

Universidad de Las Palmas de Gran Canaria
Escuela de Ingeniería Informática
Examen parcial de Programación IV. Programación Concurrente
11 de mayo 2014

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>
- 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 auto-servicio de préstamo de bicicletas. Las clases a crear serán Bicicleta, Estacion y Usuario. Un objeto de la clase Bicicleta representa una bicicleta que puede ser prestada. Un objeto de la clase Estacion representa una estación de bicicletas con capacidad para un número fijo de bicicletas y donde están almacenadas las bicicletas disponibles y donde se pueden alquilar o devolver bicicletas. Los usuarios son los elementos activos que pueden ir a una estación alquilar una bicicleta y devolverla en otra. El detalle de dichas clases se muestra a continuación.

Clase Bicicleta

Los objetos de la clase Bicicleta disponen de un identificador numérico (int). Se tiene un constructor al que se le pasa el identificador numérico y se dispone de un método getId que nos devuelve su identificador.

Clase Estacion

Debe tener un constructor con dos parámetros: un identificador de estación (int) y un número con la capacidad de bicicletas de la estación. El constructor genera un número de bicicletas igual a su capacidad de almacenamiento. Los identificadores de las bicicletas creadas serán el identificador de la estación multiplicado por 1000 más una secuencia de números desde 0 hasta la capacidad de la estación menos 1. La estación debe disponer de tres métodos: getId, alquila y devuelve. El método getId nos devuelve el identificador de la estación. Alquila se le pasa un identificador de usuario (para mostrarlo) y nos devuelve una bicicleta de las disponibles, en caso de que no exista ninguna disponible, espera hasta 10 segundos a que haya una disponible y si no la hay devuelve null. Al método devuelve se le pasa una bicicleta y un identificador de usuario y almacena la bicicleta cuando haya algún hueco disponible (esperando lo que fuera necesario).

Clase Usuario

La clase Usuario dispone de un constructor al que se le pasa un identificador y dos estaciones de bicicletas una de salida y otra de llegada. El usuario, al ejecutar el método start, ejecutará en un hilo propio la siguiente secuencia:

- 1) Intenta alquilar una bicicleta en la estación de salida. Si no obtiene la bicicleta abandona el proceso y finaliza su ejecución.
- 2) Simulando un paseo, hace una espera de entre 4000 y 7000 milisegundos elegidos aleatoriamente.
- 3) Intenta devolver la bicicleta en la estación de llegada. Finalizando su ejecución cuando lo consiga.

Se dispone de una clase de mostrado llamada Log, ver código suministrado. Se suministra hecha la clase Bicicleta y la clase Main. La clase main se encarga de realizar una simulación de crear dos estaciones y 50 usuarios con intervalos de entre 0 y 2000 milisegundos.

Implemente las clases Estacion y Usuario como se describe arriba.