

<b>EXAMEN DE PROGRAMACIÓN</b> <b>18 de Enero de 2010</b> <b>GRADO EN INGENIERÍA INFORMÁTICA</b> <b>Leganés</b>		 Universidad Carlos III de Madrid	
<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 datos personales como 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 3 horas
- Se permiten apuntes y/o libros para la realización del examen
- Para las respuestas a las cuestiones, utilice exclusivamente estas hojas de test. No se recogerá ninguna otra hoja adicional.
- Para las respuestas a los problemas, utilice las hojas adicionales que se le han entregado.

**NO PASE DE ESTA PÁGINA HASTA QUE SE LE INDIQUE**

**PARTE 1: CUESTIONES**

---

**Pregunta 1 (1 Punto).- Explicar** si las siguientes afirmaciones son o no ciertas

- 1.1. (0,25 puntos) Se puede ejecutar una clase (Clase1) previamente compilada usando el comando `javac Clase1` de Java SDK.
- 1.2. (0,25 puntos) Java es capaz de realizar automáticamente algunas conversiones entre tipos.
- 1.3. (0,25 puntos) La única manera de realizar comentarios en Java es colocando `//`. Desde el inicio de las dos barras hasta el final de línea Java reconocerá todo el código como comentario.
- 1.4. (0,25 puntos) Una variable creada dentro de un método se puede utilizar sin problemas en cualquier otro método, incluido el método `main`.

- 1.1. Falso, para ejecutar una clase usando directamente java SDK hay que poner `java Clase1`. Previamente habrá que haber compilado la clase poniendo `javac Clase1.java`
- 1.2. Verdadero. Siempre que el rango del destino sea mayor que el del origen se hace la conversión automática. Por ejemplo para pasar de `int` a `float`, de `float` a `double`, de `long` a `float`, etc.
- 1.3. Falso. Esa es una manera de realizar comentarios en java pero hay otras dos: `/* ... */` y `/** ... */` que comentan todo lo que hay entre los símbolos `/*` y `*/`.
- 1.4. Falso. Una variable creada dentro de un método sólo existe en ese método (se dice que ese método es su ámbito) y por lo tanto no se puede usar en otro método distinto, ni siquiera en el método `main`.

**Pregunta 2 (1 Punto).- Explicar** el resultado de ejecutar el método main de la clase Pregunta2.

```
public class Pregunta2 {
    public static void main(String[] args) {
        int [] numerosChungos = {4, 8, 15, 16, 23, 42};
        boolean esChungo = false;
        for(int n = 1; n <= 50 && !esChungo; n++) {
            int i;
            for(i = 0; i < numerosChungos.length && !esChungo; i++) {
                if(n == numerosChungos[i]) {
                    esChungo = true;
                }
            }
            if(esChungo) {
                System.out.println(n+ " es un número chungo en la
posición "+i);
            }
            else {
                System.out.println( n + " no es un número chungo");
            }
        }
    }
}
```

Empezamos definiendo el array de números chungos y luego utilizamos un bucle que va de 1 a 50 para ir recorriendo números (variable n) y en el que permanecemos mientras esChungo sea false. A continuación en el segundo for recorremos el array de números chungos para cada valor de n (también mientras esChungo sea false). Si el valor de n coincide con el de algún número de numerosChungos, esChungo pasará a ser true. En cada ciclo del bucle n comprobamos si esChungo es verdadero o falso.

Para n=1:

n	i	esChungo
1	0	false
1	1	false
1	2	false
1	3	false
1	4	false
1	5	false

Por lo que imprime "1 no es un número chungo"

Lo mismo ocurre para n= 2 y n = 3, pero para n=4, resulta que es el primer número chungo y por lo tanto n==i[0], por lo que esChungo=true, actualiza i (con el i++) y comprueba la condición, que ya no se cumple porque !esChungo es falso. Así que imprime "4 es un número chungo en la posición 1" y termina también el otro bucle por la misma razón.

