

EXAMEN DE PROGRAMACIÓN Junio 2012 GRADO EN 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 3 horas
- Se permiten apuntes y/o libros para la realización del examen

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Encontrar y **explicar** los 3 errores de compilación que aparecen en el siguiente código Java (por comodidad se ha numerado cada línea). ¿Cómo los resolvería?

```
1. public class Preguntal {
2.     public static void main(String[] args) {
3.         int a = 12;
4.         boolean b;
5.         do {
6.             b = !b;
7.             a++;
8.         } while(a < 12);
9.         for(int j=0; j<10; j++) {
10.            int a = calcula(j);
11.            System.out.println(a);
12.        }
13.    }
14.    public static int calcula(long var) {
15.        return 2*var;
16.    }
17. }
```

- 1) En la línea 6, no se ha inicializado el valor de b, por lo que no puede utilizarse. Se solucionaría dándole valor al declararla en la línea 4:
 boolean b=true;
- 2) En la línea 10, no se puede declarar una variable llamada a porque ya existe una en ese contexto. Se solucionaría llamando a la variable con otro nombre.
- 3) En la línea 15, al multiplicar un long por dos, el resultado es un long. El método debe devolver un entero, por lo que o se hace un casting a entero del resultado, o se cambia el tipo de la variable var a int.

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

2.1. (0,25 puntos) El siguiente bucle se ejecuta infinitamente:

```
boolean b;  
b = true;  
while(true) {  
    b = !b;  
    if(b) {  
        break;  
    }  
}
```

Falso, el bucle se ejecuta **dos veces**. La primera vez se niega el valor de b, por lo que vale false y por tanto no entra por el if. La segunda vez se vuelve a negar, por lo que b vale true, y por tanto entra por el if, y se sale del bucle con la sentencia break.

2.2. (0,25 puntos) Un bucle de tipo for siempre se ejecuta al menos una vez:

Falso, si la condición del bucle no se cumple ni siquiera la primera vez, no se entra por el bucle.

2.3. (0,25 puntos) Si en el siguiente código sustituimos XXXX por cualquier valor numérico entre -100 y +100, el código imprime siempre el valor de la variable tipo por pantalla, independientemente del valor que hayamos elegido:

```
int tipo = XXXX; // Cambiar XXXX por cualquier valor  
switch(tipo) {  
    case 1:  
    case 2:  
        System.out.println("Tipo: " + tipo);  
        break;  
    case 3:  
    default:  
        System.out.println("Tipo: " + tipo);  
}
```

Verdadero, cualquier valor por el que sustituyamos xxxx hará que al entrar por el switch se ejecute una de las dos sentencias que imprimen el tipo por pantalla.

2.4. (0,25 puntos) El siguiente código NO compila:

```
public static void main(String[] args) {  
    if(args == 1) {  
        System.out.println("Vale uno");  
    }  
}
```

Verdadero, no se puede comparar un array de String con un valor entero.

Pregunta 3 (1 Punto).- Dado el siguiente método `main`, y sabiendo que imprime *false* por pantalla, **explicar** a cuál o cuáles de los métodos se puede estar llamando y por qué se descartan los otros:

```
public static void main(String[] args) {  
    int a=8;  
    boolean c = true;  
    c = elMetodo (a);  
    System.out.println(c);  
}
```

a)

```
public boolean elMetodo(float var) {  
    return false;  
}
```

Este NO puede ser, porque le falta la palabra clave "static".

b)

```
public static boolean elMetodo(short var) {  
    int uno = 0;  
    int dos = 1;  
    boolean ret = (uno < dos);  
    return ret;  
}
```

Este NO puede ser, porque recibe como parámetro un "short" y se llama con un "int" por lo que no puede haber casting automático.

c)

```
public static boolean elMetodo(float var) {  
    return !true;  
}
```

Este SI puede ser, porque la negación de "true" es "false".

d)

```
public static boolean elMetodo(int var) {  
    return (10 > 12);  
}
```

Este SI puede ser, como 10 no es mayor que 12 devuelve false.

e)

```
public static boolean elMetodo(int var) {  
    boolean uno = false, dos = true;  
    boolean tres = uno || dos;  
    return tres;  
}
```

Este NO puede ser, se hace un "OR" de true y false, por lo que devuelve "true".

PARTE 2: PROBLEMAS

Problema 1 (1,25 puntos).- Implementar el código de los siguientes métodos estáticos:

- 1) (0,5 puntos) Un método denominado `esTriangulo`, que recibe tres enteros con las medidas de los valores de los tres lados de un triángulo, y devuelve `true` si es un triángulo, y `false` si no lo es.

*Nota: Para comprobar si se trata de un triángulo, hay que comprobar que la suma de dos de los lados es siempre mayor que la longitud del otro lado.

