


EXAMEN DE PROGRAMACIÓN Enero 2009 GRADO EN INGENIERÍA INFORMÁTICA Leganés y Colmenarejo		 Universidad Carlos III de Madrid	
Apellidos		Nombre	
Firma		NIA	Grupo

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 2 horas y 30 minutos
- Se permiten apuntes y/o libros para la realización del examen
- 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 las siguientes afirmaciones son o no ciertas, y **explicar** brevemente por qué.

- 1.1. **(0,5 puntos)** En java, podemos decidir en qué momento se ejecuta el recolector de basura.
- 1.2. **(0,5 puntos)** Nunca se puede acceder a un método de una clase sin haber creado antes un objeto de esa clase.

Pregunta 2 (1 Punto).- Indicar si las siguientes afirmaciones son o no ciertas, y **explicar** brevemente por qué.

- 2.1. **(0,5 puntos)** "Al final del siguiente programa i tendrá valor igual a 8."

```
public class Pregunta2 {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i<8) {  
            i=i+1;  
            i++;  
        }  
    }  
}
```

- 2.2. **(0,5 puntos)** La siguiente es una declaración válida de una clase

```
public class Pregunta2 {  
    public static int a = 0;  
    public static void main(String[] args) {  
        if(args.length > a) {  
            System.out.println("Existen atributos");  
        } else {  
            System.out.println("NO Existen atributos");  
        }  
    }  
}
```

Pregunta 3 (1 Punto).- Marcar la respuesta correcta como resultado de ejecutar el método main de la clase Pregunta3 y explicar brevemente por qué.

```
public class Pregunta3 {  
    public static void main(String[] args){  
        int b = 3, a = 4;  
        if (compara(a,b) & --a == 3){  
            System.out.print(a--);  
        } else {  
            System.out.print(b++);  
        }  
        System.out.print (a+b);  
    }  
    public static boolean compara (int a, int b){  
        if(b==3 | ++a/2>2){  
            System.out.print(a);  
            System.out.print(++b);  
            return true;  
        }  
        else return false;  
    }  
}
```

a) 4435
b) 5437
c) 4535
d) 5435
e) 37
f) Ninguna de las anteriores.

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{
    public byte x;
    public boolean y;
    public char z;

    public Pregunta4 (boolean y1, char z1){
        x=12.0;
        y=true;
        z=z1;}

    public Pregunta4 (byte x1, char z1){
        x=x1;
        if (x1>10) y=true;
        else y=false;
        z=z1;}

    public Pregunta4 (byte x1, char z1, boolean y1){
        this(y1,z1);
        x=x1;}

    public Pregunta4 (byte x, char z){
        this.x=x;
        y=false;
        this.z=z;}

    public int getX(){
        return x;}

    public setX(byte x1){
        x=x1;}}
```

PARTE 2: PROBLEMAS

Problema 1 (4 Puntos).- Crear una clase pública denominada `Naranja` con las siguientes características:

- (0,2 puntos) Deberá pertenecer al paquete `fruta` e importar la clase `java.lang.Math`
- (0,2 puntos) Deberá tener los siguientes atributos públicos:
 - `origen` de tipo `String`
 - `numeroGajos` de tipo `int`
 - `exprimida` de tipo `boolean`
- (0,2 puntos) Deberá tener un atributo de tipo `int` común a todos los objetos de la clase, denominado `numeroNaranjas` que en todo momento almacene cuántos objetos `Naranja` se han creado.
- (0,4 puntos) Crear un método `getOrigen` que devuelva el valor del atributo `origen` y un método `setOrigen` que reciba un valor y establezca ese valor como nuevo `origen`. Hacer métodos similares para `numeroGajos`. Se debe comprobar que el número de gajos está entre 0 y 15 y en el caso de que sean 0, `exprimida` debe ser verdadero.
- (0,4 puntos) Crear un método `exprimir` que no devuelva nada ni reciba parámetros y al final de su ejecución la naranja está exprimida y el `numeroGajos` es cero.
- (0,4 puntos) Crear un método `setGajos` que no reciba parámetros y que establezca el `numeroGajos` de la naranja aleatoriamente entre 0 y 15 (en el caso de que sean 0 `exprimida` debe ser verdadero).
- Crear los siguientes constructores
 - (0,2 puntos) Un constructor por defecto, que cree una naranja de Valencia con 10 gajos.
 - (0,4 puntos) Un constructor que reciba valores para los tres atributos. Deberá comprobar que el `numeroGajos` está entre 0 y 15. Si se sale de rango deberá usar los valores por defecto para ese atributo.
 - (0,4 puntos) Un constructor que sólo reciba valores para el `origen` y el `numeroGajos`. **Debe** usar el anterior constructor.
 - (0,4 puntos) Un constructor de copia, que reciba un objeto de tipo `Naranja` y cree otro con los mismos valores. **Debe** usar el segundo constructor creado.
- (0,4 puntos) Crear un método denominado `equals` para comparar si dos naranjas son iguales. Recibirá como parámetro un objeto `Naranja` y devolverá `true` si la naranja recibida tiene los mismos valores de `origen`, `numeroGajos` y `exprimida` que la `Naranja` sobre la que se invoca.
- (0,4 puntos) Crear un método denominado, `mostrar`, que devuelva la siguiente cadena de caracteres: "Tengo una naranja de "+`origen`+" con "+`numeroGajos`+" gajos" si la naranja no está exprimida y en caso contrario devuelva "Soy un zumo de naranja de "+`origen`.

Problema 2 (1 Punto).- Crear una clase pública denominada `UsoNaranja` con las siguientes características:

- (0,1 puntos) Debe contener un método `main`.
- (0,3 puntos) Declarar dentro del método `main` dos variables de tipo `Naranja`. Crearlas usando un constructor distinto para cada una.
- (0,3 puntos) Crear un array de tipo `Naranja` de 2 posiciones. Crear los elementos del array usando los dos constructores que no se usaron en el punto anterior.
- (0,3 puntos) Escribir el código que compare 2 objetos `Naranja` de los creados e imprima por pantalla si son iguales (`true`) o no (`false`)

Problema 3 (1 Punto).- Crear un método que reciba como parámetro un array con todas las letras del alfabeto (en minúsculas y mayúsculas) y lo catalogue separándolo en 2 arrays distintos: el primero con las **vocales** y el segundo con las **consonantes**, y los muestre por pantalla. Nota: El array recibido ya está inicializado con: {"a", "A", "b", "B",

“c”, “C”, “d”, “D”, ..., “z”, “Z”}