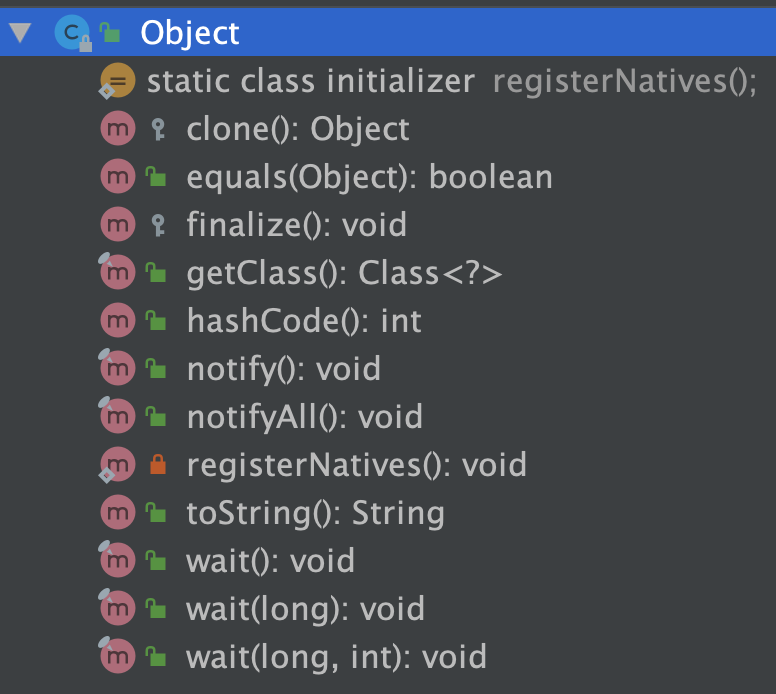
Java语言是一种单继承结构语言，Java中所有的类都有一个共同的祖先。这个祖先就是Object类。如果一个雷没有用extends明确指出继承于某个类，那么它默认继承Object类。Object的方法我们在平时基本都会用到。

#### 分析

Object类是Java中所有类的基类。位于java.lang包中，一共有13个方法。



#### 1、Object()

Object类的构造方法。

#### 2、registerNatives()

private static native void registerNatives();

static {  
 *registerNatives*();  
}

为了使JVM发现本机功能，他们被一定的方式命名。例如，对于java.lang.Object.registerNatives，对应的C函数命名为Java\_java\_lang\_Object. registerNatives。

通过使用registerNatives（或者更准确地说，JNI函数RegisterNatives），可以命名任何你想要你的C函数。

#### 3、clone()

protected native Object clone() throws CloneNotSupportedException;

clone函数的用途是用来另存一个当前存在的对象，只有实现了Cloneable接口才可以调用该方法，否则抛出CloneNotSupportedException异常。（注意：回答这里时可能会引出设计模式的提问）

#### 4、getClass()

public final native Class<?> getClass();

final修饰，用于获得运行时的类型。该方法返回的是此Object对象的类对象向/运行时类对象Class。效果与Object.class相同。（注意：回答这里时可能会引出类加载，反射等知识点的提问）

#### 5、equals()

equals用来比较两个对象的内容是否相等。默认情况下（继承自Object类），equals和==是一样的，除非被覆写（override）了。（注意：这里可能引出更常问的“equals与==的区别”及hashmap实现原理的提问）

#### 6、hashCode()

public native int hashCode();

该方法用来返回其所在对象的物理地址（哈希码值），常会和equals方法同时重写，确保相等的两个对象拥有相等的hashCode。（同样，可能引出hashmap实现原理的提问）

#### 7、toString()

public String toString() {

return getClass().getName() + "@" + Integer.*toHexString*(hashCode());  
}

返回该对象的字符串表示。

#### 8、wait()

public final void wait() throws InterruptedException {

wait(0);  
}

导致当前的线程等待，直到其他线程调用此对象的notify()方法或notifyAll()方法。（引出线程通信及“wait和sleep的区别”的提问）

#### 9、wait(long timeout)

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

导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法，或者超过指定的时间量。（引出线程通信及“wait和sleep的区别”的提问）

#### 10、wait(long timeout, int nanos)

public final void wait(long timeout, int nanos) throws InterruptedException

{  
 if (timeout < 0) {  
 throw new IllegalArgumentException("timeout value is negative");  
 }  
  
 if (nanos < 0 || nanos > 999999) {  
 throw new IllegalArgumentException(  
 "nanosecond timeout value out of range");  
 }  
  
 if (nanos > 0) {  
 timeout++;  
 }  
  
 wait(timeout);  
}

导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法，或者其他某个线程中断当前线程，或者已超过某个实际时间量。（引出线程通信及“wait和sleep的区别”的提问）

#### 11、notify()

public final native void notify();

唤醒再次对象监视器上等待的单个线程。（引出线程通信的提问）

#### 12、notifyAll()

public final native void notifyAll();

唤醒在此对象监视器上等待的所有线程。（引出线程通信的提问）

#### 13、finalize()

public final native Class<?> getClass();

当垃圾回收器确定不存在对该对象的更多引用时，由对象的垃圾回收器调用此方法。（非重点，但小心引出垃圾回收器的提问）

#### 引申常见问题

equals() 与 == 的区别是什么？

hashCode() 和 equals() 之间有什么联系？

wait()方法与sleep()方法的区别

为什么重写了equals就必须重写hashCode？

HashMap的实现原理？

谈谈类加载机制？