P131 容器Collection

Set:无序不可重复.

List:有序可重复.(比Collection多了一些索引相关的操作)

Map:通过一个对象找另一个对象.(键值对). //底层实现是数组加链表.

Java.util.Collection

Iterator<E> iterator(); //迭代器,帮助我们便利容器.

Object[] toArray(); //把容器里的对象,放到一个Object[] 数组里面.

boolean add(E e); //把一个对象放入容器里面.

boolean remove(Object o); //将容器里的Object o移除出容器.

boolean containsAll(Collection<?> c); //容器是否包含另一个容器的所有元素.

boolean addAll(Collection<? extends E> c); //把指定容器里的所有元素加到该容器里面.

boolean removeAll(collection<?> c); //将该容器内与指定容器相同的元素移除.

boolean retainAll(Collection<?> c); //将该容器与指定容器内相同的元素取出来.

void clear(); //把该容器内所有对象全部清除.

boolean equal(object o ); //判断两个对象是否相等.

int hashCode();

P132 ArrayList;LinkedList;Vector

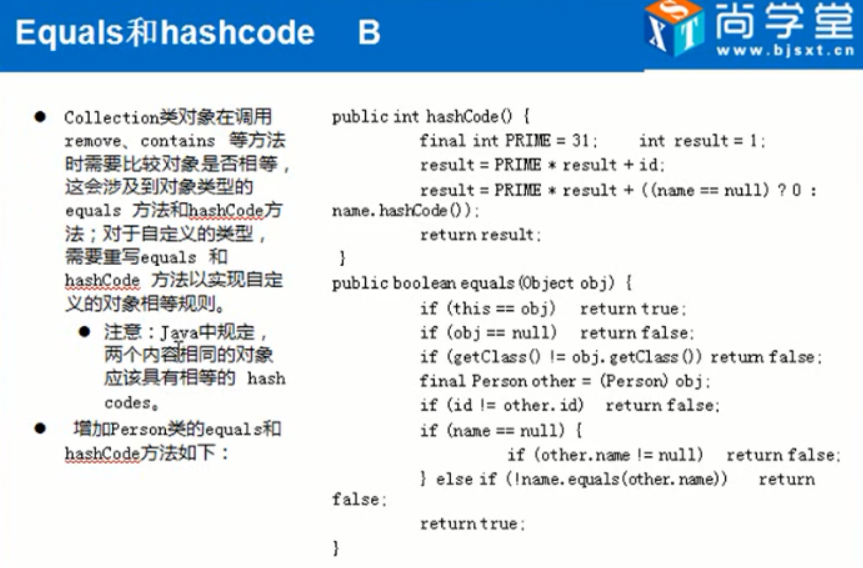
//ArrayList:底层实现是数组,线程不安全,效率高.所以,查询快.修改/插入/删除慢.

//LinkedList:底层实现是链表,线程不安全,效率高.所以,查询慢.修改/插入/删除快.

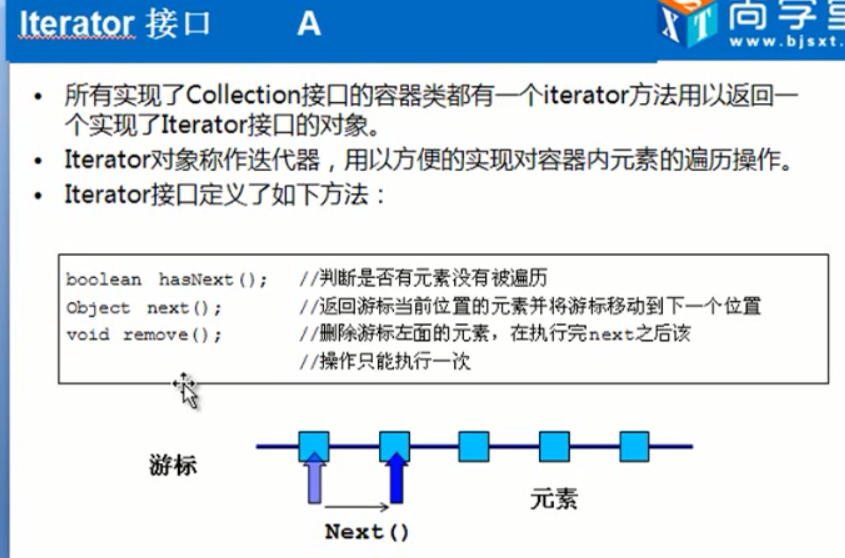
//Vector:底层实现是数组,线程安全,效率低.

