

Interface SortedMap<K, V>

26

- SortedMap é um Map que mantém as suas entradas em ordem ascendente, e de acordo com a ordem natural das chaves, ou então de acordo com o comparador que tenha sido fornecido aquando da sua criação
- Bastante utilizado em colecções ordenadas de pares chave-valor, como é o caso de dicionários e listas telefónicas
- Para além das operações de Map, permite ainda
 - Aplicar operações a zonas delimitadas do Map
 - Devolver a primeira e a última chave do mapa ordenado
 - Devolver o iterador que percorre os elementos em ordem ascendente
 - Devolver o comparador utilizado na ordenação, se este existir

```
public interface SortedMap<K,V> extends Map<K,V>
```

Interface SortedMap<K, V>

27

Métodos

K firstKey()	Devolve a primeira chave (mais pequena) do mapa
K lastKey()	Devolve a última chave (maior) do mapa
SortedMap<K,V> headMap(K toKey)	Devolve uma vista para o mapa só com pares cuja chave é menor que toKey
SortedMap<K,V> subMap (K fromKey, K toKey)	Devolve uma vista para o mapa só com pares cuja chave é maior ou igual a fromKey e menor que toKey
SortedMap<K,V> tailMap(K fromKey)	Devolve uma vista para o mapa só com pares cuja chave é maior ou igual a fromKey
Comparator<? super K> comparator()	Devolve o comparador usado na ordenação (se existir)

Classe `TreeMap<K, V>`

28

- A classe `TreeMap` é uma implementação da interface `SortedMap` baseada numa árvore binária de pesquisa
 - Permite também `null` como valor e como chave
 - Também suporta acesso eficiente
 - Garante ordem

```
public class TreeMap<K,V> extends AbstractMap<K,V>  
                                implements NavigableMap<K,V>, ...
```

K, o tipo das chaves **V**, o tipo dos valores mapeados

29

Conjuntos



Interface Set<E>

30

- Um Set modela o conceito matemático de conjunto
 - Ex: baralho de cartas
- A interface Set estende Collection com as seguintes restrições
 - Não são admitidos elementos duplicados
 - Não estabelece ordem entre elementos
- A interface não adiciona métodos para além dos que são herdados de Collection



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

Interface Set<E>

31

Métodos

`boolean add(E e)`

Adiciona o elemento indicado ao conjunto se ainda não existir no conjunto

`boolean remove(Object o)`

Remove o elemento indicado do conjunto, se este existir no conjunto

`void clear()`

Remove todos os elementos do conjunto

`boolean contains (Object o)`

Devolve true se o conjunto contém o elemento indicado

`boolean isEmpty()`

Devolve true se o conjunto não contém elementos

`int size()`

Devolve o número de elementos do conjunto

`Iterator<E> iterator()`

Devolve um iterador sobre os elementos do conjunto

Classe HashSet<E>

32

- A classe `HashSet` é uma implementação geral e eficiente da interface `Set`, baseada numa tabela de dispersão
 - Permitem também `null` como valor e como chave
 - Não garante a ordem de iteração ao longo do tempo

```
public class HashSet<E> extends AbstractSet<E>  
    implements Set<E>, ...
```

Interface SortedSet<E>

33

- SortedSet é um conjunto que permite estabelecer o conceito de ordem total entre os seus elementos
- Os elementos são ordenados de acordo com a sua ordem natural, através da implementação da interface Comparable, ou então de acordo com o comparador que tenha sido fornecido aquando da sua criação
- As novas operações que utilizam a ordenação são as seguintes:
 - Aplicar operações a zonas delimitadas do Set
 - Devolver o primeiro e o último elemento do conjunto ordenado
 - Devolver o iterador que percorre o conjunto por ordem ascendente
 - Devolver o comparador utilizado na ordenação, se este existir

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


Interface SortedSet<E>

34

Métodos

E first()

Devolve o primeiro elemento (mais pequeno) do conjunto

E last ()

Devolve o último elemento (maior) do conjunto

SortedSet<E> headSet(E toElem)

Devolve uma vista para o conjunto só com elementos menores que toElem

SortedSet<E> subSet(E fromElem, E toElem)

Devolve uma vista para o conjunto só com elementos maiores ou iguais a fromElem e menores que toElem

SortedSet<E> tailSet(E fromElem)

Devolve uma vista para o conjunto só com elementos maiores ou iguais a fromElem

Comparator<? super E> comparator()

Devolve o comparador usado na ordenação (se existir)

Classe `TreeSet<E>`

35

- A classe `TreeSet` é uma implementação geral e eficiente da interface `SortedSet`, baseada numa árvore binária de pesquisa
 - Permitem também `null` como valor e como chave
 - Garante a ordem de iteração ao longo do tempo

```
public class TreeSet<E> extends AbstractSet<E>  
                           implements SortedSet<E>, ...
```

Java Framework

36

			Implementações			
			Resizable array	Linked list	Balanced tree	Hash table
Interfaces	Collection	Set				HashSet
		SortedSet			TreeSet	
		List	ArrayList	LinkedList		
		Map				HashMap
		SortedMap			TreeMap	

Java Framework

37

