

Programación Orientada a Objetos

Examen 24 de junio 2004

3º Ingeniería Informática

Observación: Las preguntas se deberán entregar en dos bloques: primer bloque para preguntas 1 a 3 y segundo bloque para preguntas 4 a 6.

Bloque 1

- 3 pt. 1. Explica brevemente si las siguientes afirmaciones son ciertas o falsas, indicando por qué.
- a) Un objeto es un tipo abstracto de datos.
 - b) La ocultación de variables y métodos permite establecer la separación entre interfaz e implementación de las clases.
 - c) Las clases pueden ser públicas, privadas o del package.
 - d) Algunas clases no pertenecen a ningún package.
 - e) La clase java.net.Socket no es public.
 - f) Es obligatorio definir un constructor para cada clase.

- 3 pt. 2. ¿Cuál es la salida del siguiente programa?

```
class Punto3D {
    double x, y, z;
    Punto3D (double xx, double yy, double zz) { x = xx; y = yy; z = zz; }
}

class Esfera {
    Punto3D centro;
    double radio;
    Esfera (Punto3D p, double r) { centro = p; radio = r; }
    public boolean equals (Object obj) {
        if (!(obj instanceof Esfera)) return false;
        Esfera e = (Esfera) obj;
        return centro.equals (e.centro) && radio == e.radio;
    }
}

class Plano3D {
    double a, b, c, d;
    Plano3D (double aa, double bb, double cc, double dd) { a = aa; b = bb; c = cc; d = dd; }
    public boolean equals (Plano3D p) {
        return a * p.b == b * p.a && a * p.c == c * p.a && a * p.d == d * p.a;
    }
}

class Main {
    public static void main (String a[]) {
        Punto3D x = new Punto3D (0, 0, 0);
        Esfera e = new Esfera (x, 3);
        Esfera f = new Esfera (x, 3);
        Esfera g = new Esfera (new Punto3D (0, 0, 0), 3);
        Plano3D p = new Plano3D (2, 0, -2, 4);
        Plano3D q = new Plano3D (3, 0, -3, 6);
        System.out.println (e.equals (f));
        System.out.println (e.equals (g));
        System.out.println (p.equals (q));

        java.util.ArrayList lista = new java.util.ArrayList ();
        lista.add (f);
        lista.add (g);
        lista.add (p);
        lista.add (q);
        System.out.println (lista.indexOf (e)); // indexOf devuelve la posición de la
        System.out.println (lista.lastIndexOf (e)); // primera aparición del objeto en la
        System.out.println (lista.indexOf (p)); // lista (-1 si no aparece). lastIndexOf
        System.out.println (lista.lastIndexOf (p)); // devuelve la posición de la última
                                                    // aparición del objeto (-1 si no aparece).
    }
}
```