P140 Equals和Hashcode

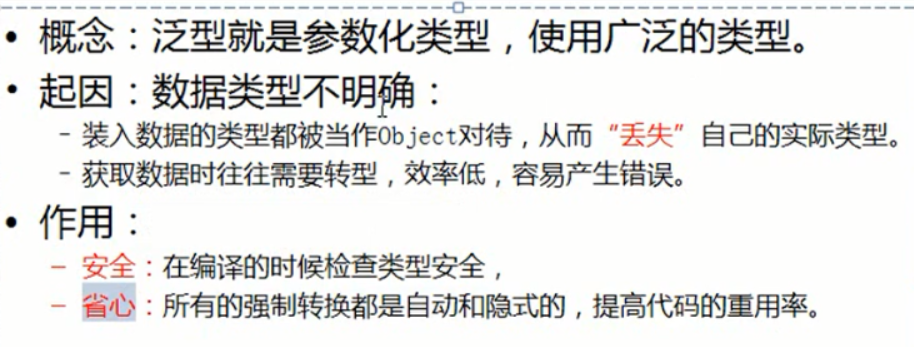
一个对象Equals另一个对象,他们的Hashcode一定相等.两个对象的Hashcode相等,这两个对象不一定Equals.

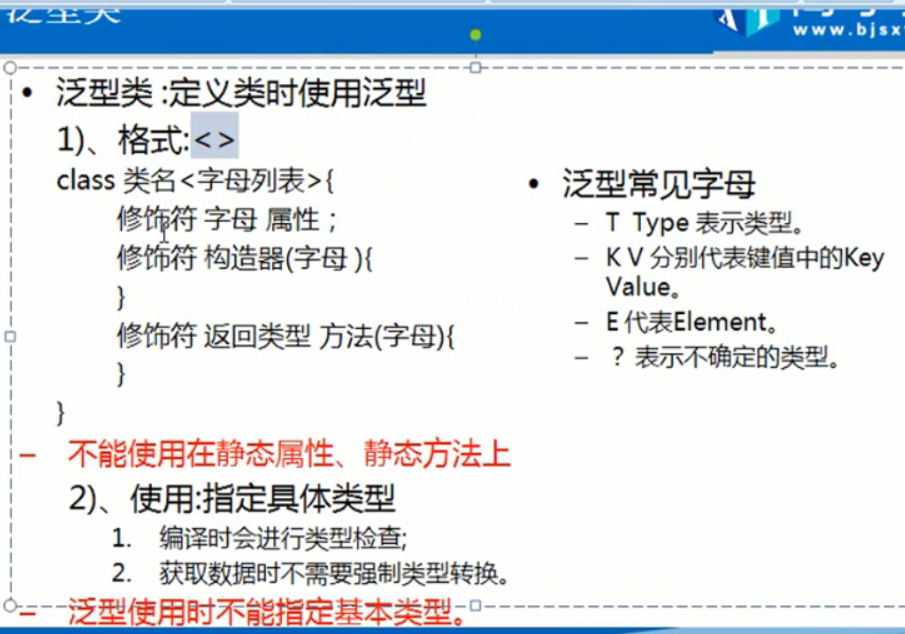


P146 Iterator接口(迭代器)



