Returns a new array longer by one of this array with *obj* in the first position.

slice

```
public java.lang.Object slice(WNumber from,  
                               WNumber to)
```

Returns the specified range of this array as a new array. The value at *from* is placed at the first position in the new array. The length of the returned array will be *to* - *from*.

clone

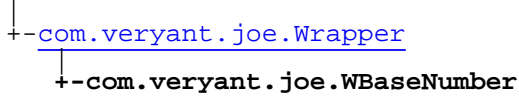
```
public java.lang.Object clone()
```

Returns a copy of this array

com.veryant.joe

Class WBaseNumber

java.lang.Object



All Implemented Interfaces:

[WNumber](#), InternalObject

Direct Known Subclasses:

[WLong](#), [WDouble](#), [WBigDecimal](#)

public abstract class **WBaseNumber**
 extends [Wrapper](#)
 implements InternalObject, [WNumber](#)

This class defines the methods that are common to all the numbers i.e int, short, long, double and BigDecimal.. Consider that in JOE some operators symbols are equivalent to methods names, i.e.

add	+
subtract	-
multiply	*
divide	/
remainder	%
equals	=
gt	>
lt	<
ge	>=
le	<=
ne	<>

Constructor Summary

public	WBaseNumber ()
--------	---------------------------------

Method Summary

java.math.BigDecimal	add (WNumber n)
java.math.BigDecimal	divide (WNumber n)
boolean	equals (java.lang.Object n)
WBoolean	equals (WNumber n)
WBoolean	ge (WNumber n)
WBoolean	gt (WNumber n)

WBoolean	le (WNumber n)
WBoolean	lt (WNumber n)
java.math.BigDecimal	multiply (WNumber n)
WBoolean	ne (WNumber n)
java.math.BigDecimal	remainder (WNumber n)
java.math.BigDecimal	subtract (WNumber n)
WString	toChar ()

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

WBaseNumber

```
public WBaseNumber()
```

Methods

add

```
public java.math.BigDecimal add(WNumber n)
```

subtract

```
public java.math.BigDecimal subtract(WNumber n)
```

multiply

```
public java.math.BigDecimal multiply(WNumber n)
```

(continued from last page)

divide

```
public java.math.BigDecimal divide(WNumber n)
```

remainder

```
public java.math.BigDecimal remainder(WNumber n)
```

equals

```
public boolean equals(java.lang.Object n)
```

equals

```
public WBoolean equals(WNumber n)
```

ne

```
public WBoolean ne(WNumber n)
```

lt

```
public WBoolean lt(WNumber n)
```

le

```
public WBoolean le(WNumber n)
```

gt

```
public WBoolean gt(WNumber n)
```

ge

```
public WBoolean ge(WNumber n)
```

(continued from last page)

toChar

```
public WString toChar( )
```

com.veryant.joe

Class WBigDecimal

```

java.lang.Object
├── com.veryant.joe.Wrapper
│   ├── com.veryant.joe.WBaseNumber
│   │   └── com.veryant.joe.WBigDecimal

```

All Implemented Interfaces:

[WNumber](#), [InternalObject](#), [WNumber](#)

```

public class WBigDecimal
extends WBaseNumber
implements WNumber, InternalObject, WNumber

```

Constructor Summary

public	WBigDecimal (java.math.BigDecimal n)
public	WBigDecimal (java.lang.Integer n)
public	WBigDecimal (java.lang.Double n)
public	WBigDecimal (java.lang.String n)

Method Summary

java.math.BigDecimal	abs ()
java.math.BigDecimal	bigDecimalValue ()
byte	byteValue ()
char	charValue ()
double	doubleValue ()
float	floatValue ()
java.lang.Object	getWrapped ()
int	intValue ()
long	longValue ()

java.math.BigDecimal	movePointLeft (WInteger newScale)
java.math.BigDecimal	movePointRight (WInteger newScale)
java.math.BigDecimal	negate ()
java.math.BigDecimal	pow (WInteger n)
int	precision ()
int	scale ()
java.math.BigDecimal	setScale (WInteger newScale)
java.math.BigDecimal	setScale (WInteger newScale, WInteger roundingMode)
short	shortValue ()
int	signum ()
java.math.BigDecimal	stripTrailingZeros ()
java.lang.String	toString ()
Wrapper.Type	type ()

Methods inherited from class [com.veryant.joe.WBaseNumber](#)

[add](#), [divide](#), [equals](#), [equals](#), [ge](#), [gt](#), [le](#), [lt](#), [multiply](#), [ne](#), [remainder](#), [subtract](#), [toChar](#)

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

(continued from last page)

WBigDecimal

```
public WBigDecimal(java.math.BigDecimal n)
```

WBigDecimal

```
public WBigDecimal(java.lang.Integer n)
```

WBigDecimal

```
public WBigDecimal(java.lang.Double n)
```

WBigDecimal

```
public WBigDecimal(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

bigDecimalValue

```
public java.math.BigDecimal bigDecimalValue()
```

byteValue

```
public byte byteValue()
```

charValue

```
public char charValue()
```

(continued from last page)

shortValue

```
public short shortValue()
```

intValue

```
public int intValue()
```

longValue

```
public long longValue()
```

floatValue

```
public float floatValue()
```

doubleValue

```
public double doubleValue()
```

toString

```
public java.lang.String toString()
```

abs

```
public java.math.BigDecimal abs()
```

negate

```
public java.math.BigDecimal negate()
```

pow

```
public java.math.BigDecimal pow(WInteger n)
```

precision

```
public int precision()
```

scale

```
public int scale()
```

movePointLeft

```
public java.math.BigDecimal movePointLeft(WInteger newScale)
```

movePointRight

