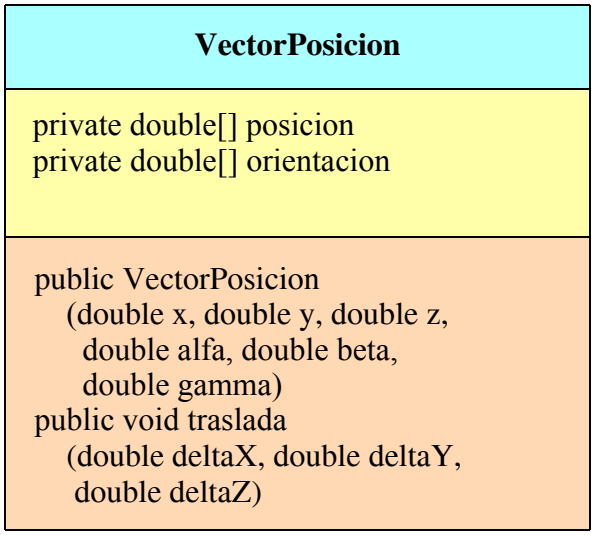
Examen Final de Fundamentos de Computadores y Lenguajes (Licenciado en Física)

Junio 2009

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

1) Escribir una clase que permita almacenar la posición (coordenadas) y orientación (ángulos) de un cuerpo en el espacio tridimensional. Cada uno de estos dos atributos es un array de tres números reales. Escribir un método constructor al que se le pasen 6 números reales como parámetros y los almacene en los atributos (los tres primeros como coordenadas de la posición y los tres últimos como ángulos de la orientación). Escribir también un método que permita trasladar la posición: a este método se le pasan tres valores reales que deben sumarse a cada una de sus respectivas coordenadas. La clase debe responder al diagrama que se muestra en la figura.



2) Escribir un método que responda al siguiente pseudocódigo y permita determinar si un texto que se le pasa como parámetro es simétrico, es decir que se puede leer de la misma manera de izquierda a derecha o de derecha a izquierda (por ejemplo, `"abccba"`).

```
método esSimetrico(String s) retorna booleano
    entero i=0;
    entero j=s.longitud -1
    mientras i<j hacer
        si s.caracter(i)!=s.caracter(j) entonces
            // hemos encontrado caracteres diferentes en posiciones simétricas
            retorna false
        fin de si
        i=i+1;
        j=j-1;
    fin de mientras
    // todos los caracteres en posiciones simétricas son iguales
    retorna true
fin de método
```

3) Escribir un método que escribe en un fichero de texto todos los valores de dos arrays de números reales `x[]` e `y[]` que se le pasan como parámetros. El nombre del fichero es `"datos.txt"`. Los datos se escribirán de manera que en cada línea aparece un valor del array `x`, una coma, y un valor del array `y`. En la línea siguiente los dos valores de `x` e `y`

siguientes, y así sucesivamente. Si los arrays no tienen la misma longitud, lanzar la excepción `DistintaLongitud`, de la clase que se muestra a continuación.

```
public class DistintaLongitud extends Exception {}
```

4) Escribir un método estático al que se le pase un valor enumerado de la clase `Funcion` que se muestra abajo y un número real x, y que retorne el cuadrado de x, la raíz cuadrada de x, el logaritmo neperiano de x, o e^x, respectivamente segun el valor de `func`. El método debe responder a la interfaz que se muestra debajo.

```
public enum Funcion {
    cuadrado, raizCuadrada, logaritmo, exp
}

public static double calcula(Funcion func, double x) {...}
```

5) Indicar el tiempo de ejecución del método de la cuestión 2, usando la notación O(n). Razonar la respuesta.

Examen de Fundamentos de Computadores y Lenguajes
(Licenciado en Física)

Junio 2009

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

Se dispone de una clase ya realizada llamada `Rayo` que contiene los datos relativos a la caída de un rayo detectada por un satélite, y que obedece a la siguiente interfaz:

```
public class Rayo {

    /**
     * Retorna el tipo de rayo, positivo o negativo
     */
    public TipoRayo tipo() {...}

    /**
     * Retorna la coordenada X del rayo, en kilómetros
     */
    public double coordX() {...}

    /**
     * Retorna la coordenada Y del rayo, en kilómetros
     */
    public double coordY() {...}

    /**
     * Retorna la hora a la que cayó el rayo (entre 0 y 23)
     */
    public int hora() {...}

    /**
     * Retorna el minuto en el que cayó el rayo (entre 0 y 59)
     */
    public int minuto() {...}
}
```

El tipo de rayo es un valor enumerado definido así:

```
public enum TipoRayo {positivo, negativo }
```

Se dispone también de una clase ya realizada que contiene una lista de los datos de los rayos caídos en una determinada fecha en territorio nacional. La clase responde a la siguiente interfaz:

```
public class MapaRayos {

    /**
     * Recibe por un sistema de telecomunicación un mapa de rayos y lo
     * retorna. Si no hay comunicación lanza NoHayComunicacion
     */
    public static MapaRayos recibe() throws NoHayComunicacion {...}

    /**
     * Numero de rayos caídos en 24 horas
     */
    public int numRayos() {...}

    /**
     * Retorna el rayo numero i, siendo i un numero entre 0 y numRayos()-1
     * Retorna null si i es menor que cero o mayor que numRayos()-1
     */
    public Rayo rayo(int i) {...}
}
```

Por último se dispone de las siguientes clases que definen excepciones:

```
public class NoHayComunicacion extends Exception {}

public class FormatoIncorrecto extends Exception {}
```

Lo que se pide es implementar la clase `InfoRayos` de acuerdo con la siguiente especificación, y con el único atributo que se menciona:

```
public class InfoRayos {

    private MapaRayos mapa; // objeto con una lista de datos de los rayos

    /**
     * Constructor que obtiene un mapa de rayos con el método recibe
     * de la clase MapaRayos y lo guarda en el atributo mapa; Si se
     * lanza NoHayConexion pone un mensaje de error en pantalla
     */
    public InfoRayos() {...}

    /**
     * Calcula el numero de rayos caídos por hora en el intervalo de
     * tiempo entre horaInicial:minutoInicial y horaFinal:minutoFinal.
     * Lanza formato incorrecto si las horas inicial o final no están
     * entre 0 y 23, o si los minutos iniciales o finales no están
     * entre 0 y 59.
     */
    public int rayosPorHora
        (int horaInicial, int minutoInicial,
         int horaFinal, int minutoFinal) throws FormatoIncorrecto  {...}

    /**
     * Calcula el numero de rayos positivos caídos en un entorno
     * circular del radio indicado, alrededor de las coordenadas
     * x e y indicadas
     */
    public int rayosPositivosCercaPosicion (double x, double y, double radio)
    {...}

    /**
     * Retorna true si hay algún rayo negativo y false si no hay
     * ninguno
     */
    public boolean hayRayosNegativos() {...}
}
```

Para comparar horas transformar cada pareja (*horas*:*minutos*) a su equivalente en minutos, usando la expresión: *horas**60+*minutos*

Para calcular la distancia entre dos puntos de coordenadas (x1,y1) y (x2,y2) usar la fórmula:

$$dist = \sqrt{(x1-x2)^2 + (y1-y2)^2}$$

Nota: se valorarán los métodos a desarrollar de la siguiente forma:

- `rayosPorHora`: 2 puntos
- constructor y resto de los métodos: 1 punto cada uno