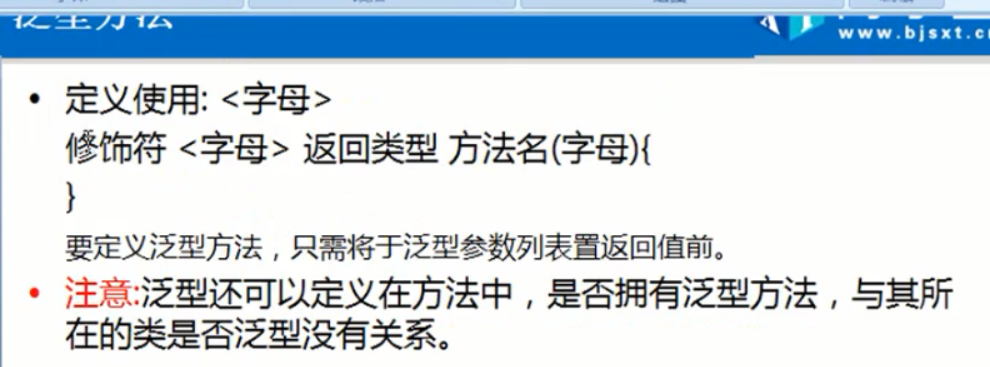
P148泛型



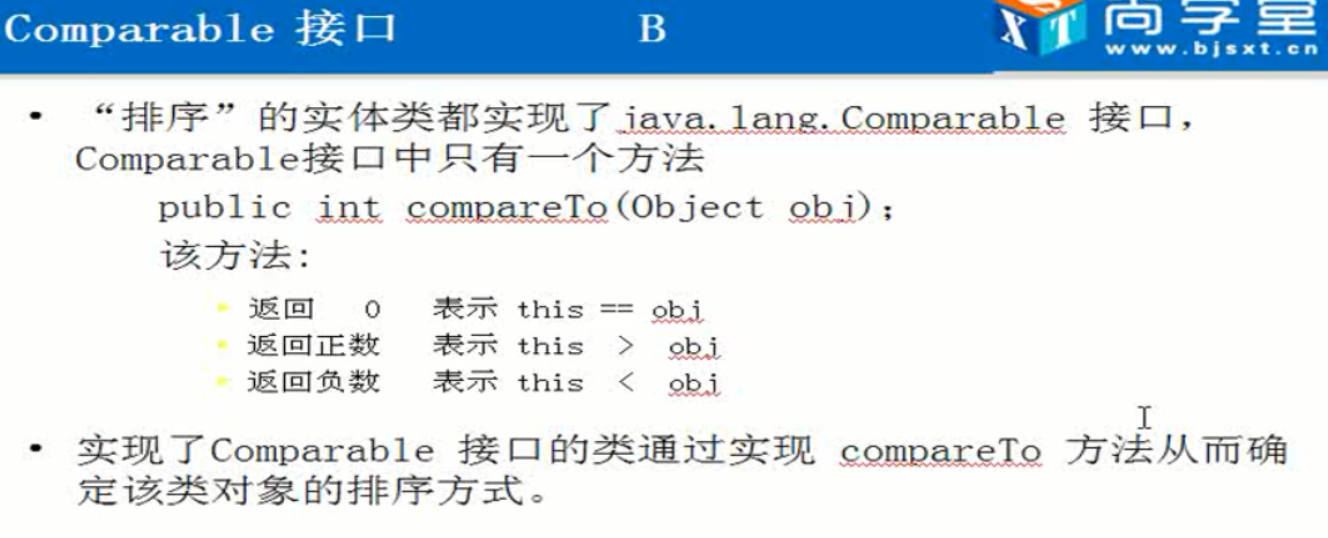




泛型方法



P156 引用类型,内置类比较;Comparable接口;



1、整数、小数Integer Float Double直接比较基本数据类型的大小

2、字符:比较unicode码之差;

3、字符串：

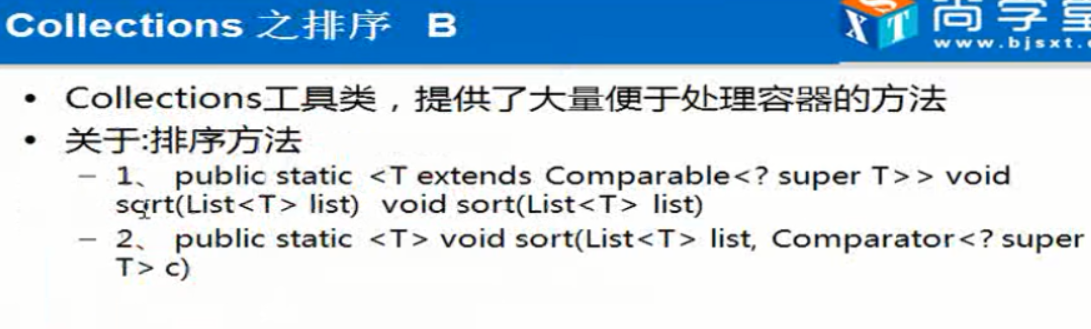
1）如果其中一个是另外一个起始开始的字串，返回长度之差；

2）否则返回第一个不相等的unicode码之差；

4、java.util.Date:根据日期的长整形数比较

P157引用类型,内置类比较; Comparable接口B;





