com.sun.btrace

Class BTraceUtils

java.lang.Object
    com.sun.btrace.BTraceUtils

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

This class is an all-in-one wrapper for BTrace DSL methods　　**所有功能于一身的BTrace DSL方法包装器**

Author:

A. Sundararajan, Jaroslav Bachorik

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static class	BTraceUtils.Aggregations
static class	BTraceUtils.Atomic
static class	BTraceUtils.Collections
static class	BTraceUtils.Counters
static class	BTraceUtils.D
static class	BTraceUtils.Export
static class	BTraceUtils.Numbers
static class	BTraceUtils.Profiling Profiling support.
static class	BTraceUtils.References
static class	BTraceUtils.Reflective
static class	BTraceUtils.Speculation
static class	BTraceUtils.Strings
static class	BTraceUtils.Sys
static class	BTraceUtils.Threads Namespaced methods
static class	BTraceUtils.Time

Method Summary

All Methods | Static Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method and Description
static java.lang.String	$(int n) Returns n'th command line argument.
static int	$length() Returns the number of command line arguments.
static int	accessFlags(java.lang.Class clazz) Returns the access flags of the given Class.
static int	accessFlags(java.lang.reflect.Field field) Returns the access flags of the given Field.
static int	addAndGet(java.util.concurrent.atomic.AtomicInteger ai, int delta) Atomically adds the given value to the current value.
static long	addAndGet(java.util.concurrent.atomic.AtomicLong al, long delta) Atomically adds the given value to the current value.
static <V> void	addLast(java.util.Deque<V> queue, V value)
static void	addToAggregation(Aggregation aggregation, AggregationKey key, long value) Adds a value to the aggregation with a grouping key.

static void	**addToAggregation**(**Aggregation** aggregation, long value) Adds a value to the aggregation with no grouping key.
static long	**availableProcessors**() Returns the number of processors available to the Java virtual machine.
static java.lang.String	**bootClassPath**() Returns the boot class path that is used by the bootstrap class loader to search for class files.
static java.lang.Boolean	**box**(boolean b) Returns a Boolean instance representing the specified boolean value.
static java.lang.Byte	**box**(byte b) Returns a Byte instance representing the specified byte value.
static java.lang.Character	**box**(char c) Returns a Character instance representing the specified char value.
static java.lang.Double	**box**(double d) Returns a Double instance representing the specified double value.
static java.lang.Float	**box**(float f) Returns a Float instance representing the specified float value.
static java.lang.Integer	**box**(int i) Returns a Integer instance representing the specified int value.
static java.lang.Long	**box**(long l) Returns a Long instance representing the specified long value.
static java.lang.Short	**box**(short s) Returns a Short instance representing the specified short value.
static java.lang.Class	**classForName**(java.lang.String name) Returns Class object for given class name.
static java.lang.Class	**classForName**(java.lang.String name, java.lang.ClassLoader cl) Returns the Class for the given class name using the given class loader.
static java.lang.Class	**classOf**(java.lang.Object obj) Returns the runtime class of the given Object.
static java.lang.String	**classPath**() Returns the Java class path that is used by the system class loader to search for class files.
static <K,V> void	**clear**(java.util.Map<K,V> map)
static void	**clearAggregation**(**Aggregation** aggregation) Resets values within the aggregation to the default.
static void	**commit**(int id) Commits the speculative buffer associated with id.
static long	**committed**(java.lang.management.MemoryUsage mu) Returns the amount of memory in bytes that is committed for the Java virtual machine to use.
static boolean	**compare**(java.lang.Object obj1, java.lang.Object obj2) Indicates whether two given objects are "equal to" one another. **指示“两个给定的对象是否相等”**
static boolean	**compareAndSet**(java.util.concurrent.atomic.AtomicInteger ai, int expect, int update) Atomically sets the value of given AtomitInteger to the given updated value if the current value == the expected value.
static boolean	**compareAndSet**(java.util.concurrent.atomic.AtomicLong al, long expect, long update) Atomically sets the value to the given updated value if the current value == the expected value.
static int	**compareTo**(java.lang.String str1, java.lang.String str2) Compares two strings lexicographically.
static int	**compareToIgnoreCase**(java.lang.String str1, java.lang.String str2) Compares two strings lexicographically, ignoring case differences.
static java.lang.String	**concat**(java.lang.String str1, java.lang.String str2) Concatenates the specified strings together. **将“指定的字符串”连接在一起**

