



# Università degli Studi dell'Aquila



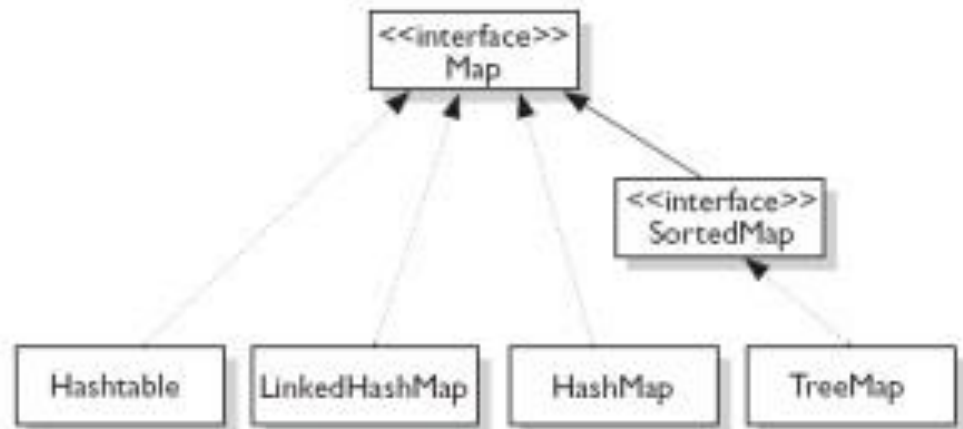
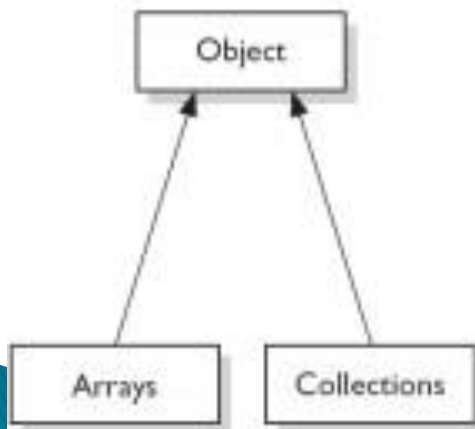
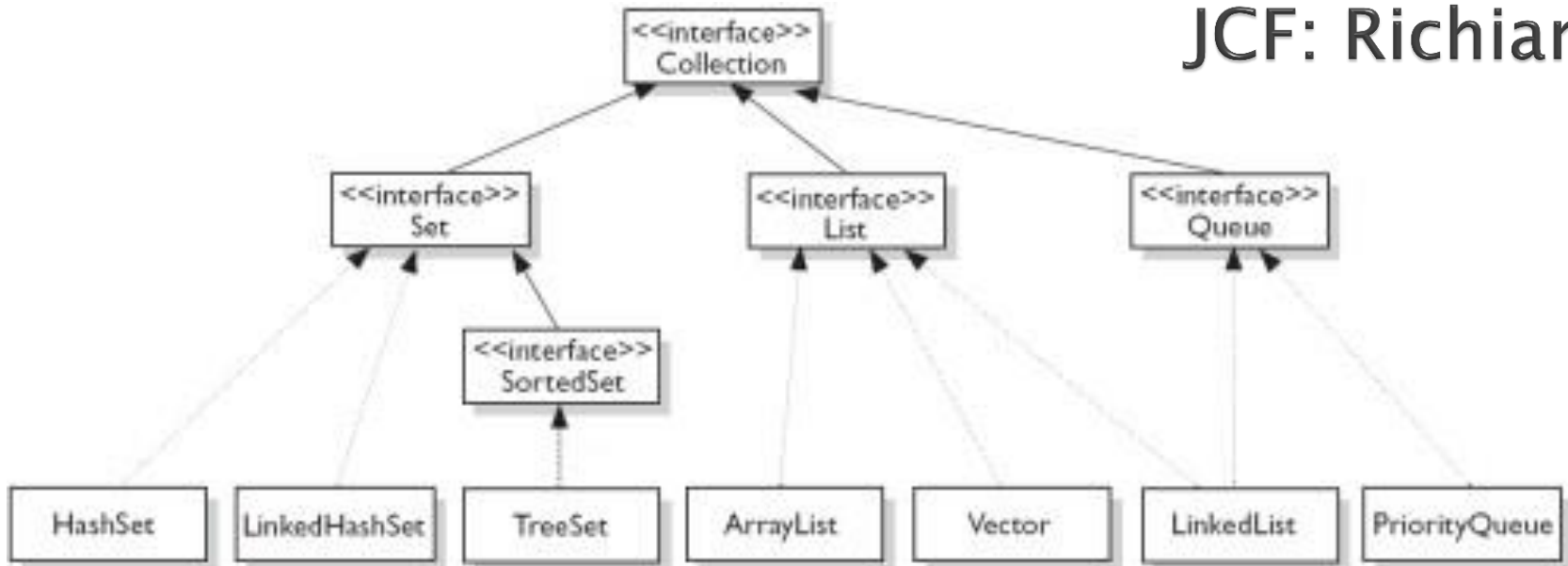
Dipartimento di Ingegneria e Scienze  
dell'Informazione e Matematica

