**Polimorfismo**

El polimorfismo está relacionada con la herencia. Es la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos y estos responden de acuerdo a su implementación.

Puede entenderse esto si tomamos en cuenta que un objeto padre puede almacenar un objeto hijo (Principio de sustitución de Liskov) .

Ejemplo:

Tenemos una clase padre “Figura”, este tiene 2 hijos, cuadrado y triangulo. Estos objetos tienen un método que sobrescriben del padre llamado área. Cada uno lo implementa de forma distinta.

Nosotros podríamos crear un arreglo de figuras y guardar dentro triángulos y cuadrados.

//Creamos el arreglo  
Figura arregloFiguras[] = new Figura[2];

// Cargamos el arreglo. Al ser cuadrado y triangulo hijos de figura podemos usar el objeto figura para referirnos a ellos (por el principio de Liskov)

// Acá hay que notar que estamos guardando un hijo dentro de un padre. No es posible hacerlo al revés.  
arregloFiguras[0] = new Triangulo(baseTriangulo, alturaTriangulo):  
arregloFiguras[1] = new Cuadrado(baseCuadrado, alturaCuadrado):

Después podríamos recorrer el arreglo y calcular el área de los cuadrados y triángulos. Sabemos que la forma de calcular el área es distinta en cuadrados y triángulos. Esta implementación está dentro de las respectivas clases.  
Gracias al polimorfismo nosotros no tenemos que preocuparnos por esto, solo debemos llamar al método área… De acuerdo al tipo de objeto se procesara de una u otra manera.

Siguiendo con el ejemplo podemos recorrer el arreglo con un foreach e imprimir el area.

for (Figura figura : arregloFiguras) {  
System.out.println(“Area” + figura.area());  
}