static <E> boolean	**contains**(java.util.Collection<E> coll, java.lang.Object obj)
static boolean	**contains**(java.lang.Object[] array, java.lang.Object value)
static <K,V> boolean	**containsKey**(java.util.Map<K,V> map, java.lang.Object key)
static <K,V> boolean	**containsValue**(java.util.Map<K,V> map, java.lang.Object value)
static java.lang.ClassLoader	**contextClassLoader**() Returns the current context class loader
static java.lang.Thread	**currentThread**() 返回“当前正在执行的线程对象的引用” Returns a reference to the currently executing thread object.
static long	**currentThreadCpuTime**() Returns the total CPU time for the current thread in nanoseconds.
static long	**currentThreadUserTime**() Returns the CPU time that the current thread has executed in user mode in nanoseconds.
static long	**daemonThreadCount**() Returns the current number of live daemon threads.
static void	**deadlocks**() Prints the Java level deadlocks detected (if any).
static void	**deadlocks**(boolean stackTrace) 如果“死锁”存在的话，则打印“检测到的死锁信息” Prints deadlocks detected (if any).
static java.lang.Class	**declaringClass**(java.lang.reflect.Field field) Returns the Class object representing the class or interface that declares the field represented by the given Field object.
static int	**decrementAndGet**(java.util.concurrent.atomic.AtomicInteger ai) Atomically decrements by one the current value of given AtomicInteger.
static long	**decrementAndGet**(java.util.concurrent.atomic.AtomicLong al) Atomically decrements by one the current value.
static java.lang.Object	**deref**(java.lang.ref.Reference ref) Returns the given reference object's referent.
static void	**discard**(int id) Discards the speculative buffer associated with id.
static int	**dtraceProbe**(java.lang.String str1, java.lang.String str2) BTrace to DTrace communication chennal.
static int	**dtraceProbe**(java.lang.String str1, java.lang.String str2, int i1) BTrace to DTrace communication chennal.
static int	**dtraceProbe**(java.lang.String str1, java.lang.String str2, int i1, int i2) BTrace to DTrace communication channel.
static void	**dumpHeap**(java.lang.String fileName) 转储“Java堆快照”到以hprof二进制格式的文件 Dump the snapshot of the Java heap to a file in hprof binary format.
static void	**dumpHeap**(java.lang.String fileName, boolean live) Dump the snapshot of the Java heap to a file in hprof binary format.
static boolean	**endsWith**(java.lang.String s, java.lang.String end)
static void	**exit**() This is same as exit(int) except that the exit code is zero.
static void	**exit**(int exitCode) 退出“客户端追踪会话”，即终止“追踪行为方法” Exits the BTrace session -- note that the particular client's tracing session exits and not the observed/traced program! After exit call, the trace action method terminates immediately and no other probe action method (of that client) will be called after that.
static double	**exp**(double a) Returns Euler's number *e* raised to the power of a double value.
static java.lang.reflect.Field 返回指定类中的特定字段对象	**field**(java.lang.Class clazz, java.lang.String name) Returns a Field object that reflects the specified declared field of the class or interface represented by the given Class object.
static java.lang.reflect.Field	**field**(java.lang.Class clazz, java.lang.String name, boolean throwException)