```
public java.math.BigDecimal movePointRight(WInteger newScale)
```

setScale

```
public java.math.BigDecimal setScale(WInteger newScale)
```

setScale

```
public java.math.BigDecimal setScale(WInteger newScale,  
    WInteger roundingMode)
```

signum

```
public int signum()
```

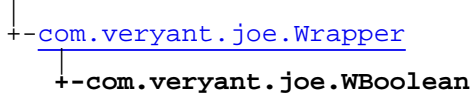
stripTrailingZeros

```
public java.math.BigDecimal stripTrailingZeros()
```

com.veryant.joe

Class WBoolean

java.lang.Object



All Implemented Interfaces:
InternalObject

public class **WBoolean**
extends [Wrapper](#)

Field Summary

public static final	FALSE
public static final	TRUE

Method Summary

WBoolean	and (Block m)
WBoolean	and (WBoolean b)
boolean	booleanValue ()
boolean	equals (java.lang.Object b)
java.lang.Object	getWrapped ()
java.lang.Object	ifFalse (Block bTrue)
java.lang.Object	ifFalse (Block bTrue, Block bFalse)
java.lang.Object	ifTrue (Block bTrue)
java.lang.Object	ifTrue (Block bTrue, Block bFalse)
java.lang.Object	iif (java.lang.Object oTrue, java.lang.Object oFalse)
boolean	ne (WBoolean b)
WBoolean	not ()

WBoolean	or (Block m)
WBoolean	or (WBoolean b)
java.lang.String	toString ()
Wrapper.Type	type ()
WBoolean	xor (Block m)
WBoolean	xor (WBoolean b)

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Fields

TRUE

public static final com.veryant.joe.WBoolean **TRUE**

FALSE

public static final com.veryant.joe.WBoolean **FALSE**

Methods

type

public Wrapper.Type **type**()

getWrapped

public java.lang.Object **getWrapped**()

booleanValue

public boolean **booleanValue**()

equals

```
public boolean equals(java.lang.Object b)
```

ne

```
public boolean ne(WBoolean b)
```

and

```
public WBoolean and(WBoolean b)
```

and

```
public WBoolean and(Block m)  
    throws JOEException
```

or

```
public WBoolean or(WBoolean b)
```

or

```
public WBoolean or(Block m)  
    throws JOEException
```

xor

```
public WBoolean xor(WBoolean b)
```

xor

```
public WBoolean xor(Block m)  
    throws JOEException
```

not

```
public WBoolean not()
```

iif

```
public java.lang.Object iif(java.lang.Object oTrue,  
    java.lang.Object oFalse)  
    throws JOEException
```

ifTrue

```
public java.lang.Object ifTrue(Block bTrue)  
    throws JOEException
```

ifTrue

```
public java.lang.Object ifTrue(Block bTrue,  
    Block bFalse)  
    throws JOEException
```

ifFalse

```
public java.lang.Object ifFalse(Block bTrue,  
    Block bFalse)  
    throws JOEException
```

ifFalse

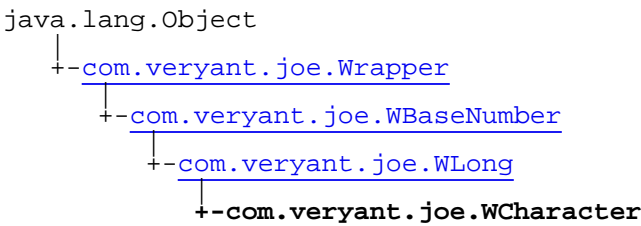
```
public java.lang.Object ifFalse(Block bTrue)  
    throws JOEException
```

toString

```
public java.lang.String toString()
```

com.veryant.joe

Class WCharacter



All Implemented Interfaces:
InternalObject, [WNumber](#)

public class **WCharacter**
extends [WLong](#)

Constructor Summary	
public	WCharacter (long n)
public	WCharacter (java.lang.String n)

Method Summary	
java.lang.String	add (java.lang.Object s)
java.lang.String	add (WString s)
java.lang.Object	getWrapped ()
java.lang.String	toString ()
Wrapper.Type	type ()

Methods inherited from class com.veryant.joe.WLong
abs , add , add , add , and , and , bigDecimalValue , byteValue , charValue , divide , divide , divide , doubleValue , equals , equals , equals , floatValue , ge , ge , ge , getWrapped , gt , gt , gt , intValue , le , le , le , longValue , lt , lt , lt , multiply , multiply , multiply , ne , ne , ne , negate , not , or , or , pow , remainder , remainder , remainder , shifta , shiftr , shiftr , shortValue , signum , subtract , subtract , subtract , toBinaryString , toHexString , toString , type , xor , xor

Methods inherited from class com.veryant.joe.WBaseNumber
add , divide , equals , equals , ge , gt , le , lt , multiply , ne , remainder , subtract , toChar

Methods inherited from class com.veryant.joe.Wrapper
--

[getWrapped](#), [newInstance](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

WCharacter

