**Sesión # 10: 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.
* Mostrar: devuelva una cadena descriptiva con la información del producto separada por guión medio (-).

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

Cada subclase debe sobreescribir el método mostrar de la super clase Producto, adicionando los parámetros adicionales a mostrar en cada subclase.

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:

* Consultar producto: solicite el código de un producto a vender, el tipo (calzado o prenda) y muestre la información del mismo.
* 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  talla 40 | Código = 122  pCompra = 25000  pVenta = 35000  cBodega = 30  cMinRequerida = 12  cMaxPermitida 80  desc formal  talla 38 | Código=123  pCompra=30000  pVenta=38000  cBodega=15  cMinRequerida=25  cMaxPermitida 30  desc sandalia  talla 36 |

0 1 2

ALERTA NO SE GENERA ALERTA ALERTA

3

121

20000

30000

18

20

50

Tenis

40

122

25

35000

30

12

80

Formal

38

123

30000

38000

15

25

30

sandalia

36

productosP

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

0 1 2

ALERTA NO SE GENERA ALERTA ALERTA

3

12

10000

20000

10

15

20

Blusa

s

13

15000

22000

20

12

80

Jean

m

14

20000

28000

5

10

50

Camisa

l

Ejemplo 1.

/\*\*

Herencia

Partes del programa

1. Definición de variables

2. Creacion de objetos v1, v2, y v3 instanciando la SuperClase Vehiculo en sus

tres constructores, respectivamente.

3. Creación de objetos t1, t2, y t3 instanciando la SubClase Taxi en sus tres

constructores, respectivamente.

4. Creación de objeto c1 instanciando la SubClase Camioneta y aplica el concepto

de sobre-escritura de métodos en el método getnumSillas para aumentar su

numero de sillas sobre la base de 3 que le asigna el constructor.

Notas: En este ejemplo se usan:

a. Operador condicional ?:

b. Operador booleano || para determinar true o false en la variable aa. Evita

el uso de un if.

c. el método charAt(0) para leer un carácter de teclado y no un String.

\*\*/

/\* Herencia: SuperClase Vehículo

Acciones:

Se define la SuperClase Vehículo con los atributos

String: matricula

int: numSillas

Se definen tres constructores:

Vehiculo() sin paso de paso de parámetros. Los paramétros se inician de forma fija dentro del constructor.

Vehiculo(String matricula) con paso de parámetro matrícula y se fija el número de sillas.

Vehiculo(String matricula, int numSillas) con paso de todos los parámetros

Se definen los métodos set y get para cada una de las variables de instancias

\*/

/\* SubClase Taxi - Uso de super (diferentes formas)

Se define la SubClase Taxi con los atributos

boolean: AA

y los parámetros de la SuperClase que son heredados

String: matricula

int: numSillas

Se definen tres constructores:

Taxi(boolean AA) se usa super sin paso de parámetros para llamar al constructor de la SuperClase sin paso de paso de parámetros. El parámetro AA se inicia en el constructor Taxi.

Taxi(String matricula, boolean AA) se usa super con paso de parámetro matrícula para llamar al constructor de la SuperClase que utiliza este parámetro. El parámetro AA se inicia en el constructor Taxi.

Taxi(String matricula, int numSillas) se usa super con paso de parámetro matrícula y numSillas para llamar al constructor de la SuperClase que utiliza todos los parámetros. El parámetro AA se inicia en el constructor Taxi.

Se definen los métodos set y get para la variable de instancia AA de Taxi

\*/

/\* SubClase Camioneta - Uso de super y sobre-escritura de métodos

Se define la SubClase Camioneta con los atributos

int sillasAdicionales; Nuevo atributos

boolean: AA

y los parámetros de la SuperClase que son heredados

String: matricula

int: numSillas

Se definen un constructor:

Camioneta (String matricula, boolean AA) se usa super con paso de parámetro matrícula para llamar al constructor de la SuperClase que utiliza este parámetro.

El númerode sillas por defecto es 3.

El parámetro AA se inicia en el constructor Camioneta.

Se definen los métodos set y get para la variable de instancia AA de Camioneta

Se definen los métodos set y get para la variable de instancia sillasAdicionales

y con el método get se sobre-escribe el método getnumSillas de la SuperClase Vehiculo.

\*/