

As we all know, the Object method in Java is used very frequently. All classes in Java have a common source, that is the Object class, and subclasses will inherit all the public methods in the Object class, so let’s take a look at these 8 methods today!

**Method 1: getClass**

As you can see, for the final method, the runtime class object that gets the object, the class object is the object that describes the class to which the object belongs. I believe everyone is familiar with the use of this method. In general, it will be used in conjunction with the Java reflection mechanism.  
*renders as follows:*

public final native Class<?> getClass();

**Method 2: equals**

For comparing two objects, returns true if the two object references refer to the same object, false otherwise.  
*renders as follows:*

public boolean equals(Object obj) { return (this == obj);}

A picture containing grass, outdoor, sky, field

Description automatically generated

Photo by [Dim Hou](https://unsplash.com/@dimhou?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral" \t "_blank)

**Method 3: notify**

This method is relatively simple, it is mainly used to wake up a thread waiting on the object.  
*renders as follows:*

public final native void notify();

**Method 4: hashCode**

For this method, I believe many people have also used it, it is mainly used to obtain the hash value of the object. In Object, this method returns the heap memory address of the object by default.  
*renders as follows:*

public native int hashCode();



Photo by [Chang Qing](https://unsplash.com/@lee0201?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral" \t "_blank)

**Method 5: clone**

This method is a shallow copy of the implementation object. This method can only be called if the Cloneable interface is implemented, otherwise a CloneNotSupportedException exception will be thrown, so this is actually a protection method.  
*renders as follows:*

protected native Object clone() throws CloneNotSupportedException;

**Method 6: wait**

This method is to make the current thread wait for the lock of the object, the current thread must be the owner of the object, that is, the lock with the object  
*renders as follows:*

public final native void wait(long timeout) throws InterruptedException;



Photo by [Adriel Kloppenburg](https://unsplash.com/@adriel?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral" \t "_blank)

**Method 7: notify**

This method is also relatively clear and simple. We can know from a general understanding that it is mainly used to wake up a thread waiting on the object.  
*renders as follows:*

public final native void notify();

**Method 8: finalize**

It is mainly used to be called again during GC. If we implement this method, the object may be resurrected again in this method to avoid being reclaimed by GC. This needs to be distinguished from clone  
*renders as follows:*

protected void finalize() throws Throwable { }