

Ejercicios POO Agregacion y Composicion

- **Ejercicio de Agregacion**

En una universidad llamada "Universidad XYZ", se desea gestionar la información de sus estudiantes. Cada estudiante tiene un nombre, edad y número de estudiante. La universidad puede tener un máximo de 100 estudiantes.

Debes implementar un programa en Java utilizando programación orientada a objetos que cumpla con los siguientes requisitos:

Define una clase llamada "Universidad" que tenga los siguientes atributos:

- a. Un nombre de tipo String.
- b. Un vector de estudiantes de tipo Estudiante.
- c. Un contador de cantidad de estudiantes.

La clase "Universidad" debe tener los siguientes métodos:

- d. Un constructor que reciba el nombre de la universidad y que inicialice el vector de estudiantes y el contador.

- e. Un método llamado "agregarEstudiante" que reciba un objeto de tipo Estudiante y lo agregue al vector de estudiantes si hay capacidad disponible.
- f. Un método llamado "getEstudiantes" que retorne el vector de estudiantes.
- g. Un método llamado "getCantidadEstudiantes" que retorne la cantidad actual de estudiantes en la universidad.

Define una clase llamada "Estudiante" que tenga los siguientes atributos:

- h. Un nombre de tipo String.
- i. Una edad de tipo int.
- j. Un número de estudiante de tipo int.

La clase "Estudiante" debe tener los siguientes métodos:

- k. Un constructor que reciba el nombre, la edad y el número de estudiante.

En el método principal (main) del programa, realiza lo siguiente:

- l. Crea una instancia de la clase "Universidad" llamada "universidad" con el nombre "Universidad XYZ".

- m. Crea al menos dos instancias de la clase "Estudiante" con datos ficticios.
- n. Agrega los estudiantes creados a la universidad utilizando el método "agregarEstudiante" de la clase "Universidad".
- o. Imprime en la consola el nombre de la universidad.
- p. Recorre el vector de estudiantes utilizando un bucle y muestra en la consola la información de cada estudiante (nombre, edad y número de estudiante).

Recuerda utilizar vectores en lugar de listas para almacenar los estudiantes en la clase "Universidad" según los requisitos del ejercicio.

- **Ejercicio de Composicion**

Se desea modelar una relación de composición entre las clases "Casa" y "Habitación" utilizando vectores en lugar de listas. Cada casa está compuesta por varias habitaciones, y si la casa se destruye, las habitaciones también desaparecen. Cada habitación tiene un número y un tamaño en metros cuadrados.

Debes implementar un programa en Java utilizando programación orientada a objetos que cumpla con los siguientes requisitos:

Define una clase llamada "Casa" que tenga los siguientes atributos:

- Un vector de habitaciones de tipo Habitación.
- Un contador de cantidad de habitaciones.
- Una constante llamada "CAPACIDAD_MAXIMA" con valor 100 para indicar la capacidad máxima de habitaciones en una casa.

La clase "Casa" debe tener los siguientes métodos:

- Un constructor que inicialice el vector de habitaciones y el contador.
- Un método llamado "agregarHabitacion" que reciba un objeto de tipo Habitación y lo agregue al vector de habitaciones si hay capacidad disponible.
- Un método llamado "destruirCasa" que elimine todas las habitaciones de la casa.
- Un método llamado "mostrarInformacionCasa" que muestre en la consola la información de cada habitación en la casa.

Define una clase llamada "Habitación" que tenga los siguientes atributos:

- Un número de habitación de tipo int.
- Un tamaño de habitación en metros cuadrados de tipo double.

La clase "Habitación" debe tener los siguientes métodos:

- Un constructor que reciba el número y el tamaño de la habitación.
- Métodos getter para acceder al número y al tamaño de la habitación.

En el método principal (main) del programa, realiza lo siguiente:

- Crea una instancia de la clase "Casa" llamada "casa".
- Crea al menos dos instancias de la clase "Habitación" con datos ficticios.
- Agrega las habitaciones creadas a la casa utilizando el método "agregarHabitacion" de la clase "Casa".
- Llama al método "mostrarInformacionCasa" de la clase "Casa" para mostrar la información de las habitaciones en la consola.
- Llama al método "destruirCasa" de la clase "Casa" para simular la destrucción de la casa y vuelve a llamar al método "mostrarInformacionCasa" para verificar que las habitaciones se han eliminado.