En este ejercicio tendrás que programar una versión simplificada de un problema clásico de programación concurrente. En este programa se quiere simular a personas que acuden a un bar a tomar una cerveza para después devolver el vaso y volver a pedir otra.



Lee detenidamente todo el examen antes de comenzar con la implementación.

**FR1 [2 puntos]. Clase VasoCerveza:** representa el elemento que será consumido por los Clientes y preparado por los Camareros.

* **1.1. Atributos:**
  + *id:* int (identificador del vaso) - Valores aceptados: 0,1,2,3, ...
  + *tipo:* int - Valores aceptados: 0 media pinta, 1 pinta
* **1.2. Métodos:**
  + *Constructor*
  + *Getters y setters*
  + *toString*

**FR2 [2 puntos]. Clase Camarero:** simula la persona encargada de servir y devolver vasos de cerveza. Deberá recibir un nombre como argumento en el constructor.

* **2.1. Atributos:**
* *listaVasos:* lista que contendrá los Vasos de Cerveza.
* **2.2. Métodos:**
  + *Constructor:* deberá crear 3 vasos (de tipo aleatorio (0 o 1)) y añadirlos a la lista así como asignarse a sí mismo un nombre.
  + *servirCerveza:* elegirá un vaso aleatoriamente de la lista, lo sacará de ella y lo entregará al cliente para que pueda beber su cerveza.
  + *devolverCerveza:* inserta de nuevo en la lista el vaso devuelto.
  + *contarVasos:* imprime los vasos disponibles en el bar

**FR3 [2 puntos].** Clase HilosClientes: que extienda **Thread**. Deberá recibir un nombre como argumento en el constructor y asignarlo al hilo usando la función adecuada. El método **run()** deberá implementar el siguiente algoritmo:

* Indicar que el hilo está ejecutándose
* Infinitamente repetir:
  + Pedir un vaso de Cerveza
  + Beber la rica y deliciosa Cerveza
  + Ir contabilizando la cantidad total de cerveza bebida (en LITROS)
  + Devolver el vaso de Cerveza
  + Esperar antes de pedir otra Cerveza (dormir al cliente un tiempo aleatorio entre 250 ms y 1000 ms)

**FR4 [2 puntos].** Crea una **Aplicación (main)** que genere un **Camarero**, de nombre *Mou* y creará los siguientes **Clientes**: *Homer, Barney, Carl, Lenny y Lurleen*. Cada *Cliente* recibirá el objeto compartido *Camarero*. Adicionalmente, se deberá sincronizar el uso del objeto compartido y se deberán realizar varias pruebas de ejecución para garantizar que el sistema implementado funciona correctamente.

**FR5 [2 puntos].** Se debe cumplir con las siguientes características adicionales de implementación:

* Todos los métodos (de todas las clases) deben imprimir por pantalla información respecto a su estado de ejecución.
* Implementa el control de errores.
* Documenta el código.