

# UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Industrial y de Sistemas



N° de Ensayo

- Ensayo N° 05

Título del ensayo realizado

- "Herencia y Polimorfismo"

Nombre del profesor:

- Rony Jordan Hancoco Carpio

Nombre del alumno:

- Flores Tambo, Diego Alonso.....20132716H

## HERENCIA

La herencia, vendría a ser una forma de reutilización de código, en la cual, una clase denominada sub-clase absorbe los métodos y los atributos de otra clase, denominada superclase.

Una sub-clase generalmente agrega sus propios métodos y atributos, que le dan un comportamiento diferente al de la superclase.

Las relaciones de herencia forman estructuras jerárquicas en forma de árbol. Una clase se convierte, ya sea, en una superclase proporcionando métodos y atributos a las demás clases, o en una sub-clase, absorbiendo los métodos o atributos de la clase que extienden.



EJEMPLO: Jerarquía de herencia para objetos MiembroDeLaComunidad universitaria

Por ejemplo: Podemos tener una superclase cuadriláteros, de la cual sub-clases como rectángulo, cuadrilátero, paralelogramo, etc. van a heredar sus características.

Una superclase puede tener muchas sub-clases, pero la sub-clase no puede heredar las características de solamente una superclase.

Los objetos de toda clase que se extiende de una superclase común, puede ser tratado como objeto de esa superclase. Sin embargo, los objetos de una superclase, no pueden ser tratados como objetos de una subclase.

Los miembros private de una clase son accesibles dentro de la misma clase, por lo tanto, las subclases que se extiendan de esa superclase no heredaran dichos miembros. Producto de ello, utilizaremos el modificador de acceso protected.

La extensión de una clase tiene la siguiente sintaxis:

```
public class nombre – subclass extends nombre – superclass{  
  
}
```

Super, llama al constructor de la clase padre, para poder coger todos los métodos y atributos que contenga la superclase. La palabra clave Super, usualmente va en la primera línea de código después de haber definido el constructor, y contiene los argumentos del constructor de la superclase.

## Polimorfismo

El polimorfismo se define como la capacidad de un lenguaje de programación de elegir entre uno u otro método según el contexto en el que se encuentre.

Por ejemplo, si suponemos que tenemos diversas sub-clases que se extienden de una superclase, de la cual heredan diversos métodos. Cuando cada una de las sub-clases llame a estos métodos, no necesariamente te tiene que votar los mismos resultados, sino tiene muchas formas de resultado.

Con el polimorfismo podemos usar el mismo nombre y la misma firma del método para que ocurran distintas acciones, dependiendo el tipo de objeto en el que se invoca el método

Por lo tanto, en la práctica, un mismo identificador puede tener distintas formas (distinto cuerpo de función, distintos comportamientos) dependiendo, en general, del contexto en el que se halla insertado.

El polimorfismo se subdivide en dos ramas:

- La SOBRECARGA: Se refiere al mismo identificador u operador en distintos contextos y con distintos significados.

La sobrecarga de funciones conduce a que un mismo nombre pueda representar distintas funciones con distinto tipo y numero de argumentos.

Por Ejemplo:

```
public void asignarValor (int x)  
public void asignarValor(double x)  
public void asignarValor(int x, int y)
```

La sobrecarga de funciones equivale a que un mismo mensaje puede ser enviado a varios objetos de diferentes clases de forma que cada objeto respondería al mensaje apropiadamente.

- La REDEFINICION.

## Bibliografía:

- Como Programar en Java Séptima edición. P. J. Deitel.
- C++ / OOP Un enfoque práctico. Devis Botella.
- Fundamentos de Programación en Java. Ladrón de Guevara.
- Notas de Introducción al Lenguaje de Programación en Java. Ortega Arjona.