

PRUEBA 2 PROGRAMACIÓN Diciembre 2008 INGENIERÍA INFORMÁTICA Leganés		 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 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.

“Si creamos una clase sin ningún constructor no podremos crear objetos de esa clase”

Falso. Java crea un constructor por defecto, sin parámetros que se puede usar para crear objetos. Este constructor da a los atributos sus valores por defecto (0 para números, el carácter vacío para char, false para booleanos y null para objetos, incluido String)

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

“Una clase podrá utilizar un objeto de otra clase distinta si ambas están en el mismo paquete”

Verdadero. Una clase puede crear objetos de otra que se encuentre en su mismo paquete, independientemente de que se haya declarado o no como public, siempre que exista al menos un constructor que no se haya declarado como privado. No obstante el acceso a los atributos o métodos de ese objeto dependerá de la visibilidad con la que se hayan declarado.

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 long atributo2=4;

    public void compara (int a1, long atributo2){
        if (a1>atributo2)
            {atributo2= atributo2/2;
            }
        }
}

public class UsoPregunta3 {
    public static void main (String [] args){
        Pregunta3 a = new Pregunta3();
        a.compara(5,6);
        System.out.println(a.atributo2);
    }
}
```

Imprime 4 que era el valor que se le había dado al atributo2 en la declaración, ya que al tener el método compara un parámetro que se llama igual que el atributo2, lo oculta, y todas las operaciones que se realicen dentro del método se harán sobre el parámetro y no sobre el atributo por lo que su valor no se verá alterado.

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

    public getA (){
        return a;}}
```

- 1) El primer constructor y el tercero reciben parámetros del mismo tipo, aunque con distintos nombres. Habría que quitar uno de los dos constructores.
- 2) En el segundo constructor, el `this` debe ser la primera instrucción.
- 3) En el método `get` falta especificar el tipo que devuelve, habría que poner `public int getA ()`

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 sean el doble de los originales. Ejemplo: recibe {1,2,3,4} y devuelve {2,4,6,8}

Primera solución, cambiando el propio array que se le pasa por parámetro

```
public int [] dobla (int lista []){
    for (int i=0; i<lista.length; i++)
        lista[i]=2*lista[i];
    return lista;
}
```

Segunda solución, creando un array auxiliar para respetar el original:

```
public int [] dobla2 (int lista []){
    int [] aux = new int [lista.length];
    System.arraycopy(lista, 0, aux, 0, lista.length);
    for (int i=0; i<aux.length; i++)
        aux[i]=2*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 < 9){  
            for (int i=3; i>1;i--)  
                if (a%3==0)  
                    System.out.println(a+i);  
            a++;  
        }  
    }  
}
```

Es un bucle `for` anidado dentro del bucle `while`. En la primera ejecución, como `a` es menor que 6, se ejecuta el `for`, e `i` va tomando los valores 3, 2. Dentro del `for` se comprueba si `a` es divisible por 3, en ese caso se imprime `a+i`. Por lo tanto los valores son:

a	i	a%3	a+i
3	3	0	6
3	2	0	5

A continuación sale del `for`, incrementa `a` y vuelve al `while`, como `a<6` vuelve a entrar en el `for`, pero ahora como `a= 4` no imprime nada. Lo mismo ocurre para `a=5`. Para `a=6` vuelve a ser divisible entre 3 y por lo tanto:

a	i	a%3	a+i
6	3	0	9
6	2	0	8

Para `a= 7` y `a=8` no hace nada y para `a=9` termina.

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

- a) No pueden porque tienen parámetros del mismo tipo, da igual que se llamen de manera distinta, Java no sabría a cuál de los dos nos referimos.
- b) No pueden porque sólo se diferencian en el valor de retorno y tienen que diferenciarse en el tipo y/o número de parámetros.
- c) No pueden, es una combinación de a) y b)
- d) Si pueden, uno recibe un `int` y el otro un `long`, Java sabe a cuál estamos llamando viendo el tipo del parámetro.

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

Crear una clase llamada Gato que tendrá las siguientes características:

- (0,2 puntos) Debe tener 3 atributos privados denominados nombre, edad y raza.
- (0,2 puntos) Hacer un método denominado getNombre que devuelva el nombre del gato.
- (0,4 puntos) Hacer un método setEdad que reciba como parámetro la edad del gato. Deberá comprobar que el valor recibido es válido.
- Hacer los siguientes constructores (todos deberán comprobar que la edad recibida es correcta)
 - o (0,5 puntos) Uno que reciba valores para todos los atributos de la clase Gato.
 - o (0,4 puntos) Uno por defecto, sin parámetros que usando el anterior cree un Gato de 14 años, llamado Garfield de raza Mestiza.
 - o (0,4 puntos) Uno que sólo reciba valores para el nombre y la raza y ponga la edad a 1 año.
 - o (0,4 puntos) Uno de copia que reciba como parámetro un objeto de tipo Gato y cree otro con los mismos atributos.

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

```
public class Gato {
    private String nombre, raza;
    private int edad;

    public String getNombre(){
        return nombre;
    }
    public void setEdad (int e){
        //Un gato vive siempre menos de 30 años
        if (e>=0 && e<30) edad = e;
    }
    // Constructor completo
    public Gato (String n, String r, int e){
        raza = r;
        nombre = n;
        setEdad(e);
    }
    // Constructor sin parámetros
    public Gato(){
        this ("Garfield","Mestizo",14);
    }
    //Tercer constructor
    public Gato (String n, String r){
        this (n,r,1);
    }
    //Constructor de copia
    public Gato (Gato g){
        this(g.nombre,g.raza,g.edad);
    }
}

public class UsoGato {
    public static void main(String[] args) {
        Gato g1,g2,g3,g4;
        g1 = new Gato ("Sultán","Persa",5);
        g2 = new Gato ();
        g3 = new Gato ("Silvestre","Mestizo");
        g4 = new Gato (g2);
    }
}
```