

Parcial II Programación I

Nombres:

I Parte

1. (0.6) ¿Cuál de las siguientes formas de nombrar una variable, no es un dato primitivo en Java?
 - a. char
 - b. boolean
 - c. String
 - d. double
2. (0.6) ¿Cuál de las siguientes variables se encuentra **mal inicializada**, dado que se está utilizando un **valor primitivo**:
 - a. public boolean a = false
 - b. protected Long a = 123456789
 - c. byte a = 123
 - d. private short a = 12345
3. (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)
 - a. If Else anidados o switch
 - b. for
 - c. while
4. (0.6) Si se requiere desarrollar un programa que determine si un número es un número de Armstrong. 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 $371 = 3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$. Este programa puede dividirse en 2 partes, una lógica donde 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
 - a. Lógica(while) y verificación(if Else)
 - b. Lógica(if else) y verificación(while)
 - c. Lógica(switch) y verificación(while)
 - d. Lógica(switch) y verificación(for)
5. (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:
 - a. Parte 1 (switch y if) y Parte 2 (switch)
 - b. Parte 1 (switch y for) y Parte 2 (if else)
 - c. Parte 1 (Do while y if) y Parte 2 (if else)
 - d. 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 enviarla 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, sólo si la respuesta no se encuentra dentro de las seleccionadas.

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.

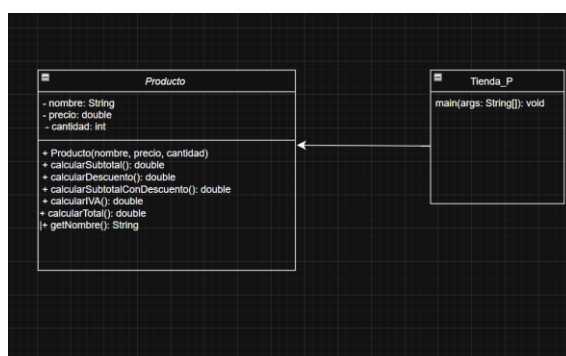
15% si compra más de 4 unidades de un mismo tipo de prenda.

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:

Producto	Valor unitario	Cantidad	Subtotal	IVA	Descuento	Total
Camisetas	\$25.000	5	\$125.000	\$23.750	\$18.750	\$130.000
Pantalones	\$45.000	2	\$90.000	\$17.100	\$0	\$107.100
Chaquetas	\$65.000	3	\$195.000	\$37.050	\$19.500	\$212.550
Total a pagar						\$449.650

Para el anterior problema:

1. (0.5) Escriba un algoritmo en java para implementar el anterior problema (lógica de Negocio).
R// EN EL REPOSITORIO DE GITHUB https://github.com/Danielito-2/Parcial_2
2. (0.5) Para el problema anterior identifique cuales pueden ser los motivos por los que pueda fallar la implementación.
R// Error en el cálculo del descuento, el descuento debe aplicarse sobre el subtotal, antes del IVA, si el orden de cálculo no es el correcto, los resultados finales serían incorrectos, por eso la diferencia del código y de los resultados finales de las tablas, si se aplica el IVA al subtotal completo en vez del subtotal después del descuento, el total final sería mayor al que sería el correcto.
3. (0.5) ¿Construya el diagrama UML de clases y de objetos del anterior ejercicio?
Lo coloque en 2 clases, que la Tienda_P tenga una relación directa con el producto para hacer los cálculos, tal cual como esta en el código, pero también se puede separar en 3 archivos. Subo Jpg también a GitHub



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..

Camisetas

Producto	Camisetas
Valor unitario	\$25.000
Cantidad	5
Subtotal	\$125.000
Descuento	\$18.750
IVA	\$23.750
Total	\$130.000

Pantalones

Producto	Pantalones
Valor unitario	\$45.000
Cantidad	2
Subtotal	\$90.000
Descuento	\$0
IVA	\$17.100
Total	\$107.100

Chaquetas

Producto	Chaquetas
Valor unitario	\$65.000
Cantidad	3
Subtotal	\$195.000
Descuento	\$19.500
IVA	\$37.050
Total	\$212.550



Total

Total a pagar	\$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 además de hacer la prueba de escritorio agregar el archivo de prueba en java, se debe incluir en el mismo repositorio de la nota 1.