

**EXAMEN DE PROGRAMACIÓN**  
**Junio 2014**  
**GRADO EN INGENIERÍA INFORMÁTICA**  
**Leganés**



Universidad  
Carlos III de Madrid

**LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:**

- Rellene todas las hojas a bolígrafo, no utilice lápiz ni bolígrafo rojo
- El tiempo máximo de realización es de 3 horas
- Se permiten apuntes y/o libros para la realización del examen pero no dispositivos electrónicos de ningún tipo

**Pregunta 1 (1 punto).**- Encontrar los 4 errores de compilación que aparecen en este programa y **explicar** si se podrían solucionar y cómo.

```
public class Pregunta1 {  
  
    public static metodo1 (int a){  
        if (a>0)  
            System.out.println("positivo");  
        else  
            System.out.println("negativo");  
    }  
  
    public void metodo2 (double b){  
        for (int ii=0; ii<b; ii++)  
            System.out.println(b+2);  
    }  
  
    public static void main(String[] args) {  
        int a = 4, b = 8;  
        double c = a+b;  
        float d = c-3;  
        metodo1(b);  
        metodo2(a);  
        if (ii>0) System.out.println("positivo");  
    }  
}
```

**Solución:**

El primer error se encuentra en la definición del metodo1, en la cabecera no especificamos el tipo de dato que devuelve. Como no devuelve nada (no hay ningún return) debería ser void y por lo tanto habría que haberlo definido de la siguiente manera: **public static void metodo1 (int a)**

El segundo error está en la declaración del float d. Le asignamos el valor de un double menos 3, pero un double (64 bits) no cabe en un float (32 bits), por lo que habría que hacer un casting del tipo **float d = (float) (c-3)**

El tercer error está en la llamada a metodo2(a). metodo2 no se ha definido como static y por lo tanto no se puede llamar desde el método main que es static. Para poder llamarlo habría que haberlo declarado también como static: **public static void metodo2 (double b)**

El último error está en el if, la variable ii fue declarada dentro del for del metodo2 y solo existe dentro de ese for, por lo que no está declarada en el main. Una posibilidad para usar su valor fuera del método sería que metodo2 devolviera el valor de ii (cambiando el

`void` por un `int` y añadiendo un `return ii`). También se podría declarar la variable `ii` en el método `main`.

**Nota:** Las siguientes preguntas están relacionadas entre sí, no obstante se pueden realizar de manera independiente. Es decir, aunque no se sepa cómo realizar un determinado método o una clase se pueden usar sin problemas en otras clases o métodos.

**Pregunta 2 (0,5 puntos).**- Crear un tipo enumerado denominado `Clasificacion` con los siguientes componentes: `FASE_GRUPOS`, `OCTAVOS`, `CUARTOS`, `SEMIFINALES`, `FINAL`, `CAMPEON`

```
public enum Clasificacion {FASE_GRUPOS, OCTAVOS, CUARTOS, SEMIFINALES, FINAL, CAMPEON}
```

**Pregunta 3 (2,5 puntos).**- Crear un nuevo tipo (una nueva clase) denominado `Seleccion` con las siguientes características:

- (0,3 puntos)** Atributos privados: `nombre`, que no podrá cambiar, de tipo `String`, `golesFavor`, `golesContra` y `puntos` de tipo entero, y `ronda` de tipo `Clasificacion`.
- (0,2 puntos)** Crear un método `getRonda` y un método `setRonda`, que devuelva la ronda máxima a la que ha llegado una `Seleccion` o cambie el valor de ese atributo, respectivamente. **Asumir** si hace falta que existen métodos similares para el resto de atributos.
- (0,2 puntos)** Un constructor sin parámetros que ponga como nombre "desconocido", ronda a `FASE_GRUPOS` y el resto de atributos a cero.
- (0,3 puntos)** Un constructor que reciba el nombre de la `Seleccion` y ponga el resto de atributos a los mismos valores que el constructor anterior.
- (0,5 puntos)** Un método `void sumaPuntos` que recibirá dos parámetros de tipo entero. El primero serán los goles que ha marcado la `Seleccion` en un partido y el segundo los que ha recibido. Deberá comprobar que ambos son mayores o iguales a cero, si no lo son, no hará nada. Si son correctos calculará los puntos que ha obtenido esa `Seleccion` en ese partido (3 si ha ganado, 1 si ha empatado y 0 si ha perdido) y se los sumará al atributo `puntos`. Finalmente actualizará `golesFavor` y `golesContra`.
- (0,5 puntos)** Un método `toString` que devuelva lo siguiente:
  - Si el equipo está en la fase de grupos: "El equipo <nombre> ha conseguido <puntos> puntos, con <golesFavor> goles a favor y <golesContra> goles en contra".
  - Si el equipo está en otra fase: "El equipo <nombre> ha llegado a <ronda>"Los valores entre <> deben cambiarse por el valor del atributo correspondiente.
- (0,5 puntos)** Un método `compareTo` que reciba como parámetro otra `Seleccion` y que servirá para comparar cuál ha quedado mejor en la fase de grupos. Devolverá -1 si la primera ha quedado mejor que la segunda, 1 si la segunda ha quedado

