

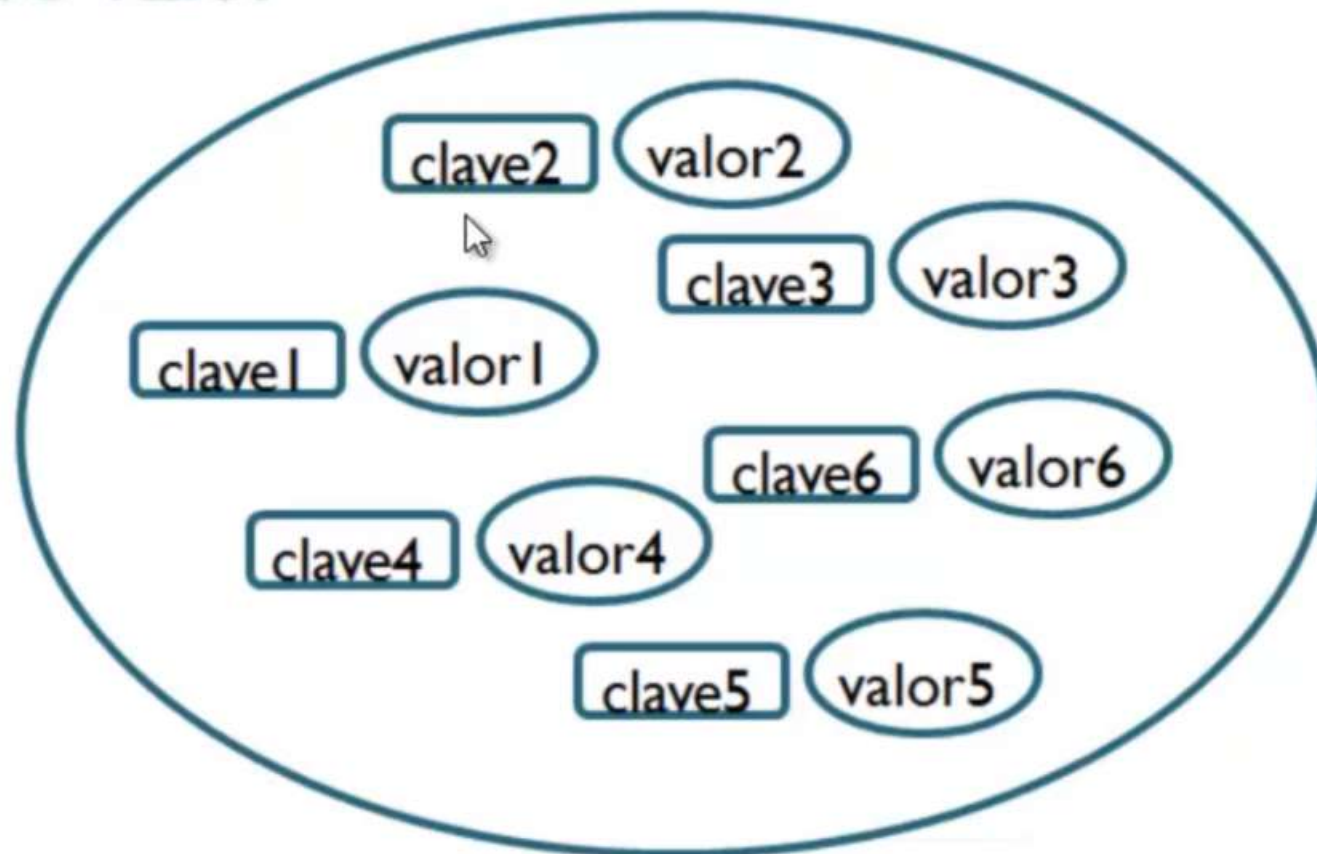
**Sección: Clases de uso general**

# 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

- **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

# Enunciado

➤ Realizar un programa para la gestión de una agenda de contactos en el que se mostrará el siguiente menú:

- 1.- Anadir contacto
- 2.- Buscar contacto
- 3.- Eliminar contacto
- 4.- Mostrar todos
- 5.- Salir

➤ Cada contacto está formado por tres datos: nombre, email y edad. El campo email será el campo clave del contacto, de modo que no podrá haber dos contactos con el mismo email

# Enunciado(II)

➤ Al elegir la opción 1, se solicitarán los tres datos del contacto y si el email no está repetido se añadirá. Cuando se elija la opción 2, se solicitará el email para realizar la búsqueda, si se encuentra, se muestran todos los datos del contacto. Con la opción 3, se solicita el email y se elimina el contacto que tenga dicha clave asociada. Con la opción 4 se muestran los datos de todos los contactos y con la 5 se abandona el programa

