

Universidad de Las Palmas de Gran Canaria
Escuela de Ingeniería Informática
Examen de Programación IV. Programación Concurrente
18 de mayo 2015

Nombre y apellidos:

Notas:

- La duración del examen es de 60 minutos.
- Permanezca en su lugar. Para hablar con el profesor, levante la mano.
- Debe tener visible sobre la mesa un documento que le identifique.
- No se permite el acceso a otra información que no sean las transparencias de clase y la documentación de referencia de java en <http://docs.oracle.com/javase/7> o <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>
- El examen se realizará utilizando únicamente la página de edición correspondiente en el servidor aulaga.
- Sólo se permite un lápiz o bolígrafo durante la realización del examen.
- Si necesita escribir anotaciones, use el reverso de esta hoja, que deberá entregar al finalizar.
- Antes de comenzar el examen escriba su nombre y apellidos arriba.

Descripción

Se desea construir un conjunto de clases coordinadas que representen un mercado de compraventa de gofio. Las clases a crear serán **Tienda**, **Cliente** y **Agricultor**. Se quiere hacer una simulación de la evolución del sistema para 4 clientes, 2 agricultores y una tienda. Se suministra un boceto de estas clases, donde se establecen los métodos a implementar. Se tiene ya implementada una clase **Main** que coordina la simulación. Además, se dispone de una clase **Log** que deberá usar para mostrar los eventos de la simulación y otra clase **ValoresSimulacion**, cuyos métodos estáticos, que se mencionan a continuación, nos dan los valores a emplear en la simulación.

Los objetos de la clase **Cliente** en un bucle (en un hilo propio) se encargarán de comprar en la tienda de gofio *cantidadAComprar()* y consumirlo a un ritmo indicado por *tiempoConsumoKilo()*. Si la tienda no les suministra la compra, devuelve falso el método comprar de la tienda y terminan el hilo. Los clientes deben contabilizar el total de gofio que compran.

Los objetos de la clase **Agricultor** realizarán 5 iteraciones (en un hilo propio) de cosechar tras un tiempo dado por *tiempoCosecha()* se encargarán de vender en la tienda de gofio *cantidaCosechada()*, en caso de el método vender de la tienda devuelva falso quiere decir que no se ha podido realizar la venta descartandola, aunque siguen las iteraciones. Los agricultores deben contabilizar el total de gofio que venden realmente a la tienda. Al finalizar las 5 iteraciones termina el hilo.

La clase **Tienda** dispone de una capacidad máxima de almacenamiento de gofio en kilos que se establece en su constructor. Su método *vender* se encarga de intentar recibir un número de sacos de los agricultores, si se almacenan devuelve true, si no logra almacenarlos (por falta de espacio) durante *esperaVenta()* milisegundos devuelve false. Su método *comprar* se encarga dar a los clientes los kilos pedidos, si lo consigue devuelve true, si no logra dar la cantidad solicitada (por falta de stock) durante *esperaCompra()* milisegundos devuelve false.

Estudie las clases Log, ValoresSimulacion y Main para realizar correctamente el examen.
Implemente las clases Cliente, Agricultor y Tienda como se describe arriba.