Tarea 2 – Clases – Constructores (por defecto y parametrizados)

Enunciado

1. Crear una clase llamada ***ClienteVip***. Debe tener 4 campos ***nombre***, ***limiteCredito***, ***email*** y ***tipoVip*** que serán String, double, String respectivamente y tipoVip será de tipo enum con las siguientes opciones: GOLD, SILVER, NORMAL.
   Crea tres constructores:
   a. El primero será el constructor sin parámetros también llamado constructor por defecto. Este constructor llamará al constructor con 4 parámetros con valores por defecto.
   b. El segundo constructor tendrá 2 parámetros y llamará al constructor de 4 parámetros con dos valores por defecto.
   c. El tercer constructor incluirá todos los campos, es decir, tendrá 4 argumentos.
   Codifica todos los getters y setters de la clase.
   Genera tú código de prueba en el método main para probar la clase y sus métodos.

2. Codifica una clase con el nombre ***Muro***. La clase necesita dos campos (variables de instancia) con nombre ***ancho*** y ***alto*** de tipo double.
   La clase debe tener dos constructores. El primer constructor (constructor por defecto sin parámetros). El segundo constructor tiene los parámetros de ancho y alto de tipo double y debe inicializar los campos. En caso de que el ancho sea menor que 0, debe establecer el valor del campo de ancho en 0, en caso de que el parámetro de altura sea menor que 0, debe establezca el valor del campo de altura en 0.
   Escribe los siguientes métodos (métodos de instancia):
   a. Método llamado ***getAncho*** sin ningún parámetro, devuelve el valor del campo ancho.
   b. Método llamado **getAlto** sin ningún parámetro, devuelve el valor del campo alto.
   c. Método llamado **setAncho** con un parámetro de tipo double, establece el valor del campo ancho. Si el parámetro es menor que 0, establece el valor del campo ancho en 0.
   d. Método llamado **setAlto** con un parámetro de tipo double, establece el valor del campo alto. Si el parámetro es menor que 0, establece el valor del campo alto en 0.
   e. Método llamado **getArea** sin ningún parámetro, devuelve el área del muro.
   CÓDIGO TEST

```
Muro muro = new Muro(5,4);
System.out.println("area= " + muro.getArea());

muro.setHeight(-1.5);
System.out.println("ancho= " + muro.getWidth());
System.out.println("alto= " + muro.getHeight());
System.out.println("area= " + muro.getArea());
```

   → OUTPUT:
   area= 20.0
   ancho= 5.0
   alto= 0.0
   area= 0.0

3. Tienes que representar un punto en el espacio 2D. Escribe una clase con el nombre ***Punto***. La clase debe tener dos campos (variables de instancia) con el nombre ***x*** e ***y*** de tipo int.

La clase tiene dos constructores, el primer constructor no tiene parámetros (constructor sin argumentos), el segundo constructor tiene los parámetros x e y de tipo int y debe inicializar los campos.
Codifica los siguientes métodos (métodos de instancia):

a. Método llamado **getX** sin ningún parámetro, devuelve el valor del campo x.
b. Método llamado **getY** sin ningún parámetro, devuelve el valor del campo y.
c. Método llamado **setX** con un parámetro de tipo int, establece el valor del campo x.
d. Método llamado **setY** con un parámetro de tipo int, establece el valor del campo y.
e. Método llamado **distancia** sin ningún parámetro, devuelve la distancia entre el Punto this (el objeto actual) y el Punto 0,0, con valor x=0 y=0 como doble.
f. Método llamado **distancia** con dos parámetros x e y ambos de tipo int, devuelve la distancia entre el Punto (this) y el Punto x e y, como doble.
g. Método llamado distancia con parámetro otro de tipo Punto, devuelve la distancia entre este Punto y otro Punto como doble.
   Ayuda. ¿Cómo encontrar la distancia entre dos puntos? Para encontrar la distancia entre los puntos A (xA, yA) y B (xB, yB), usamos la fórmula:

   d (A, B) = $\sqrt{(xB - xA) * (xB - xA) + (yB - yA) * (yB - yA)}$

   Utiliza el método ***Math.sqrt(valor)*** para calcular la raíz cuadrada √.
   CÓDIGO TEST

```
Punto primero = new Punto(6, 5);
Punto segundo = new Punto(3, 1);
System.out.println("distancia(0,0)= " + primero.distancia());
System.out.println("distancia(segundo)= " + primero.distancia(segundo));
System.out.println("distancia(2,2)= " + primero.distancia(2, 2));
Punto punto = new Punto();
System.out.println("distancia()= " + punto.distancia());
```

