

0101010
0100101
1101010

UD8.3- ESTRUCTURES DE DADES DINÀMIQUES Col·leccions (Mapes)

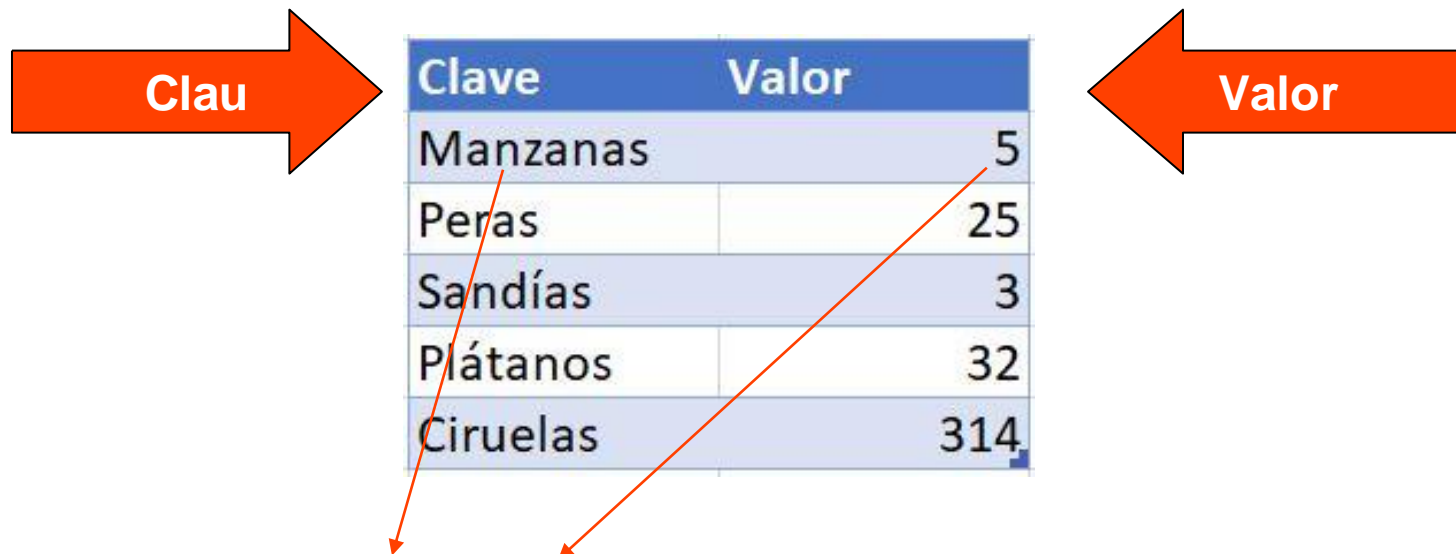
Programació – 1er DAW/DAM

CONTINGUTS

- **Map (HashMap)**
 - Declaració i creació
 - Inserció, modificació i eliminació d'elements
 - Recorregut

1. La interfície Map

- Els diccionaris (**Mapes**) permeten l'emmagatzematge de dades en memòria en forma de parells **clau-valor**.
 - La clau (**key**) ens permet accedir al contingut (**value**)
 - Similar als arrays però en aquest cas, la **key** pot ser un **objecte** de qualsevol **tipus**.



Clave	Valor
Manzanas	5
Peras	25
Sandías	3
Plátanos	32
Ciruelas	314

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

1. La interfície Map

- En general, un **mapa** és gestionat com un **conjunt** (Set) en què els **elements** són les **claus** i cada clau té un **valor associat**.
 - Les claus permeten l'accés, la modificació i l'esborrament de l'element.

```
// > Creació d'un mapa buit: associa cadenes amb sencers
HashMap<String, Integer> mapa =new HashMap<String, Integer>();

// > Inserció d'entrades (clau, valor)
mapa.put("dos",2);
mapa.put("tres",3);

// > Recuperació d'entrades per clau
System.out.println(mapa.get("tres"));
```

2. La implementació HashMap

- Java disposa de diferents implementacions:

- **HashMap**<K,V>: Els elements no estan ordenats.
- **TreeMap**<K,V>: Els elements es troben sempre ordenats (s'utilitza la interfície **compareTo**).
- **LinkedHashMap**<K,V>: Implementació basada en una llista ordenada.

