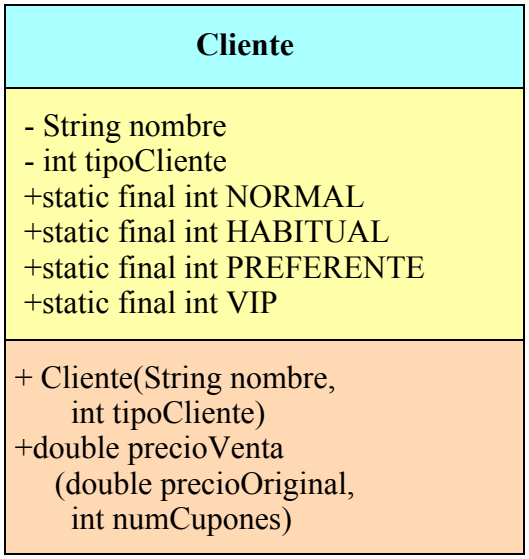
Examen de Introducción al Software
(Ingeniería Informática)

Febrero 2014

Primera parte (5 puntos, 50% nota del examen)

1) Se dispone de la clase Cliente cuyo diagrama se muestra en la figura.



Escribir el método precioVenta() que calcula y retorna el precio de un producto de un supermercado a partir del parámetro precioOriginal. A este precio se le aplica el descuento indicado en la tabla inferior, según el valor del atributo tipoCliente. Finalmente, al precio obtenido se le aplica una reducción por los cupones de promoción entregados por el cliente. El número de cupones se indica en el parámetro numCupones. Por cada cupón se reduce el precio 1 euro, hasta un máximo de 3 cupones (es decir, si se aportan más de tres cupones se reduce el precio solo en 3 euros).

tipoCliente	descuento (%)
NORMAL	0%
HABITUAL	5%
PREFERENTE	7%
VIP	10%

Nota: usar las instrucciones condicionales más apropiadas a cada caso

2) Escribir un método en Java al que se le pasa como parámetro un array de números reales, y que calcula y retorna la suma de todos los que sean positivos.

3) Escribir un método al que se le pasa como parámetro un String str que contiene una frase. El método debe calcular y retornar el substring que contiene los caracteres de str desde el primero hasta la posición del primer espacio en blanco que se encuentre en la segunda mitad de str. En el substring retornado no se incluirá ese espacio en blanco.

   Si no hay ningún espacio en blanco que cumpla lo indicado se retornará el string str completo.

4) Suponemos que estamos en un computador que no dispone de operación "elevar a". Para calcular x^n siendo n un entero positivo es posible aplicar esta definición recursiva:

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ (x^{n/2})^2 & \text{si } n \text{ es par} \\ x \cdot (x^{n/2})^2 & \text{si } n \text{ es impar} \end{cases}$$

Nota: el cociente $n/2$ se hace con números enteros y redondeando el resultado por abajo.

Escribir el *pseudocódigo* de un método *recursivo* que calcule y retorne la potencia entera de un número real x elevado a n, usando la definición indicada; x y n serán parámetros del método. En este método establecemos un caso directo que corresponde a n=0, en el que se retorna 1. En el caso recursivo se distingue si n es par o impar. En el primer caso se retorna $x^{n/2}$ multiplicado por sí mismo, mientras que en el caso impar se retorna lo mismo pero multiplicado por x.

5) Contestar *razonadamente* a las siguientes preguntas. Utilizar un *máximo* de *3 líneas* para cada respuesta:

   a. En una hoja de cálculo, ¿qué diferencias hay entre una celda con una etiqueta o con una fórmula?

   b. Escribir una fórmula para una hoja de cálculo que calcule la suma de las 10 primeras celdas de la columna B de otra hoja del mismo fichero llamada "hoja3".

   c. Se parte de una hoja de cálculo con diversas columnas de datos ya metidos. Describe brevemente los pasos a realizar para añadir una gráfica que represente esos datos.

   d. ¿Cuáles son los principales tipos de datos que podemos almacenar en un registro de una base de datos? Indicar unos cinco tipos.

   e. ¿Qué funcionalidad útil encuentras en una base de datos frente a la de una hoja de cálculo?

Nota: en esta cuestión, las respuestas correctas suman 0.2 puntos, las incompletas o las que pasen de tres líneas ni suman ni restan y las erróneas restan 0.1 puntos. Se valora la *precisión* de la respuesta.