- 2) (0,75 puntos) Un método denominado `esTrianguloRectangulo`, que recibe tres enteros con las medidas de los valores de los tres lados de un triángulo, y:
 - o Comprueba si efectivamente se trata de un triángulo, llamando al método del punto anterior. Si no es un triángulo devuelve `false`.
 - o Si efectivamente es un triángulo, entonces comprueba si se trata de un triángulo rectángulo, devolviendo `true` si lo es, y `false` si no lo es.

*Nota: Para comprobar si un triángulo es rectángulo, basta con comprobar que la suma de los catetos al cuadrado es igual a la hipotenusa al cuadrado.

*Nota: Se puede hacer uso del siguiente método de la librería `Math`, que calcula la raíz cuadrada de un número:

```
public static double sqrt(double a)
```

Problema 2 (2,25 puntos).- Crear una clase denominada `Problema2` que contendrá los siguientes métodos estáticos:

- (0,65 puntos) Método `imprimir`
 - o Parámetros: array de enteros de dos dimensiones
 - o Acción: deberá imprimir el array por pantalla en formato legible, con los elementos de la misma fila separados por coma y un salto de línea entre filas.
 - o Nota: El método debe ser general, debe funcionar para cualquier tamaño de array.
- (0,8 puntos) Método `sumar`
 - o Parámetros: array de enteros de dos dimensiones
 - o Acción: deberá sumar los elementos del array cuya suma del valor de su fila y su columna sea impar
 - o Devuelve: un `int` con la suma anterior
 - o Nota: El método debe ser general, debe funcionar para cualquier tamaño de array.
 - o Ejemplo: si el array creado es
 - 1,2,3
 - 4,5,6
 - 7,8,9
 - 1,2,3

El método suma devolverá la suma de $2 + 4 + 6 + 8 + 1 + 3$, es decir, 24

- (0,8 puntos) Método `main`
 - o Acción:
 - Crea un array de enteros de 2 dimensiones, `m` por `n`, donde `m` y `n` son números aleatorios entre 4 y 20.
 - Rellena todo el array con números aleatorios entre 1 y 10.
 - Imprime el array por pantalla llamando al método `imprimir`
 - Llama al método `sumar` e imprime la suma.

Problema 3 (3,5 puntos).- Para organizar mi biblioteca de libros, quiero hacer un programa en Java que me permita tenerlos colocados y poder localizarlos rápidamente. Para ello, necesito que realices las siguientes tareas:

(1 punto) Crear un nuevo tipo de dato (una clase), denominada `Libro`, que deberá tener:

- a) **(0,2 puntos)** Tres atributos públicos:
 - a. `titulo`: representa el título del libro
 - b. `numeroDePaginas`: representa el número de páginas total.
 - c. `leido`: indica si ya se ha leído el libro o no
- b) **(0,2 puntos)** Un constructor que reciba valores para sus tres atributos.
- c) **(0,2 puntos)** Un constructor por defecto que crea el libro “*El Quijote*” de 1800 páginas (todavía no lo he leído).
- d) **(0,2 puntos)** Métodos `set` y `get` para los tres atributos (se debe comprobar que el número de páginas sea mayor que cero).
- e) **(0,2 puntos)** Un método `leerLibro` que marca el libro como `leido`, haciendo uso del método `set` creado anteriormente.

(2 puntos) Crear una clase denominada `Biblioteca` que nos permitirá tener organizados los libros. La clase deberá tener:

- a) **(0,2 puntos)** Un atributo público denominado `losLibros` que almacenará un array con los libros de la biblioteca (array de `Libro`)
- b) **(0,2 puntos)** Dos métodos `setLosLibros` y `getLosLibros` que permiten establecer el valor del array y devolver el valor del array respectivamente.
- c) **(0,2 puntos)** Un constructor que reciba como parámetro el array de los libros de la `Biblioteca`
- d) **(0,2 puntos)** Un método `setLibro` que recibe un `Libro` y una posición y coloca ese libro en la posición indicada en el array de libros.
- e) **(0,2 puntos)** Un método `dameTitulo` que recibe una posición, y devuelve el título del `Libro` situado en esa posición del array de libros de la `Biblioteca`.
- f) **(1 punto)** Un método denominado `dameLibrosOrdenados` que devuelve una copia del array de libros de la biblioteca, ordenados ascendentemente según su número de páginas (se puede utilizar cualquier algoritmo de ordenación).

(0,5 puntos) Crear una clase denominada `Pruebas` que contendrá un método `main` que hará lo siguiente:

- a) **(0,2 puntos)** Crear tres libros que todavía no están leídos:
 - a. “*El Quijote*”, utilizando el constructor por defecto.
 - b. “*El Lazarillo de Tormes*” que tiene 125 páginas
 - c. “*Don Juan Tenorio*” que tiene 100 páginas
- b) **(0,1 puntos)** Crear una biblioteca, que contiene los tres libros anteriores
- c) **(0,2 puntos)** Imprimir por pantalla los números de páginas de los libros de la biblioteca (separados por coma), ordenados ascendentemente.

