

由于 3 个消费者是并发执行的，因此当某个消费者打印时，其他消费者可能还没有给相应数据赋值，所以会出现类似 `data1= 'null'` 的情况。

如果要将 4 个属性全部赋值之后再打印，应该如何操作？很遗憾，`RingBuffer`、`EventHandler`、`BatchEventProcessor` 及下面要讲的 `WorkerPool`、`WorkHandler` 等 API 都没有直接提供这种设置依赖顺序的方法。因此，如果项目中存在这种依赖顺序的问题，就只能使用之前讲的 `Disruptor` 对象来实现了。当然，读者也可以通过自己编写 `Future`、`CountDownLatch` 等闭锁代码来实现。

### 9.3.2 ▶ WorkerPool+WorkHandler 使用案例

“`EventHandler+BatchEventProcessor`”方式是使用 `WorkHandler` 模拟消费者，并且将 `EventHandler`、`RingBuffer`（环形缓冲区）等对象封装到 `WorkerPool` 中。最后创建一个线程去执行 `WorkerPool` 对象中的任务。

#### 范例 9-6 使用 JDK 提供的 API 实现观察者模式

本次用“`WorkerPool+WorkHandler`”方式再次实现范例 9-5 的要求。

本例的实现代码如下所示。

#### 1. 数据类

与“`EventHandler+BatchEventProcessor`”方式的相同。

#### 2. 第一个消费对象

» 【源码：`demo/ch09/disruptor/ringbuffer2/Data1EventHandler.java`】

```
...
public class Data1EventHandler implements WorkHandler<DataEvent> {
    @Override
    public void onEvent(DataEvent event) throws Exception {
        event.setData1(UUID.randomUUID().toString().substring(0,4));
        System.out.println("Data1EventHandler:"+event);
    }
}
```

#### 3. 第二个消费对象

» 【源码：`demo/ch09/disruptor/ringbuffer2/Data2EventHandler.java`】

```
...
public class Data2EventHandler implements WorkHandler<DataEvent> {
    @Override
    public void onEvent(DataEvent event) throws Exception {
```