```
public WCharacter(long n)
```

WCharacter

```
public WCharacter(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

add

```
public java.lang.String add(WString s)
```

add

```
public java.lang.String add(java.lang.Object s)
```

toString

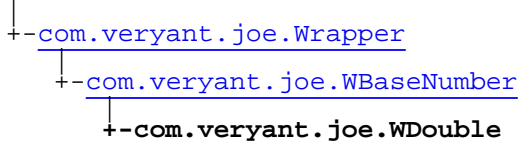
```
public java.lang.String toString()
```

(continued from last page)

com.veryant.joe

Class WDouble

java.lang.Object



All Implemented Interfaces:

[WNumber](#), [InternalObject](#), [WNumber](#)

public class **WDouble**

extends [WBaseNumber](#)

implements [WNumber](#), [InternalObject](#), [WNumber](#)

Constructor Summary

public	WDouble (double n)
public	WDouble (java.lang.String n)

Method Summary

WDouble	abs ()
WDouble	add (WDouble n)
WDouble	add (WInteger n)
WDouble	add (WLong n)
java.math.BigDecimal	bigDecimalValue ()
byte	byteValue ()
char	charValue ()
WDouble	divide (WDouble n)
WDouble	divide (WInteger n)
WDouble	divide (WLong n)
double	doubleValue ()

WBoolean	equals(WDouble n)
WBoolean	equals(WInteger n)
WBoolean	equals(WLong n)
float	floatValue()
WBoolean	ge(WDouble n)
WBoolean	ge(WInteger n)
WBoolean	ge(WLong n)
java.lang.Object	getWrapped()
WBoolean	gt(WDouble n)
WBoolean	gt(WInteger n)
WBoolean	gt(WLong n)
int	intValue()
WBoolean	le(WDouble n)
WBoolean	le(WInteger n)
WBoolean	le(WLong n)
long	longValue()
WBoolean	lt(WDouble n)
WBoolean	lt(WInteger n)
WBoolean	lt(WLong n)
WDouble	multiply(WDouble n)
WDouble	multiply(WInteger n)
WDouble	multiply(WLong n)
WBoolean	ne(WDouble n)
WBoolean	ne(WInteger n)

WBoolean	ne (WLong n)
WDouble	negate ()
WDouble	pow (WInteger e)
WDouble	remainder (WDouble n)
WDouble	remainder (WInteger n)
WDouble	remainder (WLong n)
short	shortValue ()
int	signum ()
WDouble	subtract (WDouble n)
WDouble	subtract (WInteger n)
WDouble	subtract (WLong n)
java.lang.String	toString ()
Wrapper.Type	type ()

Methods inherited from class [com.veryant.joe.WBaseNumber](#)

[add](#), [divide](#), [equals](#), [equals](#), [ge](#), [gt](#), [le](#), [lt](#), [multiply](#), [ne](#), [remainder](#), [subtract](#), [toChar](#)

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

(continued from last page)

WDouble

```
public WDouble(double n)
```

WDouble

```
public WDouble(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

byteValue