	Returns a Field object that reflects the specified declared field of the class or interface represented by the given Class object.
static java.lang.reflect.Field	field(java.lang.String clazz, java.lang.String name) Returns a Field object that reflects the specified declared field of the class or interface represented by the given Class object.
static java.lang.reflect.Field	field(java.lang.String clazz, java.lang.String name, boolean throwException) Returns a Field object that reflects the specified declared field of the class or interface represented by the given Class object.
static long	finalizationCount() Returns the approximate number of objects for which finalization is pending.
static long	freeMemory()　返回“JVM中的空闲内存大小” Returns the amount of free memory in the Java Virtual Machine.
static void	gc() Runs the garbage collector.
static int	get(java.util.concurrent.atomic.AtomicInteger ai) Gets the current value of the given AtomicInteger.
static long	get(java.util.concurrent.atomic.AtomicLong al) Gets the current value the given AtomicLong.
static java.lang.Object	get(java.lang.reflect.Field field) Gets the value of a static reference field.
static java.lang.Object	get(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance reference field.　获取“给定实例引用字段”的值
static <K,V> V	get(java.util.Map<K,V> map, java.lang.Object key)
static int	getAndAdd(java.util.concurrent.atomic.AtomicInteger ai, int delta) Atomically adds the given value to the current value.
static long	getAndAdd(java.util.concurrent.atomic.AtomicLong al, long delta) Atomically adds the given value to the current value.
static int	getAndDecrement(java.util.concurrent.atomic.AtomicInteger ai) Atomically decrements by one the current value of given AtomicInteger.
static long	getAndDecrement(java.util.concurrent.atomic.AtomicLong al) Atomically decrements by one the current value.
static int	getAndIncrement(java.util.concurrent.atomic.AtomicInteger ai) Atomically increments by one the current value of given AtomicInteger.
static long	getAndIncrement(java.util.concurrent.atomic.AtomicLong al) Atomically increments by one the current value.
static int	getAndSet(java.util.concurrent.atomic.AtomicInteger ai, int newValue) Atomically sets to the given value and returns the old value.
static long	getAndSet(java.util.concurrent.atomic.AtomicLong al, long newValue) Atomically sets to the given value and returns the old value.
static boolean	getBoolean(java.lang.reflect.Field field) Gets the value of a static boolean field.
static boolean	getBoolean(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance boolean field.
static byte	getByte(java.lang.reflect.Field field) Gets the value of a static byte field.
static byte	getByte(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance byte field.
static char	getChar(java.lang.reflect.Field field) Gets the value of a static char field.
static char	getChar(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance char field.

static java.lang.Class	**getComponentType**(java.lang.Class clazz) returns component type of an array Class.
static double	**getDouble**(java.lang.reflect.Field field) Gets the value of a static double field.
static double	**getDouble**(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance double field.
static java.util.Map<java.lang.String,java.lang.String>	**getenv**()　返回“当前系统环境的一个不可修改的映射表视图” Returns an unmodifiable string map view of the current system environment.
static java.lang.String	**getenv**(java.lang.String name) Gets the value of the specified environment variable.
static float	**getFloat**(java.lang.reflect.Field field) Gets the value of a static float field.
static float	**getFloat**(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance float field.
static int	**getInt**(java.lang.reflect.Field field) Gets the value of a static int field.
static int	**getInt**(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance int field.
static long	**getLong**(java.lang.reflect.Field field) Gets the value of a static long field.
static long	**getLong**(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance long field.
static int	**getpid**() Returns the process id of the currently BTrace'd process.
static short	**getShort**(java.lang.reflect.Field field) Gets the value of a static short field.
static short	**getShort**(java.lang.reflect.Field field, java.lang.Object obj) Gets the value of an instance short field.
static java.lang.Class	**getSuperclass**(java.lang.Class clazz) Returns the Class representing the superclass of the entity (class, interface, primitive type or void) represented by the given Class.
static long	**getTotalGcTime**()　返回“垃圾回收所消耗的总时长” Returns the total amount of time spent in GarbageCollection up to this point since the application was started.
static int	**hash**(java.lang.Object obj) Returns a hash code value for the object.
static java.lang.management.MemoryUsage	**heapUsage**()　返回“堆内存的使用情况” Returns heap memory usage
static boolean	**holdsLock**(java.lang.Object obj) Returns true if and only if the current thread holds the monitor lock on the specified object.
static int	**identityHashCode**(java.lang.Object obj) Returns the same hash code for the given object as would be returned by the default method hashCode(), whether or not the given object's class overrides hashCode().
static java.lang.String	**identityStr**(java.lang.Object obj) Returns identity string of the form class-name@identity-hash
static int	**incrementAndGet**(java.util.concurrent.atomic.AtomicInteger ai) Atomically increments by one the current value of given AtomicInteger.
static long	**incrementAndGet**(java.util.concurrent.atomic.AtomicLong al) Atomically increments by one the current value.
static int	**indexOf**(java.lang.String str1, java.lang.String str2)
static long	**init**(java.lang.management.MemoryUsage mu) Returns the amount of memory in bytes that the Java virtual machine initially requests from the operating system for memory management.
static boolean	**isArray**(java.lang.Class clazz) Determines if the given Class object represents an array class.

