



《Android 多线程编程》习题

一、选择题

1. Android 四大组件包括 ()。

A Activity B service

C content provider D SQLite

2. 启动线程时需要调用的方法是 ()。

A run() B start()

C execute() D create()

3. Android 的异步消息处理机制中，发送消息一般使用 Handler 的 () 方法。

A sendMessage () B handleMessage ()

C loop () D MessageQueue ()

4. 在使用 AsyncTask 时，() 方法中的所有代码都会在子线程中运行。

A onPreExecute()

B doInBackground(Params...)

C onProgressUpdate(Progress...)

D onPostExecute(Result)

二、问答题

1. Android 中的异步消息处理主要由几部分组成，简单介绍这几部分。

2. Android 中如何创建和使用多线程？



《Android 多线程编程》习题答案

一、选择题

1. 答案 ABC

解析：Android 四大组件分别为 activity、service、content provider、broadcast receiver。

2. 答案 B 3. 答案 A

解析：handleMessage() 是对具体的 Message 进行处理。loop() 是 Looper 的循环方法。MessageQueue() 不是方法，MessageQueue 是消息队列。

4. 答案 B

二、问答题

1. 答案

Android 中的异步消息处理主要由四个部分组成，Message、Handler、MessageQueue 和 Looper。下面我就对这四个部分进行一下简要的介绍。

(1) Message

Message 是在线程之间传递的消息，它可以在内部携带少量的信息，用于在不同线程之间交换数据。上一小节中我们使用到了 Message 的 what 字段，除此之外还可以使用 arg1 和 arg2 字段来携带一些整型数据，使用 obj 字段携带一个 Object 对象。

(2) Handler

Handler 顾名思义也就是处理者的意思，它主要是用于发送和处理消息的。发送消息一般是使用 Handler 的 sendMessage() 方法，而发出的消息经过一系列地辗转处理后，最终会传递到 Handler 的 handleMessage() 方法中。

(3) MessageQueue

MessageQueue 是消息队列的意思，它主要用于存放所有通过 Handler 发送的消息。这部分消息会一直存在于消息队列中，等待被处理。每个线程中只会拥有一个 MessageQueue 对象。

(4) Looper

Looper 是每个线程中的 MessageQueue 的管家，调用 Looper 的 loop() 方法后，就会进入到一个无限循环当中，然后每当发现 MessageQueue 中存在



一条消息，就会将它取出，并传递到 Handler 的 handleMessage() 方法中。每个线程中也只会会有一个 Looper 对象。

2. 答案

(1) 继承 Thread 类实现多线程

Android 多线程编程和 Java 多线程编程基本都是使用相同的语法。需要新建一个类继承自 Thread，然后重写父类的 run() 方法，并在里面编写耗时逻辑即可，如下所示：

```
class MyThread extends Thread {  
  
    @Override  
  
    public void run() {  
        // 处理具体的逻辑  
    }  
}
```

启动这个线程需要 new 出 MyThread 的实例，然后调用它的 start() 方法，这样 run() 方法中的代码就会在子线程当中运行了，如下所示：

```
new MyThread().start();
```

(2) 实现 Runnable 接口的实现多线程

新建一个类实现 Runnable 接口，实现 run() 方法。

```
class MyThread implements Runnable {  
  
    @Override  
  
    public void run() {  
        // 处理具体的逻辑  
    }  
}
```

启动线程的方法为：

```
MyThread myThread = new MyThread();  
new Thread(myThread).start();
```

由于 Runnable 接口没有 start() 方法，一个实现了 Runnable 接口的对象传入到 Thread 的构造函数里。接着调用 Thread 的 start() 方法，run() 方法中的代码就会在子线程当中运行了。也可以使用匿名类的方式，如下所示：

```
new Thread(new Runnable() {
```



```
@Override  
public void run() {  
    // 处理具体的逻辑  
}  
}  
).start();
```