



TALLER PRACTICO HERENCIA ADSI 57

1. Implementar una super clase llamada Producto, la cual tenga los siguientes atributos y métodos:

Atributos:

- Código.
- Descripción.
- Precio de compra.
- Precio de venta.
- Cantidad en bodega.
- Cantidad mínima requerida en bodega.
- Cantidad máxima de inventario permitida.
- Porcentaje de Descuento.

Métodos:

- Solicitar pedido: devuelva true si debe solicitar el producto al proveedor y false en caso contrario.
- Calcular total a pagar : devuelva la cantidad total a pagar al proveedor dado una cantidad de unidades de compra.

Adicionalmente se desea dos subclases para los siguientes tipos de productos:

- Prendas de vestir (como lo son blusas, jeans, camisas, etc) el cual debe tener los siguientes parámetros adicionales:
 - Talla: S, M, L, etc
 - Permite planchado: verdadero o falso.
- Calzado (como lo son tenis, calzado formal, sandalias, etc) el cual debe tener el siguiente parámetro adicional:
 - Talla: 35, 36, 37, etc

Diseñar un programa que:

- Consulte el número de productos de tipo de prendas de vestir a manejar.
- Consulte el número de productos de tipo calzado a manejar.
- Cree un vector de productos de prendas de vestir en el cual se guardarán las instancias de cada uno de ellos.
- Cree un vector de productos de calzado en el cual se guardarán las instancias de cada uno de ellos.
- Solicite los datos requeridos para cada tipo de producto a guardar.

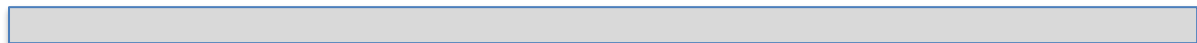
Implemente en la clase principal un menú con las siguientes opciones:

- Verificar productos a pedir: recorrer los vectores de productos y arrojar una alerta en caso tal se deba solicitar el pedido al proveedor.
- Calzado con mayor cantidad de unidades: indicar cuál es el código del producto que tiene mayor cantidad de unidades en bodega, si hay varios que cumplen con la



- condición, indicar el primero de ellos.
- Prenda con mayor cantidad de unidades: indicar cuál es el código del producto que tiene mayor cantidad de unidades en bodega, si hay varios que cumplen con la condición, indicar el primero de ellos.
- Modificar cantidad mínima requerida en bodega: leer código de producto, el tipo (calzado o prenda) y el nuevo valor para la cantidad de unidades mínimas requeridas, validar en el método set de la variable correspondiente que la nueva cantidad no sea menor a 0, en caso de serlo establecer la cantidad mínima como 0.
- Vender producto: solicite el código de un producto a vender, el tipo (calzado o prenda), y las unidades correspondientes, calcule el valor de la factura con descuento y sin descuento. Validar que existan las unidades necesarias para la venta y modificar las unidades existentes después de la venta utilizando el método set de la variable correspondiente.

Para tener en cuenta: el porcentaje de descuento es el mismo para todos los productos, y es del 1%.



2. Eres un ingeniero de software independiente a quién, el Zoológico de Cali, ha contactado para ayudarles a mantener un software que ellos tienen parcialmente implementado, el cual les ayuda a ver el IMC (índice de masa corporal) de los animales que ellos tienen en cautiverio. Ellos te explican que van a añadir gorilas a el zoológico, por lo tanto, deben de añadir soporte para gorilas en el software que ellos han venido usando.

El administrador del zoológico te muestra el diagrama de clases del software que ellos usan con el fin de que tú te enteres de cómo funciona y cómo vas a implementar lo que ellos te solicitan:

Dada la situación planteada, el administrador del Zoológico de Cali te explica que, para los gorilas, hay que tenerles en cuenta el peso de los alimentos que han ingerido, dado que sus sistemas digestivos son muy lentos, al momento de hacer el cálculo del IMC. El IMC está dado por la operación $\text{peso}/\text{altura}^2$.

El administrador también explica que desean que el software que les realices tenga un método que devuelva el peso total de todos los alimentos ingeridos, en GRAMOS. Él hace especial énfasis en que la unidad DEBE tratarse en gramos debido a que el software que ya tienen maneja la información del peso de cada alimento en GRAMOS.

