**Sesión # 9: Componente Práctico**

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 (String)
  + Permite planchado: verdadero o falso. (boolean)
* Calzado (como lo son tenis, calzado formal, sandalias, etc) el cual debe tener el siguiente parámetro adicional:
  + Talla: 35, 36, 37, etc (String)

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%.

productosC

|  |  |  |
| --- | --- | --- |
| Código = 121  pCompra = 20000  pVenta = 30000  cBodega = 18  cMinRequerida = 20  cMaxPermitida=50  desc tenis | Código = 122  pCompra = 25000  pVenta = 35000  cBodega = 30  cMinRequerida = 12  cMaxPermitida 80  desc formal | Código=123  pCompra=30000  pVenta=38000  cBodega=15  cMinRequerida=25  cMaxPermitida 30  desc sandalia |

0 1 2

ALERTA NO SE GENERA ALERTA ALERTA

3

121

20000

30000

18

20

50

tenis

122

25

35000

30

12

80

formal

123

30000

38000

15

25

30

sandalia

productosP

|  |  |  |
| --- | --- | --- |
| Código = 12  pCompra = 10000  pVenta = 20000  cBodega = 10  cMinRequerida = 15  cMaxPermitida=20  desc blusa | Código = 13  pCompra = 15000  pVenta = 22000  cBodega = 20  cMinRequerida = 12  cMaxPermitida 80  desc jean | Código=14  pCompra=20000  pVenta=28000  cBodega=5  cMinRequerida=10  cMaxPermitida 50  desc camisa |

0 1 2

ALERTA NO SE GENERA ALERTA ALERTA

3

12

10000

20000

10

15

20

blusa

13

15000

22000

20

12

80

jean

14

20000

28000

5

10

50

camisa

Ejemplo3.

Ejercicio

Durante la sesión de hoy vamos a trabajar herencia con la clase Persona. Define la clase Persona con los atributos: **nombre, apellido, identificación, celular**.

Y crea una clase Empleado que implemente Persona, tenga los siguientes atributos adicionales: **código interno, banco, número de cuenta, mes a facturar.**

Ambas clases deben tener los métodos get y set de todos sus atributos.

Atributos de la clase **Persona**:

nombre - String

apellido - String

identificacion – int

celular – String

Orden de los parámetros de entrada para el constructor de la clase Persona: nombre, apellido,

identificacion, celular

Atributos adicionales de la clase **Empleado**:

codigo - String

banco - String

nroCuenta - int

mesFacturacion - int

Orden de los parámetros de entrada para el constructor de la clase Empleado: codigo, banco,

nroCuenta, mesFacturacion nombre, apellido, identificacion, celular