

<b>PRUEBA 2 PROGRAMACIÓN</b> <b>Diciembre 2008</b> <b>GRADO SISTEMAS COMUNICACIÓN</b> <b>Leganés</b>		 <b>Universidad Carlos III de Madrid</b>	
<b>Apellidos</b>		<b>Nombre</b>	
<b>Firma</b>		<b>NIA</b>	<b>Grupo</b>

**LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:**

- Rellene todas las hojas a bolígrafo, tanto los datos personales como las respuestas
- No utilice lápiz ni bolígrafo rojo
- No olvide rellenar el NIA y el grupo real al que pertenece
- El tiempo máximo de realización es de 1 hora
- El único material permitido sobre la mesa es la hoja de test y un bolígrafo
- Utilice exclusivamente esta hoja de test para las respuestas, use las caras posteriores para contestar si lo necesita. No se recogerá ninguna otra hoja adicional.

### **PARTE 1: CUESTIONES**

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

*“Aunque no creemos ningún constructor en una clase, podremos crear objetos de esa clase porque Java define un constructor por defecto”*

Verdadero, Java define un constructor por defecto sin parámetros que desaparece si nosotros definimos uno en esa clase.

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

*“Para crear una lista dinámica de objetos, basta con crear un objeto de las clases que Java nos proporciona, como java.util.Vector, o java.util.Hashtable o java.util.Stack”*

Falso. No basta sólo con crear un objeto de esas clases, hay que importar la clase que vayamos a utilizar.

```
import java.util.Vector;
```

```
Vector miVector = new Vector();
```

---

**Pregunta 3 (1 Punto).**- Dada la clase Pregunta3 indicar cuál es el resultado de ejecutar el método main de la clase UsoPregunta3, y **explicar** brevemente por qué.

```
public class Pregunta3 {
    public boolean at1;

    public void mayor (boolean at1, int a2){
        if (at1) {
            at1= a2/2 > 0;
        }
    }
}

public class UsoPregunta3 {
    public static void main (String [] args){
        Pregunta3 a = new Pregunta3();
        a.at1 = false;
        a.mayor(true,3);
        System.out.println(a.at1);
    }
}
```

Resultado: False

Pintamos por pantalla el valor del atributo del objeto a, que es de Clase Pregunta3. El método mayor, no modifica el valor del atributo, si no el del parámetro at1

**Pregunta 4 (1 Punto).**- Encontrar y **explicar** los 3 errores de compilación que aparecen en el siguiente código Java. ¿Cómo los resolvería?

```
public class Pregunta4 {
    int a;
    private boolean b;
    protected char c;

    public Pregunta4 (int a1, char c1){
        a=a1;
        b=true;
        c=c1;
    }
    public Pregunta4 (int a1, char c1, boolean b1){
        b=b1;
        this(a1,c1);
    }

    public void Pregunta4 (int a, boolean b, char c){
        a=a;
        this.b=b;
        this.c=c;
    }
    public Pregunta4 (int a2, char c2){
        a=a2;
        b=false;
        c=c2;
    }
    public Pregunta4(){
        this(1, true);
    }
}
```

ERROR 1: Hay dos constructores con el mismo tipo y número de parámetros:

```
public Pregunta4 (int a1, char c1){
    public Pregunta4 (int a2, char c2){
```

Se solucionaría eliminando uno de ellos.

ERROR 2: No existe ningún constructor que reciba un int y un boolean, por lo que no puedo hacer llamada a this(1, true)

```
public Pregunta4(){
    this(1, true);
}
```

Se solucionaría utilizando un constructor que exista.

ERROR 3: No puedo hacer this(a1, c1) después de b=b1. Tiene que ser la primera línea.

```
public Pregunta4 (int a1, char c1, boolean b1){
    this(a1,c1);
    b=b1;
}
```

Atención: a=a1 no es un error, es un warning!

