Java SE Lesson 6

1. 当向 ArrayList 添加一个对象时，实际上就是将该对象放置到了 ArrayList 底层所维护的数组当中；当向 LinkedList 中添加一个对象时，实际上 LinkedList 内部会生成一个 Entry 对象，该 Entry 对象的结构为：

```
Entry
{
    Entry previous;
    Object element;
    Entry next;
}
```

其中的 Object 类型的元素 element 就是我们向 LinkedList 中所添加的元素，然后 Entry 又构造好了向前与向后的引用 previous、next，最后将生成的这个 Entry 对象加入到了链表当中。换句话说，**LinkedList 中所维护的是一个个的 Entry 对象**。

2. 关于 Object 类的 equals 方法的特点

a) 自反性：x.equals(x)应该返回 true

b) 对称性：x.equals(y)为 true，那么 y.equals(x)也为 true。

c) 传递性：x.equals(y)为 true 并且 y.equals(z)为 true，那么 x.equals(z)也应该为 true。

d) 一致性：x.equals(y)的第一次调用为 true，那么 x.equals(y)的第二次、第三次、第 n 次调用也应该为 true，前提条件是在比较之间没有修改 x 也没有修改 y。

e) 对于非空引用 x，x.equals(null)返回 false。

3. 关于 Object 类的 hashCode()方法的特点：

a) 在 Java 应用的一次执行过程当中，对于同一个对象的 hashCode 方法的多次调用，他们应该返回同样的值（前提是该对象的信息没有发生变化）。

b) 对于两个对象来说，如果使用 equals 方法比较返回 true，那么这两个对象的 hashCode 值一定是相同的。

c) 对于两个对象来说，如果使用 equals 方法比较返回 false，那么这两个对象的 hashCode 值不要求一定不同（可以相同，可以不同），但是如果不同则可以提高应用的性能。

d) 对于 Object 类来说，不同的 Object 对象的 hashCode 值是不同的（Object 类的 hashCode 值表示的是对象的地址）。

4. 当使用 HashSet 时，hashCode()方法就会得到调用，判断已经存储在集合中的对象的 hash code 值是否与增加的对象的 hash code 值一致；如果不一致，直接加进去；如果一致，再进行 equals 方法的比较，equals 方法如果返回 true，表示对象已经加进去了，就不会再增加新的对象，否则加进去。

5. 如果我们重写 equals 方法，那么也要重写 hashCode 方法，反之亦然。

6. Map（映射）：Map 的 keySet()方法会返回 key 的集合，因为 Map 的键是不能重复的，因此 keySet()方法的返回类型是 Set；而 Map 的值是可以重复的，因此 values()方法的返回类型是 Collection，可以容纳重复的元素。

7. **作业：参见 Lesson 4 的要求，使用集合实现，不允许使用数组。**

8. **策略模式（Strategy Pattern）。通过查询资料掌握策略模式的原理。**

9. **阅读 TreeMap 的帮助文档，自己写一个程序，练习 TreeMap 的使用方式并且自己定义一个 Comparator。**