

Herencia

Luis Ángel Serrano Catalá | Prof. Ricardo Vegas Morales

Programación Orientada a Objetos, Unidad 3.

LIDTS 2ºP, Universidad Autónoma de Chiapas.

Herencia

Es uno de los conceptos fundamentales teóricos del que la programación orientada a objetos se basa.

Es el hecho de pasar atributos y métodos de una clase general a una derivada de. Esto permite crear una estructura jerárquica de clases cada vez más especializada.

Clases

Todo programa puede ser representado como una colección de "planos de construcción" describiendo los atributos y funcionalidades de áreas en específico de ese programa.

De este plano podremos contruir (instanciar) objetos, que comparten una misma funcionalidad con diferentes datos.

Clases derivadas

Podemos reutilizar los planos clonandolos y añadiendo funcionalidades extra a estos clones/hijos.

La clase derivada/hijo tendrá todas las propiedades de la clase general/padre junto a sus propias funcionalidades.

Ejemplo

```
public class PanelDibujo extends JPanel
{
    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        final int TIMES = 15;
        final int MODIFIER = 20;
        final int SIZE = TIMES * MODIFIER;

        for (int x = 0; x < TIMES; x++) {
            g.drawLine(0, 0, x * MODIFIER, (TIMES - x) * MODIFIER);
        }

        for (int x = 0; x < TIMES; x++) {
            g.drawLine(SIZE, 0, (TIMES - x) * MODIFIER, (TIMES - x) * MODIFIER);
        }
    }
}
```

```
    }

    for (int x = 0; x < TIMES; x++) {
        g.drawLine(SIZE, SIZE, (TIMES - x) * MODIFIER, x * MODIFIER);
    }

    for (int x = 0; x < TIMES; x++) {
        g.drawLine(0, SIZE, x * MODIFIER, x * MODIFIER);
    }

    final int FONT_SIZE = 32;
    final String TEXT = "Cuadradito gang";

    g.setFont(new Font("Comic Sans MS", Font.PLAIN, FONT_SIZE));
    g.drawString(TEXT, 0, SIZE + FONT_SIZE);
}
}
```