Por pantalla:

```
1 no es un número chungo
2 no es un número chungo
3 no es un número chungo
4 es un número chungo en la posición 1
```

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

```
public class Pregunta3 {  
  
    public static void metodo1 (String b){  
        b="hola";  
        return b;  
    }  
  
    public static int metodo2 (String a, double b){  
        a="adios";  
        b=34;  
        int c= b+5;  
    }  
  
    public static void metodo3 (){  
        int a = 35;  
        double b = a;  
        short c;  
        System.out.println(a);  
    }  
    public static void main(String[] args){  
        String h = "caracola";  
        metodo1(h);  
        metodo2(h,231.1);  
        metodo3(35);  
    }  
}
```

- 1) El metodo1 es void, pero tiene un return de un String. Se arregla o quitando el return o poniendo String como valor de retorno.
- 2) El metodo2 tiene que devolver un int según su cabecera, pero no devuelve nada. Se arregla quitando el int y cambiándolo por un void, o haciendo un return.
- 3) En el metodo2, c es de tipo int, pero b es de tipo double por lo que no podemos hacer directamente c=b+5 (un double no cabe en un int). Se arregla mediante un casting: int c = (int) (b+5);
- 4) En el método main, se llama al metodo3 con un parámetro, pero en su definición no hay parámetros. Se arregla eliminando el 35 de la llamada: metodo3( )

**Pregunta 4 (1 Punto).- Explicar** cuál es el resultado del siguiente código:

```
public class Pregunta4{
    static byte x;
    static boolean y;
    static char z;

    public static void metodo1 (char a){
        z=a;
    }

    public static boolean metodo2(){
        return y;
    }

    public static int metodo3 (){
        return (x*5+2)/3;
    }

    public static void main (String [] args){
        System.out.println(z);
        char b='b';
        metodo1(b);
        System.out.println(z);
        System.out.println(metodo2());
        x=6;
        System.out.println(metodo3());
    }
}
```

Se declaran tres variables globales (static): x, y, z y tres métodos. En el método main se imprime z. Como no se le ha dado valor java le da su valor por defecto (para las variables globales Java sí da valores por defecto) por lo que se imprime un cuadradito ('\\0'). A continuación declaramos b y llamamos a metodo1(b) que toma el valor que se le pase por parámetro y se lo asigna a z. Debido a ello, al imprimir otra vez z obtenemos 'b'. Imprimimos lo que devuelve metodo2, que es el valor de y, que como no ha sido inicializado vale false. A continuación ponemos x=6 y llamamos a metodo3, que devuelve  $(6*5+2)/3=32/3=10$  al ser x de tipo int.

Por lo tanto imprime

```
 '\\0' (cuadradito)
b
false
10
```

**Pregunta 5 (1 Punto).**- Sabiendo que el siguiente código imprime "holo y odios" por pantalla, **explicar** cuál de los métodos se puede estar llamando:

```
public class Pregunta5 {  
    public static void main(String[] args) {  
        String a = "hola";  
        String b= "adios";  
        String c = metodo1 (a,b);  
        System.out.println(c);  
    }  
}
```

- a) 

```
public static void metodo1(String a, String b){  
    String c = a+" y "+b;  
    c= c.replace('a','o');  
    System.out.println (c);  
}
```
- b) 

```
public static String metodo1 (String b1, String a1){  
    String x = b1+" y "+a1;  
    x = x.replace('a','o');  
    return x;  
}
```
- c) 

```
public static void metodo1 (String a2, String b2){  
    c = a2+" y " +b2;  
    c = c.replace('a','o');  
}
```
- d) 

```
public static String metodo1 (String a3){  
    return "holo y odios";  
}
```

Por eliminación de candidatos:

a) No puede ser puesto que en el main a c se le da el valor de lo que devuelve metodo1 y a) no devuelve nada porque es void. Por la misma razón c) tampoco puede ser.

d) Tampoco puede ser porque en el main se llama a metodo1 con dos parámetros de tipo String y en d) sólo hay un parámetro.

Por lo tanto tiene que ser b), que efectivamente crea un String denominado x como resultado de sumar su primer parámetro más " y ", más su segundo parámetro. Luego se reemplazan en x las 'a' por las 'o' y se devuelve x.

