

**1. Genera un proyecto con un paquete llamado store. Descarga las clase Store.java y Product.java y StoreDemo.java de Blackboard, e impórtalas a tu proyecto.**

**2. Diseña una subclase llamada CoffeeShop que herede la clase Store. La nueva clase CoffeeShop debe tener los siguientes atributos y métodos:**

- `maxOccupancy` como una variable de tipo entero para representar el aforo máximo del restaurante.
- `menuItems` como un arreglo objetos de la clase `Product` que servirá para almacenar el nombre y los precios de cada uno de los productos en el menu.
- `menuItemsCount` como un entero que servirá para almacenar la cantidad de elementos que existen en el arreglo `menuItems`.
- **Método constructor** que reciba como parámetro de entrada los siguientes elementos:
  - `address`
  - `storeName`
  - `employeeCount`
  - `maxOccupancy`

El arreglo `menuItems` lo instanciará con espacio para almacenar 100 ítems. `menuItemsCount` comenzará con un valor de 0.

Nota: recuerda que el constructor de la clase **CoffeeShop** debe hacer siempre una llamada al constructor de la clase base.

- Un segundo método **constructor** que no reciba parámetros de entrada, pero que utilice el método constructor 1 para instanciar el objeto con valores default.

Nota: recuerda utilizar el método `this( )` para esto.

- **Método `addMenuItem(Product item)`** que agregará el producto recibido como parámetro al arreglo `menuItems`, e incrementará en 1 la variable `menuItemsCount`.
- **Método `toString()`** que retorne la información de todas las variables de instancia de la clase, incluidas aquellas variables que han sido heredadas.
  - `address`
  - `storeName`
  - `employeeCount`
  - `maxOccupancy`
  - ***cada elemento del menu y su respectivo precio***

Recuerda que puedes reutilizar la implementación de `toString()` de la clase base llamando `super.toString()`.

- **Método `openShop()`** que imprima en pantalla tres líneas con la siguiente información:
  - Mensaje de bienvenida a la tienda, que incluya el nombre de ésta.
  - El menú de los productos que vende.
- **Método `closeShop()`** que imprima en pantalla:
  - Mensaje de despedida de la tienda, que incluya el nombre de ésta.

**3. Diseña una subclase llamada ClothingStore a partir de la clase base Store. La clase debe contener:**

- 2 atributos adicionales que se te ocurran.
- **Método constructor** que instancie todas las propiedades de la clase `ClothingStore`, incluidas las propiedades heredadas.

- Método constructor sin parámetros de entrada, que únicamente inicialice las variables de instancia vacías.
- 1 método adicional único para tu clase (defínelo tu)
- Redefine el método `toString()` que imprima todos los atributos de la clase `ClothingStore`.
- Redefine los métodos `openShop()` y `closeShop()` con actividades que sean representativas de una tienda de ropa. Ejemplo:
  - We are doing an inventory count!
  - We are cleaning the Store!

#### **4. Utiliza la clase `StoreDemo.java`, adjunta en Blackboard, para probar tu clase.**

- Declara un arreglo de 3 elementos con variables de tipo `Store`.
- En el primer elemento, instancia un objeto de la clase `Store`.
- En el segundo elemento, instancia un objeto de tipo `CoffeeShop`. Agrega 3 productos al menú.
- En el tercer elemento, instancia un objeto de tipo `ClothingStore`.
- Para cada elemento del arreglo, prueba la impresión de la información del objeto, así como los métodos `openShop()` y `closeShop()`.