Università degli Studi dell'Aquila

Corso di Algoritmi e Strutture Dati con Laboratorio

La classe `PriorityQueue<E>`

# JCF: Richiami



## L'interfaccia `Queue<E>`: Richiami

- ▶ Estende l'interfaccia `Collection<E>`
- ▶ Specifica i metodi `element()` / `peek()` e `remove()` / `poll()` per l'accesso e la rimozione, rispettivamente, dell'elemento iniziale della coda
- ▶ Eredita il metodo `add()` di `Collection` e specifica il metodo `offer()` per l'inserimento
- ▶ La classe `LinkedList` implementa `Queue`
- ▶ I seguenti metodi di `LinkedList` violano la definizione di coda:
  - ✗ `remove(int index)` e
  - ✗ `add(int index, E Element)`

# Code con priorità

- ▶ È una raccolta in cui può essere rimosso soltanto l'elemento con priorità massima, dato un criterio di confronto fra gli elementi
- ▶ Java.util contiene la classe **PriorityQueue**
- ▶ In ogni esemplare della classe **PriorityQueue** gli elementi sono memorizzati in uno **heap binario** orientato al minimo

# La classe `PriorityQueue<E>`:

## Constructors and Description

- `PriorityQueue()`** Creates a `PriorityQueue` with the default initial capacity (11) that orders its elements according to their natural ordering.
- `PriorityQueue(Collection<? extends E> c)`** Creates a `PriorityQueue` containing the elements in the specified collection.
- `PriorityQueue(Comparator<? super E> comparator)`** Creates a `PriorityQueue` with the default initial capacity and whose elements are ordered according to the specified comparator.
- `PriorityQueue(int initialCapacity)`** Creates a `PriorityQueue` with the specified initial capacity that orders its elements according to their natural ordering.
- `PriorityQueue(int initialCapacity, Comparator<? super E> comparator)`** Creates a `PriorityQueue` with the specified initial capacity that orders its elements according to the specified comparator.
- `PriorityQueue(PriorityQueue<? extends E> c)`** Creates a `PriorityQueue` containing the elements in the specified priority queue.
- `PriorityQueue(SortedSet<? extends E> c)`** Creates a `PriorityQueue` containing the elements in the specified sorted set.

# La classe `PriorityQueue<E>`:

## Methods

**`boolean add(E e)`** Inserts the specified element into this priority queue.

**`void clear()`** Removes all of the elements

**`Comparator<? super E> comparator()`**

Returns the comparator used to order the elements in this queue, or null if this queue is sorted according to the natural ordering of its elements.

**`boolean contains(Object o)`** Returns true if this queue contains the specified element.

**`Iterator<E> iterator()`** Returns an iterator over the elements in this queue.

**`boolean offer(E e)`** Inserts the specified element into this priority queue.

# La classe `PriorityQueue<E>`: Methods

- E peek()** Retrieves, but does not remove, the head of this queue (returns null if the queue is empty)
- E poll()** Retrieves and removes the head of this queue, or returns null if this queue is empty.
- boolean remove(Object o)** Removes a single instance of the specified element from this queue, if it is present.  
Remark: viola la definizione di priority queue!
- int size()** Returns the number of elements in this collection.
- Object[] toArray()** Returns an array containing all of the elements in this queue.

Esempi: **`StringPriorityQueue.java`**  
**`HeapSort_ByPriorityQueue.java`**