

## PROGRAMACIÓN CON OBJETOS 2 - TP JAVA-POO

## Programación con Objetos 2 - TP JAVA-POO

## Objetivos

Aplicar conceptos de Programación Orientada a Objetos en Java.

## Lectura sugerida

Java para programadores objetosos (subida al classroom)

## Ejercicio 1

Un supermercado maneja el catálogo de los productos que vende. De cada *producto* se conoce su nombre, precio, y si el mismo es parte del programa Precios Cuidados o no. Por defecto, el producto no es parte del programa, a menos que se especifique lo contrario. Para ayudar a los clientes, el supermercado quiere realizar descuentos en productos de primera necesidad. Es por eso que al calcular el precio de un producto de "Primera Necesidad", se aplica un descuento del 10%. Es decir:

$$\text{precioProductoPrimeraNecesidad} = \text{precioBaseDelProducto} * 0.9$$

El supermercado, del cual se conoce el nombre y la dirección, desea conocer la cantidad total de productos que comercializa y la suma total de los precios de los mismos.

1. A partir de los *tests* provistos por la cátedra, descubra qué clases debe crear y los métodos que debe definir cada una. Asimismo, descubra qué funcionalidad se espera de los mismos.
2. Documente la solución utilizando un Diagrama de Clases UML.
3. Implemente la solución en Java. Asegúrese de que la misma "pase" los tests provistos por la cátedra.

## Ejercicio 2

Suponga ahora que el descuento a aplicar en cada producto de primera necesidad puede ser distinto y se debe poder definir al momento de crear el mismo. Por ejemplo, el arroz puede ser un producto de primera necesidad con un descuento del 8%, mientras que leche podría ser otro producto de primera necesidad con un descuento del 11%. Esto es sólo un ejemplo. El descuento a aplicar en cada producto de primera necesidad debe ser configurable al momento de crearlo.

1. Modifique los tests para contemplar esta variación.
2. Implemente la solución en Java.

## Ejercicio 3

El Impuesto al Trabajador es un impuesto que pagan anualmente todos los trabajadores del país. El monto a pagar del impuesto es un 2% del total del *monto imponible* de los ingresos del trabajador.

De cada trabajador se registran sus *ingresos percibidos* del año. De cada ingreso se conoce el mes de percepción, el concepto y el monto percibido. Existe un tipo de ingreso, el *Ingreso por Horas Extras*, del cual se conoce además la cantidad de horas extras realizadas, por las cuales el trabajador percibió tal ingreso.

Para un ingreso, el monto imponible al Impuesto al Trabajador es igual al monto percibido. Sin embargo, los ingresos por horas extras están exentos al impuesto. Es decir, suman cero al total del monto imponible al impuesto al trabajador.

La solución debe respetar el siguiente protocolo:

```
Trabajador>>getTotalPercibido()  
//retornar el monto total percibido por el trabajador
```

Trabajador>>getMontolImponible()

//retornar el monto imponible al impuesto al trabajador

Trabajador>>getImpuestoAPagar()

//retornar el monto a pagar por el trabajador, calculado como el 2% del monto imponible

1. Documente la solución utilizando un Diagrama de Clases UML.
2. Escriba los tests correspondientes.
3. Implemente la solución en Java

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by/4.0/)

---