



## C. F. G. S. Desarrollo de Aplicaciones Web - Distancia -

### Examen de Programación de la 1ª Evaluación. 11-12-2018

#### Unidades didácticas 1, 2 y 4.

***“ First, solve the problem. Then, write the code. ” - John Johnson –***

- 1) Hacer un programa que esté leyendo números hasta que se introduzca el valor -1. Escribir a continuación cuántos de esos valores tienen una cifra, cuántos tienen dos cifras y cuántos más de tres cifras. En la salida controlar las palabras que irán en singular o en plural.

Ejemplo de ejecución del programa:

```
Introduce un número entero, teclea el valor -1 para finalizar el programa: 2
Introduce un número entero, teclea el valor -1 para finalizar el programa: 45
Introduce un número entero, teclea el valor -1 para finalizar el programa: 3
Introduce un número entero, teclea el valor -1 para finalizar el programa: 1112
Introduce un número entero, teclea el valor -1 para finalizar el programa: 29
Introduce un número entero, teclea el valor -1 para finalizar el programa: 54
Introduce un número entero, teclea el valor -1 para finalizar el programa: -1
```

```
Has tecleado 2 números de una cifra
Has tecleado 3 números de dos cifras
Has tecleado 1 número de más de tres cifras
```

- 2) Escribir un programa que pida por teclado un número entero n, positivo y comprendido entre 3 y 40, ambos incluidos. Hay que obligar al usuario a que introduzca un valor correcto, visualizando un mensaje de error en caso contrario y volviendo a pedir el número, hasta que este sea válido. Escribir en pantalla una figura formada por los caracteres 'A' y '.' con el formato siguiente, siendo n la longitud de la línea más larga (la central).

```
...A
..AA
.AAA      Figura para n=4
AAAA
..AA
...A
```

Ejemplo de ejecución del programa:

```
Introduce un número entero comprendido entre 3 y 40: 2
Número fuera de rango, introduce otro: 4
```

```
...A
..AA
.AAA
AAAA
..AA
...A
```



- 3) Se dice que un número natural es deficiente cuando la suma de sus divisores propios es menor que el propio número. Por ejemplo, 16 es un número deficiente ya que sus divisores propios son 1, 2, 4 y 8 y se cumple que  $1+2+4+8=15$ , que es menor que 16.

Hacer un programa que indique si un número leído por teclado es o no deficiente.

- 4) Hacer un programa que lea un número por teclado y que escriba en orden descendente todos los valores comprendidos entre el número leído y 4 que sean divisibles por 4. Separar por comas los números escritos, sin escribir una coma al final.
- 5) Ramón se pasa el día de Nochevieja contando los minutos que faltan para que den las uvas. Ayúdalo haciendo un programa, que a partir de una hora escriba los minutos que faltan hasta las campanadas. La hora de entrada al programa está formada por las horas y los minutos separados por ':' y utilizando siempre dos dígitos (leemos un String). Se utiliza una representación en formato 24 horas, es decir desde 00:00 a 23:59.

Ejemplo de ejecución del programa:

```
Introduce la hora actual: 21:30  
Faltan 150 minutos hasta las campanadas
```

NOTA:

El método `parseInt` de la clase `Integer` convierte un `String` a un valor `int`. La siguiente sentencia de escritura escribe 26.

```
String a = "13";  
System.out.println(Integer.parseInt(a) * 2);
```

#### Puntuación de cada ejercicio:

- Ejercicio 1: 2 puntos  
Ejercicio 2: 3 puntos  
Ejercicio 3: 1.5 puntos  
Ejercicio 4: 1.5 puntos  
Ejercicio 5: 2 puntos

#### Criterios de corrección y cómo entregar el examen:

Aunque un programa no funcione, entregarlo igual, ya que puede ocurrir que esté bien planteado y falle por un pequeño detalle. De la misma manera, si se pide hacer un programa con una determinada sentencia para implementar un bucle y no os sale, hacerlo de otra manera. Evidentemente no puntuará lo mismo, pero al menos podéis completar el ejercicio.

Hay que subir la solución del examen como tarea, en formato de procesador de texto en un único documento. El documento tiene que incluir el código fuente completo de cada uno de los ejercicios. Nombrar al documento con vuestro apellido y nombre.



## AYUDA PARA EL EXAMEN

A la hora de programar, netbeans nos ayuda a escribir código de manera más rápida mediante decenas de atajos (shortcuts). NetBeans incluye un archivo pdf con la lista completa de dichos atajos. Esta lista se encuentra dentro del directorio de instalación de NetBeans en una carpeta llamada "nb". Allí encontraremos 2 pdfs:

shortcuts.pdf

shortcuts\_mac.pdf

El primero contiene los atajos para PC (Windows/Linux), el segundo para los usuarios de Mac.

Acordaros dónde está el fichero shortcuts.pdf para poder consultarlo el día del examen. Os recomiendo que os familiaricéis con el uso de estos atajos. De todas maneras, os pongo a continuación algunos de los atajos más útiles:

1. `sout + Tab`, genera la línea: `System.out.println("");`
2. `psvm` genera un método main

```
public static void main(String[] args) {  
  
}
```

3. `sw + Tab`, genera una sentencia switch

```
switch (var) {  
    case val:  
  
        break;  
    default:  
        throw new AssertionError();  
}
```

4. `for`, `while`, `if` + `ctrl+spacio` dan sugerencias para construir cualquiera de estas estructuras de control

Tampoco es necesario saberse de memoria todos los métodos de cada una de las clases que tiene el lenguaje. Para eso podemos consultar la API del lenguaje y los IDEs nos ayudan. Si no recordamos como se calcula el valor absoluto de un valor, simplemente escribiendo `Math`. (clase utilizada en los apuntes), nos aparecen todos los métodos de la clase, con lo que podemos localizar el que nos interesa utilizar. De la misma manera, si después de una variable de tipo `String` pongo el punto también me aparecen todos los métodos (extraer un carácter, convertir a mayúsculas.....).



Math.

<b>E</b>	double
<b>PI</b>	double
<b>IEEEremainder</b> (double d, double dl)	double
<b>abs</b> (double d)	double
<b>abs</b> (float f)	float
<b>abs</b> (int i)	int
<b>abs</b> (long l)	long
<b>acos</b> (double d)	double
<b>asin</b> (double d)	double
<b>atan</b> (double d)	double
<b>atan2</b> (double d, double dl)	double
<b>cbrt</b> (double d)	double
<b>ceil</b> (double d)	double
<b>cos</b> (double d)	double

String nombre;

nombre.

<b>charAt</b> (int i)	char
<b>codePointAt</b> (int i)	int
<b>codePointBefore</b> (int i)	int
<b>codePointCount</b> (int i, int il)	int
<b>compareTo</b> (String string)	int
<b>compareToIgnoreCase</b> (String string)	int
<b>concat</b> (String string)	String
<b>contains</b> (CharSequence cs)	boolean
<b>contentEquals</b> (CharSequence cs)	boolean

### Plantilla de un programa que lee un valor por teclado.

```
import java.util.Scanner;
```

```
public class Plantilla {
```

```
    public static void main(String[] args) {
```

```
        Scanner teclado = new Scanner(System.in);
```

```
        int numero;
```

```
        System.out.println("Introduce un primer número:");
```

```
        numero = teclado.nextInt();
```

```
    }
```

```
}
```