

集合框架

Java Platform Standard Edition

Java教学部

课程目标

CONTENTS

ITEMS **1** 集合的概念

ITEMS **2** Collection体系集合

ITEMS **3** List接口与实现类

ITEMS **4** Set接口与实现类

ITEMS **5** Map接口与实现类

ITEMS **6** 泛型集合与工具类

什么是集合

- 概念：对象的容器，存储对象的对象，可代替数组。
- 特点：容器的工具类，定义了对多个对象进行操作的常用方法。
- 位置：java.util.*;

Collection体系集合

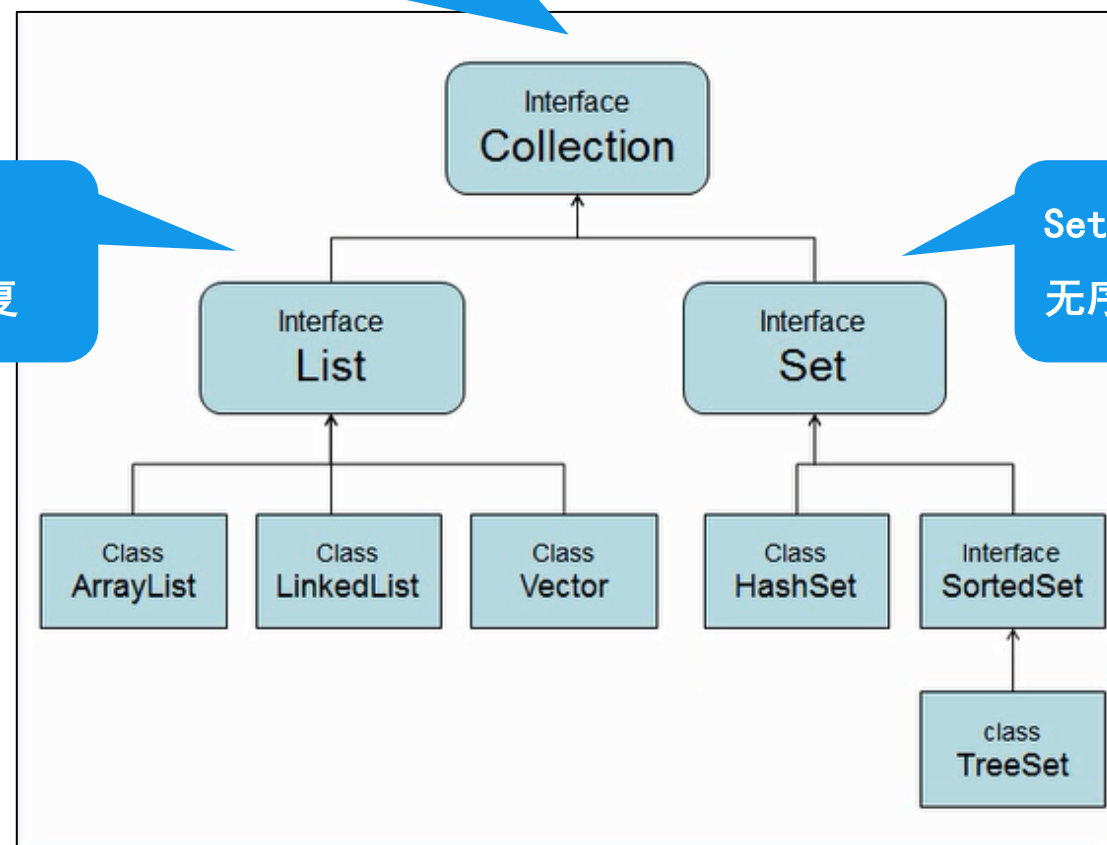
Java Platform Standard Edition

Collection体系集合

该体系结构的根接口，代表一组对象，称为“集合”。

List接口的特点：
有序、有下标、元素可重复

Set接口的特点：
无序、无下标、元素不能重复



- 特点：代表一组任意类型的对象，无序、无下标。
- 方法：
 - `boolean add(Object obj)` //添加一个对象。
 - `boolean addAll(Collection c)` //将一个集合中的所有对象添加到此集合中。
 - `void clear()` //清空此集合中的所有对象。
 - `boolean contains(Object o)` //检查此集合中是否包含o对象
 - `boolean equals(Object o)` //比较此集合是否与指定对象相等。
 - `boolean isEmpty()` //判断此集合是否为空
 - `boolean remove(Object o)` //在此集合中移除o对象
 - `int size()` //返回此集合中的元素个数。
 - `Object[] toArray()` //将此集合转换成数组。

List集合

Java Platform Standard Edition

- 特点：有序、有下标、元素可以重复。
- 方法：
 - `void add(int index, Object o)` //在index位置插入对象o。
 - `boolean addAll(int index, Collection c)` //将一个集合中的元素添加到此集合中的index位置。
 - `Object get(int index)` //返回集合中指定位置的元素。
 - `List subList(int fromIndex, int toIndex)` //返回fromIndex和toIndex之间的集合元素。

- ArrayList【重点】：
 - 数组结构实现，查询快、增删慢；
 - JDK1.2版本，运行效率高、线程不安全。
- Vector：
 - 数组结构实现，查询快、增删慢；
 - JDK1.0版本，运行效率慢、线程安全。
- LinkedList：
 - 链表结构实现，增删快，查询慢。