```
public byte byteValue()
```

charValue

```
public char charValue()
```

shortValue

```
public short shortValue()
```

intValue

```
public int intValue()
```

longValue

```
public long longValue()
```

(continued from last page)

floatValue

```
public float floatValue()
```

doubleValue

```
public double doubleValue()
```

bigDecimalValue

```
public java.math.BigDecimal bigDecimalValue()
```

signum

```
public int signum()
```

abs

```
public WDouble abs()
```

negate

```
public WDouble negate()
```

pow

```
public WDouble pow(WInteger e)
```

add

```
public WDouble add(WDouble n)
```

add

```
public WDouble add(WInteger n)
```

add

```
public WDouble add(WLong n)
```

(continued from last page)

subtract

```
public WDouble subtract(WDouble n)
```

subtract

```
public WDouble subtract(WInteger n)
```

subtract

```
public WDouble subtract(WLong n)
```

multiply

```
public WDouble multiply(WDouble n)
```

multiply

```
public WDouble multiply(WInteger n)
```

multiply

```
public WDouble multiply(WLong n)
```

divide

```
public WDouble divide(WDouble n)
```

divide

```
public WDouble divide(WInteger n)
```

divide

```
public WDouble divide(WLong n)
```

(continued from last page)

remainder

```
public WDouble remainder(WDouble n)
```

remainder

```
public WDouble remainder(WInteger n)
```

remainder

```
public WDouble remainder(WLong n)
```

equals

```
public WBoolean equals(WDouble n)
```

equals

```
public WBoolean equals(WInteger n)
```

equals

```
public WBoolean equals(WLong n)
```

lt

```
public WBoolean lt(WDouble n)
```

lt

```
public WBoolean lt(WInteger n)
```

lt

```
public WBoolean lt(WLong n)
```

gt

```
public WBoolean gt(WDouble n)
```

(continued from last page)

gt

```
public WBoolean gt(WInteger n)
```

gt

```
public WBoolean gt(WLong n)
```

ge

```
public WBoolean ge(WDouble n)
```

ge

```
public WBoolean ge(WInteger n)
```

ge

```
public WBoolean ge(WLong n)
```

le

```
public WBoolean le(WDouble n)
```

le

```
public WBoolean le(WInteger n)
```

le

```
public WBoolean le(WLong n)
```

ne

```
public WBoolean ne(WDouble n)
```

(continued from last page)

ne

```
public WBoolean ne(WInteger n)
```

ne

```
public WBoolean ne(WLong n)
```

toString

```
public java.lang.String toString()
```

com.veryant.joe

Class WInteger

```

java.lang.Object
├── com.veryant.joe.Wrapper
│   ├── com.veryant.joe.WBaseNumber
│   │   ├── com.veryant.joe.WLong
│   │   └── com.veryant.joe.WInteger

```

All Implemented Interfaces:
InternalObject, [WNumber](#)

public class **WInteger**
extends [WLong](#)

Constructor Summary

public	WInteger (long n)
public	WInteger (java.lang.String n)

Method Summary

WInteger	abs ()
WInteger	add (WInteger n)
WInteger	and (WInteger n)
WLong	and (WLong n)
WInteger	divide (WInteger n)
java.lang.Object	getWrapped ()
WInteger	multiply (WInteger n)
WInteger	negate ()
WInteger	not ()
WLong	or (WInteger n)
WLong	or (WLong n)

WInteger	pow (WInteger e)
WInteger	remainder (WInteger n)
WLong	shiftd (WInteger n)
WLong	shiftdl (WInteger n)
WLong	shiftr (WInteger n)
int	signum ()
WInteger	subtract (WInteger n)
java.lang.String	toBinaryString ()
java.lang.String	toHexString ()
Wrapper.Type	type ()
WLong	xor (WInteger n)
WLong	xor (WLong n)

Methods inherited from class [com.veryant.joe.WLong](#)

[abs](#), [add](#), [add](#), [add](#), [and](#), [and](#), [bigDecimalValue](#), [byteValue](#), [charValue](#), [divide](#), [divide](#), [divide](#), [doubleValue](#), [equals](#), [equals](#), [equals](#), [floatValue](#), [ge](#), [ge](#), [ge](#), [getWrapped](#), [gt](#), [gt](#), [gt](#), [intValue](#), [le](#), [le](#), [le](#), [longValue](#), [lt](#), [lt](#), [lt](#), [multiply](#), [multiply](#), [multiply](#), [ne](#), [ne](#), [ne](#), [negate](#), [not](#), [or](#), [or](#), [pow](#), [remainder](#), [remainder](#), [remainder](#), [shiftd](#), [shiftdl](#), [shiftr](#), [shortValue](#), [signum](#), [subtract](#), [subtract](#), [subtract](#), [toBinaryString](#), [toHexString](#), [toString](#), [type](#), [xor](#), [xor](#)

Methods inherited from class [com.veryant.joe.WBaseNumber](#)

[add](#), [divide](#), [equals](#), [equals](#), [ge](#), [gt](#), [le](#), [lt](#), [multiply](#), [ne](#), [remainder](#), [subtract](#), [toChar](#)

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

(continued from last page)

Constructors

WInteger

```
public WInteger(long n)
```

WInteger

```
public WInteger(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

add

```
public WInteger add(WInteger n)
```

subtract

```
public WInteger subtract(WInteger n)
```

multiply

```
public WInteger multiply(WInteger n)
```

divide

```
public WInteger divide(WInteger n)
```

remainder

```
public WInteger remainder(WInteger n)
```

and

```
public WLong and(WLong n)
```

and

```
public WInteger and(WInteger n)
```

or

```
public WLong or(WLong n)
```

or

```
public WLong or(WInteger n)
```

xor

```
public WLong xor(WLong n)
```

xor

```
public WLong xor(WInteger n)
```

not

```
public WInteger not()
```

shifl

```
public WLong shifl(WInteger n)
```

shiftr

```
public WLong shiftr(WInteger n)
```

shifla

```
public WLong shifla(WInteger n)
```

toHexString

```
public java.lang.String toHexString()
```

toBinaryString

```
public java.lang.String toBinaryString()
```

signum

```
public int signum()
```

abs

```
public WInteger abs()
```

negate

```
public WInteger negate()
```

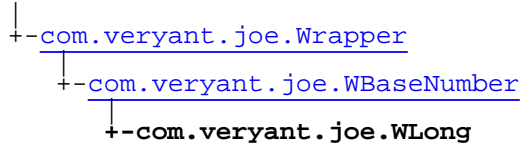
pow

```
public WInteger pow(WInteger e)
```


com.veryant.joe

Class WLong

java.lang.Object



All Implemented Interfaces:

InternalObject, [WNumber](#)

Direct Known Subclasses:

[WShort](#), [WInteger](#), [WCharacter](#)

public class **WLong**
 extends [WBaseNumber](#)

Constructor Summary

public	WLong (long n)
public	WLong (java.lang.String n)

Method Summary

WLong	abs ()
WDouble	add (WDouble n)
WLong	add (WInteger n)
WLong	add (WLong n)
WLong	and (WInteger n)
WLong	and (WLong n)
java.math.BigDecimal	bigDecimalValue ()
byte	byteValue ()
char	charValue ()
WDouble	divide (WDouble n)
WLong	divide (WInteger n)

