

PRUEBA 1 PROGRAMACIÓN
Noviembre 2011
INGENIERÍA INFORMÁTICA
Leganés



Universidad
Carlos III de Madrid

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 15 minutos
- 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.

Apellidos	Nombre	
Firma	NIA	Grupo

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Dadas las siguientes declaraciones de variables, explicar cuáles son incorrectas. ¿Cómo se pueden resolver, si es que se puede?

a) `{int a = 0;}`
`double a = 3;`

b) `int a = 0;`
`{double a = 3;}`

c) `String a = "avión", b = "casa", c = a + b + a, d = c + c;`

d) `int [] arr = new int [3];`
`arr[3] = 8;`

- a) Correcta, la primera variable llamada a de tipo int se declara dentro de un bloque, al terminarse ese bloque se puede volver a declarar una variable con el mismo nombre.
- b) Incorrecta. La variable a de tipo int sigue existiendo dentro del segundo bloque, por lo que no podemos declarar otra variable con el mismo nombre. Se puede arreglar cambiando de nombre una de las dos variables.
- c) Correcta. El operador + se puede usar con String y el resultado es la concatenación de las cadenas.
- d) Incorrecta. Declaramos un array de 3 posiciones e intentamos acceder a la posición 3, que no existe, las posiciones del array van de la 0 a la 2.

Pregunta 2 (1 Punto).- Encontrar y explicar los 3 errores de compilación que aparecen en el siguiente código Java. ¿Cómo los resolvería?

```
public static void main(String[] args) {  
    int [] a = new int [];  
    float [] A;  
    A = new float [44];  
    double b = A[12];  
    if (b>8 & A[18]<4){  
        int c = 33;  
    }  
    else c=8;  
    switch (b){  
    case 0: System.out.println("Cero");  
    default: System.out.println("No es cero");  
    }}  
}
```

Primer error, declaramos un array de `int` llamado `a`, pero al crearlo no decimos de cuántas posiciones va a ser. Habría que decir el número de posiciones que tiene.

Segundo error, `c` se declara dentro del bloque del `if`, por lo tanto sólo existe en ese bloque y no se le puede dar valor en el `else`. Habría que declarar `c` antes del `if`.

Tercer error, `b` es de tipo `double` y no se puede utilizar en un `switch`. La variable de un `switch` sólo puede ser de tipo `int` o compatible con `int`. Podríamos hacer un casting para convertir `b` en `int`.

PARTE 2: PROBLEMAS

Problema 1 (1 Punto).- Crear un tipo enumerado denominado `Astros` compuesto por {TIERRA, MARTE, LUNA, SATURNO, SOL, HALLEY, TITAN}. Crear un programa que reciba como parámetro del método `main` uno de los valores anteriores y mediante un `switch` imprima por pantalla si es una estrella, un planeta, un cometa o un satélite. Por ejemplo, si el valor recibido como parámetro es `Astros.TITAN`, deberá imprimir "Es un satélite".

```
enum Astros {TIERRA, MARTE, LUNA, SATURNO, SOL, HALLEY, TITAN};  
public class Problema1 {  
  
    public static void main(String[] args) {  
        Astros astro = Astros.valueOf(args[0]);  
        switch (astro){  
        case LUNA:  
        case TITAN: System.out.println("Es un satélite"); break;  
        case SOL: System.out.println("Es una estrella"); break;  
        case HALLEY: System.out.println("Es un cometa"); break;  
        default: System.out.println("Es un planeta");break;  
        }  
    }  
}
```

Problema 2 (2 Puntos).- Hacer un programa que reciba dos números como parámetros del método `main`.

- Deberá comprobar que se han recibido al menos dos parámetros. En caso contrario imprimirá "Error en los parámetros: se necesitan dos".
- Crear dos variables de tipo `int` y convertir los dos parámetros recibidos a `int` guardándolos en esas variables. Comprobar que ambos son mayores que cero, si no lo son imprimirá "Error en los parámetros: deben ser mayores que cero".
- Si los parámetros son correctos, crear un array de `long` de tantas posiciones como el valor pasado como primer parámetro del método `main` y rellenar todas sus posiciones con el valor pasado como segundo parámetro.
- Imprimir el array resultante.

```
public class Problema2 {  
  
    public static void main(String[] args) {  
        if (args.length<2) System.out.println("Error en los parámetros: se  
necesitan dos");  
        else {  
            int var1, var2;  
            var1 = Integer.parseInt(args[0]);  
            var2 = Integer.parseInt(args[1]);  
            if (var1>0 && var2>0){  
                long [] array = new long [var1];  
                for (int ii=0; ii<var1; ii++)  
                    array[ii]= var2;  
                //También se podría imprimir en el for anterior  
                for (int elemento: array)  
                    System.out.print(elemento+",");  
            }  
            else System.out.println ("Error en los parámetros: deben ser  
mayores que cero");  
        }  
    }  
}
```

