

Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2012

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

- 1) Suponiendo que existe una clase enumerada definida del siguiente modo:
- ```
public enum Orden {RETROCEDER, PARAR, AVANZAR, GIRAR_DERECHA,
 GIRAR_IZQUIERDA}
```

Escribir mediante instrucciones Java el siguiente pseudocódigo de movimiento de un vehículo del tipo oruga, utilizando la instrucción condicional `switch`:

```
si
orden==RETROCEDER ->
 velolzqda=-10.0; veloDcha=-10.0; mover(velolzqda,veloDcha);
orden==PARAR ->
 velolzqda=0.0; veloDcha=0.0; mover(velolzqda,veloDcha);
orden==AVANZAR ->
 velolzqda=10.0; veloDcha=10.0; mover(velolzqda,veloDcha);
orden==GIRAR_DERECHA ->
 velolzqda=10.0; veloDcha=0.0; mover(velolzqda,veloDcha);
orden==GIRAR_IZQUIERDA ->
 velolzqda=0.0; veloDcha=10.0; mover(velolzqda,veloDcha);
fin si
```

Suponer que las variables `orden`, `veloIzqda` y `veloDcha` ya existen, así como el método `mover()`.

- 2) Escribir un método con la siguiente cabecera, que crea y retorna un array de dos dimensiones, de tamaño  $n \times n$ , con todas sus casillas a valor `Double.NaN`

```
public static double[][] creaMatriz(int n)
```

- 3) Escribir el pseudocódigo de un método que calcula y retorna la amplitud de un conjunto de valores reales almacenados en un array que se pasa como parámetro. La amplitud es la diferencia entre el valor máximo de los elementos del array y el valor mínimo.

- 4) Escribir en Java un método recursivo que sirve para dar la vuelta (poner en el orden contrario) a los elementos de un array `a` que se pasa como parámetro, trabajando a partir del elemento número `i`. El método tendrá la siguiente cabecera:

```
public static void invertir(double[] a, int i)
```

Llamaremos `n` a la longitud de `a`. El caso directo ocurre cuando `i` es mayor o igual que  $n/2$  (cociente calculado como división entera). En ese caso no hay que hacer nada. En el caso recursivo hay que intercambiar las casillas `i` y  $n-1-i$  del array `a` y luego invocar al mismo método para los parámetros `(a, i+1)`.

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

- a. En una hoja de cálculo indica qué fórmula hay que escribir para obtener el número de casillas que hay en la columna D con valores numéricos. Se puede suponer que el número total de filas está limitado.

- b. Escribir una fórmula que calcule la suma de las celdas A1, A3 y A7 de una hoja de cálculo, de modo que si se copia esta fórmula en otra casilla siempre se sumen esas tres celdas.
- c. En una hoja de cálculo es posible utilizar un formulario para introducir datos. ¿Se pueden usar también las tablas para ello? En caso afirmativo ¿qué diferencia hay entre las tablas y los formularios a la hora de introducir datos?
- d. Indica la diferencia entre una clave primaria y una clave extranjera en una base de datos.
- e. ¿Qué diferencia hay entre mySQL y SQL?

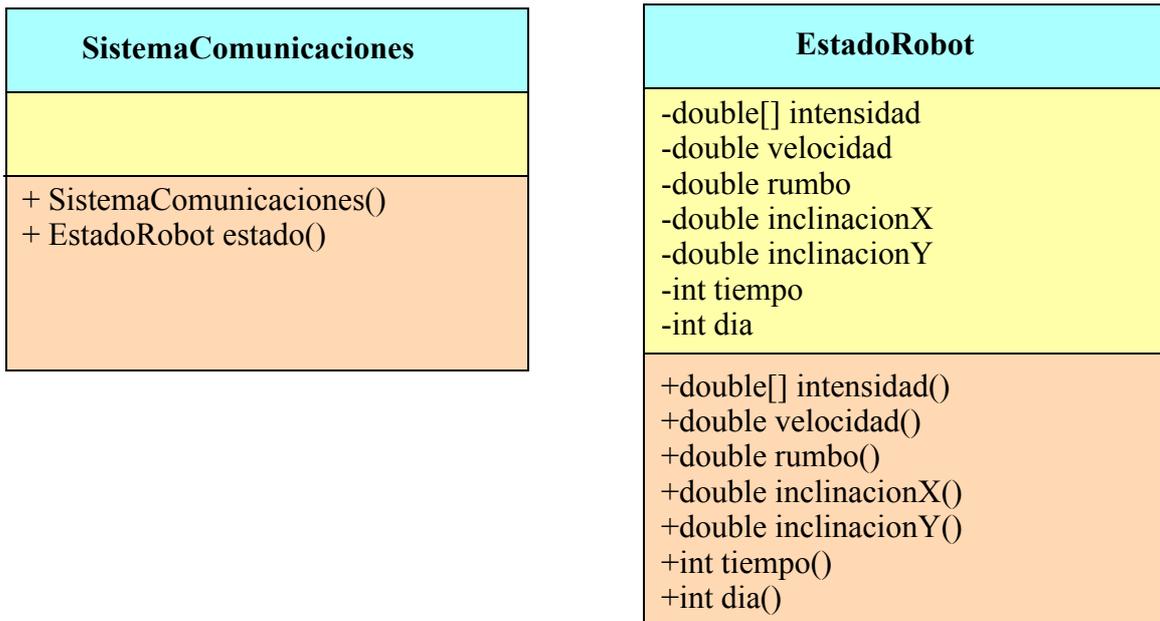
*Nota:* en esta cuestión, las respuestas correctas suman 0.2 puntos y las incorrectas restan 0.1 puntos. Se valora la *precisión* de la respuesta.

## Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2012

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

Se desea realizar una parte del software perteneciente a un sistema de monitorización de un robot móvil que se halla en el planeta Marte. Para ello, se dispone de las siguientes clases, ya realizadas, que funcionan de acuerdo a los siguientes diagramas de clases:



La clase `EstadoRobot` almacena el estado en que se encuentra el robot en un instante determinado. Sus métodos son observadores de los atributos del mismo nombre y son:

- `intensidad()`: retorna un array de 6 números reales, con la intensidad de cada uno de los 6 motores del robot, en amperios
- `velocidad()`: retorna la velocidad lineal del robot en m/s
- `rumbo()`: retorna el rumbo del robot medido en grados respecto al norte geográfico
- `inclinacionX()`: retorna la inclinación del robot en la dirección de avance, en grados
- `inclinacionY()`: retorna la inclinación del robot en la perpendicular a la dirección de avance, en grados
- `tiempo()`: retorna los milisegundos transcurridos desde la medianoche marciana
- `dia()`: retorna el día marciano, medido desde el inicio de las operaciones

La clase `SistemaComunicaciones` es la implementación informática de un sistema de comunicaciones con el robot. Tiene un método llamado `estado()` que retorna un objeto de la clase `EstadoRobot` conteniendo el estado actual del robot. Retorna `null` si momentáneamente no hay comunicación con el robot.

Lo que se pide es construir la clase `RegistroRobot` que almacene en un `ArrayList` los sucesivos estados del robot de un día marciano concreto. La clase tendrá el siguiente diagrama

de clases:

| RegistroRobot                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - ArrayList<EstadoRobot> lista<br>- int día<br>- SistemaComunicaciones sc                                                                                   |
| + RegistroRobot(SistemaComunicaciones sc, int día)<br>+ void recogeEstados(int n)<br>+ double distanciaRecorrida()<br>+ int intensidadIncorrecta(int motor) |

La clase `RegistroRobot` debe almacenar en `lista` los estados del robot obtenidos del sistema de comunicaciones `sc`. Asimismo debe almacenar en `día` el día marciano al que corresponden los estados y en `sc` el sistema de comunicaciones a emplear. Sus métodos son:

- *Constructor*: Almacena el sistema de comunicaciones y el día que se pasan como parámetros en los atributos `sc` y `día`, respectivamente. Además, crea la `lista` vacía.
- `recogeEstados()`: Llama `n` veces al método `estado()` del objeto `sc`. Añade a la `lista` cada estado así obtenido que sea distinto de `null` y cuyo día sea igual al atributo `día`. Para cada estado igual a `null` pone un mensaje de error en pantalla.
- `distanciaRecorrida()`: Calcula y retorna la distancia recorrida por el robot, en metros. Para el cálculo se usa esta fórmula:

$$\sum_{i=1}^{n-1} \left( \frac{v_i + v_{i-1}}{2} \right) \cdot (t_i - t_{i-1})$$

siendo  $n$  el tamaño de la lista,  $v_i$  la velocidad de la casilla  $i$  de la lista en m/s y  $t_i$  el tiempo de la casilla  $i$  de la lista en segundos.

- `intensidadIncorrecta()`: Hace una búsqueda en la lista para buscar estados en los que la intensidad de motor número `motor` sea demasiado alta. Se considera que la intensidad es demasiado alta si supera en valor absoluto los 5 amperios. Si se encuentra una intensidad demasiado alta para ese motor se retorna el número de la casilla de la lista en que se encuentra. Si no se encuentra ninguna, se retorna -1.

Valoración:

- 1) Constructor y atributos: 0.5 puntos
- 2) Resto de los métodos: 1.5 puntos cada uno