ESTRUCTURA DE DATOS Y ALGORITMOS **24-Enero-2008**
Titulación: Ingeniero Técnico en Informática de Gestión **Curso: 2º**

NOTA FINAL: Nota Practica (1 punto) + Nota Examen (9 punto)
Es indispensable aprobar el examen (4,5 puntos) para aprobar la asignatura (5 puntos)
La práctica es opcional
Duración: 3 horas
No está permitido el uso de apuntes, libros o móviles
Publicación de las notas: 11-2-2008
Revisión del examen: 12-2-2008, 10:00 - 13:00

1. Ejercicio (1 puntos)

Dado el array A={8, 3, 7, 1, 4, 9, 5, 2, 6}

Describir cual es el estado del array A después de cada paso principal del algoritmo: `SelectionSort`.

2. Ejercicio (1'5 puntos)

Implementar un método que dado un array de *`Persona`*s, ordene el array por el nombre de la persona y en orden descendente, utilizando el algoritmo **genérico** `insertionSort`. La clase `Persona` contiene la información del nombre y la edad. Implementa también el algoritmo `insertionSort`.

3. Ejercicio (1 punto)

Dado un array que contiene los elementos {8, 13, 17, 26, 44, 56, 88, 97} y utilizando el algoritmo de *búsqueda binaria*, **trazar** las etapas necesarias para encontrar el número 88.

4. Ejercicio. (2'5 puntos)

Se pide:

1. **Definir** las clases necesarias para definir la estructura de listas doblemente enlazadas para almacenar cualquier tipo de contenido.
2. **Realizar** una figura que describa el siguiente contenido 8, 1, 5, 3, en el mismo orden descrito.
3. **Realizar** una figura que describa una lista vacía.
4. **Implementar** el método `fusionar`, que extiende la funcionalidad de la lista doblemente enlazada (y con acceso por tanto a las referencias internas) y al que se le pasa otra segunda lista como parámetro y que, cambiando sólo un máximo de 3 referencias y sin duplicar las celdas de las listas, encadene esta segunda lista al final de la lista representada por el objeto actual (el invocador).

ESTRUCTURA DE DATOS Y ALGORITMOS **24-Enero-2008**
Titulación: Ingeniero Técnico en Informática de Gestión **Curso: 2º**

5. Ejercicio (3 puntos)

Desarrollar un sistema informático que se ocupe de la gestión de una tienda que vende libros por Internet. Para ello suponemos disponibles los módulos que implementan a las siguientes clases, todas ellas equipadas con operaciones de de igualdad (i.e. *`equals`*) y orden (i.e. *`compareTo`*):

- `Cliente`, que sirve para almacenar y gestionar la información sobre un cliente: su nombre, dirección, …
- `Libro`, que sirve para almacenar y gestionar la información sobre un libro: su título, autor, tema, ISBN, …
- `Pais`, que sirve para representar los países a los que pertenecen los clientes. `Cliente` está equipada con la observadora (método `get`) país que devuelve el país donde reside el cliente.
- `Tema`, que sirve para representar los temas de los libros. La clase `Libro` está equipada con la observadora (método `get`) tema que devuelve el tema al que pertenece el libro.

Se pide implementar la clase `Tienda`, cuya misión es gestionar la información sobre los libros disponibles y sobre los pedidos de los clientes. Esta clase ofrece las siguientes operaciones públicas:

public Tienda(): inicializa una tienda vacía.
public void insertaLibro(Libro l, int numero): añade a una tienda un número determinado de ejemplares de un cierto libro. El libro podía estar ya disponible, con lo que esta operación sólo modificará el número de ejemplares, o puede que se trate de un libro nuevo.
public void pedido(Cliente cliente, LinkedList libros): esta operación añade a una tienda la información de un pedido que ha realizado un cliente determinado. La información sobre el pedido está dada como una secuencia de libros. Aunque el envío de los libros no se realiza automáticamente al recibirse el pedido, los libros solicitados se consideran reservados, por lo que esta operación se deberá encargar de actualizar adecuadamente el número de ejemplares disponibles.
public LinkedList envíoLibros(String pais): esta operación consulta y modifica una tienda para obtener la información sobre los pedidos pendientes de envío que tienen como destino a un país determinado, y darlos por realizados. Esta operación devuelve una secuencia donde cada elemento contiene los datos de un cliente y la secuencia de libros que ese cliente ha pedido (y tiene reservados).

Se pide:
a.) **Desarrollar** en Java una implementación de la clase `Tienda` basada en otras estructuras de datos conocidas, **optimizando la complejidad** temporal de las operaciones. **Argumenta**, brevemente, la forma de estructura que has seleccionado para organizar la información de la tienda. Se debe implementar cada una de las operaciones indicadas en base a la estructura seleccionada, **controlando los posibles errores** que se puedan ocasionar.

Nota: Las clases `Cliente, Libro, Pais y Tema` no tienen ninguna relación entre sí. Si necesitas cualquier otra clase, aparte de las estructuras de datos conocidas, impleméntala.