static boolean	**isAssignableFrom**(java.lang.Class<?> a, java.lang.Class<?> b) Determines if the class or interface represented by the first Class object is either the same as, or is a superclass or superinterface of, the class or interface represented by the second Class parameter.
static boolean	**isBootClassPathSupported**() Tests if the Java virtual machine supports the boot class path mechanism used by the bootstrap class loader to search for class files.
static <E> boolean	**isEmpty**(java.util.Collection<E> coll)
static <K,V> boolean	**isEmpty**(java.util.Map<K,V> map)
static boolean	**isInfinite**(double d) Returns true if the specified number is infinitely large in magnitude, false otherwise.
static boolean	**isInfinite**(float f) Returns true if the specified number is infinitely large in magnitude, false otherwise.
static boolean	**isInstance**(java.lang.Class clazz, java.lang.Object obj) Determines if the specified Object is assignment-compatible with the object represented by the specified Class.
static boolean	**isInterface**(java.lang.Class clazz) Determines if the specified Class object represents an interface type.
static boolean	**isInteruppted**() Tests whether this thread has been interrupted.
static boolean	**isNaN**(double d) Returns true if the specified number is a Not-a-Number (NaN) value, false otherwise.
static boolean	**isNaN**(float f) Returns true if the specified number is a Not-a-Number (NaN) value, false otherwise.
static boolean	**isPrimitive**(java.lang.Class clazz) Returns whether the given Class represent primitive type or not.
static void	**jstack**()　　**打印“当前线程的Java调用栈信息”** Prints the java stack trace of the current thread.
static void	**jstack**(int numFrames) Prints the java stack trace of the current thread.
static void	**jstack**(java.lang.Throwable exception)　**打印“给定异常对象的调用栈信息”** Prints the stack trace of the given exception object.
static void	**jstack**(java.lang.Throwable exception, int numFrames) Prints the stack trace of the given exception object.
static void	**jstackAll**()　　**打印“所有Java线程的调用栈信息”** Prints Java stack traces of all the Java threads.
static void	**jstackAll**(int numFrames) Prints Java stack traces of all the Java threads.
static java.lang.String	**jstackAllStr**() Returns the stack traces of all Java threads as a String.
static java.lang.String	**jstackAllStr**(int numFrames) Returns atmost given number of frames in stack traces of all threads as a String.
static java.lang.String	**jstackStr**() Returns the stack trace of current thread as a String.
static java.lang.String	**jstackStr**(int numFrames) Returns the stack trace of the current thread as a String but includes atmost the given number of frames.
static java.lang.String	**jstackStr**(java.lang.Throwable exception) Returns the stack trace of given exception object as a String.
static java.lang.String	**jstackStr**(java.lang.Throwable exception, int numFrames) Returns stack trace of given exception object as a String.
static int	**lastIndexOf**(java.lang.String str1, java.lang.String str2)

static void	**lazySet**(java.util.concurrent.atomic.AtomicInteger ai, int newValue) Eventually sets to the given value to the given AtomicInteger.
static void	**lazySet**(java.util.concurrent.atomic.AtomicLong al, long newValue) Eventually sets to the given value to the given AtomicLong.
static int	**length**(java.lang.String str) Returns the length of the given string.
static java.lang.String	**libraryPath**() Returns the Java library path.
static java.lang.ClassLoader	**loader**(java.lang.Class clazz) Returns the class loader for the given class.
static double	**log**(double a) Returns the natural logarithm (base *e*) of a double value.
static double	**log10**(double a) Returns the base 10 logarithm of a double value.
static boolean	**matches**(java.util.regex.Pattern regex, java.lang.String input) Matches the given (precompiled) regular expression and attempts to match the given input against it.
static boolean	**matches**(java.lang.String regex, java.lang.String input) Compiles the given regular expression and attempts to match the given input against it.
static long	**max**(java.lang.management.MemoryUsage mu) Returns the maximum amount of memory in bytes that can be used for memory management.
static long	**maxMemory**()　返回“JVM可使用的最大内存容量” Returns the maximum amount of memory that the Java virtual machine will attempt to use.
static java.lang.String	**name**(java.lang.Class clazz)　返回“给定类对象的名称” Returns the name of the given Class object.
static java.lang.String	**name**(java.lang.reflect.Field field) Returns the name of the Field object.
static java.lang.String	**name**(java.lang.Thread thread) Returns the name of the given thread.　返回“给定线程对象的名称”
static **Aggregation**	**newAggregation**(**AggregationFunction** type) Creates a new aggregation based on the given aggregation function type.
static **AggregationKey**	**newAggregationKey**(java.lang.Object element1) Creates a grouping aggregation key with the provided value.
static **AggregationKey**	**newAggregationKey**(java.lang.Object element1, java.lang.Object element2) Creates a composite grouping aggregation key with the provided values.
static **AggregationKey**	**newAggregationKey**(java.lang.Object element1, java.lang.Object element2, java.lang.Object element3) Creates a composite grouping aggregation key with the provided values.
static **AggregationKey**	**newAggregationKey**(java.lang.Object element1, java.lang.Object element2, java.lang.Object element3, java.lang.Object element4) Creates a composite grouping aggregation key with the provided values.
static java.util.concurrent.atomic.AtomicInteger	**newAtomicInteger**(int initialValue) Creates a new AtomicInteger with the given initial value.
static java.util.concurrent.atomic.AtomicLong	**newAtomicLong**(long initialValue) Creates a new AtomicLong with the given initial value.
static <V> java.util.Deque<V>	**newDeque**()
static <K,V> java.util.Map<K,V>	**newHashMap**() Operating on maps
static <K,V> java.util.Map<K,V>	**newWeakMap**()
static java.lang.management.MemoryUsage	**nonHeapUsage**()　返回“非堆内存的使用情况” Returns non-heap memory usage

