

Polimorfismo – Práctica Java.

Introducción

El siguiente documento tiene como objetivo explicar el funcionamiento del código Java, que implementa el concepto de polimorfismo a través de una jerarquía de clases que representan diferentes formas geométricas.

Estructura del Código

Clase Principal

La clase Principal contiene el método main, que es el punto de entrada del programa. En este método, se crea una lista de objetos de tipo Forma, que es una clase de la cual heredan otras clases representando formas específicas como círculos, rectángulos, etc. Se instancian diversas formas y se añaden a la lista formas.

Clase Forma

La clase Forma es una clase que define un método para las clases que heredan de ella, y así puedan implementar los métodos calcularArea y dibujar. También sobrescribe el método toString para devolver el nombre de la clase de la instancia.

Clase Círculo

La clase Círculo extiende de Forma y proporciona la implementación específica para el cálculo del área y el dibujo de un círculo. Utiliza el atributo radio para estas operaciones.

Clase Cuadrado

Siguiendo el mismo patrón que Círculo, la clase Cuadrado extiende de Forma y sobrescribe los métodos para calcular el área y dibujar, utilizando el atributo lado.

Clase Elipse

La clase Elipse es similar a Círculo, pero tiene dos radios (eje mayor y eje menor) que usa para calcular el área y para dibujar la elipse.

Clase Rectángulo

La clase Rectángulo extiende de Forma y usa los atributos ancho y alto para las operaciones de cálculo de área y dibujo.

Clase Rombo

La clase Rombo utiliza las diagonales (mayor y menor) para calcular el área y para representar el dibujo del rombo.

Clase Trapecio

La clase Trapecio extiende de Forma y calcula el área con base en sus bases (mayor y menor) y su altura.

Clase Triángulo

La clase Triángulo, por último, extiende de Forma y usa la base y la altura para sus cálculos y representación gráfica.

Funcionamiento del Programa

El programa utiliza el polimorfismo para tratar a todas las formas de manera uniforme dentro de la lista formas. Al iterar sobre esta lista, se llama a los métodos calcularArea y dibujar de cada forma, sin necesidad de conocer el tipo específico de forma que se está procesando. Esto demuestra el poder del polimorfismo: la capacidad de procesar objetos de diferentes clases a través de una interfaz común.