Repaso/Consulta 2

Supongamos el siguiente sistema, el cual puede crear un objeto *Calculadora* o *CalculadoraCientifica* y realizar operaciones a través de la invocación de métodos de dichos objetos. Decimes que “Una *CalculadoraCientifica* es una *Calculadora*”: por lo tanto estamos ante una relación de herencia.



Codifiquemos en Java

|  |
| --- |
| ***class Calculadora {***  ***public double sumar(double a, double b) {***  ***return a + b;***  ***}***  ***public double restar(double a, double b) {***  ***return a - b;***  ***}***  ***public double multiplicar(double a, double b) {***  ***return a \* b;***  ***}***  ***public double dividir(double a, double b) {***  ***return a / b;***  ***}***  ***}***  ***class CalculadoraCientifica extends Calculadora {***  ***public double logaritmo(double a) {***  ***return Math.log(a);***  ***}***  ***public double potencia(double a, double b) {***  ***return Math.pow(a, b);***  ***}***    ***public double seno(double a) {***  ***return Math.sin(a);***  ***}***  ***public double coseno(double a) {***  ***return Math.cos(a);***  ***}***    ***public double dividir(double a, double b) {***  ***if(b==0) {***  ***System.out.println("No se puede dividir por 0");***  ***return 0;***  ***}***  ***else***  ***return a / b;***  ***}***  ***}***  ***// Prograqma/Clase Principal-----------------------------------***  ***public class Prueba {***    ***public static void main(String[] args) {***  ***Calculadora c1 = new Calculadora(); //con el objeto c1 solo podremos realizar o0peraciones simples***  ***CalculadoraCientifica c2 = new CalculadoraCientifica();***    ***//ejemplo de uso***  ***System.out.println("El seno de 0 es: " + c2.seno( 0 ) );***  ***}***  ***}*** |

La clase *CalculadoraCientifica*, a través de la herencia, crea una clase más específica y especializada que *Calculadora*, a través de:

1. Agregar métodos para el cálculo del logaritmo, potencia, seno y coseno.
2. IMPORTANTE: Sobrescribir el método *dividir* de la clase *Calculadora*, con lo cual:
   1. El método heredado queda oculto por el nuevo método sobrescrito.
   2. El método sobrescrito tiene la posibilidad de controlar que no se realicen divisiones por cero.

El método dividir sobrescrito en *CalculadoraCientifica* también podría escribirse como:

|  |
| --- |
| *public double dividir(double a, double b) {*  *if(b==0) {*  *System.out.println("No se puede dividir por 0");*  *return 0;*  *}*  *else*  *return* ***super.dividir(a, b);***  *}* |

A través de la palabra ***super***, podemos acceder al método dividir de la superclase *Calculadora*, sin la necesidad de reescribir la expresión de división.