

Taller de practica para POO final

En este taller, trabajaremos en la creación de una aplicación sencilla para gestionar una lista de estudiantes de una escuela. Cada estudiante tiene un nombre, un número de identificación único y una calificación.

Utiliza los conceptos de programación orientada a objetos para modelar esta aplicación.

1. Clase **Estudiante**

Crea una clase llamada Estudiante con los siguientes atributos:

- Nombre (String)
- Número de Identificación (int)
- Calificación (double)
- Crea un constructor que permita inicializar estos atributos al crear una instancia de la clase Estudiante.
- Implementa métodos get y set para todos los atributos.

2. Clase **Escuela**

Crea una clase llamada Escuela que represente una lista de estudiantes. Utiliza una colección Set para almacenar los objetos Estudiante. Inicializa la colección en el constructor.

Implementar los siguientes métodos:

- agregarEstudiante(): Agrega un estudiante a la lista validando que el número de identificación no esté repetido.
- buscarEstudiante(int numeroidentificacion): Busca y devuelve un estudiante por su número de identificación.
- actualizarEstudiante(int numeroidentificacion, String nuevoNombre, double nuevaCalificacion): Actualiza los datos de un estudiante existente.
- eliminarEstudiante(int numeroidentificacion): Elimina un estudiante de la lista por su número de identificación.

3. Método Principal (Main)

- En el método main, crea una instancia de la clase Escuela.
- Agrega al menos tres estudiantes a la lista utilizando el método agregarEstudiante.

4. Interacción con el Usuario

Implementa un menú de texto en el método main que permita al usuario realizar las siguientes operaciones:

- Agregar un nuevo estudiante.
- Buscar un estudiante por número de identificación.
- Salir del programa.

5. Validación de Entrada

Agrega validación de entrada para garantizar que los datos proporcionados por el usuario sean correctos. Por ejemplo, asegúrate de que el número de identificación sea único para cada estudiante y que las calificaciones estén dentro de un rango válido (por ejemplo, de 0 a 10).

Sistema de Gestión de Figuras Geométricas

Desarrollar un sistema que permita crear y gestionar diferentes tipos de figuras geométricas, calculando áreas, perímetros y permitiendo moverlas en un plano. Utilizaremos clases abstractas, interfaces y polimorfismo.

1. Clase Abstracta Figura

Crea una clase abstracta llamada Figura con los siguientes atributos:

- **Color** (String)
- **Posición** (coordenadas x, y)

Define dos métodos abstractos:

- `calcularArea()`: Calcula el área de la figura.
- `calcularPerimetro()`: Calcula el perímetro de la figura.

Implementa los métodos get y set para los atributos comunes.

2. Interfaces

Crea las siguientes interfaces:

- **Movable**: Define los métodos `moverHorizontalmente(int distancia)` y `moverVerticalmente(int distancia)`.
- **Dibujable**: Define el método `dibujar()`, y el método `rotar()` para rotar la figura.

3, Clases Concretas

Crea las siguientes clases concretas que hereden de Figura e implementen las interfaces necesarias:

1. **Círculo:**

- Atributo adicional: radio (double).
- Implementa los métodos para calcular el área y el perímetro.

2. **Rectángulo:**

- Atributos adicionales: base (double) y altura (double).
- Implementa los métodos para calcular el área y el perímetro .

3. **Triángulo:**

- Atributos adicionales: base (double) y altura (double).
- Implementa los métodos para calcular el área.

En el método main(), realizar las siguientes acciones:

1. Crear instancias de las clases **Círculo**, **Rectángulo**, y **Triángulo**. Los datos (como radio, base y altura) pueden ser ingresados por el usuario a través de la consola o ser inicializados directamente en el código.
2. Llamar a los métodos para calcular el **área** y el **perímetro** de cada figura.
3. Utilizar las funcionalidades de **mover** las figuras en el plano. Llama a los métodos de la interfaz Movable para desplazar las figuras horizontal o verticalmente.
4. Dibujar y rotar las figuras utilizando la interfaz Dibujable.
5. Implementar un menú que permita al usuario seleccionar opciones como:
 - Crear una nueva figura (círculo, rectángulo o triángulo).
 - Calcular el área y perímetro de una figura.
 - Mover una figura.
 - Dibujar una figura.
 - Salir del programa.