



Coleções de Objetos

Interface Map

Classe HashMap

Profa. Karen Selbach Borges



Interface Map



- Manipula grupos de objetos, onde cada objeto está associado a uma chave de busca.
- Não permite duas chaves iguais.
- Exemplo de uso :
 - Catálogo de produtos (cod x descrição)





Métodos da Interface Map

- **V put (K key, V value)** : insere um novo par chave-valor na coleção. Se já existir um valor associado à chave, esse será substituído pelo novo valor e retornado pelo método.
- **Exemplo :**

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
String i = m.put(123, "Ana"); // i = Maria
```



Métodos da Interface Map



- **void putAll (Map<K,V>, m) :** faz a união entre duas coleções
- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234, "Jose");  
m.put(123, "Maria");  
Map<Integer,String> m2 = new HashMap <Integer,String> ();  
m2.put(567, "Pedro");  
m2.put(543, "Ana");  
m.putAll(m2);
```



Métodos da Interface Map



- **V get(Object key)** : retorna o valor associado à chave indicada como argumento
- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
System.out.println(m.get(234)); // José
```



Métodos da Interface Map



- **V remove (Object key)** : remove um valor da coleção, identificado pela chave indicada como argumento

- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
m.remove(123); // Remove Maria
```



Métodos da Interface Map



- **void clear()** : apaga todo o conteúdo da coleção.
- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
m.clear();
```



Métodos da Interface Map



- **boolean containsKey(Object key) :**
retorna True se a coleção contém a chave indicada como argumento

- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
System.out.println(m.containsKey(123)); // True
```





Métodos da Interface Map

- **boolean containsValue(Object value) :**
retorna True se a coleção contém uma ou mais entradas com o valor indicado como argumento

- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
System.out.println(m.containsValue("Maria")); // True
```



Métodos da Interface Map



- **int size()** : retorna o número de entradas na coleção.
- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
System.out.println(m. size()); // 2
```



Métodos da Interface Map



- **boolean isEmpty()** : retorna True se a coleção não contiver objetos

- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
System.out.println(m.isEmpty()); // False
```



Métodos da Interface Map



- **Set keySet()** : retorna um conjunto de todas as chaves contidas no mapa.
- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
Set chaves = m.keySet(); // chaves = {234,123}
```



Métodos da Interface Map



- **Set entrySet()** : retorna um conjunto de todas as entradas do mapa (par chave-valor)

- Exemplo :

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
Set valores = m.entrySet();  
// valores = {[234,"Jose"],[123, "Maria"]}
```



Métodos da Interface Map



- **Collection values()** : retorna uma coleção de todos os valores contidos no mapa.

Exemplo :

```
Map<Integer,String> m = new HashMap  
    <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
Collection c = m.values();
```



Classe Map.Entry



- Representa um par chave/valor.
- Principais métodos :
 - K **getKey()**: retorna a chave
 - V **getValue()**: retorna o valor
 - V **setValue(V valor)**: muda o valor no mapa associado para o novo valor e retorna o valor antigo.



Classe HashMap



- Possui todos os métodos de Map e tem os seguintes construtores :
 - **HashMap()**: 16 posições iniciais
 - **HashMap(Map entradas)**
 - **HashMap(int capacidadeInicial)**



Classe HashMap



- Para percorrer um HashMap, é necessário utilizar a classe Map.Entry
- Exemplo

```
Map<Integer,String> m = new HashMap <Integer,String> ();  
m.put(234,"Jose");  
m.put(123, "Maria");  
for (Map.Entry<Integer,String> entry : m.entrySet())  
    System.out.println("Chave = " + entry.getKey() + " Valor = " +  
entry.getValue());
```

