

## 1、AtomicInteger类

在java.util.concurrent.atomic包里，多了一批原子处理类。AtomicBoolean、AtomicInteger、AtomicLong、AtomicReference。主要用于在高并发环境下的高效程序处理，来帮助我们简化同步处理。

AtomicInteger一个提供原子操作的Integer的类。在Java语言中，++i和i++操作并不是线程安全的，在使用的时候，不可避免的会用到synchronized关键字。而AtomicInteger则通过一种线程安全的加减操作接口。

先来看看AtomicInteger提供了什么接口：

```
public final int get() //获取当前的值
public final int getAndSet(int newValue)//获取当前的值，并设置新的值
public final int getAndIncrement()//获取当前的值，并自增
public final int getAndDecrement() //获取当前的值，并自减
public final int getAndAdd(int delta) //获取当前的值，并加上预期的值
```

## 2、AtomicLong类

java.util.concurrent.atomic.AtomicLong类提供了可以被原子地读取和写入的底层long值的操作，并且还包含高级原子操作。AtomicLong支持基础long类型变量上的原子操作。它具有获取和设置方法，如在volatile变量上的读取和写入。也就是说，一个集合与同一变量上的任何后续get相关联。原子compareAndSet方法也具有这些内存一致性功能。

### AtomicLong类中的方法

以下是AtomicLong类中可用的重要方法的列表。

序号	方法	描述
1	public long addAndGet(long delta)	将给定值原子地添加到当前值。
2	public boolean compareAndSet(long expect, long update)	如果当前值与预期值相同，则将该值原子设置为给定的更新值。
3	public long decrementAndGet()	当前值原子减1。
4	public double doubleValue()	以double形式返回指定数字的值。
5	public float floatValue()	以float形式返回指定数字的值。
6	public long get()	获取当前值。
7	public long getAndAdd(long delta)	自动将给定值添加到当前值。
8	public long getAndDecrement()	当前值原子减1。
9	public long getAndIncrement()	当前值原子增加1。
10	public long getAndSet(long newValue)	将原子设置为给定值并返回旧值。

11	public long incrementAndGet()	原子上增加一个当前值。
12	public int intValue()	以int形式返回指定数字的值。
13	public void lazySet(long newValue)	最终设定为给定值。
14	public long longValue()	返回指定数字的值为long类型。
15	public void set(long newValue)	设置为给定值。
16	public String toString()	返回当前值的String表示形式。
17	public boolean weakCompareAndSet(long expect, long update)	如果当前值与预期值相同，则将该值原子设置为给定的更新值。

### 3、AtomicBoolean类

java.util.concurrent.atomic.AtomicBoolean类提供了可以原子读取和写入的底层布尔值的操作，并且还包含高级原子操作。AtomicBoolean支持基础布尔变量上的原子操作。它具有获取和设置方法，如在volatile变量上的读取和写入。也就是说，一个集合与同一变量上的任何后续get相关联。原子compareAndSet方法也具有这些内存一致性功能。

#### AtomicBoolean类中的方法

以下是AtomicBoolean类中可用的重要方法的列表。

序号	方法	描述
1	public boolean compareAndSet(boolean expect, boolean update)	如果当前值==期望值，则将该值原子设置为给定的更新值。
2	public boolean get()	返回当前值。
3	public boolean getAndSet(boolean newValue)	将原子设置为给定值并返回上一个值。
4	public void lazySet(boolean newValue)	最终设定为给定值。
5	public void set(boolean newValue)	无条件地设置为给定的值。
6	public String toString()	返回当前值的String表示形式。
7	public boolean weakCompareAndSet(boolean expect, boolean update)	如果当前值==期望值，则将该值原子设置为给定的更新值。

### 4、AtomicReference类

java.util.concurrent.atomic.AtomicReference类提供了可以原子读取和写入的底层对象引用的操作，还包含高级原子操作。AtomicReference支持对底层对象引用变量的原子操

作。它具有获取和设置方法，如在易变的变量上的读取和写入。也就是说，一个集合与同一变量上的任何后续get相关联。原子compareAndSet方法也具有这些内存一致性功能。

### AtomicReference类的方法

以下是AtomicReference类中可用的重要方法的列表。

序号	方法	描述
1	public boolean compareAndSet(V expect, V update)	如果当前值==期望值，则将该值原子设置为给定的更新值。
2	public boolean get()	返回当前值。
3	public boolean getAndSet(V newValue)	将原子设置为给定值并返回上一个值。
4	public void lazySet(V newValue)	最终设定为给定值。
5	public void set(V newValue)	无条件地设置为给定的值。
6	public String toString()	
7	public boolean weakCompareAndSet(V expect, V update)	如果当前值==期望值，则将该值原子设置为给定的更新值。