P158 自定义数据排序

P159 TreeSet与TreeMap



1. TreeSet:数据元素可以排序且不可重复

对比:Set接口:HashSet,元素必须重写HashCode和equals方法.

去重:比较等于0即重复

1. 元素可以排序:java.lang.Comparable +compareTo

new TreeSet();

1. 排序业务类 java.util.Comparator +compare

new TreeSet(Comparator<? SuperE> comparator)

注意:在添加数据时排序,数据更改不会影响原来的顺序,不要修改数据,否则可能重复.

P160 工具栏Collections;

binarySearch(List<? Extends Comparable<? super T>> list, T key) 容器有序

sort(List<?> list)

sort(List<?> list, Comparator<super T> c)

reverse(List<?> list)

shuffle(List<?> list) 洗牌

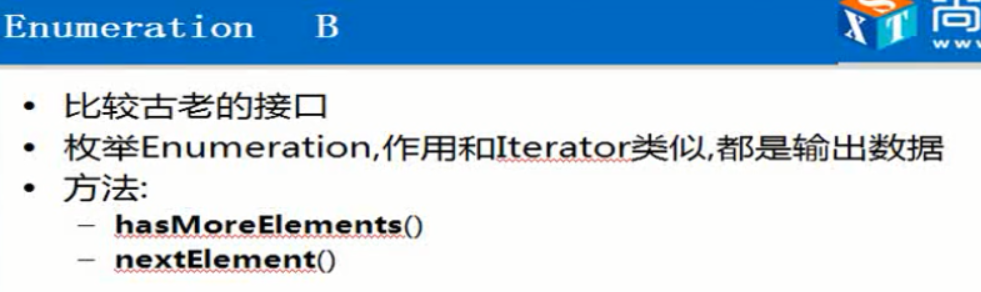
swap(List<?> list, int i ,int j) 交换list中i和j的顺序

P161 Queue接口,单向队列





P162 Enumeration接口



应用:

Vector:elements()

StringTokenizer

P163 Hashtable

1. Hashtable与HashMap的区别(面试题)
2. 主要:Hashtable线程安全,同步,效率相对低下

HashMap线程不安全,非同步,效率相对高

1. 父类:Hashtable是Dictionary HashMap是AbstractMap
2. null: Hashtable键与值不能为null

HashMap键最多一个null,值可以多个null

1. Properties
2. 作用:读写资源配置文件
3. 键与值只能为字符串
4. 方法:

[**setProperty**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#setProperty(java.lang.String, java.lang.String))([String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) key, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) value)

[**getProperty**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#getProperty(java.lang.String))([String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) key)

[**getProperty**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#getProperty(java.lang.String, java.lang.String))([String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) key, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) defaultValue)

后缀:properties

[**store**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#store(java.io.OutputStream, java.lang.String))([OutputStream](mk:@MSITStore:C:\\Users\\Hasee\\Desktop\\JAVA_API_1.7中文.chm::/java/io/OutputStream.html" \o "class in java.io) out, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) comments)

[**store**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#store(java.io.Writer, java.lang.String))([Writer](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Writer.html) writer, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) comments)

[**load**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#load(java.io.InputStream))([InputStream](mk:@MSITStore:C:\\Users\\Hasee\\Desktop\\JAVA_API_1.7中文.chm::/java/io/InputStream.html" \o "class in java.io) inStream)

[**load**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#load(java.io.Reader))([Reader](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Reader.html) reader)

.xml

[**storeToXML**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#storeToXML(java.io.OutputStream, java.lang.String))([OutputStream](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html) os, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) comment) :UTF-8字符集

