

Map和collection是接口

public interface Queue<E> extends [Collection](http://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<E>

public class LinkedList<E>

extends [AbstractSequentialList](http://docs.oracle.com/javase/7/docs/api/java/util/AbstractSequentialList.html)<E>

implements [List](http://docs.oracle.com/javase/7/docs/api/java/util/List.html)<E>, [Deque](http://docs.oracle.com/javase/7/docs/api/java/util/Deque.html)<E>, [Cloneable](http://docs.oracle.com/javase/7/docs/api/java/lang/Cloneable.html), [Serializable](http://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html) （deque实现了queue）

collection继承iterable

Collection: 存放独立元素的序列。

Map：存放key-value型的元素对。从类体系图中可以看出，Collection定义了Collection类型数据的最基本、最共性的功能接口，而List对该接口进行了拓展。

其中各个类的适用场景有很大的差别，在使用时，应该根据需要灵活的进行选择。此处介绍最为常用的四个容器：

**LinkedList** ：其数据结构采用的是链表，此种结构的优势是删除和添加的效率很高，但随机访问元素时效率较ArrayList类低。

**ArrayList**：其数据结构采用的是线性表，此种结构的优势是访问和查询十分方便，但添加和删除的时候效率很低。

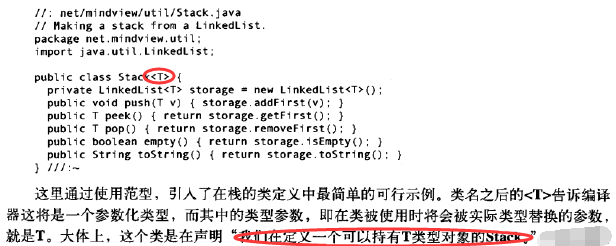
**HashSet**: Set类不允许其中存在重复的元素（集），无法添加一个重复的元素（Set中已经存在）。HashSet利用Hash函数进行了**查询效率上的优化**，其contain（）方法经常被使用，以用于判断相关元素是否已经被添加过。（treeset排序-红黑树）

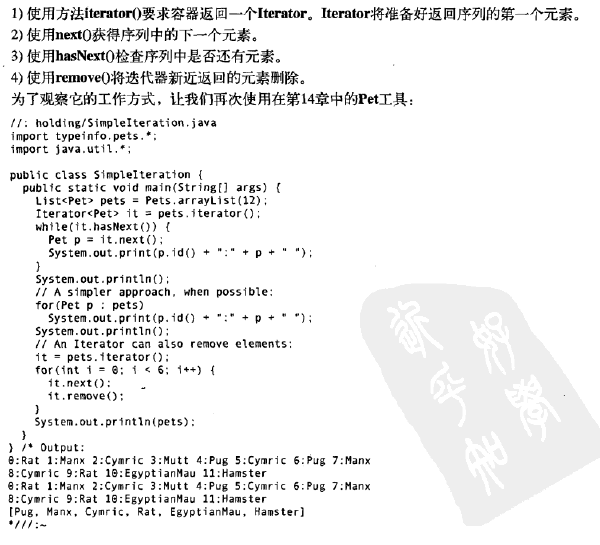
**HashMap**: 提供了key-value的键值对数据存储机制，可以十分方便的通过键值查找相应的元素，而且通过Hash散列机制，查找十分的方便。

**List的remove，removeall,indexof，containsall,retainsall(取交集)这些操作都是基于equals做的，所谓的equals是基于类自己的方法判断，若不存在则是调用基类的方法判断引用是否一致。**

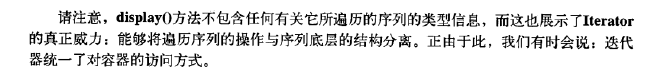
**因为list判重基于引用，所以不保存重复元素的set更适合查找，Hashset则是对查找有优化。**

**linkedlist**可作为**栈 队列**使用！

栈：



**Iterator：屏蔽容器类类型**



**小问题：**

List <Integer> b = new ArrayList<Integer>(Arrays.asList(1,3,45,2));  
b.add(2);  
System.***out***.print(b);

**对的**

List <Integer> b =Arrays.asList(1,3,45,2);  
b.add(2);  
System.***out***.print(b);