static java.lang.ClassLoader	**parentLoader**(java.lang.ClassLoader loader) Returns the parent class loader of the given loader.
static boolean	**parseBoolean**(java.lang.String s) Parses the string argument as a boolean.
static byte	**parseByte**(java.lang.String s) Parses the string argument as a signed decimal byte.
static double	**parseDouble**(java.lang.String s) Returns a new double initialized to the value represented by the specified String, as performed by the valueOf methcod of class Double.
static float	**parseFloat**(java.lang.String s) Returns a new float initialized to the value represented by the specified String, as performed by the valueOf method of class Float.
static int	**parseInt**(java.lang.String s) Parses the string argument as a signed decimal integer.
static long	**parseLong**(java.lang.String s) Parses the string argument as a signed decimal long.
static short	**parseShort**(java.lang.String s) Parses the string argument as a signed decimal short.
static java.util.regex.Pattern	**pattern**(java.lang.String regex) This is synonym for "regexp".
static java.util.regex.Pattern	**pattern**(java.lang.String regex, int flags) This is synonym for "regexp".
static long	**peakThreadCount**() Returns the peak live thread count since the Java virtual machine started or peak was reset.
static <V> V	**peek**(java.util.Deque<V> queue)
static <V> V	**peekFirst**(java.util.Deque<V> queue)
static <V> V	**peekLast**(java.util.Deque<V> queue)
static long	**perfInt**(java.lang.String name) accessing jvmstat (perf) int counter
static long	**perfLong**(java.lang.String name) accessing jvmstat (perf) long counter
static java.lang.String	**perfString**(java.lang.String name) accessing jvmstat (perf) String counter
static <V> V	**poll**(java.util.Deque<V> queue)
static void	**print**(boolean b) Prints a boolean value.
static void	**print**(char c) Prints a character.
static void	**print**(double d) Prints a double-precision floating-point number.
static void	**print**(float f) Prints a floating-point number.
static void	**print**(int i) Prints an integer.
static void	**print**(long l) Prints a long integer.
static void	**print**(java.lang.Object obj)
static void	**printAggregation**(java.lang.String name, **Aggregation** aggregation) Prints the aggregation.
static void	**printAggregation**(java.lang.String name, **Aggregation** aggregation, java.lang.String format) Prints aggregation using the provided format
static void	**printArray**(java.lang.Object[] array) Prints the elements of the given array as comma separated line bounded by '[' and

