

GUIA POO N° III

COLLECTIONS - JAVA



EJERCICIOS DE APRENDIZAJE

1- Realizar en la clase Main, una lista de nombres dados por el usuario, tanto la cantidad de nombres a ingresar y los nombres por ingresar.

Luego eliminaremos el nombre en la ubicación (guiarnos por el índice), numero tres y uno.

Vamos a ordenar la lista alfabéticamente. ✂ Llamamos a la clase Collections.sort(nombreLista);

Después lo mostramos por pantalla el tamaño de la lista (recordemos el método size), y la lista completa de los nombres; aplicar la lógica correspondiente para dicho caso.

2- En el main, realizaremos una lista de contactos, en la cual alojaremos números de contactos dados por el usuario.

Se ingresaran los respectivos números de contactos mediante un bucle, y luego se guardaran en la lista, aplicar la lógica respectiva para estos manejos de datos. También se imprimirán o visualizarán en orden por pantalla/ consola una vez que el usuario decida que ya no se agreguen más números a la lista.

3- Diseñar un programa que lea y guarde razas de perros en un ArrayList de tipo String.

El programa pedirá una raza de perro en un bucle, el mismo se guardará en la lista y después se le preguntará al usuario si quiere guardar otro perro o si quiere salir. Si decide salir, se mostrará todos los perros guardados en el ArrayList.

4- Continuando el ejercicio anterior, después de mostrar los perros, al usuario se le pedirá un perro y se recorrerá la lista con un Iterator, se buscará el perro en la lista.

Si el perro está en la lista, se eliminará el perro que ingresó el usuario y se mostrará la lista ordenada. Si el perro no se encuentra en la lista, se le informará al usuario y se mostrará la lista ordenada.

📁 Extras: <https://spa.myservername.com/java-iterator-learn-use-iterators-java-with-examples>
<https://youtu.be/32DRqt6i1u0>

5- Crear una clase llamada Alumno que mantenga información sobre las notas de distintos alumnos. La clase Alumno tendrá como atributos, su nombre, una lista de tipo Integer con sus 3 notas, y una lista de tipo Alumno para listar los alumnos.

Realizar constructores respectivos el primero debe ser de esta forma :

```
public Alumno() {  
    this.listaAlumnos = new ArrayList();  
}
```

después con los otros atributos, y crear métodos Set y Get.

Métodos:

El primer método se llamara CrearAlumno en el cual pasaremos como parámetro el nombre y sus tres notas.

Dentro del método:

- Crearemos una instancia de la misma clase Alumno.
- Y una ArrayList de tipo Integer para alojar las notas.
- Con el objeto de la clase Alumno, vamos a ocupar el método set para pasarle el nombre
- Con el objeto de la Lista vamos a pasarle las tres notas con el método add()
- Nuevamente con el objeto de la clase Alumno vamos a utilizar el método set para pasarle el atributo notas.

Método llamado `añadirAlumno`, será el encargado de añadir todos los alumnos que iremos creando. Debe verse de esta forma:

```
public void añadirAlumno(Alumno alumno) {  
    this.listaAlumnos.add(alumno);  
}
```

✂ Su función: este método lo vamos a invocar o llamar también dentro del método `crearAlumno`, y dentro de sus paréntesis vamos a pasar el objeto de la clase `Alumno`, para guardarlo ya con el nombre y sus respectivas notas.

Por último un método llamado `verListado`, donde implementaremos lo aprendido (for each) para mostrar la lista de alumnos cargada.

📁 No nos olvidemos del método `ToString`, que era el encargado de desenscriptar los atributos si trabajamos en diferentes clases.

En el main, vamos a instanciar la clase `Alumno`, creando un objeto.

Definiremos nuevamente los atributos `String nombre`, `Integer nota1`, `nota2`, `nota3`;

Con el objeto de la instancia de la clase `Alumno` invocaremos al método `crearAlumno` y creamos 3 alumnos para la lista de manera directa y manual, es decir de esta manera:

```
listaAlumno.CrearAlumno("Fer", 10, 9, 8);  
listaAlumno.CrearAlumno("Maru", 10, 10, 10);
```

Dato 🤖 - **listaAlumno** seria el ejemplo del nombre o como le llamemos al objeto de la instancia de la clase Alumno creado en el main.

Luego aplicaremos la lógica necesaria para crear un bucle, en el cual crearemos un alumno. Tiene que ser dinámico por lo tanto ocuparemos un Scanner, preguntando el nombre y las 3 notas del alumno.

Importante: el bucle deberá preguntar al usuario si desea ingresar otro alumno mas, en caso que no, saldrá de el y almacenara el alumno creado en la lista.

De esta forma debería verse el alojamiento de datos pasados por teclado:

```
listaAlumno.CrearAlumno(nombre, nota1, nota2, nota3);
```

Por ultimo con nuestro objeto de la clase Alumno llamaremos al método verListado, donde nos imprimirá todos los alumnos de la lista con sus respectivos nombres y notas.

Deberá verse algo así: listaAlumno.verListado();

6- Un cine necesita implementar un sistema en el que se puedan cargar películas. Para esto, tendremos una clase Película con el título, director y duración de la película (en horas). Implemente las clases y métodos necesarios para esta situación, teniendo en cuenta lo que se pide a continuación:

- En el main deberemos tener un bucle que crea un objeto Película pidiéndole al usuario todos sus datos y guardándolos en el objeto Película.
- Después, esa Película se guarda una lista de Películas y se le pregunta al usuario si quiere crear otra Película o no.

Después de ese bucle realizaremos las siguientes acciones:

- Mostrar en pantalla todas las películas.
- Mostrar las películas mayores a 2 horas.
- Mostrar las películas menores a 2 horas.

7- Una tienda necesita un programa en el que se pueda cargar productos. Para esto, tendremos una clase Producto, con el tipo, cantidad, precio por unidad. Implementar clases y métodos necesarios para esta situación, teniendo en cuenta lo que se pide a continuación:

- En el main deberemos tener un bucle que crea un objeto Producto pidiéndole al usuario todos sus datos y guardándolos en el objeto Producto.
- Después, ese Producto se guarda una lista y se le pregunta al usuario si quiere crear otro tipo de Producto o no.

Después de ese bucle realizaremos las siguientes acciones:

- Mostrar en pantalla todos los productos por orden alfabético.
- Mostrar los productos mayores a 20\$.
- Mostrar los productos menores a 100\$.