不同结构实现方式

list.add(5,"G");

ArrayList

A	B	C	D	E	F	null	null	null	null
0	1	2	3	4	5	6	7	8	9
		B							F
		1		D					6
A						E			
0			C			4		G	
			2					5	

LinkedList

list.add(5,"G");

ArrayList: 必须开辟连续空间, 查询快, 增删慢。
LinkedList: 无需开辟连续空间, 查询慢, 增删快。

- 概念：参数化类型、类型安全的集合，强制集合元素的类型必须一致。
- 特点：
 - 编译时即可检查，而非运行时抛出异常。
 - 访问时，不必类型转换（拆箱）。
 - 不同泛型之间引用不能相互赋值，泛型不存在多态。

- 概念：集合工具类，定义了除了存取以外的集合常用方法。
- 方法：
 - `public static void reverse(List<?> list)` //反转集合中元素的顺序
 - `public static void shuffle(List<?> list)` //随机重置集合元素的顺序
 - `public static void sort(List<T> list)` //升序排序（元素类型必须实现Comparable接口）

Set集合

Java Platform Standard Edition

Set子接口

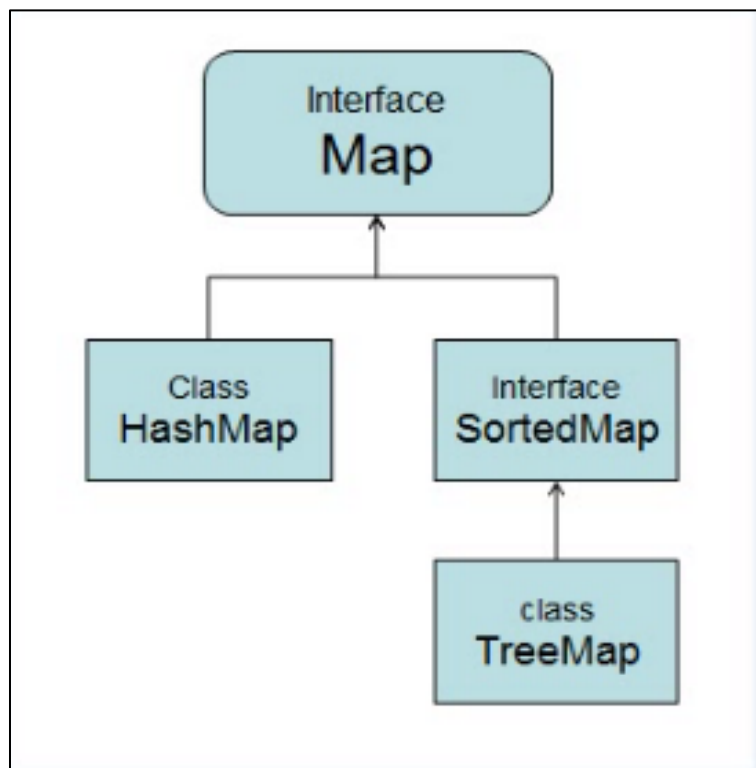
- 特点：无序、无下标、元素不可重复。
- 方法：全部继承自Collection中的方法。

- **HashSet【重点】：**
 - 基于HashCode实现元素不重复。
 - 当存入元素的哈希码相同时，会调用equals进行确认，如结果为true，则拒绝后者存入。
- **LinkedHashSet：**
 - 链表实现的HashSet，按照链表进行存储，即可保留元素的插入顺序。
- **TreeSet：**
 - 基于排列顺序实现元素不重复。
 - 实现了SortedSet接口，对集合元素自动排序。
 - 元素对象的类型必须实现Comparable接口，指定排序规则。
 - 通过CompareTo方法确定是否为重复元素。

Map体系集合

Java Platform Standard Edition

Map结构



Map接口的特点：

- ① 用于存储任意键值对 (Key-Value)
- ② 键：无序、无下标、不允许重复 (唯一)
- ③ 值：无序、无下标、允许重复

- 特点：存储一对数据(Key-Value)，无序、无下标，键不可重复，值可重复。
- 方法：
 - `V put(K key,V value)` //将对象存入到集合中，关联键值。key重复则覆盖原值。
 - `Object get(Object key)` //根据键获取对应的值。
 - `Set<K>` //返回所有key。
 - `Collection<V> values()` //返回包含所有值的Collection集合。
 - `Set<Map.Entry<K,V>>` //键值匹配的Set集合。

- **HashMap【重点】：**
 - JDK1.2版本，线程不安全，运行效率高；允许用null作为key或是value。
- **Hashtable：**
 - JDK1.0版本，线程安全，运行效率慢；不允许null作为key或是value。
- **Properties：**
 - Hashtable的子类，要求key和value都是String。通常用于配置文件的读取。
- **TreeMap：**
 - 实现了SortedMap接口(是Map的子接口)，可以对key自动排序。

- 集合的概念：
 - 对象的容器，存储对象的对象，定义了对多个对象进行操作的常用方法。
- List集合：
 - 有序、有下标、元素可以重复。（ ArrayList、LinkedList、Vector ）
- Set集合：
 - 无序、无下标、元素不可重复。（ HashSet、LinkedHashSet、TreeSet ）
- Map集合：
 - 存储一对数据，无序、无下标，键不可重复，值可重复。（ HashMap、HashTable ）
- Collections：
 - 集合工具类，定义了除了存取以外的集合常用方法。