mejor que la primera y 0 si ambas han quedado igual. Para ello se tendrá en cuenta lo siguiente:

- Una *Seleccion* ha quedado mejor que otra si tiene más puntos que ella.
- Si tienen los mismos puntos, una *Seleccion* ha quedado mejor si su diferencia de goles (goles a favor menos goles en contra) es mayor.
- Si también empatan en diferencia de goles, quedará mejor la que tenga más goles a favor.
- Si tampoco hay diferencia en eso, se considerará que han empatado.

```
public class Seleccion {
    //Apartado a)
    private final String nombre;
    private int golesFavor;
    private int golesContra;
    private int puntos;
    private Clasificacion ronda;

    //Apartado b)
    public void setRonda(Clasificacion ronda) {
        this.ronda = ronda;
    }

    public Clasificacion getRonda() {
        return ronda;
    }

    //Apartado c)
    public Seleccion () {
        nombre = "desconocido";
        ronda = Clasificacion.FASE_GRUPOS;
        //El resto no hace falta ponerlos a 0 porque Java ya lo hace
    }

    //Apartado d)
    public Seleccion (String n){
        nombre = n;
        ronda = Clasificacion.FASE_GRUPOS;
        //El resto no hace falta ponerlos a 0 porque Java ya lo hace
    }

    //Apartado e)
    public void sumaPuntos (int favor, int contra){
        if (favor>=0 && contra>=0){
            if (favor>contra)
                puntos = puntos + 3;
            else if (favor == contra)
                puntos = puntos + 1;
            //Actualizamos goles a favor y en contra
            golesFavor = golesFavor + favor;
            golesContra = golesContra + contra;
        }
    }

    //Apartado f)
    public String toString () {
        if (ronda==Clasificacion.FASE_GRUPOS)
            return "El equipo "+nombre+" ha conseguido "+puntos+" puntos, con
            "+golesFavor+" goles a favor y "+golesContra+" goles en contra";
    }
}
```

```

        else
            return "El equipo "+nombre +" ha llegado a "+ronda;
    }

    //Apartado g)
    public int compareTo (Seleccion otra){
        if (puntos>otra.puntos) return -1;
        else if (puntos<otra.puntos) return 1;
        //Si llega aquí es que están empatadas a puntos
        //Este else se podría quitar, pero lo dejamos por claridad
        else {
            int dif1 = golesFavor - golesContra;
            int dif2 = otra.golesFavor - otra.golesContra;
            if (dif1>dif2) return -1;
            else if (dif1<dif2) return 1;
            //Si llega aquí también están empatadas a diferencia de goles
            else if (golesFavor > otra.golesFavor) return -1;
            else if (golesFavor < otra.golesFavor) return 1;
            else return 0;
        }
    }

    //Métodos que asumimos que existen
    public int getGolesFavor() {
        return golesFavor;
    }
    public void setGolesFavor(int golesFavor) {
        this.golesFavor = golesFavor;
    }
    public int getGolesContra() {
        return golesContra;
    }
    public void setGolesContra(int golesContra) {
        this.golesContra = golesContra;
    }
    public String getNombre() {
        return nombre;
    }
    public int getPuntos() {
        return puntos;
    }
    public void setPuntos(int puntos) {
        this.puntos = puntos;
    }
}

```

**Pregunta 4 (1,5 puntos).**- Crear un nuevo tipo (una nueva clase) denominado Partido con las siguientes características