   OUTPUT

   distancia(0,0)= 7.810249675906654
   distancia (segundo)= 5.0
   distancia (2,2)= 5.0
   distancia ()= 0.0

4. La compañía Mundo Alfombra te ha pedido que escribas una aplicación que calcule el precio de las alfombras para habitaciones rectangulares. Para calcular el precio, se multiplica el área del piso (ancho por largo) por el precio por metro cuadrado de alfombra. Por ejemplo, el área de un piso que mide 12 metros de largo y 10 metros de ancho es 120 metros cuadrados, cubrir el piso con una alfombra que cuesta 8€ por metro cuadrado costaría 960€.
   Vamos a crear las siguientes clases con sus constructores y métodos:
   a. Codifica una clase con el nombre ***Suelo***. La clase necesita dos campos (variables de instancia) con nombre ***ancho*** y ***largo*** de tipo double.
      La clase debe tener un constructor con parámetros **ancho** y **largo** de tipo double y debe inicializar los campos.
      En caso de que el parámetro de ancho sea menor que 0, se establecerá el valor del campo ancho en 0, en caso de que el parámetro largo sea menor que 0, se establecerá el valor del campo largo en 0.
      Escribe los siguientes métodos (métodos de instancia):
      i. Método llamado ***getArea*** sin ningún parámetro, devuelve el área calculada (ancho * largo).

b. Desarrolla una clase llamada ***Alfombra*** con un campo llamado ***coste*** de tipo double. Incluye el constructor de la clase con un parámetro para inicializar el campo coste. En caso de que el parámetro coste sea menor que 0, se establecerá el valor del campo coste en 0. Añade el siguiente método:
   i. Método llamado ***getCoste*** sin ningún parámetro, devuelve el valor del campo coste.

   d. Escribe una clase con el nombre ***Calculadora*** con dos campos (variables de instancia) ***suelo*** de tipo Suelo y ***alfombra*** de tipo Alfombra. Codifica el constructor parametrizado con los parámetros de suelo de tipo Suelo y alfombra de tipo Alfombra e inicializa los campos. Añade el siguiente método en la clase:
   ii. Método ***getCosteTotal*** sin ningún parámetro, devuelve el coste total calculado para cubrir el piso con una alfombra.

→ TEST:

```
Alfombra alfombra = new Alfombra(3.5);
Suelo suelo = new Suelo(2.75, 4.0);
Calculadora calculadora = new Calculadora(suelo, alfombra);
System.out.println("total= " + calculadora.getCosteTotal());
alfombra = new Alfombra(1.5);
suelo = new Suelo(5.4, 4.5);
calculadora = new Calculadora(suelo, alfombra);
System.out.println("total= " + calculadora.getCosteTotal());
```

→ OUTPUT

```
total= 38.5
total= 36.45
```

Juega con el código y crea situaciones de prueba cuando tenemos ancho y largo con valores negativos. Y cuando nos dan un precio de coste negativo. Muestra las salidas por pantalla.

5. Es tu turno. Crea más de una clase, como hicimos en el ejercicio anterior donde definas las clases, campos, constructores y métodos para cada una de ellas. Las clases deben estar relacionadas y realizar alguna función coherente que necesite de ellas.
   Opción extra. Intenta incluir un campo de tipo enum en tu clase.

Entrega

- Los métodos NO deben ser public static.
- Realiza capturas con las salidas del programa en un PDF.
- Copia el código de las clases java.