

**UNIVER MILENIUM**

**PLATEL ECATEPEC**

**Ingeniería en Sistemas Computacionales**

**Programación Orientada a Objetos**

**Práctica 4**

**Plantel-Ecatepec**

**Alumno**

Alcantar Garcia Olaez Jorge Aaron

6232000085

**Profesor**

M. en C.C. José Alejandro Pineda Aguillón

**Periodo**

21-3 – Módulo II

MISC-301

1. **Preguntas Críticas**

**¿Qué es la Herencia en POO?**

La **herencia** es un pilar importante de POO (Programación Orientada a Objetos). Es el mecanismo en Java por el cual una clase permite heredar las características (atributos y métodos) de otra clase

**Terminologías**

**Superclase:** la clase cuyas características se heredan se conoce como superclase (o una clase base o una clase principal).

**Subclase:** la clase que hereda la otra clase se conoce como subclase (o una clase derivada, clase extendida o clase hija). La subclase puede agregar sus propios campos y métodos, además de los campos y métodos de la superclase.

**Reutilización:** la herencia respalda el concepto de “reutilización”, es decir, cuando queremos crear una clase nueva y ya hay una clase que incluye parte del código que queremos, podemos derivar nuestra nueva clase de la clase existente. Al hacer esto, estamos reutilizando los campos/atributos y métodos de la clase existente.

**¿Qué importancia tiene la Herencia en la POO?**

Permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

**¿Cuándo se debe aplicar herencia en POO?**

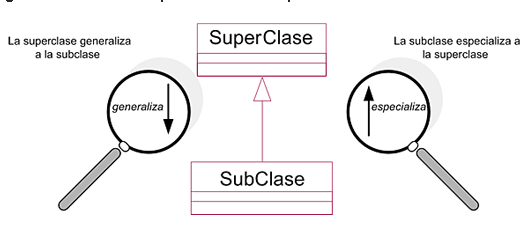
Esta forma de diseñar Herencia en POO nos permite que el resultado sea un sistema mucho más flexible en tiempo de programación e incluso en tiempo de ejecución.

**¿A qué se refiere la generalización de clases?**

Una generalización muestra que una clase hereda de otra clase. La clase que hereda se denomina descendiente. La clase heredada se denomina predecesor.

\*  La generalización se extrae de la clase descendiente a la clase predecesora.

**¿A qué se refiere la especialización de clases?**

Una clase es más específica que otra si todas las instancias que la componen son a su vez instancias de la otra clase. La clase más específica es una subclase de la otra clase. Esta última, más general, recibe el nombre de superclase.

**¿Qué cosas se pueden heredar en POO?**

\*Herencias que se pueden hacer en POO

**Herencia única:** donde las subclases heredan las características de solo una superclase.

**Herencia Multinivel:** una clase derivada heredará una clase base y, además, la clase derivada también actuará como la clase base de otra clase.

**Herencia Jerárquica:** una clase sirve como una superclase (clase base) para más de una subclase.

**Herencia Múltiple (a través de interfaces):** una clase puede tener más de una superclase y heredar características de todas las clases principales. Pero Java no admite herencia múltiple con clases, así que para lograrlo tenemos que usar Interfaces.

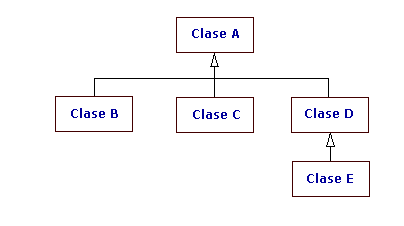
**Herencia Híbrida (a través de Interfaces):** Es una mezcla de dos o más tipos de herencia anteriores. Como Java no admite herencia múltiple con clases, la herencia híbrida tampoco es posible con clases, pero como en el ejemplo anterior, podemos lograr el mismo resultado a través de Interfaces.

**¿Qué representa una clase “Padre”?**

En esta estructura jerárquica, cada clase tiene sólo una **clase padre**. La clase padre de cualquier clase es conocida como su superclase. La clase hija de una superclase es llamada una  subclase.

**\***Una superclase puede tener cualquier número de subclases.

\*Una subclase puede tener sólo una superclase.



**¿Qué son las clases “hijas o derivadas”**

Las **clases hijas o derivadas** quedan implícitamente vinculadas en una relación de herencia con las clases de las cuales derivan. Como consecuencia de esto, las modificaciones en las clases “padre” se verán reflejadas en las “hijas”, con algunas excepciones. Las modificaciones consisten en cambiar los valores de alguna de las propiedades de la clase.

**¿Qué es la herencia sucesiva?**

Consiste en crear nuevas clases partiendo de las clases existentes, que tendrán todas los atributos y los métodos de su superclase o clase padre y además se le podrían añadir otros atributos y métodos propios

**¿Cómo se aplica herencia en java? Coloca un ejemplo**

**Reutilización del código**: En aquellos casos donde se necesita crear una clase que, además de otros propios, deba incluir los métodos definidos en otra, la herencia evita tener que reescribir todos esos métodos en la nueva clase.

