Examen de Fundamentos de Computadores y Lenguajes

Examen Final. Junio 2001

Cuestiones (5 cuestiones, 5 puntos en total)

1) Escribir una clase denominada `Caja` que permita almacenar las medidas de una caja (largo, ancho, y alto, del tipo `double`) y que tenga las siguientes operaciones:
   - constructor, al que se le pasan como parámetros las tres medidas, y las almacena
   - operación que retorna el volumen de la caja

2) Se dispone de la clase `CuentaCorriente` que tiene entre otras una operación para averiguar el saldo, y cuya interfaz es:

```
public long saldo() throws NoDisponibleException {...}
```

   Donde `NoDisponibleException` es una excepción definida dentro de la clase `CuentaCorriente`. Suponiendo que se dispone de un objeto de la clase `CuentaCorriente` denominado `miCuenta`, escribir un fragmento de programa que muestre en la pantalla (usando el objeto `System.out`) el saldo de ese objeto, o ponga el mensaje "No Disponible" si se arrojase la excepción `NoDisponibleException`.

3) Contestar brevemente: ¿Puede una clase no tener operaciones? ¿Para qué serviría una clase así?

4) Escribir un fragmento de programa que, mediante un lazo, vaya sumando números enteros consecutivos, empezando por 1, hasta que al sumar el valor *i-ésimo* la suma sea mayor que 100; terminado el lazo, mostrar en pantalla el valor de *i*.

5) Una matriz cuadrada A de números reales con $N>0$ filas y $N>0$ columnas se dice simétrica si para cualesquiera valores de los índices $1 \le i,j \le N$ se verifica la identidad $A[i][j]=A[j][i]$.

   Se puede diseñar fácilmente un algoritmo basado en el esquema de búsqueda para determinar si una matriz dada es simétrica. La idea es la siguiente. Se utilizan dos contadores i, j uno para las filas y otro para las columnas, haciéndose la búsqueda sobre los elementos de la matriz situados por encima de la diagonal principal, de manera que si en un instante el elemento en curso es el de índices i,j, en el instante siguiente el elemento a considerar es el de índices $i=i$, $j=j+1$ si $j \neq N$; y es el de índices $j=i+1$, $i=i+1$ si $j=N$. Los contadores i,j se inicializan en $i=1$, $j=1$. Si $A[i][j] \neq A[j][i]$ la matriz no es simétrica y el algoritmo termina anticipadamente retornando *falso.* Si nunca se detecta la situación anterior, la condición de salida del lazo es $i=N$ y $j=N$, que es lo que caracteriza al último elemento de la diagonal principal. Si se sale del lazo llegando hasta el final, el algoritmo retornará *verdad*, para indicar que la matriz es simétrica.

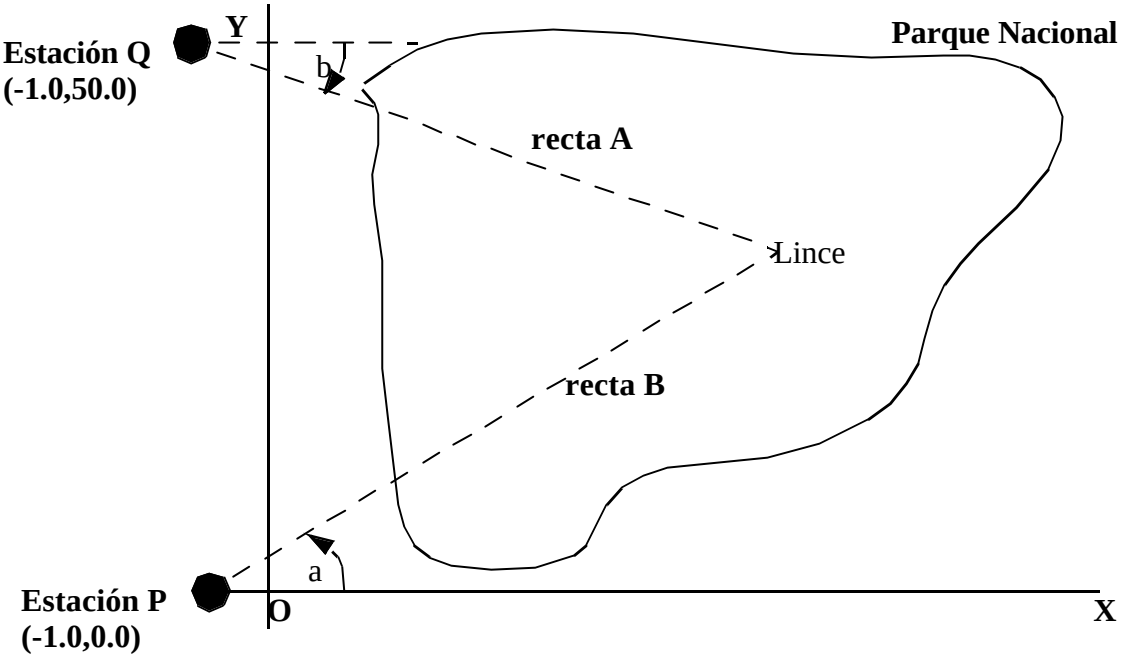
   Se pide especificar y diseñar en pseudocódigo el algoritmo descrito anteriormente. Escribir el orden de magnitud del coste en función de N.

