Nombre y Apellido: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cantidad de hojas entregadas: \_\_ Nota:\_\_\_\_\_\_\_\_

|  |
| --- |
| **Lea atentamente TODO el enunciado antes de empezar a desarrollar, prestar mucha atención a las firmas de las funciones, algunas pueden tener nombre plural y otras en singular.** |

1. Se solicita crear la clase Kiosco, la misma contiene 4 estanterías con productos diferentes (bebidas, caramelos, alfajores y helados) (6ptos).

Se solicita crear la clase Kiosco, Bebida, Caramelo, Colacion y Helado. Y que el Kiosco contenga una dependencia en forma asociativa a ellas (Composición). Es decir, la clase principal almacenara un conjunto de objetos de las clases secundarias, mediante ArrayList<Tipo>.  
A cada uno de estas listas, de aquí en adelante, las llamaremos estanterías.

Todas las clases de productos deben almacenar: marca, nombreProducto y precio(tipo double). Todos sus atributos deben permanecer privados, con los getters correspondientes getMarca getNombreProducto y getPrecio.

Y en particular a cada producto:

* Si el nombre de la bebida es diferente de “agua”, debe contener algún atributo que indique que la bebida es azucarada/endulzada, el mismo se debe acceder por un método especial de la clase Bebida **public boolean azucarada()**
* Si la Colación es un “alfajor” se debe aumentar un 15% el precio del mismo
* Si el Helado es de marca “kibon” y es un helado “magnum” se debe aumentar un 50% el precio

Para la clase Kiosco se solicita:

1. Cada estantería debe contar con un método para agregar un producto a la misma.  
   Es decir, **public void agregarCaramelo(Caramelo c)** debe agregar un caramelo a la estantería de caramelos, así, con las 4. Si dos productos son exactamente iguales en la estantería lanzar IllegalArgumentException()
2. Dos métodos diferentes para añadir descuentos a bebidas azucaradas y no azucaradas. Mediante los métodos **void setDescuentoBebidasAzucaradas(int porcentaje)** y **void setDescuentoBebidasNoAzucaradas(double porcentaje)**. El parámetro es un porcentaje de descuento los productos de la estantería
3. Un método que calcule el precio de cada estantería. la firma debe ser **double calcularPrecioCaramelos()**
4. Una función que calcule dada una marca de Helados, el precio de todos sus productos de la estantería. la firma debe ser **double calcularPreciosHeladosDe(String marca)**