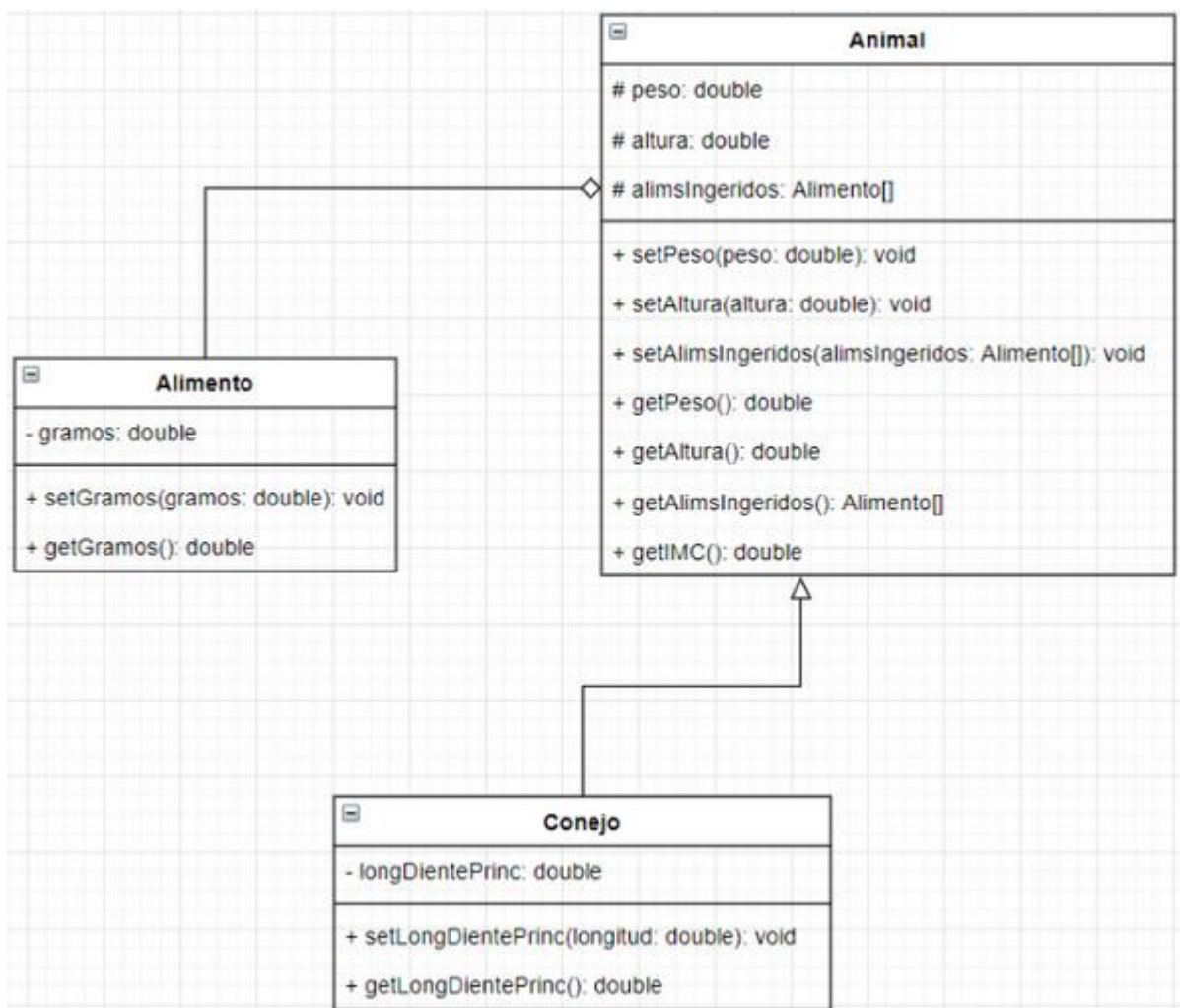
Acciones de aprendizaje

- Analizar e identificar las variables que considere necesarias para la correcta ejecución del programa.
- Determinar el algoritmo correcto que permita imprimir los datos de salida solicitados.



Escriba el código de la clase Gorila que cumpla los siguientes requisitos:

- Debe heredar de la clase Animal
- Debe hacer un constructor para la clase Gorila que reciba como primer parámetro el peso, en kilogramos, y la altura en metros, respectivamente. Recuerde que el orden de los parámetros importa.
- Debe tener un método llamado `getPesoAlims` que devuelva el peso total de los alimentos ingeridos en GRAMOS.
- Debe sobrescribir el método `getIMC` para que tenga en cuenta el peso de total de los alimentos ingeridos junto con el del gorila. Tenga cuidado con las unidades, ya que el peso de los animales está dado en kilogramos y el de los alimentos en gramos.





	Test	Esperado	Se obtuvo	
✓	<pre> Alimento banana = new Alimento(120); Alimento raices = new Alimento(60); Alimento platano = new Alimento(78.5); Gorila gorila = new Gorila(150.5, 1.67); Alimento alimentos[] = new Alimento[3]; alimentos[0] = banana; alimentos[1] = raices; alimentos[2] = platano; gorila.setAlimsIngeridos(alimentos); System.out.println(gorila.getPeso()); System.out.println(gorila.getPesoAlims()); System.out.println(Math.round(gorila.getIMC())); </pre>	150.5 258.0 54	150.5 258.0 54	✓

✓	<pre> Gorila gorila = new Gorila(190, 1.8); Alimento banana = new Alimento(155.9); Alimento raices = new Alimento(50); Alimento platano = new Alimento(89); Alimento alimentos[] = new Alimento[3]; alimentos[0] = banana; alimentos[1] = raices; alimentos[2] = platano; gorila.setAlimsIngeridos(alimentos); System.out.println(gorila.getPeso()); System.out.println(gorila.getPesoAlims()); System.out.println(Math.round(gorila.getIMC())); </pre>	190.0 294.0 59	190.0 294.0 59	✓
---	---	----------------------	----------------------	---



✓	<pre>Gorila gorila = new Gorila(160, 1.7); Alimento banana = new Alimento(90); Alimento raices = new Alimento(120); Alimento platano = new Alimento(69); Alimento alimentos[] = new Alimento[3]; alimentos[0] = banana; alimentos[1] = raices; alimentos[2] = platano; gorila.setAlimsIngeridos(alimentos); System.out.println(gorila.getPeso()); System.out.println(gorila.getPesoAlims()); System.out.println(Math.round(gorila.getIMC()));</pre>	160.0 279.0 55	160.0 279.0 55	✓
Todas las pruebas superadas. ✓				