Examen de Fundamentos de Computadores y Lenguajes

Examen final. Junio 2001

Problema (5 puntos)

Se desea construir una aplicación para el seguimiento de una población de linces en un parque nacional. Los linces levan adherido un pequeño repetidor de radio que responde cuando recibe una señal de radio, dirigida a ese repetidor concreto, desde una de las dos estaciones de seguimiento. La estación de seguimiento capta la respuesta y determina la dirección de la que procede. Al tener dos estaciones, a partir de los ángulos α y β es posible determinar por triangulación la posición del lince. La figura inferior muestra un plano del parque y la situación de las dos estaciones de seguimiento, denominadas P y Q:



Las coordenadas (x,y) de cada estación se muestran en el plano, y están en kilómetros. El punto O es el origen de coordenadas. Los ángulos se miden respecto al sistema de coordenadas en grados (0º corresponde al eje X).

El funcionamiento de la estación de seguimiento está integrado en la clase `Antena`, ya realizada, y cuya interfaz se muestra a continuación. Los comentarios de documentación describen cada operación de esta clase.

```
public class Antena {

 /** Excepción lanzada por recibeAngulo si no
  *  hay respuesta del repetidor */
  public class NoHayRespuesta extends Exception {}

  /** Constructor. Requiere las coordenadas de la antena,
   *  en Kilómetros */
  public Antena(double coordx, double coordy) {...}
```

```
   /** Retorna la coordenada X de la antena (en kilómetros) */
   public double coordenadaX() {...}

   /** Retorna la coordenada Y de la antena (en kilómetros) */
   public double coordenadaY() {...}

   /** Emite una señal dirigida al repetidor del animal
    * indicado por el string nombreAnimal */
   public void emite (String nombreAnimal) {...}

   /** Retorna el ángulo (en grados) con el que se ha detectado desde
    *  la antena el repetidor que lleva el animal para el que se
    *  emitió la última señal (con emite()) */
   public double recibeAngulo() throws NoHayRespuesta {...}
}
```

Además, se cuenta con la clase `Mapa`, también realizada ya, que muestra en una ventana en la pantalla del computador información gráfica sobre la posición de los linces, así como mensajes de error. La interfaz de esta clase se muestra a continuación.

```
public class Mapa {

   /** Dibuja en la pantalla el lince indicado por nombreAnimal
    *  en las coordenadas (coordx,coordy) */
   public void dibuja (String nombreAnimal,
                       double coordx, double coordy) {...}

   /** Escribe un mensaje de error que contiene el nombreAnimal y
    *  el mensaje indicados */
   public void muestraError (String nombreAnimal, String mensaje)
   {...}

   /** Activa una alarma visual en la pantalla que contiene
    *  el nombreAnimal y el mensaje indicados */
   public void alarma (String nombreAnimal, String mensaje) {...}
}
```

Lo que se pide es construir una clase que contenga dos métodos estáticos y el programa principal `main`(). Los dos métodos deberán obedecer a la siguiente interfaz:

```
   public static double cruceX(double px, double py, double qx,
          double qy, double alpha, double beta) {...}
   public static double cruceY(double px, double py, double qx,
          double qy, double alpha, double beta) {...}
```

La descripción de estos métodos es:

- `cruceY`: retorna la coordenada Y del punto de cruce de las dos rectas A y B (ver figura), a partir de las coordenadas de las estaciones P (p_x,p_y) y Q (q_x,q_y) y de los ángulos α y β según la ecuación de la intersección de dos rectas:

$$y = \frac{p_y a_x b_y - p_x a_y b_y - q_y b_x a_y + q_x b_y a_y}{a_x b_y - b_x a_y}$$

donde a_x=cos(α), a_y=sen(α), b_x=cos(β), b_y=sen(β).

- `cruceX`: retorna la coordenada X del punto de cruce mencionado, según la ecuación:

$$x = \frac{(y - p_y)a_x}{a_y} + p_x$$

donde *y* es el valor retornado por `cruceY`.

El programa principal deberá hacer lo siguiente:

a) Declaraciones de al menos los siguientes datos:
   - Array con los nombres de los linces, que serán: Paco, Sara, Payaso, Gris.
   - Array de números enteros para almacenar el número de veces consecutivas que el repetidor de cada lince no devuelve respuesta
   - Dos objetos de la clase `Antena`, uno para cada una de las estaciones de seguimiento. Inicializarlos con sus coordenadas respectivas (ver figura)
   - Un objeto de la clase `Mapa`, para presentar información en la pantalla

b) Repetir indefinidamente un lazo que se ejecuta para cada uno de los linces, y que hace lo siguiente:

- Emite una señal por la estación `P` y luego recibe el ángulo detectado, mediante `recibeAngulo`()
- Lo mismo para la estación `Q`
- Si ha ido todo bien:
  - Calcula el punto de cruce (es decir, la posición del lince) mediante `cruceX`() y `cruceY`()
  - Dibuja la posición del lince en el objeto de la clase `Mapa`, mediante la operación `dibuja`()
  - Pone el número de fallos de ese lince a cero
- Pero si se ha arrojado `NoHayRespuesta`, entonces:
  - Se incrementa el número consecutivo de fallos de ese lince
  - Si este número es superior a cinco, se activa una alarma con la operación `alarma`()
  - Si no, se pone en pantalla un mensaje de error con `muestraError`()

Nota: cada parte del problema se valorará en función de su complejidad.