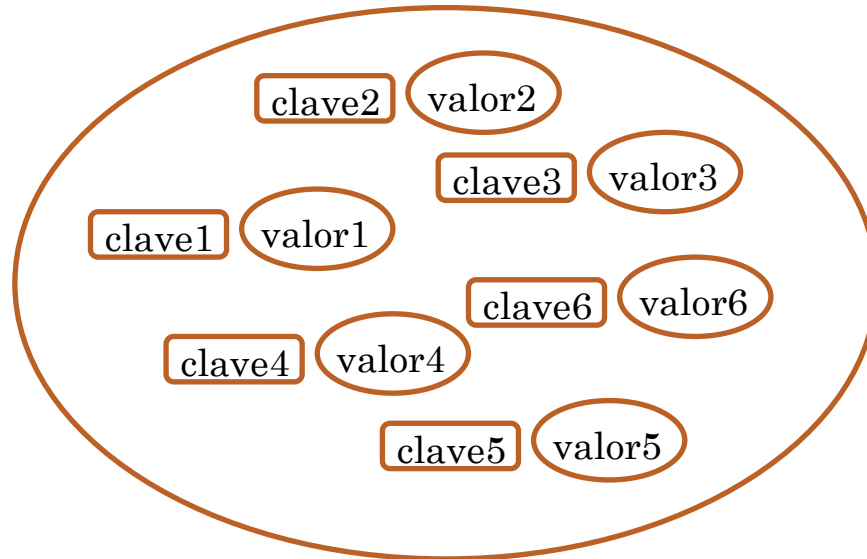


COLECCIONES II



COLECCIONES TIPO TABLA

➤ Los elementos no tienen posición ni orden, pero si una clave única asociada. La información se almacena en parejas clave-valor



LA CLASE HASHMAP

➤ Se trata de la clase de colección de tipo tabla más utilizada.

➤ Para crear un objeto de la misma:

```
HashMap<T_clave, T_valor> variable=new HashMap<>();
```

➤ Al igual que ArrayList, HashMap es de tipos genéricos. Al crear un objeto debemos indicar el tipo de la clave y el del dato o valor. En este ejemplo creamos una tabla de cadenas de caracteres con clave entera:

```
HashMap<Integer,String> tabla=new HashMap<>();
```



MÉTODOS HASHMAP I

➤ **T put(K clave, T dato).** Añade el dato a la colección y le asocia la clave indicada como primer parámetro. Si ya existiera esa clave, el valor existente será sustituido por el nuevo:

```
HashMap<Integer,String> tabla=new HashMap<>();  
tabla.put(200, "dato1");  
tabla.put(400, "dato2");  
tabla.put(200,"dato3"); //dato1 es sustituido por dato3
```

➤ **T get(K clave).** Devuelve el dato que tenga asociada dicha clave. Si no hay ninguno, devolverá null.

➤ **int size().** Devuelve el tamaño de la colección



MÉTODOS HASHMAP II

➤ **T remove(Object clave).** Elimina el dato que tenga dicha clave asociada y devuelve el elemento eliminado.

```
System.out.println(tabla.remove(400)); //muestra dato2
```

➤ **Collection<T> values().** Devuelve una colección solo con los valores. Puede utilizarse para recorrer el conjunto de valores con un for each:

```
Collection<String> datos=tabla.values();  
for(String s:datos){  
    System.out.println(s); //muestra cada elemento  
}
```



MÉTODOS HASHMAP III

- **Set<T> keySet().** Devuelve un conjunto con las claves
- **boolean containsKey(K clave).** Indica si hay algún elemento en la colección con dicha clave asociada.
- **boolean containsValue(T valor).** Indica si el elemento está presente en la colección

