**Parcial II Programación I**

**Nombres:**

**Samuel David Idrobo Victoria**

**I Parte**

1. (0.6)¿Cuál de las siguientes formas de nombrar una variable, no es un dato primitivo en Java?
   1. char
   2. boolean
   3. String
   4. Double
2. (0.6) Cuál de las siguientes variables se encuentra **mal inicializada**, dado que se está utilizando un

# valor primitivo:

* 1. public boolean a = false
  2. protected Long a = 123456789
  3. byte a = 123
  4. private short a = 12345

1. (0.6)Si usted desarrolla un programa que calcule el índice de masa corporal de una persona y dado el valor encontrado debe arrojar los siguientes mensajes: si el cálculo da <16 debe arrojar “delgadez severa”; si da entre [16-17) “delgadez moderada”; si da entre [17-18.5) “delgadez leve”; si da entre [18.5-25) “peso normal”; si da entre [25-30) “sobrepeso”; si da entre [30-35) “Obesidad leve”; si da entre [35-40) “Obesidad moderada” y >=40 “obesidad mórbida”. Este programa debe servir para realizar un solo cálculo y ese cálculo debe arrojar el mensaje que corresponda según lo mostrado anteriormente.. Según esto ¿qué estructuras serían necesarias para su implementación? (estructura principal utilizada para arrojar el mensaje)
   1. If Else anidados o switch
   2. for
   3. while
2. (0.6)Si se requiere desarrollar un programa que determine si un número es un número de Amstrong. Un número de Armstrong es un número que es igual a la suma de sus dígitos elevados a la potencia de su

número de cifras. Por ejemplo:

Este programa puede dividirse en 2 partes, una lógica dónde se harán los cálculos correspondientes para la suma de las potencias y otra parte para verificar si el resultado de este número corresponde al numerosos Armstrong o no. Respectivamente con qué estructuras se debe implementar este sistema

* 1. Lógica(while) y verificación(If Else)
  2. Lógica(if else) y verificación(while)
  3. Lógica(switch) y verificación(while)
  4. Lógica(switch) y verificación(for)

1. (0.6) Se quiere desarrollar un programa que determine si un número es un número perfecto. Un número perfecto es aquel que es igual a la suma de sus divisores positivos. Por ejemplo, el número 28 es un número perfecto ya que sus divisores son: 1, 2, 4, 7 y 14, y la suma de estos números es 28. Este problema se puede dividir en 2 partes. Parte 1: calcular la suma de todos los divisores y parte 2: verificar si la suma de los divisores de dicho número es igual al número. Según esto qué estructuras se debería utilizar en cada parte del problema:
   1. Parte 1 (switch y if) y Parte 2 (switch)
   2. Parte 1 (switch y for) y Parte 2 (if else)
   3. Parte 1 (Do while y if) y Parte 2 (if else).
   4. Parte 1 (for y if) y Parte 2 (switch)

**Nota:** si para resolver los puntos 3, 4 y 5 utiliza una estructura diferente a la presentada en las opciones implementarla y envíarla mediante un repositorio. Recuerde estructurar la respuesta en fases tal como se presentan las opciones de cada uno de los puntos. Ejemplo.

opción e. parte 1(for y for) parte 2 (while y for).

Esto dado que los puntos pueden tener múltiples respuestas, por eso es importante enfocarse en lo solicitado en cada fase.

Esto debe demostrarse mediante un repositorio en Github, sòlo si la respuesta no se encuentra dentro de las seleccionada.

# II Parte

Una tienda de ropa requiere un programa que facilite el cálculo del monto total de las compras de sus clientes. Este cálculo debe incluir tanto el Impuesto al Valor Agregado (IVA) como los descuentos por cantidad de unidades adquiridas. Al iniciar la aplicación, se debe presentar el siguiente menú de opciones:

Menú de Compras:

Camisetas – Valor: $25.000 Pantalones – Valor: $45.000 Chaquetas – Valor: $65.000 Salir.

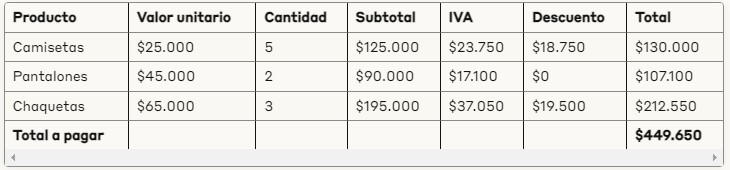
Digite su opción:

Después de elegir una opción, el programa solicitará al cliente la cantidad de prendas que desea comprar. El valor total se calculará considerando:

* IVA del 19%, aplicado sobre el costo total de las prendas adquiridas.
* Descuentos por cantidad comprada:

10% si compra 3 o 4 unidades de un mismo tipo de prenda. 15% si compra más de 4 unidades de un mismo tipo de prenda.

Es importante destacar que los descuentos se aplican sobre el subtotal antes del IVA, garantizando que los clientes obtengan beneficios al comprar en mayor cantidad. A continuación, se muestra un posible ejemplo del resultado:



Para el anterior problema:

1. (0.5)Escriba un algoritmo en java para implementar el anterior problema (lógica de Negocio).
2. (0.5) Para el problema anterior identifique cuales pueden ser los motivos por los que pueda fallar la implementación.
3. (0.5)¿Construya el diagrama UML de clases y de objetos del anterior ejercicio?
4. (0.5) Haga una prueba de escritorio con la tabla mostrada en el ejercicio, agregue las capturas de pantalla del ejercicio que evidencien que se está realizando el cálculo hasta obtener el total de 449.650.

**Nota1**: el punto 1 debe mostrar la lógica de negocio y se debe entregar mediante un repositorio en Github.

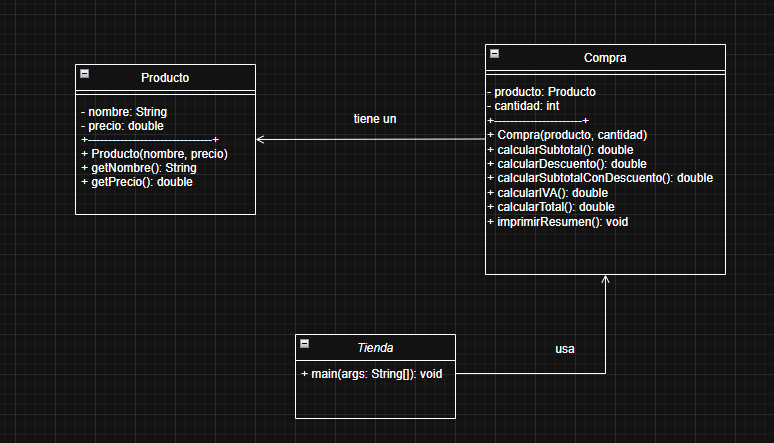
**Nota2**: Para el punto 4 se debe ademàs de hacer la prueba de escritorio agregar el archivo de prueba en java, se debe incluir en el mismo repositorio de la nota 1.

Solución

2. -Problemas quizás al querer modificar un pedido anterior, no se puede

-Por decisión propia el programa no guarda cada producto por separado (porque no se usa arrays ni listas).

- Orden incorrecto en la aplicación del descuento e IVA, algún error en un calculo hizo la diferencia de $449.650 a $442.382,5

3- 

4- Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.