

Ejercicios de Conjuntos Java



Relación de ejercicios

En los siguientes ejercicios hay que tener las siguientes consideraciones, a no ser que se indique lo contrario:

- No importa el orden de los elementos en la colección.
- No se deben permitir elementos duplicados en la colección.

Ejercicio 1. Implementa un programa que permita realizar las siguientes acciones sobre una lista de la compra. Todo el programa debe ser *case-sensitive*. Por ejemplo: “Galletas de hocolate” es distinto de “Galletas de Chocolate”. Un producto viene identificado por su nombre (cadena de texto).

MENÚ
1. Saber si la cesta contiene un producto
2. Agregar producto
3. Eliminar producto
4. Mostrar toda la cesta de productos
5. Borrar cesta
0. Salir

Ejercicio teórico-práctico 2. Implementa un programa que genere dos conjuntos (llamados ciudades1 y ciudades2) con nombres de ciudades mundiales. Por ejemplo, el conjunto ciudades1 tendrá (Madrid, Roma, Gijón y Budapest) y el conjunto ciudades2 contendrá (Madrid, Gijón y Bydgoszcz). A continuación, implementa cada una de las siguientes funcionalidades (cada una en un método).

- a) Clonar el primer conjunto de ciudades (significa crear una nueva variable de ciudades idéntica a ciudades1 pero independiente).
- b) Generar un único conjunto con las ciudades de ambos conjuntos (implica crear un conjunto que contenga todas las ciudades juntas: las de ciudades1 y ciudades2).
- c) Borrar del conjunto1 de ciudades todas las que aparezcan en el conjunto2. En nuestro ejemplo, de ciudades1 deberían desaparecer Madrid y Gijón ya que son las que también están en ciudades2).
- d) Indicar si en el segundo conjunto de ciudades (ciudades2) están otro determinado conjunto de ciudades. Por ejemplo, si ciudades2 contiene Madrid y Bydgoszcz a la vez (en nuestro

ejemplo, sí); o contiene Madrid y Málaga a la vez (en nuestro ejemplo, no, porque sí contiene Madrid pero no contiene Málaga).

- e) Indicar qué ciudades están en ambos conjuntos. En nuestro ejemplo, debería informar que las ciudades compartidas son Madrid y Gijón.

Ejercicio 3. Almacena en una colección, llamada `coches`, los nombres de 5 marcas de coches. Posteriormente, crea un menú que permita al usuario realizar las siguientes acciones.

- Mostrar por pantalla las marcas existentes junto con el número de marcas de coches que hay en la colección.
- Consultar si una marca está en la colección.
- Agregar una nueva marca de coches a la colección.
- Añadir 3 marcas de coches (pedidas al usuario) y las añada a la colección (todas las marcas a la vez, no de una en una).
- Una vez implementado lo anterior, investiga si hay alguna forma de sacar las marcas de coches ordenadas alfabéticamente.

Ejercicio 4. Almacena los siguientes datos utilizando un conjunto para cada grupo.

- *Ciudadanos*: Ana, Luis, Pepe, Sara, Miguel, Gabriel y Raquel.
- *Ciudadanos con IBI domiciliado*: Luis, Gabriel, Sara y Raquel.

Una vez creados ambos conjuntos, implementa las siguientes funcionalidades:

- a) Muestra los datos por pantalla de cada conjunto.
- b) Crea un clon del primer conjunto “ciudadanos”.
- c) Comprueba si están los `ciudadanosIBI` en el conjunto clonado.
- d) Obtén aquellos `ciudadanos` que no están en `ciudadanosIBI`.

Ejercicio 5. Implementa un programa que tenga dos opciones de menú: introducir número y listar números introducidos. La funcionalidad es la siguiente:

- El usuario puede introducir todos los números que desee y que estén entre el 1 y el 1000. Dichos números se deben ir almacenando en un conjunto. Ejemplo: el usuario introduce la siguiente secuencia 1, 100, 10, 200, 25, 50, 850 y 13.
- El usuario tiene la posibilidad de listar los números introducidos ordenados hasta un valor tope que también debe indicar. Por ejemplo, dados los números anteriores, si el valor tope es el 35, se le debe mostrar la siguiente secuencia: 1, 10, 13 y 25.