### 5、AtomicIntegerArray类

java.util.concurrent.atomic.AtomicIntegerArray类提供了可以以原子方式读取和写入的底层int数组的操作，还包含高级原子操作。AtomicIntegerArray支持对底层int数组变量的原子操作。它具有获取和设置方法，如在变量上的读取和写入。也就是说，一个集合与同一变量上的任何后续get相关联。原子compareAndSet方法也具有这些内存一致性功能。

序号	方法	描述
1	public int addAndGet(int i, int delta)	原子地将给定的值添加到索引i的元素。
2	public boolean compareAndSet(int i, int expect, int update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。
3	public int decrementAndGet(int i)	索引i处的元素原子并自减1。
4	public int get(int i)	获取位置i的当前值。
5	public int getAndAdd(int i, int delta)	原子地将给定的值添加到索引i的元素。
6	public int getAndDecrement(int i)	索引i处的元素原子并自减1，并返回旧值。
7	public int getAndIncrement(int i)	将位置i处的元素原子设置为给定值，并返回旧值。
8	public int getAndSet(int i, int newValue)	将位置i处的元素原子设置为给定值，并返回旧值。

9	public int incrementAndGet(int i)	在索引i处以原子方式自增元素。
10	public void lazySet(int i, int newValue)	最终将位置i处的元素设置为给定值。
11	public int length()	返回数组的长度。
12	public void set(int i, int newValue)	将位置i处的元素设置为给定值。
13	public String toString()	返回数组的当前值的String表示形式。
14	public boolean weakCompareAndSet(int i, int expect, int update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。

## 6、AtomicLongArray类

java.util.concurrent.atomic.AtomicLongArray类提供了可以原子读取和写入的底层long类型数组的操作，并且还包含高级原子操作。AtomicLongArray支持对基础long类型数组变量的原子操作。它具有获取和设置方法，如在变量上的读取和写入。也就是说，一个集合与同一变量上的任何后续获取相关联。原子compareAndSet方法也具有这些内存一致性功能。

### AtomicLongArray类中方法

以下是AtomicLongArray类中可用的重要方法的列表。

序号	方法	描述
1	public long addAndGet(int i, long delta)	原子地将给定的值添加到索引i的元素。
2	public boolean compareAndSet(int i, long expect, long update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。
3	public long decrementAndGet(int i)	索引i处的元素原子并自减1。
4	public long get(int i)	获取位置i的当前值。
5	public long getAndAdd(int i, long delta)	原子地将给定的值添加到索引i的元素。
6	public long getAndDecrement(int i)	索引i处的元素原子并自减1，并返回旧值。
7	public long getAndIncrement(int i)	将位置i处的元素原子设置为给定值，并返回旧值。
8	public long getAndSet(int i, long newValue)	将位置i处的元素原子设置为给定值，并返回旧值。
9	public long incrementAndGet(long i)	在索引i处以原子方式自增元素。
10	public void lazySet(int i, long newValue)	最终将位置i处的元素设置为给定值。
11	public int length()	返回数组的长度。
12	public void set(int i, long newValue)	将位置i处的元素设置为给定值。

12	public void set(int i, long new value)	将位置i处的元素设置为给定值。
13	public String toString()	返回数组的当前值的String表示形式。
14	public boolean weakCompareAndSet(int i, int expect, long update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。

## 7、AtomicReferenceArray类

java.util.concurrent.atomic.AtomicReferenceArray类提供了可以原子读取和写入的底层引用数组的操作，并且还包含高级原子操作。 AtomicReferenceArray支持对底层引用数组变量的原子操作。 它具有获取和设置方法，如在变量上的读取和写入。 也就是说，一个集合与同一变量上的任何后续获取相关联。 原子compareAndSet方法也具有这些内存一致性功能。

### AtomicReferenceArray方法

以下是AtomicReferenceArray类中可用的重要方法的列表。

序列	方法	描述
1	public boolean compareAndSet(int i, E expect, E update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。
2	public E get(int i)	获取位置i的当前值。
3	public E getAndSet(int i, E newValue)	将位置i处的元素原子设置为给定值，并返回旧值。
4	public void lazySet(int i, E newValue)	最终将位置i处的元素设置为给定值。
5	public int length()	返回数组的长度。
6	public void set(int i, E newValue)	将位置i处的元素设置为给定值。
7	public String toString()	返回数组的当前值的String表示形式。
8	public boolean weakCompareAndSet(int i, E expect, E update)	如果当前值==期望值，则将位置i处的元素原子设置为给定的更新值。