**INTRODUCCION**

Los programas se crean a partir de los objetos los cuales son instancias de clases. Alguna clases están en biblioteca de java y otras el programador.

Cuando empezamos a escribir un nuevo programa, esta metodología OO para programación significa que hemos de empezar los programas desde cero.

Considere la siguiente analogía. Suponga que desea comprar un automóvil nuevo y una agencia en donde hay una variedad de automóviles producidos en masa. A usted le gusta uno en especial que está buscando. Al igual que la descripción de una clase, al igual que un automóvil se fabrica con planos idéntico.

**HERENCIA**

Vamos a empezar con una clase similar a la que ya hemos usado varias veces en el libro para representar una esfera. La esfera tiene un radio y una posición en espacio al mostrar una esfera en la pantalla deberá aparecer como un circulo .el diámetro de la esfera esta fijo en 20px(Pixeles).

Cabe mencionar que hay varios elementos nuevos en este programa incluyendo la palabra clave PROTECTED. Esto se debe a que escribimos la clase de tal forma que se pueda utilizar para la

herencia <http://www.arrakis.es/~abelp/ApuntesJava/Herencia2.htm>



**PROTECTED**

Cuando usamos una herencia, prívate es un término demasiado privado y public es demasiado público .Si una clase necesita dar a sus subclases acceso a ciertas variables y métodos específicos pero cave evitar que las otras calases están o accedan a estos puede etiquetarlos como PROTECTED. Volviendo a la clase de esfera, necesitamos variables para describir las coordenadas y pediríamos escribirlo siguiente:

PrivateintX, Y;

Este acceso necesitara acceso a la variables X,Y que por desgracia son inaccesibles ya que se etiquetaron como Private , por lo tanto para anticiparlos a este posible uso es un futuro , podríamos decidir etiquetarlos mejor como PROTECTED.

Protectedint X, Y;

La única opción sería editar la clase y remplazarla descripción Private por Protected.



**SUPER**

Algunas veces una clase necesita llamar el método de sus superclases inmediato de algunas de las clases del nivel superior en el árbol. No hay problema con esta ya que los métodos de niveles superiores del árbol de herencia están disponibles siempre y cuando estén etiquetados como public o protected .



**FINAL**

Los procesos de heredar y de redefinir se enfocan en cambiar el compartimiento de las clases , por ejemplo es bueno saber exactamente qué es lo que hace sqlt, que es lo que hace drown line para evitar esto el programador puede describir un método como final esta significa que no se puede redefinir .



**ABTRACTAS**

Considere in programador que mantiene formas graficas de lados con tipos y tamaños como circulo, rectángulo, triangulo etc. Estos son similares a las clases que ya hemos visto en este informe. Su posición en color como en tanto en tamaño este lo declaramos a una superclase llamada de la forma que describa los datos comunes.

Por lo tanto al escribir una clase circulo o formas hemos aprovechado las herramientas que proporciona la clase forma.

