**三 面向对象**

**内部类：**

成员内部类不是静态的：

外部类.内部类 对象名=new 外部类名().new 内部类名()

成员内部类是静态的：

外部类名.内部类名=new 外部类名.内部类名()

**四 集合框架与泛型**

**List<E>接口的子类特点**

**ArraList<E>:**

底层数据结构是数组，查询快，增删慢

线程不安全，效率高

**LinkedList<E>:**

底层数据结构是链表，查询慢，增删快

线程不安全，效率高

**一般使用ArrayList<E>**

**Set<E>接口的子类特点**

**HashSet<E>**

底层数据结构是哈希表(是一个元素为链表的数组)

哈希表依赖：hashCode(),equals()（保证元素唯一性）

首先比较哈希表是否相同

相同：继续执行equals()方法

返回true：元素重复，不添加

返回false：直接把元素添加到集合

不同：直接把元素添加到集合

**TreeSet<E>**

**底层数据结构是红黑树（自平衡的二叉树）**

排序方式

自然排序：无参构造，类实现Comparable接口，重写compareTo(Object o)方法

比较器排序：有参构造，参数为实现Comparator接口的类的对象，重写compare()方法

**一般使用HashSet<E>**

**当方法重载而且参数个数相同时，可以使用泛型**

**当继承或者实现时，格式：子类名<父泛型>**

**泛型类**

格式：public class 类名<泛型类型1,…>

**泛型方法**

格式：public <泛型类型> 返回类型 方法名(泛型类型)

**泛型接口**

格式：public interface 接口名<泛型类型1,…>

**泛型类型必须时引用类型**

**集合常见功能和遍历方式**

Collection<E>:

功能：add()、remove()、contains()、iterator()、size()

遍历：增强for、迭代器

List<E>:

功能：get()

Map<K,V>:

功能：put()、remove()、containskey()、containsvalue()、keySet()、get()、value()、entrySet()、size()

遍历：根据键找值、根据键值对对象分别找键和值