打印“给定数组”的元素

	']'. **以“名称-值”对形式打印“给定对象”的所有实例字段**
static void	**printEnv**() Prints all system environment values.
static void	**printFields**(java.lang.Object obj) Print all instance fields of an object as name-value pairs.
static void	**printFields**(java.lang.Object obj, boolean classNamePrefix) Print all instance fields of an object as name-value pairs.
static void	**println**() Terminates the current line by writing the line separator string.
static void	**println**(boolean b) Prints a boolean and then terminate the line.
static void	**println**(char c) Prints a character and then terminate the line.
static void	**println**(double d) Prints a double and then terminate the line.
static void	**println**(float f) Prints a float and then terminate the line.
static void	**println**(int i) Prints an integer and then terminate the line.
static void	**println**(long l) Prints a long and then terminate the line.
static void	**println**(java.lang.Object obj) Prints the given object and then prints a newline
static void	**printMap**(java.util.Map map) Prints the given Map.
static void	**printNumber**(java.lang.String name, java.lang.Number value) Prints a number.
static void	**printNumberMap**(java.lang.String name, java.util.Map<java.lang.String,? extends java.lang.Number> data) Prints the given Map.
static void	**printProperties**() Prints all Sys properties.
static void	**printStaticFields**(java.lang.Class clazz) Print all static fields of the class as name-value pairs.
static void	**printStaticFields**(java.lang.Class clazz, boolean classNamePrefix) Print all static fields of the class as name-value pairs.
static void	**printStringMap**(java.lang.String name, java.util.Map<java.lang.String,java.lang.String> data) Prints the given Map.
static void	**printVmArguments**() Prints VM input arguments list.
static java.lang.Class	**probeClass**() **Deprecated.** Since 1.1. Use **ProbeClassName** and **Self** annotations instead
static int	**probeLine**() Returns the currently probed source line number (if available).
static java.lang.String	**probeMethod**() **Deprecated.** Since 1.1. Use **ProbeMethodName** annotation instead
static java.util.Properties	**properties**() Returns all Sys properties.
static java.lang.String	**property**(java.lang.String key) Gets the system property indicated by the specified key.
static <V> void	**push**(java.util.Deque<V> queue, V value)

static <K,V> V	**put**(java.util.Map<K,V> map, K key, V value)
static double	**random**() Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
static java.util.regex.Pattern	**regexp**(java.lang.String regex) Compiles the given regular expression into a pattern.
static java.util.regex.Pattern	**regexp**(java.lang.String regex, int flags) Compiles the given regular expression into a pattern with the given flags.
static <K,V> V	**remove**(java.util.Map<K,V> map, java.lang.Object key)
static <V> V	**removeFirst**(java.util.Deque<V> queue)
static <V> V	**removeLast**(java.util.Deque<V> queue)
static void	**runFinalization**() Runs the finalization methods of any objects pending finalization.
static void	**serialize**(java.io.Serializable obj, java.lang.String fileName) Serialize a given object into the given file.
static void	**set**(java.util.concurrent.atomic.AtomicInteger ai, int newValue) Sets to the given value to the given AtomicInteger.
static void	**set**(java.util.concurrent.atomic.AtomicLong al, long newValue) Sets to the given value.
static <E> int	**size**(java.util.Collection<E> coll)
static <K,V> int	**size**(java.util.Map<K,V> map)
static long	**sizeof**(java.lang.Object objectToSize) Returns an implementation-specific approximation of the amount of storage consumed by the specified object.
static java.lang.ref.SoftReference	**softRef**(java.lang.Object obj) Creates and returns a soft reference to the given object.
static void	**speculate**(int id) Sets current speculative buffer id.
static int	**speculation**() Returns an identifier for a new speculative buffer.
static boolean	**startsWith**(java.lang.String s, java.lang.String start)
static java.lang.String	**str**(boolean b) Returns a String object representing the specified boolean.
static java.lang.String	**str**(char c) Returns a String object representing the specified char.
static java.lang.String	**str**(double d) Returns a string representation of the double argument.
static java.lang.String	**str**(float f) Returns a string representation of the float argument.
static java.lang.String	**str**(int i) Returns a String object representing the specified integer.
static java.lang.String	**str**(long l) Returns a String object representing the specified long.
static java.lang.String	**str**(java.lang.Object obj) Returns a string representation of the object. **返回表示“给定对象”的字符串**
static java.lang.String	**strcat**(java.lang.String str1, java.lang.String str2) This is synonym to "concat".
static int	**strcmp**(java.lang.String str1, java.lang.String str2) This is synonym to "compareTo" method.
static int	**stricmp**(java.lang.String str1, java.lang.String str2) This is synonym to "compareToIgnoreCase".
static int	**strlen**(java.lang.String str) This is synonym for "length".
static int	**strstr**(java.lang.String str1, java.lang.String str2) Find String within String