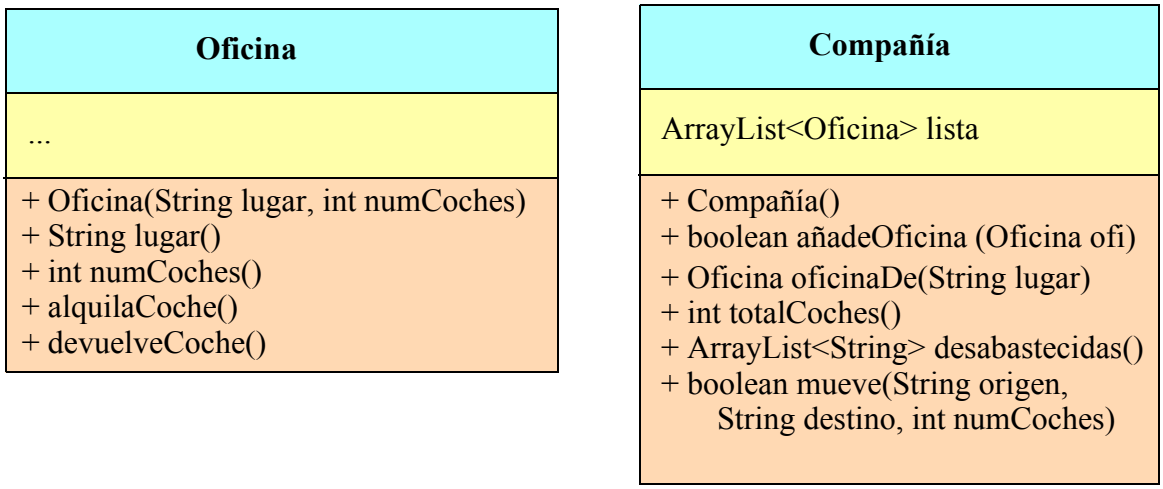
Examen de Introducción al Software
(Ingeniería Informática)

Febrero 2014

Segunda parte (5 puntos, 50% nota del examen)

Se desea realizar una parte del software perteneciente a una compañía de alquiler de coches. Para ello se dispone de la clase Oficina ya realizada, cuyos objetos almacenan el lugar de una oficina de alquiler de coches y el número de coches disponibles.

Se desea crear la clase Compañía. Los diagramas de estas clases se muestran aquí:



Los métodos de la clase Oficina (ya hecha) hacen lo siguiente:

- *Constructor*: construye la oficina a partir del lugar y el número inicial de coches, que se pasan como parámetros
- lugar(): Retorna el lugar donde está la oficina
- numCoches(): Retorna el número actual de coches disponibles en esa oficina
- alquilaCoche(): Resta uno al número de coches disponibles
- devuelveCoche(): Añade uno al número de coches disponibles

La clase Compañía (que se pide) dispondrá de un único atributo que es un ArrayList de objetos de la clase Oficina y que contiene la lista de las oficinas pertenecientes a la compañía de alquiler. Sus métodos deberán hacer lo siguiente:

- *Constructor*: Crea la lista vacía
- oficinaDe(): Busca si en la lista existe una oficina situada en el lugar indicado por el parámetro lugar (utilizar para ello uno de los esquemas de búsqueda vistos en clase). Si se encuentra, retorna esa oficina. En caso contrario, retorna null.
- añadeOficina(): Busca si en la lista existe una oficina situada en el mismo lugar que la oficina ofi (utilizar para ello el método oficinaDe()). Si se encuentra una oficina en el mismo lugar se debe retornar false. En caso contrario se añade la oficina ofi a la lista y se retorna true.
- totalCoches(): Retorna el número total de coches que hay en todas las oficinas de la lista.

- desabastecidas(): Retorna un ArrayList conteniendo los lugares de aquellas oficinas que tienen menos de 3 coches disponibles.
- mueve(): Mueve un número de coches igual al parámetro numCoches desde la oficina situada en el lugar origen, hasta la situada en destino. Retorna true si se han podido mover los coches y false en caso contrario.

Para el método mueve() utilizar el siguiente pseudocódigo:

```
método mueve(String origen, String destino, entero numCoches)
retorna booleano
    // busca la primera oficina y retorna falso si no se encuentra
    Oficina o1=oficinaDe(origen)
    si o1=null entonces
        retorna falso
    fin si
    // busca la segunda oficina y retorna falso si no se encuentra
    Oficina o2=oficinaDe(destino)
    si o2=null entonces
        retorna falso
    fin si
    // mira si en o1 hay suficientes coches
    si número de coches en o1 < numCoches entonces
        // no hay suficientes coches
        retorna falso
    fin si
    // mueve los coches de la primera oficina a la segunda, uno por uno
    para i desde 1 hasta numCoches hacer
        alquila un coche de o1
        devuelve un coche a o2
    fin para
    // el movimiento se ha podido realizar y por ello retornamos verdad
    retorna verdad
fin método
```

Valoración:

1) Encabezamiento de la clase, atributo y constructor: 0.5 puntos

2) añadeOficina: 0.5 puntos

3) Resto de los métodos: 1 punto cada uno