面试题大全【yjbo：都是网上的总结的】

1. <http://finalshares.com/read-7354>
2. <http://geek.csdn.net/news/detail/57419>

3.2016年最全的Android面试考题——链接：

[http://pan.baidu.com/s/1qX4G20c](http://pan.baidu.com/s/1qX4G20c" \t "http://finalshares.com/_blank)  密码：exwh

1. <http://wenku.baidu.com/link?url=Kqd9gcaOfRtjZ1DZfVnkYJCoVz5Bdfq2EISTSkFfYN0I5t77luhFxDsX-sbYqOtaEfNFCyWKH50LFCwBvf0Bn7Sa1BxgTmyt-hc7jO7SqDG>
2. Handler机制： <http://www.2cto.com/kf/201505/399068.html>

注意：

1. 启动模式:

这4中模式又分两类，standard和signleTop属于一类， singleTask和signleInstance属于另一类。

    standard和singleTop属性的 Activity 的实例可以属于任何任务（Task），并且可以位于Activity堆栈的任何位置。比较典型的一种情况是，一个任务的代码执行startActivity()，如果传递的 Intent 对象没有包含 FLAG\_ACTIVITY\_NEW\_TASK 属性， 指定的 Activity 将被该任务调用，从而装入该任务的Activity 堆栈中。 standard和singleTop的区别在于：standard模式的Activity在被调用时会创建一个新的实例，所有实例处理同一个Intent对象； 但对于singleTop模式的Activity，如果被调用的任务已经有一个这样的Activity 在堆栈的顶端，那么不会有新的实例创建， 任务会使用当前顶端的Activity实例来处理Intent对象，换句话说，如果被调用的任务包含一个不在堆栈顶端的 singleTop Activity， 或者堆栈顶端为 singleTop 的Activity的任务不是当前被调用的任务，那么，仍然会有一个新的Activity对象被创建。

    singleTask 和 singleInstance模式的Activity 仅可用于启动任务的情况， 这种模式的Activity总是处在Activity堆栈的最底端，并且一个任务中只能被实例化一次。两者的区别在于：对于 singleInstance模式的Activity， 任务的Activity堆栈中如果有这样的Activity，那它将是堆栈中的唯一的 Activity， 当前任务收到的 Intent 都由它处理， 由它开启的其他 Activity 将在其他任务中被启动； 对于 SingleTask模式的Activity，它在堆栈底端，其上方可以有其他Activity被创建， 但是，如果发给该Activity的Intent对象到来时该Activity不在堆栈顶端，那么该Intent对象将被丢弃，但是界面还是会切换到当前的Activity。

1. handler机制
2. Thread最多与一个looper建立联系。 2）一个looper有且仅有一个MessageQueue 3）一个handler只能与一个looper关联 4）一个Message只能与一个handler关联 这四个一对一的关系使得消息发送和处理得到正确的相应。

## Handler类：

构造函数： 1）Handler handler = new Handler(),内部会调用 public Handler(Callback callback, boolean async)方法，将Handler的成员变量mLooper和mQueue赋值为当前线程的looper对象和对应的消息队列,如果当前线程没有looper会抛出异常。 2)Handler handle = new Handler(Looper looper),内部会调用public Handler(Looper looper, Callback callback, boolean async)，将参数的looper对象传递给handler的mHandler对象，mQueue对象赋值。通过此方法可以在其它线程中为主线程创建Handler,例如：new Handler（Looper.getMainLooper()）;  
handler的sendMessage()方法，在内部会将handler对象赋值为Message的mTarget中，然后调用mQueue将消息发送到线程的消息队列中。

## Message类：

Message类内部会保存一个Handler对象mTarget。  
Looper的loop方法，取出消息，然后调用Message对象中的handler的dispatchMessage方法，dispatchMessage根据callback是否为空执行handleMessage方法。

 handler机制的原理  
　　答 ：andriod提供了 Handler 和 Looper 来满足线程间的通信。Handler 先进先出原则。Looper类用来管理特定线程内对象之间的消息交换(Message Exchange)。　　1)Looper: 一个线程可以产生一个Looper对象，由它来管理此线程里的Message Queue(消息队列)。　　2)Handler: 你可以构造Handler对象来与Looper沟通，以便push新消息到Message Queue里;或者接收Looper从Message Queue取出)所送来的消息。 　　 3) Message Queue(消息队列):用来存放线程放入的消息。　　4)线程：UI thread 通常就是main thread，而Android启动程序时会替它建立一个Message Queue。