- (0,3 puntos)** Atributos privados: local de tipo Seleccion, visitante también de tipo Seleccion y resultado que será un array de dos enteros.
- (0,2 puntos)** Un método getResultado que devuelva el resultado. **Asumir** que existen métodos get para el resto de atributos.
- (0,4 puntos)** Un constructor que reciba los valores de local y visitante y cree un array de dos elementos para resultado.
- (0,3 puntos)** Un método void jugar que no recibe parámetros. Rellenará el array resultado de la siguiente forma: se creará un número entero aleatorio entre 0 y 6 para los goles de la Seleccion local y otro similar para los goles del

visitante. El primer elemento serán los goles marcados por la Selección que juega como local y el segundo los de la que juega como visitante.

- e) **(0,3 puntos)** Un método `toString` que devuelva: "<nombre selección local>: <goles marcados> - <nombre selección visitante>: <goles marcados>". Por ejemplo "España: 3 - Holanda: 1".

```
public class Partido {
    //Apartado a)
    private Seleccion local;
    private Seleccion visitante;
    private int [] resultado;

    //Apartado b)
    public int [] getResultado() {
        return resultado;
    }

    //Apartado c)
    public Partido (Seleccion l, Seleccion v){
        local = l;
        visitante = v;
        resultado = new int [2];
    }

    //Apartado d)
    public void jugar (){
        //Ponemos 7 para que el resultado esté entre 0.0 y 6.9999 que al pasar a
        // int estará entre 0 y 6
        resultado[0] = (int) (Math.random()*7);
        resultado[1] = (int) (Math.random()*7);
    }

    //Apartado e)
    public String toString (){
        return local.getNombre()+": "+resultado[0]+" - "+visitante.getNombre()+":
"+resultado[1];
    }

    //Métodos que asumimos que existen
    public Seleccion getLocal() {
        return local;
    }
    public Seleccion getVisitante() {
        return visitante;
    }
}
```

**Pregunta 5 (3,5 puntos).**- Crear un nuevo tipo (una nueva clase) denominado Grupo con las siguientes características:

- (0,4 puntos)** Atributos: un array privado de objetos de tipo Selección de 4 elementos, denominado equipos.
- (0,3 puntos)** Métodos `getEquipos` y `setEquipos`, este último deberá comprobar que el array recibido tiene exactamente 4 elementos, en caso contrario no hará nada.
- (0,2 puntos)** Un constructor que reciba un array de equipos y use `setEquipos`.
- (0,4 puntos)** Un método `buscarSeleccion` que reciba el nombre de una

Seleccion y devuelva la posición en la que está dentro del array equipos o -1 si no está en el array.

