ModelAndView即模型与视图，通过***addObject()方法向模型中添加数据，通过setViewName()方法向模型添加视图名称。***

1. **模型**
2. **模型的本质就是HashMap**

模型就是ModelMap，而ModelMap的本质就是个HashMap，***向模型中添加数据，就是向HashMap中添加数据****。*

***向ModelAndView中添加数据，就是向HashMap中添加数据。***

1. **HashMap是一个单向查找数组**

HashMap的本质是一个***单向链表数组***。从以下源码跟踪中可以知道这点。查看HashMap的源码，其用于存放数据的底层数据结构为一个数组，而数组元素为一个Entry对象。

跟踪Entry类可知，其为HashMap类的内部类，为一个可单向链表的数据结构：因为其***只能通过next查找下一个元素，而无法查找上一个元素。***

1. **LinkedHashMap**

LinkedHashMap的本质是一个HashMap，但其将Entry内部类进行了扩展。HashMap中的Entry是单向的，只能通过next查找下一个元素。而***LinkedHashMap中的Entry变为了双向的，可以通过before查找上一个元素，通过after查找下一个元素。即从性能上说，LinkedHashMap的操作性能要高于HashMap。***

我们应该掌握的是***ModelAndView中的模型对象是ModelMap***，其本质是一个HashMap，向ModelMap中添加数据就是向HashMap中添加数据。只不过，这个ModelMap要比HashMap的性能更高。

1. **视图**

通过setViewName()指定视图名称。注意，这里的视图名称将会对应一个视图对象，一般是不会在这里直接写上要跳转的页面的。这个视图对象，将会被封装在ModelAndView中，传给视图解析器来解析，最终转换为相应的页面。但需要注意的是，这里的***View对象本质仅仅是一个String***而矣。后续的步骤中，还会继续对这个View对象进行进一步的封装。

当然，若处理器方法返回的ModelAndView中并没有数据要携带，则可直接通过ModelAndView的带参构造器将视图名称放入ModelAndView中。

