| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AtomicInteger.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicBoolean.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicIntegerArray.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/atomic/AtomicInteger.html)    [**NO FRAMES**](http://docs.google.com/AtomicInteger.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.util.concurrent.atomic**

Class AtomicInteger

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.lang.Number](http://docs.google.com/java/lang/Number.html)  
 **java.util.concurrent.atomic.AtomicInteger**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **AtomicInteger**extends [Number](http://docs.google.com/java/lang/Number.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

An int value that may be updated atomically. See the [java.util.concurrent.atomic](http://docs.google.com/java/util/concurrent/atomic/package-summary.html) package specification for description of the properties of atomic variables. An AtomicInteger is used in applications such as atomically incremented counters, and cannot be used as a replacement for an [Integer](http://docs.google.com/java/lang/Integer.html). However, this class does extend Number to allow uniform access by tools and utilities that deal with numerically-based classes.

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.util.concurrent.atomic.AtomicInteger)

| **Constructor Summary** | |
| --- | --- |
| [**AtomicInteger**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#AtomicInteger())()            Creates a new AtomicInteger with initial value 0. |
| [**AtomicInteger**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#AtomicInteger(int))(int initialValue)            Creates a new AtomicInteger with the given initial value. |

| **Method Summary** | |
| --- | --- |
| int | [**addAndGet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#addAndGet(int))(int delta)            Atomically adds the given value to the current value. |
| boolean | [**compareAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#compareAndSet(int,%20int))(int expect, int update)            Atomically sets the value to the given updated value if the current value == the expected value. |
| int | [**decrementAndGet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#decrementAndGet())()            Atomically decrements by one the current value. |
| double | [**doubleValue**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#doubleValue())()            Returns the value of the specified number as a double. |
| float | [**floatValue**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#floatValue())()            Returns the value of the specified number as a float. |
| int | [**get**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#get())()            Gets the current value. |
| int | [**getAndAdd**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#getAndAdd(int))(int delta)            Atomically adds the given value to the current value. |
| int | [**getAndDecrement**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#getAndDecrement())()            Atomically decrements by one the current value. |
| int | [**getAndIncrement**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#getAndIncrement())()            Atomically increments by one the current value. |
| int | [**getAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#getAndSet(int))(int newValue)            Atomically sets to the given value and returns the old value. |
| int | [**incrementAndGet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#incrementAndGet())()            Atomically increments by one the current value. |
| int | [**intValue**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#intValue())()            Returns the value of the specified number as an int. |
| void | [**lazySet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#lazySet(int))(int newValue)            Eventually sets to the given value. |
| long | [**longValue**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#longValue())()            Returns the value of the specified number as a long. |
| void | [**set**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#set(int))(int newValue)            Sets to the given value. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#toString())()            Returns the String representation of the current value. |
| boolean | [**weakCompareAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicInteger.html#weakCompareAndSet(int,%20int))(int expect, int update)            Atomically sets the value to the given updated value if the current value == the expected value. |

| **Methods inherited from class java.lang.**[**Number**](http://docs.google.com/java/lang/Number.html) |
| --- |
| [byteValue](http://docs.google.com/java/lang/Number.html#byteValue()), [shortValue](http://docs.google.com/java/lang/Number.html#shortValue()) |

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

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

### AtomicInteger

public **AtomicInteger**(int initialValue)

Creates a new AtomicInteger with the given initial value.

**Parameters:**initialValue - the initial value

### AtomicInteger

public **AtomicInteger**()

Creates a new AtomicInteger with initial value 0.

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

### get

public final int **get**()

Gets the current value.

**Returns:**the current value

### set

public final void **set**(int newValue)

Sets to the given value.

**Parameters:**newValue - the new value

### lazySet

public final void **lazySet**(int newValue)

Eventually sets to the given value.

**Parameters:**newValue - the new value**Since:** 1.6

### getAndSet

public final int **getAndSet**(int newValue)

Atomically sets to the given value and returns the old value.

**Parameters:**newValue - the new value **Returns:**the previous value

### compareAndSet

public final boolean **compareAndSet**(int expect,  
 int update)

Atomically sets the value to the given updated value if the current value == the expected value.

**Parameters:**expect - the expected valueupdate - the new value **Returns:**true if successful. False return indicates that the actual value was not equal to the expected value.

### weakCompareAndSet

public final boolean **weakCompareAndSet**(int expect,  
 int update)

Atomically sets the value to the given updated value if the current value == the expected value.

May [fail spuriously](http://docs.google.com/package-summary.html#Spurious) and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

**Parameters:**expect - the expected valueupdate - the new value **Returns:**true if successful.

### getAndIncrement

public final int **getAndIncrement**()

Atomically increments by one the current value.

**Returns:**the previous value

### getAndDecrement

public final int **getAndDecrement**()

Atomically decrements by one the current value.

**Returns:**the previous value

### getAndAdd

public final int **getAndAdd**(int delta)

Atomically adds the given value to the current value.

**Parameters:**delta - the value to add **Returns:**the previous value

### incrementAndGet

public final int **incrementAndGet**()

Atomically increments by one the current value.

**Returns:**the updated value

### decrementAndGet

public final int **decrementAndGet**()

Atomically decrements by one the current value.

**Returns:**the updated value

### addAndGet

public final int **addAndGet**(int delta)

Atomically adds the given value to the current value.

**Parameters:**delta - the value to add **Returns:**the updated value

### toString

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

Returns the String representation of the current value.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the String representation of the current value.

### intValue

public int **intValue**()

**Description copied from class:** [**Number**](http://docs.google.com/java/lang/Number.html#intValue()) Returns the value of the specified number as an int. This may involve rounding or truncation.

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

### longValue

public long **longValue**()

**Description copied from class:** [**Number**](http://docs.google.com/java/lang/Number.html#longValue()) Returns the value of the specified number as a long. This may involve rounding or truncation.

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

### floatValue

public float **floatValue**()

**Description copied from class:** [**Number**](http://docs.google.com/java/lang/Number.html#floatValue()) Returns the value of the specified number as a float. This may involve rounding.

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

### doubleValue

public double **doubleValue**()

**Description copied from class:** [**Number**](http://docs.google.com/java/lang/Number.html#doubleValue()) Returns the value of the specified number as a double. This may involve rounding.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AtomicInteger.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicBoolean.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicIntegerArray.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/atomic/AtomicInteger.html)    [**NO FRAMES**](http://docs.google.com/AtomicInteger.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

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

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

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