

Java基础 - Java语言的Socket类详细介绍 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Java_E5_9F_BA_E7_A1_80_c104_145633.htm 线性表，链表等是常用的数据结构，在进行Java开发时，JDK已经为我们提供了一系列相应的类来实现基本的数据结构。这些类均在java.util包中

。Collection | List | | LinkedList | | ArrayList | | Vector |
| Stack | Set Map | Hashtable | HashMap | WeakHashMap

Collection接口 Collection是最基本的集合接口，一个Collection代表一组Object，即Collection的元素（Elements）。一些Collection允许相同的元素而另一些不行。一些能排序而另一些不行。Java SDK不提供直接继承自Collection的类，Java SDK提供的类都是继承自Collection的“子接口”如List和Set。所有实现Collection接口的类都必须提供两个标准的构造函数：无参数的构造函数用于创建一个空的Collection，有一个Collection参数的构造函数用于创建一个新的Collection，这个新的Collection与传入的Collection有相同的元素。后一个构造函数允许用户复制一个Collection。如何遍历Collection中的每一个元素？不论Collection的实际类型如何，它都支持一个iterator()的方法，该方法返回一个迭代子，使用该迭代子即可逐一访问Collection中每一个元素。典型的用法如下：

```
Iterator it = collection.iterator(). // 获得一个迭代子
```

```
while(it.hasNext()) { Object obj = it.next(). // 得到下一个元素 }
```

由Collection接口派生的两个接口是List和Set。List接口 List是有序的Collection，使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引（元素在List中的位置，类似于数

组下标)来访问List中的元素,这类似于Java的数组。和下面要提到的Set不同, List允许有相同的元素。除了具有Collection接口必备的iterator()方法外, List还提供一个listIterator()方法, 返回一个ListIterator接口, 和标准的Iterator接口相比, ListIterator多了一些add()之类的方法, 允许添加, 删除, 设定元素, 还能向前或向后遍历。实现List接口的常用类有LinkedList, ArrayList, Vector和Stack。LinkedList类LinkedList实现了List接口, 允许null元素。此外LinkedList提供额外的get, remove, insert方法在LinkedList的首部或尾部。这些操作使LinkedList可被用作堆栈(stack), 队列(queue)或双向队列(deque)。注意LinkedList没有同步方法。如果多个线程同时访问一个List, 则必须自己实现访问同步。一种解决方法是在创建List时构造一个同步的List: List list = Collections.synchronizedList(new LinkedList(...)). ArrayList类ArrayList实现了可变大小的数组。它允许所有元素, 包括null。ArrayList没有同步。size, isEmpty, get, set方法运行时间为常数。但是add方法开销为分摊的常数, 添加n个元素需要O(n)的时间。其他的方法运行时间为线性。每个ArrayList实例都有一个容量(Capacity), 即用于存储元素的数组的大小。这个容量可随着不断添加新元素而自动增加, 但是增长算法并没有定义。当需要插入大量元素时, 在插入前可以调用ensureCapacity方法来增加ArrayList的容量以提高插入效率。和LinkedList一样, ArrayList也是非同步的(unsynchronized)。Vector类 Vector非常类似ArrayList, 但是Vector是同步的。由Vector创建的Iterator, 虽然和ArrayList创建的Iterator是同一接口, 但是, 因为Vector是同步的, 当一个Iterator被创建而且

正在被使用，另一个线程改变了Vector的状态（例如，添加或删除了一些元素），这时调用Iterator的方法时将抛出ConcurrentModificationException，因此必须捕获该异常。

Stack 类 Stack继承自Vector，实现一个后进先出的堆栈。Stack提供5个额外的方法使得Vector得以被当作堆栈使用。基本的push和pop方法，还有peek方法得到栈顶的元素，empty方法测试堆栈是否为空，search方法检测一个元素在堆栈中的位置。Stack刚创建后是空栈。

Set接口 Set是一种不包含重复的元素的Collection，即任意的两个元素e1和e2都有e1.equals(e2)=false，Set最多有一个null元素。很明显，Set的构造函数有一个约束条件，传入的Collection参数不能包含重复的元素。请注意：必须小心操作可变对象（Mutable Object）。如果一个Set中的可变元素改变了自身状态导致Object.equals(Object)=true将导致一些问题。

Map接口 请注意，Map没有继承Collection接口，Map提供key到value的映射。一个Map中不能包含相同的key，每个key只能映射一个value。Map接口提供3种集合的视图，Map的内容可以被当作一组key集合，一组value集合，或者一组key-value映射。

Hashtable类 Hashtable继承Map接口，实现一个key-value映射的哈希表。任何非空（non-null）的对象都可作为key或者value。添加数据使用put(key, value)，取出数据使用get(key)，这两个基本操作的时间开销为常数。Hashtable通过initial capacity和load factor两个参数调整性能。通常缺省的load factor 0.75较好地实现了时间和空间的均衡。增大load factor可以节省空间但相应的查找时间将增大，这会影响像get和put这样的操作。

100Test 下载频道开通，各类考试题目直接下载。详细请访问

