

PROGRAMACIÓN II

(Duración del examen: 2,5 horas)

NOMBRE:

Cuestión 1. (0.5 puntos)

Considere el código mostrado a continuación.

```
public class Prueba{
    public static void main (String[] args){
        Contador micontador = new Contador();
        int num = 20;
        for (int i=0;i<10;i++) incrementa (micontador, num);
        System.out.println("Contador: "+micontador.cuenta);
        System.out.println("Número: "+num);
    }

    public static void incrementa (Contador c, int itera){
        c.cuenta++;
        itera++;
    }
}

public class Contador{
    public int cuenta = 500;

    public Contador (int j){
        cuenta = j;
    }

    public Contador (){
        cuenta= 0;
    }
}
```

¿Cuál es el resultado de ejecutar dicho código?

Respuesta:

Cuestión 2. (0.75 puntos)

Considere el código mostrado a continuación:

```
package p1;
public class C1 {
    protected int entero;

    public C1 (int i){
        this.entero = i;
    }
}

package p1;
public class C2{
    public static void main (String[] args){
        C1 obj = new C1(3);
        System.out.println(2*obj.entero);
    }
}
```

(0.25 puntos) ¿Es correcto este código Java? En caso afirmativo, indique el resultado de la ejecución. En caso contrario, explique el/los error/es identificado/s.

Respuesta:

(0.25 puntos) Modifique el código Java proporcionado de forma que el programa muestre por pantalla el resultado de duplicar el valor del atributo “entero”, suponiendo que la clase C1 se ubica en el paquete p1 y la clase C2 se mueve al paquete p2. **NOTA: No se puede cambiar ninguna de las visibilidades definidas en el código inicial.**

Respuesta:

(0.25 puntos) ¿Es correcta esta nueva versión del código inicial?

```
package p1;
public class C1 {
    protected int entero;

    public C1 (int i){
        this.entero = i;
    }
}

package p1;
public class C2{
    public static void main (String[] args){
        C1 obj = new C1();
        System.out.println(2*obj.entero);
    }
}
```

Respuesta:

Cuestión 3. (0.75 puntos)

Considere el código Java indicado a continuación:

```
public class Clase1{
    private int i;

    public void metodo1 (Clase2 c2){
        metodo2 (c2);
        System.out.println("Mensaje del método 1");
    }

    public void metodo2 (Clase2 c2){
        i = c2.entero;
        System.out.println("Mensaje del método 2");
    }
}

public class Clase2{
    public int entero;
}

public class Test {
    public static void main (String[] args){
        Clase1 obj1 = new Clase1();
        Clase2 obj2 = new Clase2();
        obj2 = null;
        obj1.metodo1(obj2);
        System.out.println("Mensaje del método main");
    }
}
```

(0.25 puntos) ¿Cuál es el resultado de ejecutar el código anterior?

Respuesta:

(0.5 puntos) Añada las instrucciones necesarias en el código anterior para que la salida del programa sea únicamente "*Mensaje del método main*". **NOTA: No se puede eliminar ninguna de las instrucciones indicadas en el código inicial ni usar sentencias finally.**

Cuestión 4. (0.5 puntos)

Indique si las siguientes afirmaciones sobre características de Java son verdaderas o falsas.

NOTA: Fallar alguna de las afirmaciones supone invalidar la cuestión entera.

1. Las interfaces y las clases abstractas no tienen métodos constructores. _____
2. Una interfaz nunca puede incluir métodos abstractos. _____
3. Una clase abstracta nunca puede implementar una interfaz. _____

4. Todos los métodos de una clase abstracta deben ser abstractos; si hubiese algún método concreto, la clase se convertiría en concreta. _____
5. Una clase concreta no puede extender a una clase abstracta. _____

Problema 1. (2.5 puntos)

Dadas las clases **Persona**, **Profesor**, **Alumno** y **Prueba** mostradas a continuación, se pide:

```
public class Persona {  
    private String nombre;  
    private int edad;  
  
    ....  
}  
  
public class Profesor extends Persona {  
    private String horario_tutorias;  
    private int carga_docente;  
  
    ....  
}  
  
public class Alumno extends Persona {  
    private double nota_media;  
    private int numero_asignaturas_matriculadas;  
  
    ....  
}  
  
public class Prueba {  
    public static void main (String[] args){  
        Persona p1 = new Profesor ("Pablo López", 40, "Vi, 10 a 13h.", 12);  
        Persona p2 = new Alumno ("María Pérez", 20, 8.5, 11);  
        mostrarInfo (p1);  
        mostrarInfo (p2);  
    }  
  
    ....  
}
```

1. **(0.75 puntos)** Añadir el código necesario en las clases **Profesor**, **Alumno** y **Persona** de forma que sea posible crear los objetos p1 y p2 indicados en la clase **Prueba**.

Respuesta: Código de la clase Profesor (0.25 puntos)

Respuesta: Código de la clase Alumno (0.25 puntos)

Respuesta: Código de la clase Persona (0.25 puntos)

2. **(1 punto)** Implementar el método **mostrarInfo** perteneciente a la clase **Prueba**, sabiendo que debe presentar por pantalla el nombre y la carga docente para los profesores, y el DNI y la nota media en el caso de los alumnos.

Respuesta:

3. **(0.75 puntos)** Considere ahora la clase **OtraPrueba** cuyo código se muestra a continuación.

```
public class OtraPrueba {  
    public static void main (String[] args){  
        Profesor p3 = new Profesor ("Pablo López", 40, "Vi, 10 a 13h.", 12);  
        System.out.println (p3);  
    }  
}
```

Añadir el código necesario en las clases indicadas al inicio para que el resultado de ejecutar las sentencias Java de arriba sea:

Nombre: Pablo López

Edad: 40 años

Horario de tutorías: Vi, 10 a 13h.

Carga docente: 12 horas.

Respuesta:



