



Plataforma Java



Java Collection Framework



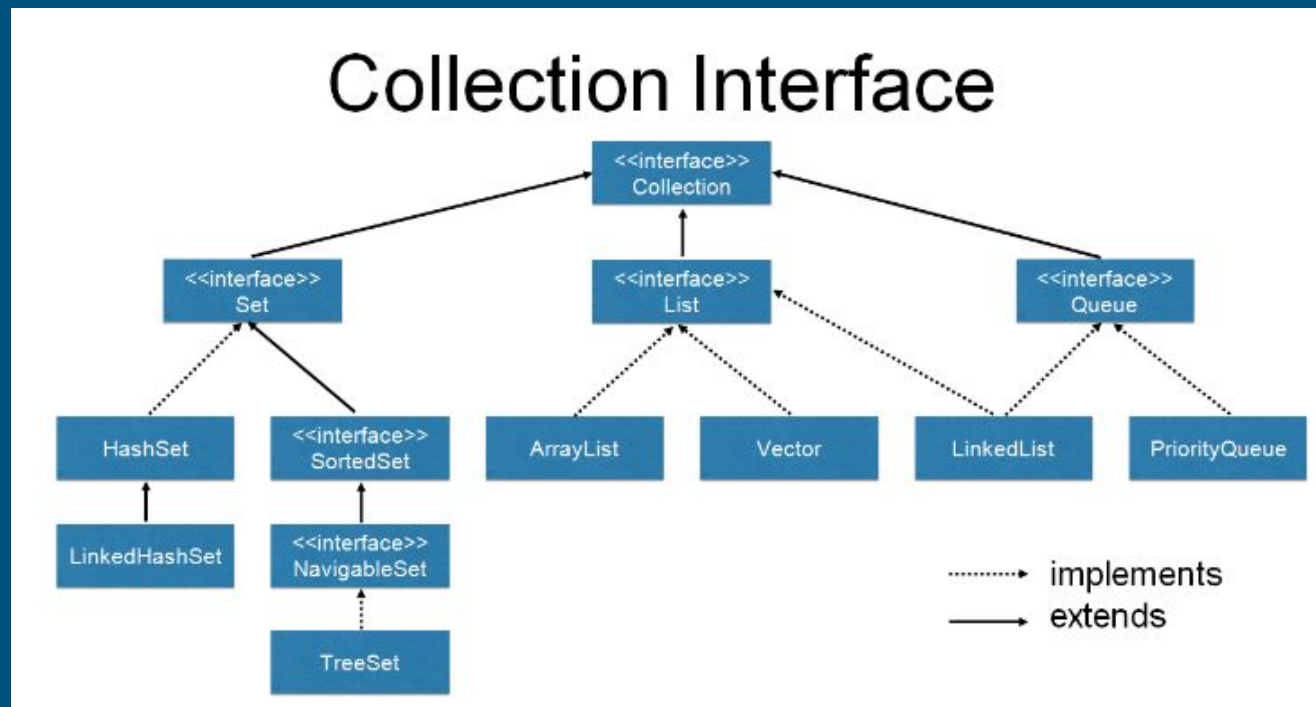
Collection Framework

1. Visão Geral
2. Set
3. List
4. Map
5. Queue
6. Utilitários



Java Collection Framework

Visão Geral



Java Collection Framework

Visão Geral

Interface	Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Set	<u>HashSet</u>		<u>TreeSet</u>		<u>LinkedHashSet</u>
List		<u>ArrayList</u>		<u>LinkedList</u>	
Deque		<u>ArrayDeque</u>		<u>LinkedList</u>	
Map	<u>HashMap</u>		<u>TreeMap</u>		<u>LinkedHashMap</u>

Java Collection Framework

java.util.Collection

```
public interface Collection<E>  
    extends Iterable<E>
```

add(E)

addAll(Collection<? Extends E>)

clear()

contains(E)

containsAll(Collection<?>)

isEmpty()

iterator()

remove(Object)

removeAll(Collection<?>)

retainAll(Collection<?>)

size()

toArray()

Java Collection Framework

Set

```
public interface Set<E>  
    extends Collection<E>
```

A interface Set define a semântica de “*Conjunto Matemático*”, ou seja, não permite itens repetidos

Permite incluir um elemento e se NÃO existe e2 tal que: $(e == \text{null} ? e2 == \text{null} : e.equals(e2))$

Java Collection Framework

HashSet

Implementação mais simples de Set

Utiliza internamente um hash table (HashMap)

Não há garantias de iterar o Set na ordem em que os elementos foram incluídos

Não há garantias de que a ordem dos itens se mantenha constante ao longo do tempo

Permite inclusão de elemento null

Oferece tempo constante nas operações básicas (add, remove, contains, size) desde que o hashCode tenha valor disperso

Java Collection Framework

LinkedHashSet

Capacidade de iterar os elementos na ordem em que foram incluídos, ou na ordem inversa

Utiliza além de uma estrutura hash table (HashMap), utiliza também uma lista duplamente encadeada

Class LinkedHashSet<E>

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractSet<E>
      java.util.HashSet<E>
        java.util.LinkedHashSet<E>
```


Java Collection Framework

Interface SortedSet

Define mecanismo para ordenação de Set pela “*Ordem Natural*” (*Comparable*) ou por um “*Comparator*”.

`Comparator<? Super E> comparator()`

`E first()`

`E last()`

`SortedSet<E> headSet(E)`

`SortedSet<E> subSet(E from, E to)`

`SortedSet<E> tailSet(E)`

Java Collection Framework

TreeSet

Implementação de NavigableSet
baseado em um TreeMap

Elementos ordenados usando a “*Ordem Natural*” (*Comparable*) ou um
Comparator

Class TreeSet<E>

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractSet<E>
      java.util.TreeSet<E>
```

Type Parameters:

E - the type of elements maintained by this set

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, NavigableSet<E>, Set<E>, SortedSet<E>

Java Collection Framework

List

Define uma coleção em que se mantém a ordem com que os itens são adicionados

Permite duplicidade de elementos

Os elementos podem ser acessado através do indice

E `get(i)`

`indexOf(e)`

`lastIndexOf(e)`

`listIterator()`

Java Collection Framework

ArrayList

Implementação de List baseada em Array

```
java.lang.Object  
    java.util.AbstractCollection<E>  
        java.util.AbstractList<E>  
            java.util.ArrayList<E>
```

Java Collection Framework

LinkedList

Implementação de List baseado numa lista duplamente encadeada

Operações que envolvem index precisaram percorrer a lista interna até acessar o elemento desejado

Class LinkedList<E>

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.AbstractSequentialList<E>
        java.util.LinkedList<E>
```

Java Collection Framework

Queue

Define uma estrutura de coleções para Filas (FIFO)

`offer(e)`

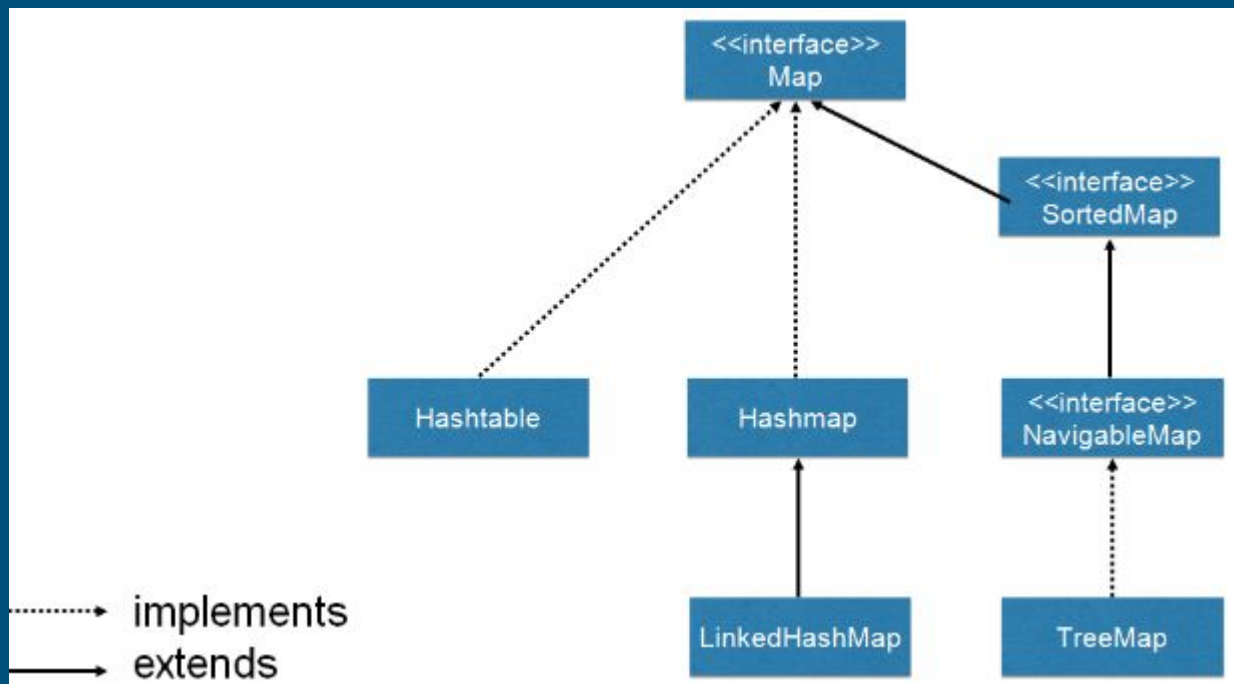
`poll()`

`peek()`

Implementações: `PriorityQueue`, `LinkedBlockingQueue`, `BlockingQueue`, `ArrayBlockingQueue`, `LinkedBlockingQueue`, `PriorityBlockingQueue`

Java Collection Framework

Map



Java Collection Framework

Hashtable

Implementação padrão para estruturas de hash table, ou seja, que mapeia a chave no objeto

Multi-thread

Class Hashtable<K,V>

java.lang.Object

java.util.Dictionary<K,V>

java.util.Hashtable<K,V>

Java Collection Framework

HashMap

Semelhante a Hashtable

Métodos Não sincronizados (não permite multi-thread)

Não mantém a ordem com que os objetos foram inseridos

Java Collection Framework

LinkedHashMap

Mantém a ordem com que os elementos foram adicionados

Lista duplamente encadeada para manter a ordem com que os elementos foram incluídos

Class LinkedHashMap<K,V>

```
java.lang.Object
    java.util.AbstractMap<K,V>
        java.util.HashMap<K,V>
            java.util.LinkedHashMap<K,V>
```

Java Collection Framework

TreeMap

Árvore Red-Black baseada na implementação de NavigableMap

Mantém a ordem natural (Comparable) ou definida por um Comparator

Class TreeMap<K,V>

java.lang.Object

java.util.AbstractMap<K,V>

java.util.TreeMap<K,V>

Java Collection Framework

Classes Utilitárias

`java.util.Iterator`

`java.lang.Iterable`

`java.lang.Comparable`

`java.lang.Comparator`