

# Programación II - 2012/2013 - Universitat Jaume I

## Evaluación continua - Módulo 2 - Teoría

29 de abril de 2013

La duración máxima de esta prueba es de 25 minutos. No puedes consultar libros ni apuntes.

APELLIDOS: ..... Marca tu grupo de teoría  
NOMBRE: .....  TE1  TE2  TE3

### Ejercicio 1 (1,5 puntos)

Considera las clases Punto y PruebaPunto que aparecen a continuación:

```
public class Punto {  
  
    // Atributos  
  
    private int x;  
    private int y;  
  
    // Constructores  
  
    public Punto(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    public Punto(Punto p) {  
        this(p.x, p.y);  
    }  
  
    // Métodos  
  
    public void mover(int incX, int incY) {  
        x += incX;  
        y += incY;  
    }  
  
    public String toString() {  
        return "(" + x + ", " + y + " )";  
    }  
}
```

```
public class PruebaPunto {  
  
    private static void mover1(Punto pto) {  
        pto.mover(1, 1);  
    }  
  
    private static void mover2(Punto pto) {  
        Punto aux = new Punto(pto);  
        aux.mover(2, 2);  
        pto = aux;  
    }  
  
    private static void mover3(Punto pto) {  
        Punto aux = pto;  
        aux.mover(3, 3);  
        pto = aux;  
    }  
  
    public static void main(String[] args) {  
        Punto p = new Punto(10, 10);  
        Punto q = new Punto(20, 20);  
        Punto r = new Punto(30, 30);  
  
        mover1(p);  
        mover2(q);  
        mover3(r);  
  
        System.out.println("p: " + p);  
        System.out.println("q: " + q);  
        System.out.println("r: " + r);  
    }  
}
```

Indica qué se escribiría en la salida estándar al ejecutar el programa PruebaPunto.java:

p: (            ,            )

q: (            ,            )

r: (            ,            )

## Ejercicio 2 (3,5 puntos)

Considera la clase Punto del ejercicio anterior y la clase `Restaurante` que aparece a continuación:

```
public class Restaurante {  
  
    private String nombre;  
    private Punto posición;  
  
    public Restaurante(String s, Punto p) {  
        nombre = s;  
        posición = new Punto(p);  
    }  
  
    public Punto getPosición() {  
        return new Punto(posición);  
    }  
  
    public String toString() {  
        return nombre + " " + posición;  
    }  
}
```

Indica qué se escribiría en la salida estándar:

a) al ejecutar el siguiente fragmento de código:

```
Punto pto1 = new Punto(10, 10);  
Punto pto2 = new Punto(20, 20);  
Restaurante[] v = { new Restaurante("Don Jamón", pto1),  
                    new Restaurante("Mr. Cheese", pto1),  
                    new Restaurante("Casa Pepe", pto2) };  
  
pto1.mover(5, 5);  
pto2.mover(5, 5);  
for (int i = 0; i < v.length ; i++)  
    v[i].getPosición().mover(i + 1, i + 1);  
  
System.out.println("pto1 es " + pto1 + " y pto2 es " + pto2);  
System.out.println(Arrays.toString(v));
```

pto1 es (      ,      ) y pto2 es (      ,      )

[Don Jamón(      ,      ) , Mr. Cheese(      ,      ) , Casa Pepe(      ,      )]

b) al ejecutar el fragmento de código del apartado (a) suponiendo que el método `getPosición` de la clase `Restaurante` se hubiera modificado del siguiente modo:

```
public Punto getPosición() {  
    return posición;  
}
```

pto1 es (      ,      ) y pto2 es (      ,      )

[Don Jamón(      ,      ) , Mr. Cheese(      ,      ) , Casa Pepe(      ,      )]

### Ejercicio 3 (2 puntos)

A continuación se describen dos excepciones y las situaciones en las que las lanza la máquina virtual Java:

- **ArithmeticException:** Indica un error aritmético, por ejemplo, una división por cero entre valores enteros.
- **IndexOutOfBoundsException:** Indica que se ha utilizado un índice fuera de rango, por ejemplo, en el acceso a los elementos de un vector.

¿Qué escribiría en la salida estándar la ejecución del programa siguiente?

```
public class TrazaExcepciones {  
  
    private static int enigma(int n) {  
        int[] vector = { 15, 17, -13 };  
        int resultado = 7;  
  
        try {  
            int índice = 10 / n;  
            resultado = vector[índice];  
            if (resultado < 0)  
                return 0;  
            System.out.println("Paso 1");  
        } catch (IndexOutOfBoundsException e) {  
            System.out.println("Paso 2");  
            resultado = -1;  
        } catch (ArithmeticException e) {  
            System.out.println("Paso 3");  
            resultado = -2;  
        }  
  
        System.out.println("Paso 4");  
        return resultado;  
    }  
  
    public static void main(String[] args) {  
        int[] datos = { 10, 5, 1, 0 };  
  
        for (int i = 0; i < datos.length; i++)  
            System.out.println("Resultado: " + enigma(datos[i]));  
    }  
}
```

Marca los mensajes que se escribirían e indica el valor del resultado obtenido en cada caso:

| <b>enigma(10)</b>               | <b>enigma(5)</b>                | <b>enigma(1)</b>                | <b>enigma(0)</b>                |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Paso 1 <input type="checkbox"/> |
| Paso 2 <input type="checkbox"/> |
| Paso 3 <input type="checkbox"/> |
| Paso 4 <input type="checkbox"/> |
| Resultado: <input type="text"/> | Resultado: <input type="text"/> | Resultado: <input type="text"/> | Resultado: <input type="text"/> |

## Ejercicio 4 (3 puntos)

Considera la siguiente definición de una clase `Canción`:

```
public class Canción {  
  
    private String título;  
    private int código;  
    private static int contador;  
  
    public Canción(String s) {  
        título = s;  
        contador++;  
        código = contador;  
    }  
  
    public String toString() {  
        return título + "(" + código + ")";  
    }  
}
```

y una clase `PruebaCanción` que hace uso de `Canción`:

```
import java.util.Arrays;  
  
public class PruebaCanción {  
    public static void main(String[] args) {  
        String[] títulos = {"Yesterday", "Help", "Girl"};  
        Canción[] misCancionesFavoritas = new Canción[títulos.length];  
  
        for (int i = 0; i < títulos.length; i++)  
            misCancionesFavoritas[i] = new Canción(títulos[i]);  
  
        System.out.println(Arrays.toString(misCancionesFavoritas));  
    }  
}
```

Indica qué se escribiría en la salida estándar:

a) al ejecutar el programa anterior:

.....

b) al ejecutar el programa anterior suponiendo que los atributos de la clase `Canción` se hubieran declarado del siguiente modo:

```
private static String título;  
private int código;  
private static int contador;
```

.....

c) al ejecutar el programa anterior suponiendo que los atributos de la clase `Canción` se hubieran declarado del siguiente modo:

```
private String título;  
private int código;  
private int contador;
```

.....