

EXAMEN DE PROGRAMACIÓN
Enero 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).- Dado el siguiente método main:

```
public static void main(String [] args) {  
    int a=1, b;  
    float d=9.2F;  
    b= metodo (a,d);  
    System.out.println(b);  
}
```

Explicar para cada uno de los métodos siguientes si podrían ser el método que se está llamando o no y por qué.

- a)

```
public static int metodo(int param1, float param2) {  
    return (int)(param1+param2);  
}
```
- b)

```
public static int metodo(float param1, float param2){  
    return (int)(param1+param2);  
}
```
- c)

```
public static byte metodo(double param1, double param2){  
    return (byte)(param1+param2);  
}
```
- d)

```
public static int metodo (float param1, int param2){  
    return (int)(param1+param2);  
}
```

Respuesta:

En la invocación en el método main, llamamos a nuestro método con dos parámetros, el primero un int y el segundo un float y guardamos el resultado en un int. Por lo tanto, cualquier método cuya cabecera sea compatible con esta configuración podría ser el método llamado.

- a) Este método recibe dos parámetros, un int y un float y devuelve un entero (resultado de hacer un casting forzado a la suma de los dos parámetros, que sin el casting sería un float). En el método main se llama a nuestro método con las variables a (int) y d (float) por lo que se podría estar llamando a este método sin problemas. El resultado se guarda en la variable b (int) que se corresponde con el tipo que devuelve el método. Por lo tanto este método se podría estar

llamando.

- b) El segundo parámetro es un `float`, como en la invocación, y devuelve un `int`. Hasta aquí todo bien. El “problema” es que este método recibe un `float` como primer parámetro y lo estamos llamando con un `int`. Pero como `int` es directamente compatible con `float`, cuando llamamos al método Java convierte (hace un casting automático) el `int` a `float` automáticamente. Por lo tanto también se podría tratar de este método.
- c) En este método los parámetros son dos `double` y lo llamamos con un `int` y un `float`, que son compatibles con `double`, por lo que no hay problema. El resultado es un `byte`, que puede guardarse sin problemas en un `int`. Por lo tanto también se podría tratar de este método.
- d) En este caso los parámetros son `float` e `int`. Para el `float` no hay problema porque lo llamamos con un `int` y Java hace la conversión automática. Pero para el `int`, al intentar llamarlo con un `float` no se hace la conversión automática, habría que hacer un casting forzado. Por lo tanto, este método no se podría estar llamando.

Pregunta 2 (2 puntos).- Los partidos políticos mayoritarios han decidido implantar un asistente automático para diputados que les indique qué deben hacer ante situaciones cotidianas de la vida política. Para ello, los políticos tendrán que responder S o N a una serie de preguntas que les hará el ordenador. Primero deben seleccionar si están ante una votación en el congreso (seleccionarán 1) o un caso de corrupción (seleccionarán 2). Se debe comprobar que las entradas del usuario son correctas, tanto en sus posibles valores como en sus tipos. En el caso de una votación se les preguntará si la votación la ha propuesto su partido, en cuyo caso deben votar SI, o la ha propuesto otro partido, por lo que deberán votar NO. En el caso de corrupción, de igual manera se les preguntará si el presunto corrupto es de su partido, en cuyo caso habrá que proclamar su inocencia, o del partido contrario, por lo que habrá que pedir su dimisión. En ambas opciones suponer que cualquier entrada que no sea una "S" o una "s" es un NO.

Ejemplos de funcionamiento:

```
Escriba el número de la opción deseada
1: Votación en el congreso
2: Presunto corrupto
ffff
¡Un 1 o un 2 por favor!
2
¿Es de nuestro partido?(S/N)
S
Proclamar inocencia
```

```
Escriba el número de la opción deseada
1: Votación en el congreso
2: Presunto corrupto
3
Escriba el número de la opción deseada
1: Votación en el congreso
2: Presunto corrupto
1
¿La hemos propuesto nosotros?(S/N)
N
Votar en contra
```

```

import java.util.Scanner;
public class Pregunta2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int opcion =0;
        do {
            System.out.println("Escriba el número de la opción deseada");
            System.out.println("1: Votación en el congreso");
            System.out.println("2: Presunto corrupto");
            //Comprobamos que sea un entero
            while (!sc.hasNextInt()){
                sc.next();
                System.out.println("¡Un 1 o un 2 por favor!");
            }
            opcion = sc.nextInt();
        }
        while(opcion!=1 && opcion!=2);

        if (opcion==1){
            System.out.println("¿La hemos propuesto nosotros?(S/N)");
            String s = sc.next();
            if (s.equalsIgnoreCase("s"))
                System.out.println("Apoyarla");
            else
                System.out.println("Votar en contra");
        }
        else {
            System.out.println("¿Es de nuestro partido?(S/N)");
            String s = sc.next();
            if (s.equalsIgnoreCase("s"))
                System.out.println("Proclamar inocencia");
            else
                System.out.println("Pedir dimisión");
        }
        sc.close();
    }
}

```

Pregunta 3 (2 puntos).- Crear un programa para calcular el precio de las entradas de cine:

- (0,5 puntos)** Crear un tipo enumerado `Dias` para guardar los días de la semana
- (0,5 puntos)** Crear un método `precioMiercoles` que reciba dos números `double`, compruebe que el primero es menor que el segundo y en caso afirmativo devuelva un número aleatorio (también `double`) comprendido entre ambos. En caso negativo devolverá 5.0
- (0,5 puntos)** Crear un método `queDiaEsHoy` que le pregunte al usuario qué día de la semana es y lo devuelva (debe devolver un valor del tipo enumerado `Dias`). Suponer que el usuario siempre va a introducir un día correcto, pero que puede hacerlo en mayúsculas o minúsculas.
- (0,5 puntos)** Crear un método `main` que pida al usuario el día de la semana (llamando al método `queDiaEsHoy`). Si el día es cualquiera menos el miércoles, las entradas costarán 9 euros. Si es miércoles tendrán un valor aleatorio entre 2,90 y 5 euros (calculado llamando al método `precioMiercoles`). Imprimir el precio de las entradas.

```

import java.util.Scanner;

enum Dias {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO}

public class Pregunta3 {

    public static double precioMiercoles (double min, double max){
        if (min<max)
            return Math.random()*(max-min)+min;
        else
            return 5.0;
    }

    public static Dias queDiaEsHoy(){
        Scanner sc = new Scanner(System.in);
        System.out.println("¿Qué día es hoy?");
        String dia =sc.next();
        sc.close();
        return Dias.valueOf(dia.toUpperCase());
    }

    public static void main(String[] args) {
        Dias d = queDiaEsHoy();
        if (d==Dias.MIERCOLES){
            double precio = precioMiercoles(2.9,5);
            System.out.println("El precio es "+precio+" euros");
        }
        else
            System.out.println("El precio es de 9 euros");
    }
}

```

Pregunta 4 (1,5 puntos).- Crear un programa que contenga:

- (0,4 puntos)** Un método `contarLetras` que reciba una palabra y una letra (ambas en forma de `String`) y devuelva cuántas veces aparece la letra dentro de la palabra independientemente de que aparezca en mayúsculas o minúsculas.
- (0,4 puntos)** Un método `contarVocales` que reciba un `String` y devuelva el número de vocales que tiene (usar el método `contarLetras` anterior).
- (0,7 puntos)** Un método `main` que le pida al usuario que introduzca por teclado tantas palabras como quiera hasta que se introduzca la palabra "fin". El programa irá guardando estas palabras en una lista (se recomienda usar un `ArrayList` aunque no es obligatorio). A continuación imprimirá la lista de palabras e imprimirá también la palabra que tenga mayor número de vocales y el número de vocales que tiene (si hay más de una se puede imprimir la que consideremos oportuno)

```

import java.util.ArrayList;
import java.util.Scanner;

public class Pregunta4 {

    public static int contarLetras(String palabra, String letra){
        int contador = 0;
        //Ponemos todo en mayúsculas
        palabra = palabra.toUpperCase();
        letra = letra.toUpperCase();
    }
}

```

```

        //Hay varias formas, una sencilla de implementar
        //es usar un bucle y charAt
        for (int ii=0; ii<palabra.length(); ii++){
            //Como letra es un String, para compararlo con un char
            //usamos también charAt
            if (palabra.charAt(ii)==letra.charAt(0))
                contador++;
        }
        return contador;
    }
    public static int contarVocales (String palabra){
        int contador = 0;
        //Usamos un array y un bucle en lugar de llamar 5 veces a contarLetras
        String [] vocales = new String []{"a","e","i","o","u"};
        for (String vocal:vocales)
            contador = contador+contarLetras(palabra,vocal);
        return contador;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        String palabra = "";
        ArrayList<String> palabras = new ArrayList<String>();
        //Pedimos las palabras, metemos también fin en el array por comodidad
        do {
            System.out.println("Introduce una palabra. Introduce \"fin\" para
terminar");
            palabra = sc.next();
            palabras.add(palabra);
        }
        while (!palabra.equals("fin"));
        int maxVocal = 0, vocales = 0;
        String p = "";
        //Imprimimos la lista y aprovechamos para calcular la que tiene más
        //vocales, no imprimimos fin
        for (int ii=0; ii<palabras.size()-1;ii++){
            System.out.println(palabras.get(ii));
            vocales = contarVocales(palabras.get(ii));
            if (vocales > maxVocal){
                p = palabras.get(ii);
                maxVocal = vocales;
            }
        }
        System.out.println("La palabra con más vocales es "+p+" con "+maxVocal+"
vocales");
        sc.close();
    }
}

```

Pregunta 5 (2 puntos).- Crear la clase Baldosa que contenga:

- (0,2 puntos)** Atributos privados: ancho y alto en **centímetros** de tipo int (valor predeterminado 25), color (valor predeterminado blanca) de tipo String, precio de tipo float (que será el precio de la baldosa sin IVA y valdrá 1 por omisión) e IVA de tipo int que será una **constante** (valor predeterminado 21).
- (0,2 puntos)** Métodos getPrecio y setPrecio (que deberá comprobar que el precio es mayor que cero). Suponer que existen métodos get y set para el resto de atributos.
- (0,1 puntos)** Un constructor sin parámetros que dé a los atributos sus valores

predeterminados.

- d) (0,3 puntos) Un constructor completo que use los métodos `set`.
- e) (0,4 puntos) Un método `numeroBaldosas` que reciba como parámetro una superficie en **metros cuadrados** y devuelva el número de baldosas que hacen falta para esa superficie. Si el número no es entero, deberá redondear hacia arriba.
- f) (0,3 puntos) Un método `precioSuperficie` que reciba como parámetro cuánto mide una superficie en metros cuadrados y devuelva cuánto cuestan las baldosas (con IVA) para esa superficie (deberá usar el método anterior).
- g) (0,3 puntos) Un método `equals` que compare dos baldosas y devuelva si son iguales (`true`) o no (`false`). Dos baldosas son iguales si su `color`, `ancho` y `alto` son iguales. El `precio` no importa.
- h) (0,2 puntos) Un método `toString` que devuelva: `<alto>x<ancho>, <color>`. Precio por metro cuadrado con IVA: `<p>`. Donde los valores entre `<>` deben ser los adecuados para cada objeto `Baldosa`.

```
public class Baldosa {
    private int alto = 25;
    private int ancho = 25;
    private String color = "blanca";
    private float precio = 1.0F;
    //todos los objetos baldosa van a llevar un IVA del 21%
    private final int IVA = 21;

    public int getAlto() {
        return alto;
    }
    public int getAncho() {
        return ancho;
    }
    public String getColor() {
        return color;
    }
    public float getPrecio() {
        return precio;
    }
    public int getIVA(){
        return IVA;
    }
    public void setAlto(int alto) {
        if (alto>0)
            this.alto = alto;
    }
    public void setAncho(int ancho) {
        if (ancho>0)
            this.ancho = ancho;
    }
    public void setColor(String color) {
        this.color = color;
    }
    public void setPrecio(float precio) {
        if (precio>0)
            this.precio = precio;
    }
    //No hay setIVA porque no puede cambiarse su valor

    public Baldosa (){}
}
```

```

    public Baldosa (int al, int an, String c, float p){
        setAlto(al);
        setAncho(an);
        setColor(c);
        setPrecio(p);
    }

    public int numeroBaldosas (float superficie){
        float superficieBaldosa = alto*ancho/10000.0F;
        return (int)Math.ceil(superficie/superficieBaldosa);
    }

    public float precioSuperficie (float superficie){
        float precioIVA = precio+precio*IVA/100;
        return numeroBaldosas(superficie)*precioIVA;
    }

    public boolean equals (Baldosa o){
        return ancho==o.ancho && alto==o.alto && color.equals(o.color);
    }

    public String toString (){
        //Calculamos el precio por 1 metro cuadrado con IVA
        float metro2 = precioSuperficie(1);
        return alto+"x"+ancho+", "+color+
            ". Precio por metro cuadrado con IVA: "+metro2;
    }
}

```

Pregunta 6 (1,5 puntos).- Crear la clase CajaBaldosas que contenga:

- (0,2 puntos)** Atributos privados: baldosas, array de Baldosa
- (0,5 puntos)** Un método setBaldosas y un método getBaldosas. El método setBaldosas deberá comprobar que todas las baldosas del array que se pasa como parámetro sean iguales (usar el método equals de la clase Baldosa). Si no, no hará nada.
- (0,2 puntos)** Un constructor que reciba como parámetro un array de Baldosa y use el método set anterior.
- (0,3 puntos)** Un método toString que devuelva: Caja de <X> baldosas de <alto>x<ancho>, <color>. Precio por metro cuadrado con IVA: <p>. Donde los valores entre <> deben ser los adecuados y el precio es por Baldosa.
- (0,3 puntos)** En un método main en otra clase distinta crear un array de dos baldosas iguales cualesquiera y una CajaBaldosas con ese array e imprimirla.

```

public class CajaBaldosas {
    private Baldosa [] baldosas;

    public void setBaldosas(Baldosa [] bal){
        boolean iguales = true;
        for (int ii=0; ii<bal.length && iguales; ii++){
            //En el momento en que una baldosa sea distinta, salimos del bucle
            iguales = bal[0].equals(bal[ii]);
        }
    }
}

```

```
//Si iguales es verdadero, asignamos el array al atributo
if (iguales) baldosas = bal;
}

public Baldosa [] getBaldosas(){
    return baldosas;
}

public CajaBaldosas (Baldosa [] bal){
    setBaldosas(bal);
    //Si el array no es correcto el atributo baldosas
    //será un array sin tamaño
}

public String toString (){
    //Como todas son iguales, imprimimos la primera
    return "Caja de "+baldosas.length+" baldosas de "+baldosas[0].toString();
}
}

public class Prueba {
    public static void main (String [] args){
        Baldosa b1 = new Baldosa (), b2 = new Baldosa();
        Baldosa [] arr = new Baldosa[]{b1,b2};
        CajaBaldosas caja = new CajaBaldosas (arr);
        System.out.println(caja);
    }
}
```