**Mantenimiento de aplicaciones existentes**: Utilizando la herencia, si tenemos una clase con una determinada funcionalidad y tenemos la necesidad de ampliar dicha funcionalidad, no necesitamos modificar la clase existente (la cual se puede seguir utilizando para el tipo de programa para la que fue diseñada) sino que podemos crear una clase que herede a la primera, adquiriendo toda su funcionalidad y añadiendo la suya propia.

//Clase para objetos de dos dimensiones

class DosDimensiones{

double base;

double altura;

void mostrarDimension(){

System.out.println("La base y altura es: "+base+" y "+altura);

}

}

//Una subclase de DosDimensiones para Triangulo

class Triangulo extends DosDimensiones{

String estilo;

double area(){

return base\*altura/2;

}

void mostrarEstilo(){

System.out.println("Triangulo es: "+estilo);

}

}

class Lados3{

public static void main(String[] args) {

Triangulo t1=new Triangulo();

Triangulo t2=new Triangulo();

t1.base=4.0;

t1.altura=4.0;

t1.estilo="Estilo 1";

t2.base=8.0;

t2.altura=12.0;

t2.estilo="Estilo 2";

System.out.println("Información para T1: ");

t1.mostrarEstilo();

t1.mostrarDimension();

System.out.println("Su área es: "+t1.area());

System.out.println();

System.out.println("Información para T2: ");

t2.mostrarEstilo();

t2.mostrarDimension();

System.out.println("Su área es: "+t2.area());

}

}

Salida:

Información para T1:

Triangulo es: Estilo 1

La base y altura es: 4.0 y 4.0

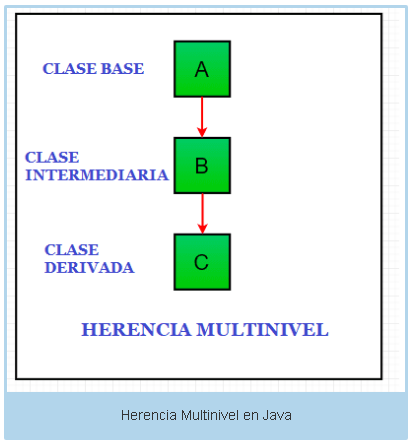
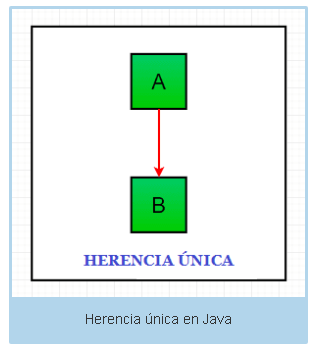
Su área es: 8.0

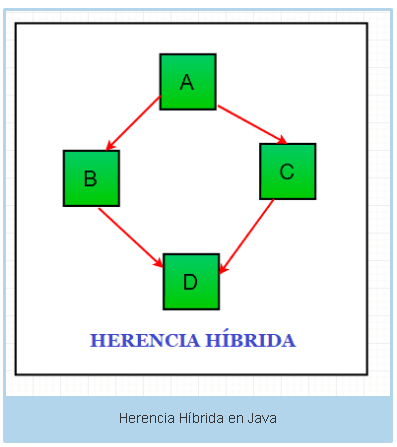
Información para T2:

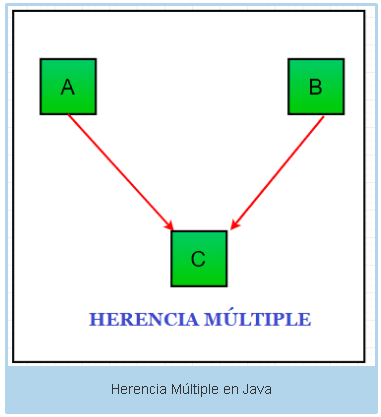
Triangulo es: Estilo 2

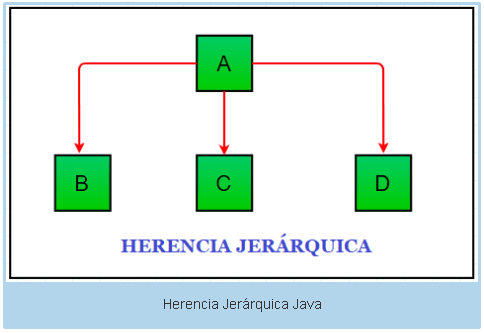
La base y altura es: 8.0 y 12.0

Su área es: 48.0

Prof por mi parte le dejo este código que se parece mucho a las act de FG pero igual no sabría qué tipo de ejemplo dejarle aunque espero que este sirva de ejemplo. De igual forma dejo más ejemplos sencillos a lo que refiere







1. **Descripción del problema a resolver**

**Instrucciones:** Implemente un esquema de herencia para resolver la siguiente situación

* Realice un programa que permita modelar distintas profesiones, oficios o deportes a partir de la especialización de la clase persona. Se debe considerar una especialización de 3 niveles que permita la aplicación de herencia sucesiva en al menos una clase derivada, considere la creación de **1 clase para profesiones**, **1 clases para deportistas** **y 1 clases para algún oficio**, determine a cuál de ellas le aplicaría herencia sucesiva para el 3er nivel de especialización.
* Considere que la clase Inicial solo lleva la ejecución del programa y no representa parte del contexto de herencia
* Cree un objeto de cada elemento incluida la superclase Persona y demuestre el estado y comportamiento de cada objeto en la clase Inicio