**Pregunta 5 (1 Punto).**- Crear un método que reciba un array de enteros de **cualquier** longitud y devuelva otro array en el que los elementos estén en orden inverso y sean el doble que sus originales. Ejemplo: recibe {'1', '2', '3', '4'} y devuelve {'8','6','4','2'}

```
public int[] inverso(int[] a){
    int[] aux = new int[a.length];
    for (int i=0; i<a.length;i++){
        aux[i] = 2*a[a.length - i - 1];
        System.out.println(aux[i]);
    }
    return aux;
}
```

**Pregunta 6 (1 Punto).**- Indicar y **explicar** cuál es el resultado por pantalla del siguiente programa:

```
public class Pregunta6 {  
    public static void main(String[] args) {  
        int a = 3;  
        while (a <= 6){  
            if (a%3==0){  
                for (int i=3; i>1;i--){  
                    System.out.println(a*i);  
                }  
            }  
            a++;  
        }  
    }  
}
```

Resultado:

9  
6  
18  
12

**Pregunta 7 (1 Punto).**- Dados los siguientes pares de métodos sobrescritos (de los que sólo se muestran las cabeceras), **explicar** cuáles pueden estar dentro de una misma clase y cuáles no.

- |  |  |
|--|--|
| a) <code>int metodo1 (int a, int b){...}</code>      | <code>void metodo1 (int d, int c){...}</code>    |
| b) <code>void metodo1 (int a) {...}</code>           | <code>long metodo1 (int a) {...}</code>          |
| c) <code>int metodo1 (int a){...}</code>             | <code>int metodo1 (long a) {...}</code>          |
| d) <code>float metodo1 (short a, int b) {...}</code> | <code>void metodo1 (short d, int i) {...}</code> |

- a) No (reciben el mismo número / tipo de parámetros)  
b) No (por lo mismo)  
c) Sí (los parámetros son distintos)  
d) No (por lo mismo)

**PARTE 2: PROBLEMAS****Problema 1 (3 Puntos).-**

Crear una clase llamada `Coche` que tendrá las siguientes características:

- (0,2 puntos) Debe tener 4 atributos privados denominados `marca`, `modelo`, `añoCompra` y `dieselOGasolina(*)`.
- (0,2 puntos) Hacer un método denominado `getMarca` que devuelva la marca del coche.
- (0,4 puntos) Hacer un método `setAñoCompra` que reciba como parámetro el año de compra. Deberá comprobar que el valor recibido es válido.
- Hacer los siguientes constructores (todos deberán comprobar que el año recibido es correcto)
  - o (0,5 puntos) Uno que reciba valores para todos los atributos de la clase `Coche`.
  - o (0,4 puntos) Uno por defecto, sin parámetros que usando el anterior cree un `Coche` comprado en 2007, marca Lancia, modelo Ypsilon y que sea Diesel.
  - o (0,4 puntos) Uno que sólo reciba valores para la marca, el modelo y el año y ponga `dieselOGasolina` a `Gasolina`.
  - o (0,4 puntos) Uno de copia que reciba como parámetro un objeto de tipo `Coche` y cree otro con los mismos atributos.

(0,5 puntos) Crear una clase Denominada `UsoCoche` que cree cuatro objetos de tipo `Coche`, uno con cada uno de los constructores anteriores.

(\*) Se valorará el uso del tipo de datos adecuado así como las etiquetas necesarias para la claridad del código.

```
public class Coche {  
    public static final boolean DIESEL = true;  
    public static final boolean GASOLINA = false;  
  
    private final int AMINIMO = 1950;  
    private final int AMAXIMO = 2008;  
    private String marca;  
    private String modelo;  
    private int aCompra;  
    private boolean dieselOGasolina;  
  
    public String getMarca(){  
        return marca;  
    }  
  
    public void setAnyoCompra(int i) {  
        if (i < AMINIMO || i > AMAXIMO) {  
            System.out.println("El parámetro año no es válido");  
        } else {  
            this.aCompra = i;  
        }  
    }  
}
```

```
public Coche(String marca, String modelo, int a, boolean dOG) {
    this.marca = marca;
    this.modelo = modelo;
    setAnyoCompra(a);
    this.dieselOGasolina = dOG;
}

public Coche() {
    this("Lancia", "Ypsilon", 2007, false);
}

public Coche(String marca, String modelo, int a) {
    this(marca, modelo, a, true);
}

public Coche(Coche c){
    this(c.getMarca(), c.modelo, c.aCompra, c.dieselOGasolina);
}
}

public class UsoCoche {
    public static void main(String[] args) {
        Coche c1 = new Coche("Seat", "Leon", 2001, Coche.DIESEL);
        Coche c2 = new Coche();
        Coche c3 = new Coche("Fiat", "Bravo", 2008);
        Coche c4 = new Coche(c1);
    }
}
```