Problema 3 (1 Punto).- Suponer que se han creado tres variables de tipo `double` y que se les ha dado valor. Crear el código que sume las 3 si la primera es par y las multiplique si es impar. Imprimir el resultado por pantalla.

```
public class Problema3 {  
    public static void main(String[] args) {  
        //Creamos tres variables de ejemplo  
        double a= 3, b= 8 , c=15;  
        //A partir de aquí la resolución del problema  
        double d;  
        if (a%2==0){  
            d= a+b+c;  
        }  
        else d= a*b*c;  
        System.out.println(d);  
    }  
}
```

Problema 4 (2 puntos).- Crear un nuevo tipo de dato denominado Telefono con los siguientes atributos: marca de tipo String, modelo de tipo String, tactil de tipo boolean y memoria de tipo int. En el main de una clase aparte llamada Problema4 crear un objeto de este nuevo tipo y dar un valor cualquiera a todos sus atributos. Crear un nuevo objeto con valores diferentes a los anteriores. Hacer un código que compare la memoria de ambos objetos y que imprima "El teléfono <marca1> <modelo1> tiene más memoria que el teléfono <marca2> <modelo2> " Donde los elementos entre <> deben sustituirse por los valores adecuados de los objetos. El código debe funcionar para cualquier valor dado a los atributos.

```
//Primero la clase Telefono para definir el nuevo tipo
public class Telefono {
    public String marca, modelo;
    public boolean tactil;
    public int memoria;
}

//Y ahora otra clase en la que usamos el nuevo tipo
public class Problema4 {
    public static void main(String[] args) {
        Telefono tel1 = new Telefono();
        tel1.marca = "Samsung";
        tel1.modelo= "Galaxy";
        tel1.tactil= true;
        tel1.memoria = 4;

        Telefono tel2 = new Telefono ();
        tel2.marca = "Sony";
        tel2.modelo = "W580";
        tel2.tactil = false;
        tel2.memoria= 1;

        if (tel1.memoria>tel2.memoria)
            System.out.println("El teléfono "+tel1.marca+" "+tel1.modelo+"
            tiene más memoria que el teléfono "+tel2.marca+" "+tel2.modelo);
        else
            System.out.println("El teléfono "+tel2.marca+" "+tel2.modelo+"
            tiene más memoria que el teléfono "+tel1.marca+" "+tel1.modelo);
    }
}
```

Problema 5 (2 puntos).- Crear una matriz de `boolean` de 100 filas. La primera fila tendrá 3 columnas, la segunda 6, la tercera 9 y así sucesivamente con múltiplos de tres (usar un bucle). Rellenar la matriz de la siguiente forma: si la columna es par, el valor del elemento será `true`, si no, será `false`. Crear una variable de tipo `int` con valor inicial cero. Recorrer la matriz de forma que cuando un elemento sea `true` se sume uno a la variable y cuando sea `false` se reste uno. Imprimir el valor resultante.

```
public class Problema5 {  
  
    public static void main(String[] args) {  
        boolean [][] matriz = new boolean [100][];  
  
        for (int filas=0; filas<matriz.length; filas++){  
            matriz [filas] = new boolean [(filas+1)*3];  
  
            for (int columna = 0; columna<matriz[filas].length;  
columna++){  
                if (columna%2==0)  
                    matriz [filas][columna]=true;  
                System.out.print(matriz[filas][columna]+" ");  
            }  
            System.out.println();  
        }  
  
        int resultado = 0;  
        for (boolean [] fila: matriz)  
            for (boolean columna: fila)  
                if (columna) resultado++;  
                else resultado--;  
        System.out.println(resultado);  
    }  
}
```