

[🏠](#) > [Els Meus Cursos](#) > [Programació-1CFSC LFP 1726475259](#) > [UT8. COL·LECCIONS. CLASSES GENÈRIQUES.](#) >
Exercicis classe

Exercicis classe

1.- Crea un ArrayList capaç d'emmagatzemar els valors: "hola", 2.5, false, "adéu"

Podrem recórrer l'ArrayList i imprimir per pantalla cadascun dels objectes? Podrem usar el mètode "length()" en els elements?

2.- Crea un programa que emmagatzeme 5 Strings i els mostre en ordre invers.

3.- Crea una classe anomenada "LlistaTemperatures" que herete de la classe ArrayList (de Doubles). Esta classe "LlistaTemperatures" haurà d'afegir als mètodes heretats el mètode "acumulat" que mostre la suma de totes les temperatures de la llista, el mètode "mitjana" que mostre la temperatura mitjana de les que apareixen en la llista, el mètode "freqüència" que donada una temperatura mostre quantes voltes apareix en la llista, el mètode "ordenar" que ordene la llista de temperatures, el mètode "desordenar" que mescle les temperatures de la llista, i el mètode "invertir" que invertisca l'ordre de les temperatures de la llista.

Comprova el funcionament de la classe creant un programa que guarde 5 temperatures en una LlistaTemperatures, i aplicant els mètodes creats.

4.- Crea una jerarquia de classes amb una classe base anomenada Persona que tinga un nom i un DNI, i dos classes derivades Alumne (curs, centre i llista de mòduls que cursa), i la classe tècnic (empresa que treballa i càrrec que exerceix).

a) En la classe Alumne has de crear mètodes per a afegir mòduls a la llista (després d'afegir-los s'ha d'ordenar la llista), per a eliminar mòduls individuals i per a esborrar la llista completa de mòduls.

b) Crea un programa amb un menú senzill que tinga una llista de persones on es puguin afegir persones, modificar persones, buscar i mostrar una persona, mostrar la llista de persones i esborrar persones.

5.- Defineix i implementa una funció "buscar" que localitze un element en un array d'elements, tots dos se li passen com a arguments. Prova-la amb enters. Després fes-la genèrica, per a qualsevol tipus d'objectes. Suposa que els objectes tenen un mètode equals(obj o) que compara l'objecte en curs (this) amb l'objecte que rep com a paràmetre. Este mètode retorna true si són iguals i false si no ho són.

6.- Implementa una classe anomenada ArrayDoble que continga dos ArrayList com a atributs. Esta classe ArrayDoble tindrà dos tipus genèrics (el primer ArrayList serà del primer tipus genèric i el segon ArrayList serà del segon tipus genèric).

Implementa el constructor, a més dels mètodes per a afegir un parell (un element en cada llista), eliminar el parell d'elements de l'índex donat, imprimir el parell d'elements de l'índex donat o bé mostrar l'ArrayDoble complet (tots els parells de valors).

7.- Dissenya i implementa la classe Poma sabent que:

- Té un atribut numèric enter anomenat sabor que indica el nivell de sabor de la poma en una escala de 0 a 10. 🍏 Una poma qualsevol té inicialment un valor mitjà de sabor, encara que l'usuari pot decidir eixe valor en crear-la. Es desitja poder decidir si una poma és millor que una altra sobre la base del seu sabor. Aquesta funcionalitat pot permetre, per exemple, crear un vector de pomes i ordenar-les de menor a major sabor.

8.- Funcionarà el següent codi? En cas negatiu, proposa una correcció:



```
public class TotalNumeros {

    public static double sumar (ArrayList <Number> lista) {
        double total = 0;
        for (Number elemento: lista) {
            total += element.doubleValue();
        }
        return total;
    }

    public static void main (String[] args) {
        Integer[] enteros = {1, 2, 3, 4};
        ArrayList<Integer> listaenteros = new ArrayList<Integer>();
        for (Integer elemento: enteros) {
            listaenteros.add (elemento);
        }
        System.out.printf("listaenteros contiene: %s \n",
listaenteros);
        System.out.printf("Total de los elementos listaenteros: %.1f
\n", sumar(listaenteros));
    }
}
```

9.- Un col·legi vol emmagatzemar el conjunt de professors que té en plantilla. Cada professor tindrà un DNI que l'identifica, un nom, uns cognoms i el nom del departament al qual pertany. Desenvolupa la classe professor i un programa principal en el qual comproves la creació i funcionament de la col·lecció de professors (tenint en compte que han d'emmagatzemar-se ordenats per DNI, que no pot haver-hi 2 professors repetits i que es considera el mateix professor si coincideix el seu DNI).

10. El col·legi anterior vol emmagatzemar també una llista d'alumnes associant cada alumne a un tutor (professor) concret. Afeg la classe Alumne que tindrà un NIA (enter) i un nom. Modifica el programa principal de l'exercici anterior per a emmagatzemar un mapa on es relacionen alumnes amb professors, comprovant que no es poden introduir claus repetides. Prova de recórrer el mapa, i a buscar claus i obtindre els seus respectius valors.