- e) **(0,7 puntos)** Un método void denominado partido que reciba los nombres de dos selecciones, compruebe que están en el grupo (con buscarSeleccion) y en caso afirmativo cree un objeto de tipo Partido y llame al método jugar de ese objeto. A continuación imprimirá el resultado del partido (usando el método toString de Partido), y actualizará los goles a favor, en contra y los puntos de cada una de las selecciones (cambiando sus valores en la posición correspondiente del array equipos usando el método sumaPuntos de Seleccion)
- f) **(0,5 puntos)** Un método void jugarGrupo que juegue todos los partidos de un Grupo, enfrentado a todos los equipos entre sí.
- g) **(0,5 puntos)** Un método void ordenarGrupo que ordene, usando el algoritmo de la burbuja las selecciones del array equipos del grupo según su clasificación. Usar para ello el método compareTo de la clase Seleccion.
- h) **(0,5 puntos)** Un método toString que devuelva los elementos del atributo equipos, utilizando para ello el método toString de la clase Seleccion.

```
public class Grupo {
    //Apartado a)
    private Seleccion [] equipos = new Seleccion [4];

    //Apartado b)
    public Seleccion [] getEquipos () {
        return equipos;
    }

    public void setEquipos (Seleccion [] eq){
        if (eq.length == 4)
            equipos = eq;
    }

    //Apartado c)
    public Grupo (Seleccion [] eq){
        setEquipos(eq);
    }

    //Apartado d)
    public int buscarSeleccion (String n){
        //recorremos el array buscando el nombre
        for (int ii=0; ii<equipos.length; ii++){
            if (equipos[ii].getNombre().equals(n)) return ii;
        }
        //Si ha llegado hasta aquí es que no está, y devolvemos -1
        return -1;
    }

    //Apartado e)
    public void partido (String sel1, String sel2){
        //Buscamos si están en el grupo y nos quedamos con las posiciones
        int pos1 = buscarSeleccion(sel1);
        int pos2 = buscarSeleccion(sel2);
        //Si ambas están en el grupo
        if (pos1>=0 && pos2>=0){
            //Creamos el objeto partido
            Partido parti = new Partido(equipos[pos1],equipos[pos2]);
        }
    }
}
```

```

        parti.jugar();
        //Imprimimos el resultado
        System.out.println(parti);
        //Sumamos los puntos y goles a favor
        equipos[pos1].sumaPuntos(parti.getResultado()[0],
parti.getResultado()[1]);
        equipos[pos2].sumaPuntos(parti.getResultado()[1],
parti.getResultado()[0]);
    }
}

//Apartado f)
public void jugarGrupo (){
    //Tenemos que recorrer el array enfrentando al primero con el resto
    //Al segundo con el tercero y cuarto, etc.
    for (int ii=0; ii<equipos.length; ii++){
        for (int jj=ii+1; jj<equipos.length; jj++){
            partido (equipos[ii].getNombre(), equipos[jj].getNombre());
        }
    }
}

//Apartado g)
//En la burbuja original recibimos la lista, aquí no hace falta porque podemos
// acceder directamente a ella porque es un atributo de esta clase
public void ordenarGrupo (){
    // definimos las variables auxiliares que necesitamos
    Seleccion aux;
    boolean cambio = true;
    //bucle exterior: (elementos-1) pasadas como máximo
    for (int i=1; i<equipos.length && cambio; i++){
        //Al inicio de cada pasada ponemos el valor de cambio a falso
        cambio = false;
        //Bucle interior (n-i comparaciones)
        for (int j=0; j<equipos.length-i; j++){
            // si el elemento de índice inferior es menor que el de
            //índice superior cambiamos
            // usamos el método compareTo
            if (equipos[j].compareTo(equipos[j+1])>0){
                //como vamos a hacer un cambio ponemos la
                //variable a verdadero
                cambio = true;
                //hacemos el cambio usando la variable auxiliar
                aux = equipos [j+1];
                equipos [j+1] = equipos [j];
                equipos [j] = aux;
            }
        }
    }
}

//Apartado h)
public String toString (){
    String result="";
    for (int ii=0; ii<equipos.length; ii++)
        result = result + equipos[ii].toString() + "\r\n";
    return result;
}
}

```

**Pregunta 6 (1 punto).**- Crear una clase con un método `main` que hará lo siguiente:

- a) (0,4 puntos) Pedirá al usuario el nombre de los 4 equipos que componen un Grupo y

- creará un objeto Grupo con ellos.
- b) (0,2 puntos) Jugará los partidos del Grupo.
  - c) (0,2 puntos) Ordenará los equipos según los resultados.
  - d) (0,2 puntos) Imprimirá el resultado final del Grupo.

```
import java.util.Scanner;

public class MetodoMain {

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        Seleccion [] sels = new Seleccion [4];
        for (int ii=0; ii<4; ii++) {
            System.out.println("Dame el nombre de la selección "+(ii+1));
            String nom = sc.next();
            sels[ii] = new Seleccion (nom);
        }
        Grupo gr = new Grupo(sels);
        gr.jugarGrupo();
        gr.ordenarGrupo();
        System.out.println(gr);

        sc.close();
    }
}
```

Ejemplo de ejecución:

Dame el nombre de la selección 1

España

Dame el nombre de la selección 2

Holanda

Dame el nombre de la selección 3

Chile

Dame el nombre de la selección 4

Australia

España: 2 - Holanda: 4

España: 2 - Chile: 2

España: 3 - Australia: 3

Holanda: 3 - Chile: 1

Holanda: 5 - Australia: 2

Chile: 5 - Australia: 0

El equipo Holanda ha conseguido 9 puntos, con 12 goles a favor y 5 goles en contra

El equipo Chile ha conseguido 4 puntos, con 8 goles a favor y 5 goles en contra

El equipo España ha conseguido 2 puntos, con 7 goles a favor y 9 goles en contra

El equipo Australia ha conseguido 1 puntos, con 5 goles a favor y 13 goles en contra