WLong	divide(WLong n)
double	doubleValue()
WBoolean	equals(WDouble n)
WBoolean	equals(WInteger n)
WBoolean	equals(WLong n)
float	floatValue()
WBoolean	ge(WDouble n)
WBoolean	ge(WInteger n)
WBoolean	ge(WLong n)
java.lang.Object	getWrapped()
WBoolean	gt(WDouble n)
WBoolean	gt(WInteger n)
WBoolean	gt(WLong n)
int	intValue()
WBoolean	le(WDouble n)
WBoolean	le(WInteger n)
WBoolean	le(WLong n)
long	longValue()
WBoolean	lt(WDouble n)
WBoolean	lt(WInteger n)
WBoolean	lt(WLong n)
WDouble	multiply(WDouble n)
WLong	multiply(WInteger n)
WLong	multiply(WLong n)

WBoolean	ne (WDouble n)
WBoolean	ne (WInteger n)
WBoolean	ne (WLong n)
WLong	negate ()
WLong	not ()
WLong	or (WInteger n)
WLong	or (WLong n)
WLong	pow (WInteger e)
WDouble	remainder (WDouble n)
WLong	remainder (WInteger n)
WLong	remainder (WLong n)
WLong	shiftd (WInteger n)
WLong	shiftdl (WInteger n)
WLong	shiftr (WInteger n)
short	shortValue ()
int	signum ()
WDouble	subtract (WDouble n)
WLong	subtract (WInteger n)
WLong	subtract (WLong n)
java.lang.String	toBinaryString ()
java.lang.String	toHexString ()
java.lang.String	toString ()
Wrapper.Type	type ()
WLong	xor (WInteger n)

[WLong](#)[xor](#)([WLong](#) n)Methods inherited from class [com.veryant.joe.WBaseNumber](#)[add](#), [divide](#), [equals](#), [equals](#), [ge](#), [gt](#), [le](#), [lt](#), [multiply](#), [ne](#), [remainder](#), [subtract](#), [toChar](#)Methods inherited from class [com.veryant.joe.Wrapper](#)[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)Methods inherited from interface [com.veryant.joe.WNumber](#)[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

WLong

```
public WLong(long n)
```

WLong

```
public WLong(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

byteValue

```
public byte byteValue()
```

(continued from last page)

charValue

```
public char charValue()
```

shortValue

```
public short shortValue()
```

intValue

```
public int intValue()
```

longValue

```
public long longValue()
```

floatValue

```
public float floatValue()
```

doubleValue

```
public double doubleValue()
```

bigDecimalValue

```
public java.math.BigDecimal bigDecimalValue()
```

signum

```
public int signum()
```

abs

```
public WLong abs()
```

negate

```
public WLong negate()
```

pow

```
public WLong pow(WInteger e)
```

add

```
public WLong add(WLong n)
```

add

```
public WLong add(WInteger n)
```

add

```
public WDouble add(WDouble n)
```

subtract

```
public WLong subtract(WLong n)
```

subtract

```
public WLong subtract(WInteger n)
```

subtract

```
public WDouble subtract(WDouble n)
```

multiply

```
public WLong multiply(WLong n)
```

multiply

```
public WLong multiply(WInteger n)
```

(continued from last page)

multiply

```
public WDouble multiply(WDouble n)
```

divide

```
public WLong divide(WLong n)
```

divide

```
public WLong divide(WInteger n)
```

divide

```
public WDouble divide(WDouble n)
```

remainder

```
public WLong remainder(WLong n)
```

remainder

```
public WLong remainder(WInteger n)
```

remainder

```
public WDouble remainder(WDouble n)
```

equals

```
public WBoolean equals(WLong n)
```

equals

```
public WBoolean equals(WInteger n)
```

equals

```
public WBoolean equals(WDouble n)
```

(continued from last page)

lt

```
public WBoolean lt(WLong n)
```

lt

```
public WBoolean lt(WInteger n)
```

lt

```
public WBoolean lt(WDouble n)
```

gt

```
public WBoolean gt(WLong n)
```

gt

```
public WBoolean gt(WInteger n)
```

gt

```
public WBoolean gt(WDouble n)
```

ge

```
public WBoolean ge(WLong n)
```

ge

```
public WBoolean ge(WInteger n)
```

ge

```
public WBoolean ge(WDouble n)
```

(continued from last page)

le

```
public WBoolean le(WLong n)
```

le

```
public WBoolean le(WInteger n)
```

le

```
public WBoolean le(WDouble n)
```

ne

```
public WBoolean ne(WLong n)
```

ne

```
public WBoolean ne(WInteger n)
```

ne

```
public WBoolean ne(WDouble n)
```

and

```
public WLong and(WLong n)
```

and

```
public WLong and(WInteger n)
```

or

```
public WLong or(WLong n)
```

or

```
public WLong or(WInteger n)
```

xor

```
public WLong xor(WLong n)
```

xor

```
public WLong xor(WInteger n)
```

not

```
public WLong not()
```

shiftl

```
public WLong shiftl(WInteger n)
```

shiftr

```
public WLong shiftr(WInteger n)
```

shifta

```
public WLong shifta(WInteger n)
```

toString

```
public java.lang.String toString()
```

toHexString

```
public java.lang.String toHexString()
```

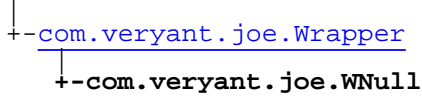
toBinaryString