11. Implementa un mètode anomenat "frecuenciaDeParaules" al qual li entra com a argument un String. Suposant que l'String que passem com a paràmetre conté unes quantes paraules (cada paraula separada per un espai en blanc), volem que el mètode retorne un mapa que continga per a cada paraula el número de vegades que apareix en l'String. Fes ús de ArrayList i TreeMap per a resoldre el problema.

----- EXERCICIS DE REPÀS -----

1. Crea un conjunt anomenat "signes".

Insereix el conjunt de signes del zodíac.

Realitza un bucle sobre els signes del conjunt i mostra els seus noms.

Consulta si en el conjunt existeix el signe «Càncer». Avisa si existeix o no.

Crea un segon conjunt "signes2" amb els signes "Àries" i "Escorpió".

Consulta si tots els elements de "signes2" existeixen en "signes".

Afegeix a "signes2" el signe "Altre".

Crea el conjunt "signes3" que siga el resultat de l'unió dels conjunts "signes" i "signes2"

Elimina del conjunt "signes" tots els elements que estiguen també en "signes2".

Elimina tots els elements del conjunt "signes2" i mostra posteriorment la quantitat d'elements que té el conjunt "signes2".

2. Donada la classe "Persona" (dni, nom i edat) volem crear un conjunt ordenat de persones, on sempre seran ordenades per



```
Persona persona1 = new Persona("1111111A", "Anna", 28);
```

```
Persona persona2 = new Persona("2222222B", "Pau", 18);  
Persona persona3 = new Persona("3333333C", "Jordi", 16);  
Persona persona4 = new Persona("4444444D", "Maria", 17);  
Persona persona5 = new Persona("5555555E", "Laura", 25);
```

Imprimeix per pantalla el conjunt

Retorna la última persona menor de 18 anys que es trobe en el conjunt. (Persona que més s'apropa als 18 per baix, sense incloure els 18)

Retorna la primera persona que tinga una edat superior o igual a 18 anys. (Persona que més s'apropa als 18 per dalt, inclòs els 18)

Retorna la primera persona que tinga una edad superior als 25 anys. (Persona que més s'apropa als 25 per dalt, sense incloure els 25)

Crea ara un ArrayList de persones que continga duplicats i no guarde ordre d'edat.

Insereix els elements de la llista en el conjunt creat anteriorment i imprimeix-lo per pantalla.

3. Ens interessa emmagatzemar en una estructura de dades la següent informació d'alumnes i notes (No és important l'ordre en que es registren les parelles):

Astor Whitehead	7
Maddison Carlyle	5
Thorley Heath	4
Moirra Norris	5
Travers Winton	7
Jem Eldred	5

- Imprimeix l'estructura de dades creada.
- Imprimeix si hi ha algun alumne amb una nota de 5 (true o false)
- Imprimeix si hi ha algun alumne amb nom John Doe (true o false)
- Imprimeix la nota que ha obtingut Travers Winton
- Imprimeix el conjunt d'alumnes (només els noms, sense nota)
- Imprimeix el conjunt de notes de forma ordenada (sense nom, només les notes, i sense duplicats).
- Imprimeix l'estructura de dades amb el següent format:

Astor Whitehead (7)

Maddison Carlyle (5)

Thorley Heath (4)

4. Desarrollar un sistema de reservas para un hotel que permita gestionar las reservas de diferentes tipos de habitaciones. El sistema debe utilizar un mapa para almacenar las reservas, donde la clave es el tipo de habitación (representado por un enumerado llamado **TipoHabitacion**) y el valor es una lista de nombres de clientes que han reservado ese tipo de habitación.

Descripción:

El sistema debe proporcionar la siguiente funcionalidad:

1. Añadir una reserva para un cliente en un tipo específico de habitación.
2. Cancelar una reservación para un cliente en un tipo específico de habitación (dando el tipo y el nombre).
3. Mostrar todas las reservas para cada tipo de habitación.

El enumerado **TipoHabitacion** debe incluir los siguientes tipos de habitaciones: SENCILLA, DOBLE, TRIPLE, SUITE.



categorías basadas en su nota final y almacene los resultados en un mapa.

Descripción:

Se proporciona una lista de 10 alumnos, cada uno con atributos dni, nombre, edad, y notaFinal. El objetivo es crear un método estático que genere un mapa con las claves INSUFICIENTE, SUFICIENTE, NOTABLE, y SOBRESALIENTE (debe mantener ese orden), correspondientes a diferentes rangos de notas finales. Cada clave tendrá asociado un Set de alumnos cuya nota final cumpla con el criterio de la clave.

Última modificació: Thursday, 2 May 2024, 11:29 AM

[← Ejemplos TreeSet con Comparator](#)

Salta a...



[Exercicis resultats >](#)

© 2024 Generalitat

Conselleria d'Educació, Cultura, Universitats i Ocupació

[Avís legal](#) | [Política de Privacitat](#) | [Política de Cookies](#) | [Contacte](#) | [Accessibilitat](#)



UNIÓ EUROPEA
Fons Europeu de Desenvolupament Regional