2.1 Creació d'un objecte HashMap

- En dues passes:

```
HashMap <Tipus_key, Tipus_value> nomDiccionari;  
nomDiccionari = new HashMap<>();
```

- En només un pas:

```
HashMap <Tipus_key, Tipus_value> nomDiccionari =new HashMap<>();
```

Exemples:

```
HashMap<String, String> diccionari = new HashMap<>();
```

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

2.2 Afegir, inserir i consultar elements

- `Object put(Object clau, Object valor)`
 - Els elements es col·loquen en un ordre determinat i s'identifiquen a través de la clau.
- `Object get(Object clau)`
 - Recupera l'element identificat per la clau

Exemple:

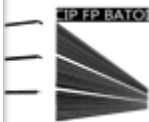
```
// Creació d'un mapa buit: associa cadenes amb cadena  
HashMap<String,String> diccionariAnglesCastella = new HashMap<>();  
  
// Inserció d'entrades (clau, valor)  
diccionariAnglesCastella.put("cat","gat");  
diccionariAnglesCastella.put("dog","gos");  
  
System.out.println(diccionariAnglesCastella .get("cat")); // gat
```

2.3 Buscar un element

- `boolean containsKey(Object elementBuscat)`
 - Retorna **true/false** si el mapa conté l'element com a clau
- `boolean containsValue(Object elementBuscat)`
 - Retorna **true/false** si el mapa conté l'element com a valor.
- Internament, utilitzarà el mètode `equals` de la classe `Object`

Exemple:

```
HashMap<Integer,String> m =new HashMap<Integer,String>();  
m.put(924,"Amalia Núñez");  
m.put(921,"Cindy Nero");  
  
if(m.containsKey(921)) {  
    System.out.println(m.get(921)); //Cindy Nero  
}
```



2.4 Recórrer una llista

- Un mapa **no és iterable**. No obstant això, el contingut es pot recórrer a través de les claus.

```
HashMap<Integer,String> m =new HashMap<Integer,String>();  
m.put(924,"Amalia Núñez");  
m.put(921,"Cindy Nero");  
  
for(Map.Entry parella: m.entrySet()) {  
    System.out.print(parella.getKey() +"\t");  
    System.out.println(parella.getValue());  
}
```

2.5 Altres operacions

- Encara que un **mapa no és una col·lecció**, ofereix operacions similars a les col·leccions:
 - `size()`
 - `isEmpty()`
 - `remove(Object clau)`
 - `clear()`
 - `putAll(Map<K, V> altreMapa)`

<https://docs.oracle.com/javase/10/docs/api/java/util/HashMap.html>

Activitats prèvies (I)

- **Activitat 9.-** Crea un **mini-diccionari espanyol-anglès** que continga, almenys, 20 paraules (amb la seua traducció corresponent). Utilitza un objecte de la classe `HashMap` per emmagatzemar les parelles de paraules. El programa demanarà una paraula en espanyol i donarà la traducció corresponent en anglès.
 - Per acabar, escriu un mètode que ens permeti mostrar tot el contingut del diccionari.
- **Activitat 9 (Continuació).-** Realitza un programa que **esculla a l'atzar 5 paraules en espanyol** del minidiccionari. El programa anirà demanant que l'usuari escrigua la traducció a l'anglès de cadascuna de les paraules i comprovarà si són correctes. En acabar, el programa haurà de mostrar quantes respostes han sigut correctes i quantes incorrectes



Activitats prèvies (II)

Activitat 10.- Implementa el **control d'accés**(login) a una àrea restringida. Per això:

- Crea una classe `LoginService` que s'encarregue de; manejar els intents d'usuari, i dur a terme les verificacions de si un usuari i password són vàlids.
- Seguidament crea una classe `TestLoginService` que **demane** un **nom d'usuari** i **contrasenya** i ens diga si podem entrar a l'aplicació o no. Has de tenir en compte:
 - L'usuari tindrà un **màxim de 3 oportunitats**. Si s'esgoten el programa dirà “*Error! l'usuari o password introduït és incorrecte*”.
 - Els noms d'usuari amb les contrasenyes corresponents han d'estar emmagatzemats en una estructura de tipus `HashMap`
 - Si l'usuari introdueix correctament les dades, el programa dirà “*Ha accedit a l'àrea restringida*”. i els reiniciaran el número d'intents.
 - Si arriben al número màxim d'intents i no hem pogut accedir a la aplicacion (esta bloquejat), el programa ha de dir. “*Usuari bloquejat. Consulte a l'administrador*”

A login form with the following elements:

- Username input field
- Password input field
- ☐ Remember Me
- A yellow button labeled **LOGIN**

LoginService
+ credencialesValidas: HashMap<String, String>
+ intentos: int
+ <u>MAXIMO_INTENTOS: int</u>
+ login(usuari:String , password:String): boolean
+ isBloqueado: boolean

Activitats Prèvies (II)

Exemple d'Execució

```
BatoiGestió aplicació 1.0
=====

Introdueix el nom d'usuari: root
Introdueix el password: pepe
Error! L'usuari o password introduït és incorrecte

Introduïu el nom d'usuari: root
Introdueix el password: 123456789
Ha accedit a l'àrea restringida
```

- Això és tot... de moment :-)