

Examen de Septiembre de 2006
Programación I (Ingeniería Informática)
Informática (Matemáticas)

Parte I.: Cuestiones: Cada una se valorará sobre 0.6 puntos.

1.- Sintaxis de expresiones. Utilizando la sintaxis del lenguaje Java indicar cuáles de las siguientes asignaciones están bien construidas siendo x, y variables de tipo **float**, i, j de tipo **int**, a, b y c de tipo **boolean**. Justificar cada respuesta.

- a) $a = (x + y = 8.0)$
- b) $b = (i = 7)$
- c) $i += j;$
- d) $a = (b \& c) \parallel (x = 1.0)$

2.- Longitud de la palabra más larga.

En una secuencia de caracteres, el carácter # se emplea como separador de bloques de caracteres (palabras). Especifíquese y diseñese un algoritmo que procese una secuencia de caracteres y obtenga la longitud de la palabra más larga de la misma (los caracteres separadores # no se consideran parte de las palabras).

Por ejemplo, si procesa la secuencia:

asdczs#fhjdfh#e#ewfertxdke#oqpwa#lsk#iq

el algoritmo devolverá el valor 10 ya que ésta es la longitud de la palabra más larga "ewfertxdke" que se puede encontrar en la secuencia anterior.

3.- Recorridos y búsquedas en tablas.

Utilizando el esquema de búsqueda sobre tablas diseñese un algoritmo para la siguiente especificación.

t: **tabla**[1..N] de entero
r: 0..N
{Pre.- $t = T$ }
{Post.- $(\forall k : r < k \leq N : t[k] \geq 0) \wedge (t[r] < 0 \vee r = 0) \wedge t = T$ }

4.- Recursividad.

La potencia n-ésima, siendo n un número natural, de un número real $X > 0$ puede definirse recursivamente del siguiente modo:

$$X^0 = 1,$$
$$X^n = (X^{n \text{ div } 2})^2, \text{ si } n \text{ par}$$
$$X^n = X^{n-1} * X, \text{ si } n \text{ impar}$$

Se pide especificar y diseñar una función recursiva que tomando n y X como parámetros retorne el valor de la potencia X^n implementando la definición anterior.

5.- Alternativas.

Se considera el siguiente fragmento de código,

```
x, y: entero;  
{Pre.- ¿?}  
si  
    x par entonces x:=x div 2  
    x impar entonces x:=x-1  
fsi  
{Post.-  $2*x+y=Z$  }
```

Se pide indicar cuáles de las siguientes precondiciones lo hacen correcto justificando la respuesta.

- a) x par && $x+y=Z$
- b) x impar && $x+y=Z$
- c) x impar && $2x+y=Z+2$

Examen de Programación (Ingeniería Informática) y de Informática (Licenciado en Matemáticas)

Septiembre 2006

Parte II. Problema (3 puntos=50% nota del examen)

Disponemos de una clase ya realizada que permite almacenar los datos asociados a un mensaje de un sistema de alarma. Estos datos son la hora y minuto en el que se produjo la alarma (enteros), el tipo de alarma (entero), y un texto explicativo. La interfaz de esta clase es:

```
public class Mensaje {
    /**
     * Construye el mensaje a partir del tipo y del texto
     */
    public Mensaje(int tipo, String texto) {}

    /** Retorna la hora a la que se genero el mensaje */
    public int hora() {...}

    /** Retorna el minuto a la que se genero el mensaje */
    public int minuto() {...}

    /** Retorna el tipo de mensaje */
    public int tipo() {...}

    /** Retorna el texto del mensaje */
    public String texto() {...}
}
```

Por otro lado, se dispone de otra clase que también está realizada y que permite almacenar una secuencia de mensajes. Responde a la interfaz de la secuencia vista en clase, tomando en este caso particular la forma siguiente:

```
public class ListadoMensajes {
    /**
     * Constructor que crea la secuencia vacia
     */
    public ListadoMensajes() {...}

    /**
     * Anadir el mensaje m al final de la secuencia
     */
    public void anade(Mensaje m) {...}

    /**
     * Reiniciar la secuencia para poder recorrerla
     */
    public void reinicia() {...}

    /**
     * obtener el mensaje actual en la secuencia
     */
    public Mensaje mensajeActual() {...}

    /**
     * Avanza al siguiente elemento en la secuencia
     */
    public void avanza() {...}

    /**
     * Indica si estamos en el final de la secuencia
     */
    public boolean finalSecuencia() {...}
}
```

Lo que se pide es realizar parte de la clase SistemaAlarma, que representa el sistema de alarma completo, y que obedece a la siguiente interfaz:

```
public class SistemaAlarma
{
    // atributos privados
    private ListadoMensajes lista;

    /**
     * Constructor que crea la lista de mensajes vacia
     */
    public SistemaAlarma(String nombre) {
        lista=new ListadoMensajes();
    }

    /**
     * Procesa los sensores de alarma e inserta mensajes en la lista
     */
    public void procesaSensores() {...}

    /**
     * Retorna el numero de mensajes del tipo tipoMensaje
     * que existen en la lista
     */
    public int numMensajesTipo(int tipoMensaje) {...}

    /**
     * Busca y retorna el texto del primer mensaje anterior
     * a la hora y minutos indicados
     */
    public String buscaMensajeAnteriorA(int hora, int minuto) {...}

    /**
     * Crea una nueva secuencia de mensajes del tipo ListadoMensajes,
     * anade a ella los mensajes anteriores a la hora y minutos
     * indicados y cuyo tipo coincide con tipoMensaje, y retorna la
     * nueva secuencia
     */
    public ListadoMensajes mensajesAnterioresDeTipo
        (int hora, int minuto, int tipoMensaje)
    {...}
}
```

Puede observarse que existe un atributo privado que es una secuencia de mensajes de la clase ListadoMensajes. Los comentarios de documentación explican lo que debe hacer cada método. De todos los métodos, sólo se piden los tres siguientes:

- numMensajesTipo: usar en su implementación el esquema de recorrido de una secuencia
- buscaMensajeAnteriorA: usar el esquema de búsqueda en secuencia
- mensajesAnterioresDeTipo: usar el esquema de recorrido sobre la secuencia del objeto (atributo lista), insertando en la nueva secuencia cada mensaje que cumpla la condición de hora, minuto y tipo indicada

Nota: para comparar horas y minutos cuando hay que determinar si un mensaje es anterior a otro usar la relación: $minutosTotales=horas*60+minutos$