Solución problema 1:

```
public static boolean esTriangulo(int a, int b, int c) {
    if(a > b+c || b > a+c || c > a+b) {
        return false;
    }
    return true;
}

public static boolean esTrianguloRectangulo(int a, int b, int c) {
    boolean res = false;

    // Comprueba que un lado no es mayor que la suma de los otros dos
    if(!esTriangulo(a, b, c)) {
        return false;
    }

    // Mira cuál es el mayor de los tres
    if(a > b && a > c) {
        if(Math.sqrt(b*b + c*c) == a) {
            res = true;
        }
    }
    else if(b > a && b > c) {
        if(Math.sqrt(a*a + c*c) == b) {
            res = true;
        }
    }
    else {
        if(Math.sqrt(a*a + b*b) == c) {
            res = true;
        }
    }
    return res;
}
```

Solución problema 2:

```
public class Problema2 {

    public static void imprimir(int[][] enteros) {
        for(int i=0; i<enteros.length; i++) {
            for(int j=0; j<enteros[i].length; j++) {
                if(j>0) {
                    System.out.print(",");
                }
                System.out.print(enteros[i][j]);
            }
            System.out.println();
        }
    }

    public static int sumar(int[][] enteros) {
        int suma = 0;
        for(int i=0; i<enteros.length; i++) {
            for(int j=0; j<enteros[i].length; j++) {
                if((i+j) % 2 == 1) {
                    suma += enteros[i][j];
                }
            }
        }
        return suma;
    }
}
```

```

    public static void main(String[] args) {
        int m, n;
        m = (int)((Math.random() * 17) + 4);
        n = (int)((Math.random() * 17) + 4);
        int[][] arr = new int[m][n];
        for(int i=0; i<arr.length; i++) {
            for(int j=0; j<arr[i].length; j++) {
                arr[i][j] = (int)((Math.random() * 10) + 1);
            }
        }
        imprimir(arr);
        System.out.println(sumar(arr));
    }
}

```

Solución problema 3:

```

public class Libro {
    public String titulo;
    public int numeroDePaginas;
    public boolean leido;
    public Libro(String tit, int numero, boolean lei) {
        titulo = tit;
        numeroDePaginas = numero;
        leido = lei;
    }
    public Libro() {
        titulo = "El Quijote";
        numeroDePaginas = 1800;
        leido = false;
    }
    public String getTitulo() {
        return titulo;
    }
    public void setTitulo(String tit) {
        titulo = tit;
    }
    public int getNumeroDePaginas() {
        return numeroDePaginas;
    }
    public void setNumeroDePaginas(int num) {
        if(num > 0) {
            numeroDePaginas = num;
        }
    }
    public boolean isLeido() {
        return leido;
    }
    public void setLeido(boolean lei) {
        leido = lei;
    }
    public void leerLibro() {
        setLeido(true);
    }
}

public class Biblioteca {
    public Libro[] losLibros;
    public Libro[] getLosLibros() {
        return losLibros;
    }
    public void setLosLibros(Libro[] libs) {
        losLibros = libs;
    }
    public Biblioteca(Libro[] libros) {
        losLibros = libros;
    }
}

```



```
}
public void setLibro(Libro lib, int posicion) {
    if (posicion < losLibros.length)
        losLibros[posicion] = lib;
}
public String dameTitulo(int posicion) {
    return losLibros[posicion].getTitulo();
}
public Libro[] dameLibrosOrdenados() {
    int tamano = losLibros.length;
    Libro[] copia = new Libro[tamano];
    System.arraycopy(losLibros, 0, copia, 0, tamano);
    for(int i=1; i<tamano; i++) {
        for(int j=0; j<tamano-i; j++) {
            if(copia[j].getNumeroDePaginas() >
copia[j+1].getNumeroDePaginas()) {
                Libro aux = copia[j];
                copia[j] = copia[j+1];
                copia[j+1] = aux;
            }
        }
    }
    return copia;
}
}

public class Pruebas {
    public static void main(String[] args) {
        Libro a = new Libro();
        Libro b = new Libro("El Lazarillo de Tormes", 125, false);
        Libro c = new Libro("Don Juan Tenorio", 100, false);
        Libro[] tmp = new Libro[] {a,b,c};
        Biblioteca bib = new Biblioteca(tmp);
        Libro[] ordenados = bib.dameLibrosOrdenados();
        for(int i=0; i<ordenados.length; i++) {
            if(i>0) {
                System.out.print(",");
            }
            System.out.print(ordenados[i].getNumeroDePaginas());
        }
    }
}
```