```
public java.lang.String toBinaryString()
```

com.veryant.joe

Class WNull

java.lang.Object



All Implemented Interfaces:
InternalObject

public class **WNull**
extends [Wrapper](#)

Field Summary

public static final	value
---------------------	-----------------------

Method Summary

boolean	ge (java.lang.Object b)
java.lang.Object	getWrapped ()
boolean	gt (java.lang.Object b)
boolean	le (java.lang.Object b)
boolean	lt (java.lang.Object b)
boolean	ne (java.lang.Object b)
java.lang.String	toString ()
Wrapper.Type	type ()

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Fields

(continued from last page)

value

```
public static final com.veryant.joe.WNull value
```

Methods

ne

```
public boolean ne(java.lang.Object b)
```

gt

```
public boolean gt(java.lang.Object b)
```

ge

```
public boolean ge(java.lang.Object b)
```

lt

```
public boolean lt(java.lang.Object b)
```

le

```
public boolean le(java.lang.Object b)
```

getWrapped

```
public java.lang.Object getWrapped()
```

type

```
public Wrapper.Type type()
```

toString

```
public java.lang.String toString()
```

com.veryant.joe Interface WNumber

All Known Implementing Classes:

[WDouble](#), [WBigDecimal](#), [WBaseNumber](#)

public interface **WNumber**
extends

Method Summary

java.math.BigDecimal	bigDecimalValue()
byte	byteValue()
char	charValue()
double	doubleValue()
float	floatValue()
int	intValue()
long	longValue()
short	shortValue()
int	signum()

Methods

byteValue

public byte **byteValue()**

charValue

public char **charValue()**

shortValue

public short **shortValue()**

intValue

```
public int intValue()
```

longValue

```
public long longValue()
```

floatValue

```
public float floatValue()
```

doubleValue

```
public double doubleValue()
```

bigDecimalValue

```
public java.math.BigDecimal bigDecimalValue()
```

signum

```
public int signum()
```

com.veryant.joe Class Wrapper

java.lang.Object

└─com.veryant.joe.Wrapper

All Implemented Interfaces:

InternalObject

Direct Known Subclasses:

[WString](#), [WNull](#), [WBoolean](#), [WBaseNumber](#), [WArray](#)

public abstract class **Wrapper**

extends java.lang.Object

implements InternalObject

Constructor Summary

public	Wrapper()
--------	---------------------------

Method Summary

abstract java.lang.Object	getWrapped()
------------------------------	------------------------------

static Wrapper	newInstance (java.lang.Object obj)
--------------------------------	--

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Wrapper

public **Wrapper**()

Methods

newInstance

public static [Wrapper](#) **newInstance**(java.lang.Object obj)

(continued from last page)

getWrapped

```
public abstract java.lang.Object getWrapped()
```


com.veryant.joe

Class WShort

```

java.lang.Object
├── com.veryant.joe.Wrapper
│   ├── com.veryant.joe.WBaseNumber
│   │   ├── com.veryant.joe.WLong
│   │   │   └── com.veryant.joe.WShort

```

All Implemented Interfaces:

InternalObject, [WNumber](#)

public class **WShort**
 extends [WLong](#)

Constructor Summary

public	WShort (long n)
public	WShort (java.lang.String n)

Method Summary

java.lang.Object	getWrapped ()
java.lang.String	toHexString ()
Wrapper.Type	type ()

Methods inherited from class [com.veryant.joe.WLong](#)

[abs](#), [add](#), [add](#), [add](#), [and](#), [and](#), [bigDecimalValue](#), [byteValue](#), [charValue](#), [divide](#), [divide](#), [divide](#), [doubleValue](#), [equals](#), [equals](#), [equals](#), [floatValue](#), [ge](#), [ge](#), [ge](#), [getWrapped](#), [gt](#), [gt](#), [gt](#), [intValue](#), [le](#), [le](#), [le](#), [longValue](#), [lt](#), [lt](#), [lt](#), [multiply](#), [multiply](#), [multiply](#), [ne](#), [ne](#), [ne](#), [negate](#), [not](#), [or](#), [or](#), [pow](#), [remainder](#), [remainder](#), [remainder](#), [shifta](#), [shifl](#), [shiftr](#), [shortValue](#), [signum](#), [subtract](#), [subtract](#), [subtract](#), [toBinaryString](#), [toHexString](#), [toString](#), [type](#), [xor](#), [xor](#)

Methods inherited from class [com.veryant.joe.WBaseNumber](#)

[add](#), [divide](#), [equals](#), [equals](#), [ge](#), [gt](#), [le](#), [lt](#), [multiply](#), [ne](#), [remainder](#), [subtract](#), [toChar](#)

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface [com.veryant.joe.WNumber](#)

[bigDecimalValue](#), [byteValue](#), [charValue](#), [doubleValue](#), [floatValue](#), [intValue](#), [longValue](#), [shortValue](#), [signum](#)

Constructors

WShort

```
public WShort(long n)
```

WShort

```
public WShort(java.lang.String n)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

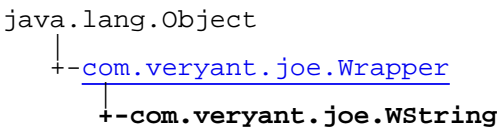
```
public java.lang.Object getWrapped()
```

toHexString

```
public java.lang.String toHexString()
```

com.veryant.joe

Class WString



All Implemented Interfaces:
InternalObject

public class **WString**
extends [Wrapper](#)

This class implements a JOE string. It implements most of the java.lang.String methods and has few more methods in order to simplify the comparison between strings, i.e. `gt`, `ge`, `lt`, `le`, `ne`, and other methods. Consider that in JOE some operators symbols are equivalent to methods names, i.e.

add	+
equals	=
gt	>
lt	<
ge	>=
le	<=
ne	<>

When a Java method outside the JOE interpreter is invoked, the instance of this object is converted in a java.lang.String.

