

**Objectius:**

- Programar Hashmaps en JAVA

**Instruccions:**

- Es tracta d'un treball en grups de dos, no s'admet cap tipus de còpia.
- Responeu a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.
- Es valorarà la presentació i els comentaris al codi

**Criteris d'avaluació:**

- Tots els exercicis tenen la mateixa puntuació
- Es valorarà la presentació i els comentaris al codi

**Entrega:**

- Un arxiu .zip anomenat: **PRx.y-NomCognom-NomCognom.zip**
  - PRx.y correspon al codi de la pràctica, per exemple PR1.1
  - NomCognom correspon al nom i primer cognom de cada participant
- L'arxiu .zip conte:
  - Aquest document emplenat en format **.pdf** anomenat **memoria.pdf**
  - Els arxius necessaris per fer anar la pràctica

**Noms i Cognoms:** Alejandro León, Pablo Mejías

**Materials:**

Necessiteu una eina per programar en JAVA

Feu servir Google per buscar els tutorials que us serveixin millor

**Tasques**, a cada exercici feu l'explicació i captures que cregueu convenientes



## Exercici 0

Partint de la base de l'API java.util, treballarem les estructures conegudes com mapes (Map). Existeixen diversos tipus però en aquest cas ens interessen dos implementacions que en un moment donat ens poden servir per emmagatzemar dades, en aquest cas indexades per una clau, i no per un número d'índex. El Map i el SortedMap són dos interfaces implementats respectivament amb HashMap i SortedMap.

Crea la classe Agenda a partir del codi que segueix:

```
import java.util.*;
public class Agenda {
    private HashMap<String, String> persones = new HashMap<String, String>();
    public static void main(String args[]){
        // Crearem una nova agenda
        Agenda a = new Agenda();
        // Afegirem una sèrie de parells <key,value> a l'agenda
        a.persones.put("Metge", "(+52)-4000-5000");
        a.persones.put("Casa", "(888)-4500-3400");
        a.persones.put("Germa", "(575)-2042-3233");
        a.persones.put("Oncle", "(421)-1010-0020");
        a.persones.put("Sogres", "(334)-6105-4334");
        a.persones.put("Oficina", "(304)-5205-8454");
        a.persones.put("Advocat", "(756)-1205-3454");
        a.persones.put("Pare", "(55)-9555-3270");
        a.persones.put("Botiga", "(874)-2400-8600");
        // Cridem un mètode que mostrarà la llista de contactes de l'agenda
        a.mostrarAgenda();
        // Definirem un Array amb determinats contactes de l'agenda
        String contactes[] = {"Oncle", "Sogres", "Advocat"};
        // Eliminar els valors continguts a l'Array
        for(int i = 0; i < contactes.length; i++){
            a.persones.remove(contactes[i]);
        }
        a.mostrarAgenda();
    }
    public void mostrarAgenda(){
        System.out.println(">Agenda amb " + this.persones.size() + " telèfons");
        for(Iterator i=this.persones.keySet().iterator(); i.hasNext();){
            String k=(String)i.next();
            String v=(String)this.persones.get(k);
```



```
        System.out.println("  "+k+ " : " +v);  
    }  
}  
}
```

Insereix "Casa" a l'array de contactes. Què succeeix?

Com t'ho faries per inserir "Casa"?

Crea un mètode que et permet inserir els valors de l'Array a l'agenda, crida'l i torna a mostrar-la. Què ha passat amb el telèfon de "Casa"?

## Exercici 1

Crea la classe Agenda2 amb el codi següent:

```
import java.util.*;  
  
public class Agenda2{  
  
    private TreeMap llista = new TreeMap();  
  
    public static void main(String args[]){  
  
        // Crearem una nova agenda ordenada  
        Agenda2 a2 = new Agenda2();  
  
        // Afegirem una sèrie de parells <key,value> a l'agenda  
        a2.llista.put("Metge", "(+52)-4000-5000");  
        a2.llista.put("Casa", "(888)-4500-3400");  
        a2.llista.put("Germa", "(575)-2042-3233");  
        a2.llista.put("Oncle", "(421)-1010-0020");  
        a2.llista.put("Sogres", "(334)-6105-4334");  
        a2.llista.put("Oficina", "(304)-5205-8454");  
        a2.llista.put("Advocat", "(756)-1205-3454");  
        a2.llista.put("Pare", "(55)-9555-3270");  
        a2.llista.put("Botiga", "(874)-2400-8600");  
  
        // Cridem un mètode que mostrarà la llista de contactes de l'agenda.  
        // L'orde establert als element inserits al TreeMap és ascendent
```



```
// del seu ordre d'inserció.  
mostrarAgenda(a2.llista);  
  
// Definirem dos subarbres utilitzant mètodes específics  
SortedMap A0=a2.llista.subMap("A", "O");  
SortedMap PZ=a2.llista.tailMap("P");  
  
System.out.println("---- Agenda A-O ----");  
mostrarAgenda(A0);  
System.out.println("---- Agenda P-Z ----");  
mostrarAgenda(PZ);  
}  
  
public static void mostrarAgenda(Map m){  
    System.out.println("> Agenda amb " + m.size() + " telefons");  
    for(Iterator i=m.keySet().iterator(); i.hasNext();){  
        String k=(String)i.next();  
        String v=(String)m.get(k);  
        System.out.println("    "+k+ " : " +v);  
    }  
}  
}
```

Què implica que el mètode 'mostrarAgenda' sigui 'static' en aquest cas?

Compara-ho amb la crida de l'exercici 0.

Permet executar aquesta funció sense necessitat d'instanciar la classe Agenda2. Per això en Agenda no fa falta passar-li paràmetres, ja que agafa els seus propis atributs.



## Exercici 2

Imagineu que treballarem en una cadena de botigues de venda de videojocs. Necessitem emmagatzemar els diferents videojocs ( en ordre ascendent ) que vendrem i el preu que tindran. És vol que tingui aquestes funcions bàsiques:

- Introduir un element (és necessari controlar que l'element existeixi dins del Maps o no, i en el cas que ja existeixi, retornar el missatge d'error "El producte ja existeix").
- Modificar el seu preu (és necessari controlar que l'element que volem modificar existeixi, en cas que no existeixi retornar el missatge "Producte desconegut")
- Eliminar un producte (comprovar si el producte existeix o no existeix com en el cas anterior, i abans d'esborrar preguntar: "Estàs segur d'esborrar el producte ABC" on ABC és el nom del producte a esborrar).
- Mostrar tots els productes ordenats alfabèticament.
- Aquestes funcions han d'estar gestionades per un menú.
- A més, hem d'assegurar-ens que no podem posar noms de videojocs (string) NULS i els preus han de permetre decimals.