

**PRUEBA 2 PROGRAMACIÓN
Mayo 2007
INGENIERÍA INFORMÁTICA**

UNIVERSIDAD CARLOS III DE MADRID

LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:

- Rellene todas las hojas a bolígrafo, tanto los datos personales como las respuestas. No use bolígrafo rojo.
- No olvide rellenar el NIA y el grupo real al que pertenece.
- El tiempo máximo de realización es de 50 minutos.
- El único material permitido sobre la mesa es la hoja de test y un bolígrafo

NO PASE DE ESTA HOJA, hasta que se le indique

Apellidos	Nombre	
Firma	NIA	Grupo

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Indicar si la siguiente afirmación es cierta, y explicar brevemente por qué.

"En Java se permite la herencia múltiple, es decir, que una clase A herede a la vez de B y C (class A extends B,C)"

Pregunta 2 (1 Punto).- Indicar si la siguiente afirmación es cierta, y explicar brevemente por qué.

"Si queremos ordenar una lista, el mejor método será siempre el método rápido, ya que es el que menos tiempo tarda y el que menos recursos consume."

Pregunta 3 (1 Punto).- Indicar si la siguiente afirmación es cierta, y explicar brevemente por qué.

Dado un método recursivo, las sucesivas llamadas recursivas deben realizarse cada vez con parámetros de inferior valor que los anteriores.

Pregunta 4 (1 Punto).- Dado el siguiente método:

```
public float metodo1 (float [] lista){  
    float [] lista2;  
    if (lista.length>1) {  
        lista2 = new float [lista.length-1];  
        System.arraycopy(lista,1,lista2,0,lista2.length);  
        return lista[0] + metodo1(lista2);  
    }  
    else return  lista[0];  
}
```

Explicar cuál sería el resultado de la siguiente invocación:

```
float miLista [] = {1.0F,3.0F,3.2F};  
System.out.println(metodo1(miLista));
```

Respuesta:

Pregunta 5 (1 Punto).- Dada la siguiente lista: (12, 8, 4, 5, 9, 6)

Explicar paso a paso cómo se ordenaría dicha lista mediante el método de inserción directa.

Respuesta:

PARTE 2: PROBLEMAS

Problema 1 (2,5 Puntos).- Dado el siguiente código java:

```
public class Tallas {  
    private int talla;  
    private final int [] listaTallas =  
        {32,34,36,38,40,42,44,46,48,50,52,54,56,58,60};  
    //Comprueba que la talla está entre las anteriores  
    public boolean tallaValida (int talla){  
        boolean resultado = false;  
        for (int i=0; i<listaTallas.length;i++){  
            if (talla==listaTallas[i]) resultado = true;  
        }  
        return resultado;  
    }  
}
```

- Crear dos constructores para la clase Tallas:
 - Uno para dar valor al atributo talla. Deberá comprobar que el dato recibido es una talla válida usando el método tallaValida. Si no lo es creará un objeto de la talla "32".
 - Otro por defecto que cree un objeto de la talla "38" utilizando el constructor anterior.
- Escribir el método public int comparar (Tallas otraTalla) que devuelve 1 si la talla del objeto que se le pasa como parámetro es mayor que la de nuestro objeto, -1 si es menor, y 0 si son iguales.
- Modificar el código del algoritmo de la burbuja para que ordene un array de objetos Tallas en lugar de un array de enteros.

o **Nota:** el código del algoritmo de la burbuja es:

```
public static void burbuja (int [] lista, int ultimo){  
    int aux = 0;  
    boolean cambio = true;  
    for (int i=1; i<=ultimo && cambio; i++){  
        cambio = false;  
        for (int j=0; j<=ultimo-i; j++){  
            if (lista[j]>lista[j+1]){  
                cambio = true;  
                aux = lista [j+1];  
                lista [j+1] = lista [j];  
                lista [j] = aux;  
            }  
        }  
    }  
}
```

Problema 2 (2,5 Puntos).- Dado el siguiente código java:

```
public class Ropa {
    private String color;
    private String fabricante;
    private float precio;
    private String tejido;
    protected static int identificador;

    public Ropa (String c, String f, float p, String t){
        color = c;
        fabricante = f;
        precio = p;
        tejido = t;
        identificador ++;
    }
    public Ropa (){
        this("sin color", "sin fabricante", 0, "sin tejido");
    }
    public void imprimir () {
        System.out.println("Color: "+color);
        System.out.println("Fabricante: "+fabricante);
        System.out.println("Precio: "+precio);
        System.out.println("Tejido: "+tejido);
        System.out.println("Identificador: "+identificador);
    }
}
```

- a) Crear la clase Camiseta, que hereda de Ropa y que tiene los siguientes atributos privados:
 - a. mangas de tipo String para indicar si son cortas o largas.
 - b. dibujo de tipo boolean que indique si tiene dibujo o no.
- b) Crear un constructor que reciba parámetros para dar valor a todos los atributos de la clase Camiseta, incluidos los heredados. Deberá usar el constructor de la clase Ropa.
- c) Crear un constructor sin parámetros similar al de la clase Ropa.
- d) Sobrescribir el método `imprimir`, para que imprima todos los parámetros de la clase Camiseta. Utilizar si es posible el heredado.