[**storeToXML**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#storeToXML(java.io.OutputStream, java.lang.String, java.lang.String))([OutputStream](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html) os, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) comment, [String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) encoding)

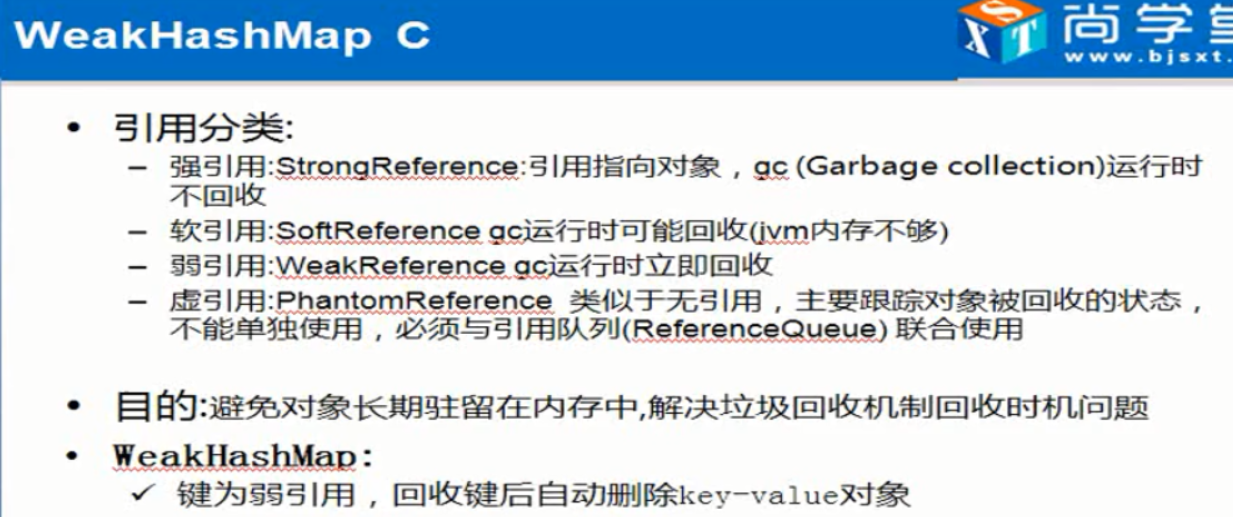
[**loadFromXML**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/util/Properties.html#loadFromXML(java.io.InputStream))([InputStream](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html) in)

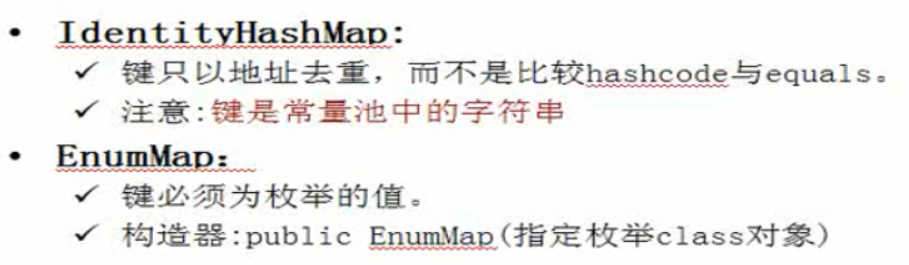
1. 相对路径与绝对路径
2. 绝对路径: 盘符: /
3. 相对路径: 当前项目,工程
4. 类路径加载资源文件

类所在的根路径

1. 类.class.getResourceAsStream("/")
2. Thread.*currentThread*().getContextClassLoader().getResourceAsStream("")

P164 WeakHashMap





P165 同步控制与只读设置



只读设置

\* 1.emptyXxx() 空的不可变的集合

\* emptyList()

\* emptyMap()

\* emptySet()

\*

\* 2.singletonXxx() 一个元素不可变的集合

\* singleton(T o)

\* singletonList(T o)

\* singletonMap(K key, V value)

\*

\* 3.unmodifiableXxx() 不可变容器

\* unmodifiableList(List<? extends T> list)

\* unmodifiableMap(Map<? extends K,? extends V> m)

\* unmodifiableSet(Set<? extends T> s)