¿QUE ES LA HERENCIA Y PARA QUE SIRVE?

**La Herencia**

Los programas se  crean a partir de objetos, los cuales son instancias de clase. La herencia consiste o constituye una manera de resolver  este problema. Con la herencia podemos usar una clase existente con la base para crear una clase modificada.

**Protected**

Cuando usamos la herencia, prívate es un termino demasiado privado y public es demasiado publico. Si una clase necesita dar a sus subclases acceso a ciertas variables o métodos específicos, pero debe evitar que otras clases accedan a estos, puede etiquetarlos como protected.

Import java.awt.\*;

Public class esfera{

Protected int x=100, y=100;

Public void set x(int nuevax){

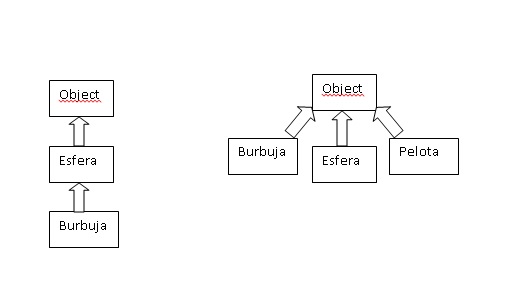
X=nuevay;}

Public void mostrar(Graphics papel){

Papel.drawoval(x,y,20,20);}}

**La Herencia**

Es común que una clase tenga una superclase, la que a su vez tiene una superclase y así sucesivamente, hasta llegar a la parte superior del árbol de herencia. No solo se heredan los elementos public y protected de la superclase inmediata, sino también todas las variables y los métodos public y protected en todas las superclases en el árbol de herencia.



**Super**

Algunas veces una clase necesita llamar a un método de superclase inmediata, o de algunas clses de nivel superior en el árbol. No hay problema con esto, ya que los métodos de todas las clases en los niveles superiores del árbol de herencia están disponibles, siempre y cuando estén etiquetados como public o protected. El único problema que puede surgir, es cuando el método de la clase actual(cuando se utiliza la redefinición o sobre escritura). Para corregir este problema hay que anteponer el nombre del método la palabra clave super.

Public class globomodificado extends globo{

Public globomodificado (int xinicial, int yinicial, int radioinicial(){

Super (xinicial, yinicial, radioinicial(); }

//resto de la clase }

**Final**

Los procesos de heredar y redefinir, se enfocan en cambiar el comportamiento de la clase y los objetos. En la POO siempre existe el peligro de que alguien extienda las clase a las que estas pertenecen y en consecuencia cambien lo que hacen. Para evitar esto el programador puede describir un método como fina. Las variables también se pueden declarar como final. Esto significa también que no podemos cambiar sus valores. Son constantes.

**Abstract**

Se usa en un programa que mantiene formas graficas de todos tipos y tamaños. Círculo, rectángulos, cuadrados, triángulos, etc.  Estas distintas formas similares, a las clases, tienen información en común: Su posición, forma y tamaño. Cada clase individual hereda esta información común.

Import java.awt.\*;

Public abstract class forma{

Protected int x, y;

Protected int tamaño;

Public void moverderecha(){

X = x + 10; }

Public abstract void mostrar(Graphics papel); }

PAGINA:blog.

<http://fer-nando-com-mx.webnode.com.co/news/la-herencia/>