



Programación Java

▶ Introducción al lenguaje Java

// Práctica integradora - Programación orientada a objetos

Objetivo

El objetivo de esta guía práctica es que podamos afianzar y profundizar los conceptos sobre encapsulamiento, herencia, sobreescritura y excepciones. Para esto vamos a plantear una serie de ejercicios simples que nos permitirán repasar los temas que estudiamos.

¿Are you ready? 😎👍



Ejercicio 1

1. Crear una clase `PracticaExcepciones` que defina los atributos `a = 0` y `b = 300` de tipo `int`. Calcular el cociente de `b/a`. Controlar la excepción que se lanza indicando el mensaje "Se ha producido un error". Al final del programa siempre deberá indicar el mensaje "Programa finalizado"
2. Modificar el programa anterior para que, al producirse el error, en vez de imprimir por consola el mensaje "Se ha producido un error", lo lance como una excepción de tipo `IllegalArgumentException` con el mensaje "No se puede dividir por cero"



Ejercicio 2

Debemos entregar un trabajo para una distribuidora de productos, por lo que vamos a generar un programa que realice diferentes operaciones. Tendremos dos categorías de productos diferentes: `Perecederos` y `No Perecederos`.

1. Crear una clase `Producto` que cuente con los siguientes atributos: `String nombre` y `double precio`, la misma debe definir los métodos `getters` y `setters` para sus atributos, un constructor que reciba todos sus atributos como parámetro y el método `toString()`. Crear el método `calcular()` al cual vamos a pasarle un parámetro de tipo `int` llamado `cantidadDeProductos`; este método tiene que multiplicar el precio por la cantidad de productos pasados.
2. Crear la clase `Perecedero`, que va a tener un atributo llamado `diasPorCaducar` de tipo `int`, al igual que para el producto, definir `setters`, `getters`, constructor que reciba todos los atributos por parámetro y el método `toString()`. Esta clase debe heredar de `Producto` y sobrescribir el método `calcular()`, el cual tiene que hacer lo mismo que la clase `Producto` (multiplicar el precio por la cantidad de productos) y adicionalmente, reducir el precio según los `diasPorCaducar`:



- Si le resta un día (1) para caducar, se reducirá 4 veces el precio final.
 - Si le restan dos días (2) para caducar, se reducirá 3 veces el precio final.
 - Si le restan 3 días (3) para caducar, se reducirá la mitad de su precio final.
3. Crear la clase NoPerecedero, la misma va a tener un atributo llamado tipo, el mismo va a ser un String, crear setters, getters, constructor y método toString(); en esta clase el método calcular() va a hacer exactamente lo mismo que en la clase Producto.
 4. Crear una clase ejecutable llamada Distribuidora donde van a crear un array de productos, imprimir el precio total al vender 5 productos de cada tipo. Crear los elementos del array con los datos que quieras.