**PARTE 2: PROBLEMAS**

**Problema 1 (1 Punto).**- Crear un método void denominado `cambiar` que reciba como parámetro un array de `char`, un `char` y un `int`, y cambie todas las letras del array, a partir de la de índice indicado por el `int`, por el nuevo `char`. Si el `int` es negativo, o mayor o igual que el número de caracteres, no se modificará el array original.

Ejemplo, si ejecutamos:

```
char [] a = new char [] { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
cambiar(a, 'k', 4);
```

el array `a` valdrá `{ 'a', 'b', 'c', 'd', 'k', 'k', 'k', 'k' }`;

```
public static void cambiar (char [] s, char c, int b){
    //comprobamos que b está dentro de los límites
    if (b>0 && b<s.length) {
        //Empezamos un bucle en b y seguimos hasta el final
        for (int k=b;k<s.length;k++)
            s[k]=c;
    }
}
```

**Problema 2 (1 Punto).**- Escribir un método que ordene números flotantes de mayor a menor mediante el algoritmo de la burbuja.

```
/**Método que implementa la burbuja de la forma más simple*/
public static void burbuja2(float [] lista){
    float aux = 0;
    //bucle exterior (elementos-1) pasadas como máximo)
    for (int i=1; i<lista.length; i++){
        //Bucle interior (n-i comparaciones)
        for (int j=0; j<lista.length-i; j++){
            // si el elemento de índice inferior es menor que el de índice
            superior cambiamos
            if (lista[j]<lista[j+1]){
                //hacemos el cambio usando la variable auxiliar
                aux = lista [j+1];
                lista [j+1] = lista [j];
                lista [j] = aux;
            }
        }
    }
}
```

**Problema 3 (1 Punto).**- Crear un método que reciba como parámetros dos matrices de dos dimensiones de `int`, y devuelva un entero con el número de elementos que son iguales en ambas matrices y están en la misma posición. Suponiendo que el número de filas y columnas de las dos matrices será siempre igual, el método tiene que funcionar para matrices de cualquier número de filas y columnas.

Ejemplo: Si las matrices son  $\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{Bmatrix}$  y  $\begin{Bmatrix} 2 & 3 & 4 \\ 5 & 5 & 6 \end{Bmatrix}$  devolverá un 2

```
public static int metodo3 (int [][] a, int [][] b){
    int resultado = 0;
    //usamos 2 bucles for para recorrer las matrices
    for (int f=0;f<a.length;f++)
        for (int c=0;c<a[f].length;c++)
            if (a[f][c]==b[f][c]) resultado++;
    return resultado;
}
```

**Problema 4 (2 Puntos).-**

- a) (1 punto) Crear un método que reciba como parámetro un array de `String` y devuelva el elemento con mayor número de caracteres. Si hay más de un elemento de igual longitud deberá devolver el que esté primero en la lista.

Ejemplo, dada la siguiente declaración:

```
String [] s1 = new String []{"hola","adios","casa","familia","caracola"};

metodo1(s1) devuelve "caracola"
```

- b) (1 punto) Crear un segundo método, usando el anterior, que reciba una matriz de `String`, para cada fila busque el elemento con mayor número de caracteres y devuelva un array de `String` formado por estos elementos.

**Nota:** recordar que cada fila de una matriz es un array de 1 dimensión

Ejemplo, dada la siguiente declaración

```
String [][] s2 = new String [][]{{"hola","adios","familia","caracola"}
                                   {"avión","casa","coche","autobús"},
                                   {"uno","dos","el","tu"}};

metodo2(s2) devuelve {"caracola","autobús","uno"}
```

- a)
- ```
public static String metodo1 (String [] s){
    String mayor="";
    int longitudMayor = 0;
    for (int i =0; i<s.length; i++){
        if (s[i].length() > longitudMayor){
            mayor = s[i];
            longitudMayor = mayor.length();
        }
    }
    return mayor;
}
```
- b)
- ```
public static String [] metodo2 (String [][] a){
    //creamos la variable para guardar el resultado
    String [] resultado;
    //su tamaño será el número de filas
    resultado = new String [a.length];
    //recorremos todas las filas
    for (int k=0; k<a.length; k++){
        //para cada una de ellas, llamamos al método anterior
        resultado[k]= metodo1(a[k]);
    }
    return resultado;
}
```