

Práctica 3

Desarrollo de Clases (parte I)

Objetivo. Definir clases para representar objetos del mundo real. Concepto de clase, estado (variables de instancia) y comportamiento (métodos). Instanciación. Mensajes.

Nota: Trabajar sobre la carpeta “tema3” del proyecto

1-A- Definir una clase para representar triángulos. Un triángulo se caracteriza por el tamaño de sus 3 lados (double), el color de relleno (String) y el color de línea (String). El triángulo debe saber:

- Devolver/modificar el valor de cada uno de sus atributos (métodos get y set)
- Calcular el área y devolverla (método `calcularArea`)
- Calcular el perímetro y devolverlo (método `calcularPerimetro`)

B- Realizar un programa que instancie un triángulo, le cargue información leída desde teclado e informe en consola el perímetro y el área.

NOTA: Calcular el área con la fórmula $\text{Área} = \sqrt{s(s-a)(s-b)(s-c)}$, donde a, b y c son los lados y $s = \frac{a+b+c}{2}$. La función raíz cuadrada es `Math.sqrt(#)`

2-A- Definir una clase para representar balanzas comerciales (para ser utilizadas en verdulerías, carnicerías, etc). Una balanza comercial sólo mantiene el monto y la cantidad de items correspondientes a la compra actual (es decir, no almacena los ítems de la compra). La balanza debe responder a los siguientes mensajes:

- `iniciarCompra()`: inicia el monto y la cantidad de ítems para la compra actual.
- `registrarProducto(pesoEnKg, precioPorKg)`: recibe el “*peso en kilos*” del ítem comprado y su “*precio por kilo*”, y actualiza el estado de la balanza.
- `getMonto()`: retorna el monto a pagar por la compra actual.
- `getResumenDeCompra()`: retorna un String del siguiente estilo “Total a pagar X por la compra de Y productos”, donde X es el monto e Y es la cantidad de ítems de la compra.

B- Realizar un programa que instancie una balanza e inicie una compra. Lea desde teclado información de los ítems comprados (peso en kg y precio por kg) hasta ingresar peso 0. Registre cada producto en la balanza. Al finalizar, informe el resumen de la compra.

3-A- Definir una clase para representar entrenadores de un club. Un entrenador se caracteriza por su nombre, sueldo básico, antigüedad y cantidad de campeonatos ganados.

- Defina métodos para obtener/modificar el valor de cada atributo.
- Defina el método `calcularEfectividad` que devuelve la efectividad del entrenador, que es el promedio de campeonatos ganados por año de antigüedad.
- Defina el método `calcularSueldoACobrar` que devuelve el sueldo a cobrar por el entrenador. El sueldo a cobrar es el sueldo básico agregando un 10% del básico por año de antigüedad, y además se adiciona un plus por campeonatos ganados (5000\$ si ha ganado entre 1 y 4 campeonatos; \$30.000 si ha ganado entre 5 y 10 campeonatos; 50.000\$ si ha ganado más de 10 campeonatos).

B- Realizar un programa que instancie un entrenador, cargándole datos leídos desde teclado. Informe el sueldo a cobrar y la efectividad del entrenador.

Taller de Programación 2022 – Módulo POO

4-A- Definir una clase para representar círculos. Los círculos se caracterizan por su radio (`double`), el color de relleno (`String`) y el color de línea (`String`). El círculo debe saber:

- Devolver/modificar el valor de cada uno de sus atributos (métodos `get` y `set`)
- Calcular el área y devolverla (método `calcularArea`)
- Calcular el perímetro y devolverlo (método `calcularPerimetro`)

B- Realizar un programa que instancie un círculo, le cargue información leída de teclado e informe en consola el perímetro y el área.

NOTA: la constante `PI` es `Math.PI`

5-A- Se dispone de la clase `Producto` (en la carpeta `tema3`). Un objeto producto puede crearse sin valores iniciales o enviando en el mensaje de creación el “*peso en kg*” y “*descripción*” (en ese orden). Un objeto producto responde a los siguientes mensajes:

<code>getDescripcion()</code>	retorna la descripción (<code>String</code>) del producto
<code>getPesoEnKg()</code>	retorna el peso en kg (<code>double</code>) del producto
<code>setDescripcion(X)</code>	modifica la descripción del producto al “ <code>String</code> ” <code>X</code>
<code>setPesoEnKg(X)</code>	modifica el peso del producto al “ <code>double</code> ” <code>X</code>

Usando la clase `Producto`. Realice las siguientes modificaciones a la clase del ejercicio 2-A. Ahora la balanza debe generar un resumen de compra más completo. Agregue a la balanza la característica *resumen* (`String`) y modifique los métodos como se indica a continuación:

- `iniciarCompra` para que además inicie el *resumen* en el `String` vacío.
- `registrarProducto` para que reciba un objeto `Producto` y su “*precio por kg*”. La operación debe actualizar el monto y cantidad de ítems de la balanza, y concatenar al *resumen* la descripción y el monto a pagar por este producto.
- `getResumenDeCompra` para que retorne un `String` del siguiente estilo: “Naranja 100\$ - Banana 40\$ - Lechuga 50\$ - Total a pagar **190** pesos por la compra de **3** productos” donde la parte subrayada es el contenido de *resumen*.

B- Realice las modificaciones necesarias en el programa principal solicitado en 2-B para corroborar el funcionamiento de la balanza.