Field Summary		
	public final	value

Constructor Summary		
	public	WString (java.lang.String s)

Method Summary		
WString	add (java.lang.Object obj)	Returns a new string that concatenates the string representation of the specified object to the end of this string.
WString	add (WString str)	Returns a new string that concatenates the specified string to the end of this string.
java.lang.String	at (WNumber index)	Returns a new string containing the character at <i>index</i> position.
java.math.BigDecimal	bigDecimalValue ()	Returns a BigDecimal if this string contains a valid representation, null otherwise.
WCharacter	charAt (WNumber index)	

WInteger	charCodeAt() Returns a integer with the codepoint of the first character of this string.
WInteger	charCodeAt(WNumber index) Returns a integer with the codepoint of the character at <i>index</i> position.
WInteger	compareTo(java.lang.String anotherString)
WInteger	compareTo(WString anotherString)
WInteger	compareToIgnoreCase(java.lang.String anotherString)
WInteger	compareToIgnoreCase(WString anotherString)
WString	concat(WString str)
WBoolean	contains(WString s)
java.lang.Double	doubleValue() Returns a double if this string contains a valid representation, null otherwise.
WBoolean	endsWith(WString s)
WBoolean	equals(WString s)
WBoolean	equalsIgnoreCase(WString s)
WBoolean	ge(WString s)
java.lang.Object	getWrapped()
WBoolean	gt(WString s)
WInteger	indexOf(WString str)
WInteger	indexOf(WString str, WNumber fromIndex)
java.lang.Integer	intValue() Returns an integer if this string contains a valid representation, null otherwise.
WBoolean	isEmpty()
WInteger	lastIndexOf(WString str)
WInteger	lastIndexOf(WString str, WNumber fromIndex)
WBoolean	le(WString s)
WInteger	length()

java.lang.Long	longValue() Returns a long integer if this string contains a valid representation, null otherwise.
WBoolean	lt(WString s)
WBoolean	matches(WString regex)
WBoolean	ne(WString s)
WString	replace(WString target, WString replacement)
WString	replaceAll(WString regex, WString replacement)
WString	replaceFirst(WString regex, WString replacement)
java.lang.String[]	split(WString regex)
WBoolean	startsWith(WString s)
WBoolean	startsWith(WString s, WNumber toffset)
WString	substring(WNumber beginIndex)
WString	substring(WNumber beginIndex, WNumber endIndex)
java.lang.String	toJava()
WString	toLowerCase()
java.lang.String	toString()
WString	toUpperCase()
WString	trim()
Wrapper.Type	type()

Methods inherited from class [com.veryant.joe.Wrapper](#)

[getWrapped](#), [newInstance](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Fields

(continued from last page)

value

```
public final java.lang.String value
```

Constructors

WString

```
public WString(java.lang.String s)
```

Methods

type

```
public Wrapper.Type type()
```

getWrapped

```
public java.lang.Object getWrapped()
```

equals

```
public WBoolean equals(WString s)
```

lt

```
public WBoolean lt(WString s)
```

gt

```
public WBoolean gt(WString s)
```

ge

```
public WBoolean ge(WString s)
```

le

```
public WBoolean le(WString s)
```

ne

```
public WBoolean ne(WString s)
```

concat

```
public WString concat(WString str)
```

add

```
public WString add(WString str)
```

Returns a new string that concatenates the specified string to the end of this string.

add

```
public WString add(java.lang.Object obj)
```

Returns a new string that concatenates the string representation of the specified object to the end of this string.

startsWith

```
public WBoolean startsWith(WString s)
```

startsWith

```
public WBoolean startsWith(WString s,  
    WNumber toffset)
```

substring

```
public WString substring(WNumber beginIndex)
```

substring

```
public WString substring(WNumber beginIndex,  
    WNumber endIndex)
```

charAt

```
public WCharacter charAt(WNumber index)
```

(continued from last page)

toLowerCase

```
public WString toLowerCase()
```

toUpperCase

```
public WString toUpperCase()
```

length

```
public WInteger length()
```

compareTo

```
public WInteger compareTo(java.lang.String anotherString)
```

compareToIgnoreCase

```
public WInteger compareToIgnoreCase(java.lang.String anotherString)
```

compareTo

```
public WInteger compareTo(WString anotherString)
```

compareToIgnoreCase

```
public WInteger compareToIgnoreCase(WString anotherString)
```

contains

```
public WBoolean contains(WString s)
```

endsWith

```
public WBoolean endsWith(WString s)
```

equalsIgnoreCase

```
public WBoolean equalsIgnoreCase(WString s)
```

indexOf

```
public WInteger indexOf(WString str)
```

indexOf

```
public WInteger indexOf(WString str,  
    WNumber fromIndex)
```

isEmpty

```
public WBoolean isEmpty()
```

lastIndexOf

```
public WInteger lastIndexOf(WString str)
```

lastIndexOf

```
public WInteger lastIndexOf(WString str,  
    WNumber fromIndex)
```

matches

```
public WBoolean matches(WString regex)
```

replace

```
public WString replace(WString target,  
    WString replacement)
```

replaceAll

```
public WString replaceAll(WString regex,  
    WString replacement)
```

replaceFirst

```
public WString replaceFirst(WString regex,  
    WString replacement)
```

split

```
public java.lang.String[] split(WString regex)
```

trim

```
public WString trim()
```

toJava

```
public java.lang.String toJava()
```

toString

```
public java.lang.String toString()
```

intValue

```
public java.lang.Integer intValue()
```

Returns an integer if this string contains a valid representation, null otherwise.

