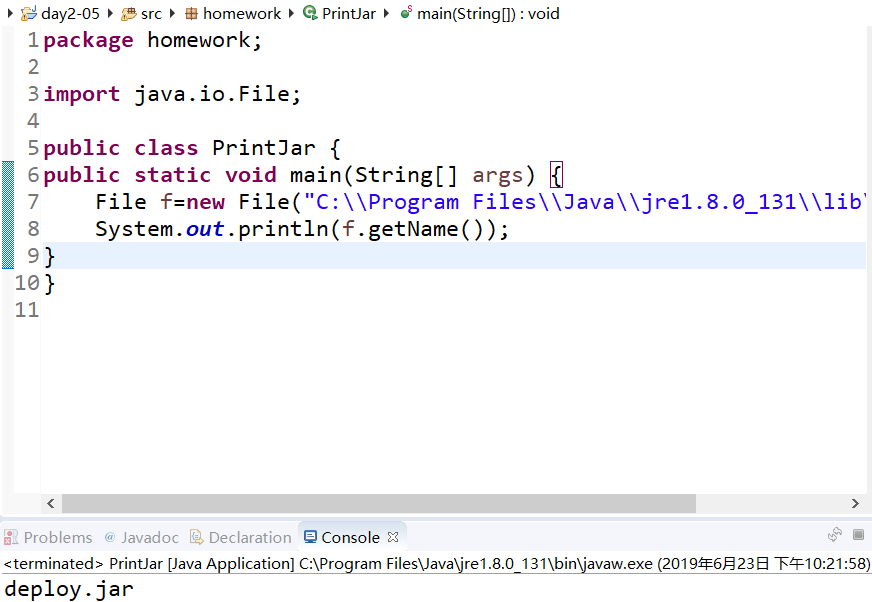
1. 打印某个地址的.jar文件名





1. 描述出HashMap与Hashtable的区别
2. 继承父类不同，HashMap继承AbstractMap类

Hashtable继承Dictionary类（已废弃）

1. 对外提供的接口不同

Hashtable比HashMap多提供了elments() 和contains() 两个方法

elements() 方法用于返回此Hashtable中的value的枚举。

contains()方法判断该Hashtable是否包含传入的value。

1. Hashtable既不支持Null key也不支持Null value，不然会出现空指针异常

HashMap中，null可以作为键，这样的键只有一个；可以有一个或多个键所对应的值为nul

4）线程安全不同

Hashtable是线程安全的，它的每个方法中都加入了Synchronize方法，在多线程并发的环境下，可以直接使用Hashtable，不需要自己为它的方法实现同步

HashMap不是线程安全的，在多线程并发的环境下，可能会产生死锁等问题，但是它的效率会比Hashtable要好很多

5）遍历方式的内部实现上不同

Hashtable、HashMap都使用了 Iterator，Hashtable还使用了Enumeration的方式 。

1. 初始容量大小和每次扩充容量大小的不同

Hashtable默认的初始大小为11，之后每次扩充，容量变为原来的2n+1。HashMap默认的初始化大小为16。之后每次扩充，容量变为原来的2倍

创建时，如果给定了容量初始值，那么Hashtable会直接使用你给定的大小，而HashMap会将其扩充为2的幂次方大小。也就是说Hashtable会尽量使用素数、奇数。而HashMap则总是使用2的幂作为哈希表的大小

1. 描述出TreeMap与TreeSet的区别

TreeMap是Map接口的常用实现类，而 TreeSet 是 Set 接口的常用实现类。

TreeMap 和TreeSet 实现的接口规范不同，但 TreeSet 底层是通过 TreeMap 来实现的（如同HashSet底层是是通过HashMap来实现的一样），因此二者的实现方式完全一样。而 TreeMap 的实现就是红黑树算法。

与HashSet完全类似，TreeSet里面绝大部分方法都是直接调用TreeMap方法来实现的。

相同点：

TreeMap和TreeSet都是有序的集合，也就是说他们存储的值都是排好序的。

TreeMap和TreeSet都是非同步集合，因此他们不能在多线程之间共享，不过可以使用方法Collections.synchroinzedMap()来实现同步

运行速度都要比Hash集合慢

不同点;

最主要的区别就是TreeSet和TreeMap分别实现Set和Map接口

TreeSet只存储一个对象，而TreeMap存储两个对象Key和Value（仅仅key对象有序）

TreeSet中不能有重复对象，而TreeMap中可以存在

TreeMap的底层采用红黑树的实现，完成数据有序的插入，排序

TreeSet要求存放的对象所属的类必须实现Comparable接口，该接口提供了比较元素的compareTo()方法，当插入元素时会回调该方法比较元素的大小

TreeMap要求存放的键值对映射的键必须实现Comparable接口从而根据键对元素进行排序。

1. 遍历出HashMap的键值对数据(至少2种)