static java.lang.String	**substr**(java.lang.String str, int start)
static java.lang.String	**substr**(java.lang.String str, int start, int length) Substring
static long	**threadCount**() Returns the current number of live threads including both daemon and non-daemon threads.
static long	**threadId**(java.lang.Thread thread) Returns the identifier of the given Thread.
static java.lang.Thread.State	**threadState**(java.lang.Thread thread) Returns the state of the given thread.
static long	**timeMillis**() Returns the current time in milliseconds.
static long	**timeNanos**() Returns the current value of the most precise available system timer, in nanoseconds.
static java.lang.String	**timestamp**() Generates a string getTimestamp (current date&time) in the default system format
static java.lang.String	**timestamp**(java.lang.String format) Generates a string getTimestamp (current date&time)
static java.lang.String	**toHexString**(int i) Returns a string representation of the integer argument as an unsigned integer in base 16.
static java.lang.String	**toHexString**(long l) Returns a string representation of the long argument as an unsigned integer in base 16.
static long	**totalMemory**()　**返回“ JVM的内存总容量 ”** Returns the total amount of memory in the Java virtual machine.
static long	**totalStartedThreadCount**() Returns the total number of threads created and also started since the Java virtual machine started.
static java.lang.String	**toXML**(java.lang.Object obj) Creates an XML document to persist the tree of the all transitively reachable objects from given "root" object.
static void	**truncateAggregation**(**Aggregation** aggregation, int count) Removes all aggregated values from the aggregation except for the largest or smallest abs(count) elements.
static java.lang.Class	**type**(java.lang.reflect.Field field) Returns the type of the Field object.
static boolean	**unbox**(java.lang.Boolean b) Returns the value of the given Boolean object as a boolean primitive.
static byte	**unbox**(java.lang.Byte b) Returns the value of the specified Byte as a byte.
static char	**unbox**(java.lang.Character ch) Returns the value of the given Character object as a char primitive.
static double	**unbox**(java.lang.Double d) Returns the double value represented by the specified Double.
static float	**unbox**(java.lang.Float f) Returns the float value represented by the specified Float.
static int	**unbox**(java.lang.Integer i) Returns the value of represented by Integer.
static long	**unbox**(java.lang.Long l) Returns the long value represented by the specified Long.
static short	**unbox**(java.lang.Short s) Returns the short value represented by Short.
static long	**used**(java.lang.management.MemoryUsage mu) Returns the amount of used memory in bytes.

static java.util.List<java.lang.String>	**vmArguments**() Returns the input arguments passed to the Java virtual machine which does not include the arguments to the main method.
static long	**vmStartTime**() Returns the start time of the Java virtual machine in milliseconds.
static long	**vmUptime**() Returns the uptime of the Java virtual machine in milliseconds.
static java.lang.String	**vmVersion**() Returns the Java virtual machine implementation version.
static boolean	**weakCompareAndSet**(java.util.concurrent.atomic.AtomicInteger ai, int expect, int update) Atomically sets the value to the given updated value if the current value == the expected value.
static boolean	**weakCompareAndSet**(java.util.concurrent.atomic.AtomicLong al, long expect, long update) Atomically sets the value to the given updated value if the current value == the expected value.
static java.lang.ref.WeakReference	**weakRef**(java.lang.Object obj) Creates and returns a weak reference to the given object.
static void	**writeDOT**(java.lang.Object obj, java.lang.String fileName) Writes a .dot document to persist the tree of the all the transitively reachable objects from the given "root" object.
static void	**writeXML**(java.lang.Object obj, java.lang.String fileName) Writes an XML document to persist the tree of the all the transitively reachable objects from the given "root" object.

Methods inherited from class java.lang.Object

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

Method Detail

isInteruppted

```
public static boolean isInteruppted()
```

Tests whether this thread has been interrupted. The *interrupted status* of the thread is unaffected by this method.

A thread interruption ignored because a thread was not alive at the time of the interrupt will be reflected by this method returning false.

Returns:

true if this thread has been interrupted; false otherwise.

jstack

```
public static void jstack()
```

Prints the java stack trace of the current thread.

jstack

```
public static void jstack(int numFrames)
```

Prints the java stack trace of the current thread. But, atmost given number of frames.

Parameters:

numFrames - number of frames to be printed. When this is negative all frames are printed.

jstackAll

```
public static void jstackAll()
```

Prints Java stack traces of all the Java threads.

jstackAll

```
public static void jstackAll(int numFrames)
```