longValue

```
public java.lang.Long longValue()
```

Returns a long integer if this string contains a valid representation, null otherwise.

doubleValue

```
public java.lang.Double doubleValue()
```

Returns a double if this string contains a valid representation, null otherwise.

bigDecimalValue

```
public java.math.BigDecimal bigDecimalValue()
```

Returns a BigDecimal if this string contains a valid representation, null otherwise.

at

```
public java.lang.String at(WNumber index)
```

Returns a new string containing the character at *index* position.

(continued from last page)

charCodeAt

```
public WInteger charCodeAt(WNumber index)
```

Returns a integer with the codepoint of the character at *index* position.

charCodeAt

```
public WInteger charCodeAt()
```

Returns a integer with the codepoint of the first character of this string.

Index

\$

\$break 25
\$case 29, 30
\$default 30
\$extends 9, 17
\$for 23
\$if 22
\$instanceof 16
\$new 8, 16
\$switch 25
\$throw 15
\$try 25
\$while 23

A

abs 42, 55, 64, 69
add 8, 33, 36, 49, 55, 62, 70, 87
addPath 16
and 46, 63, 73
areSameObject 15
array 14
arraySort 27
asc 17
at 90

B

bigDecimalValue 41, 55, 69, 78, 90
binarySearch 27
booleanValue 45
breakLoop 25
byteValue 41, 54, 68, 77

C

charAt 87
charCodeAt 90, 91
charValue 41, 54, 68, 77
chr 17
clone 8, 34
CommandBase 13

compareTo 88
compareToIgnoreCase 88
concat 87
contains 88

D

debug 17
DefaultCommand 21
divide 37, 56, 62, 71
doubleValue 42, 55, 69, 78, 90
doWhile 23
doWhileFalse 7
doWhileTrue 7

E

endSwitch 30
endsWith 88
eprint 22
eprintln 21, 22
equals 37, 46, 57, 71, 86
equalsIgnoreCase 88
exec 6, 26
execFromDir 26
execGetOut 26
execJoe 27

F

FALSE 45
floatValue 42, 54, 69, 78
foreach 23, 24, 25

G

ge 37, 58, 72, 76, 86
get 33
getArgv 8
getClass 14
getClassRef 15
getCol 9
getcwd 17
getField 14
getGlob 27

getJoeClass 9
getPath 16
getRow 9
getSetLocalVariable 10
getSetVariable 10
getStaticField 14
getURL 17
getVariable 7, 8
getVariablesNames 8
getWrapped 32, 41, 45, 49, 54, 62, 68, 76, 79, 82, 86
gt 37, 57, 58, 72, 76, 86

I

ifFalse 47
ifTrue 47
iif 47
indexOf 89
init 6
intValue 42, 54, 69, 78, 90
isConsole 21
isEmpty 89
isNull 15
isThrowable 15

J

joe 16

L

lastIndexOf 89
le 37, 58, 72, 73, 76, 86
length 33, 88
longValue 42, 54, 69, 78, 90
lookForVariable 9
lt 37, 57, 72, 76, 86

M

matches 89
movePointLeft 43
movePointRight 43
multiply 6, 36, 56, 62, 70

N

name 9
ne 37, 46, 58, 59, 73, 76, 87
negate 42, 55, 64, 69
newArray 14
newInstance 13, 14, 79
newInterface 15
nl 17
not 46, 63, 74

O

or 46, 63, 73

P

pipe 26
pow 42, 55, 64, 70
precision 42
print 22
println 21, 22

R

random 27
readLine 21
remainder 37, 56, 57, 62, 71
replace 89
replaceAll 89
replaceFirst 89
runAsBlock 26
runJoe 26

S

scale 43
set 33
setParent 9
setScale 43
setVariable 7
shift 33, 34
shifta 63, 74
shiftl 63, 74
shiftr 63, 74

shortValue 41, 54, 69, 77
showInputDialog 28
showMessageDialog 27
signum 43, 55, 64, 69, 78
size 33
sleep 27
slice 34
split 90
startsWith 87
stripTrailingZeros 43
substring 87
subtract 36, 56, 62, 70
switchTrue 25
system 26
systemExit 15
systemGetenv 27

T

toBinaryString 64, 74
toChar 37
toHexString 64, 74, 82
toJava 90
toLowerCase 87
toString 9, 17, 42, 47, 49, 59, 74, 76, 90
toUpperCase 88
trim 90
TRUE 45
type 32, 41, 45, 49, 54, 62, 68, 76, 82, 86
typename 16

U

unshift 34

V

value 75, 85
version 17

W

WArray 32
WBaseNumber 36
WBigDecimal 40, 41

WCharacter 49
WDouble 53, 54
whileFalse 7
whileTrue 7
WInteger 61, 62
WLong 68
Wrapper 79
WShort 82
WString 86

X

xor 46, 63, 74