

DAM/DAW Programación

UNIDAD 4.1

ESTRUCTURAS DE CONTROL I

1. SENTENCIAS CONDICIONALES

Escribe el PSEUDOCÓDIGO, dibuja su DIAGRAMA DE FLUJO e implementa en JAVA de los siguientes supuestos

1.1. Diseña un programa que pida dos números por teclado, determine si el primero es múltiplo del segundo y muestre el resultado de la siguiente forma:

- <num1> es múltiplo de <num2>, en caso de que sea múltiplo
- <num1> no es múltiplo de <num2>, en caso de que no lo sea

1.2. Haz un programa que solicite al usuario la entrada de un número entero y diga si es par o impar. El programa mostrará alguno de los siguientes mensajes:

- 0 es un número par, si el valor introducido es 0
- <valor introducido> es un número par, si el valor introducido es par
- <valor introducido> es un número impar, si el valor introducido es impar

PISTAS:

- Un número es par si el resto de la división entera entre 2 es 0

1.3. Haz un programa que acepte 3 números introducidos por el usuario y:

- muestre el mensaje "3 iguales" si los tres números son iguales
- muestre el mayor de los 3 en caso de que no sean iguales

1.4. Haz un programa para resolver ecuaciones de segundo grado de la forma:

$$a \cdot x^2 + b \cdot x + c = 0, \text{ donde las soluciones son } x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

El programa pedirá al usuario los coeficientes a, b y c y mostrará, o bien las soluciones con dos decimales de precisión, o un mensaje informando al usuario del error correspondiente

Debes tener en cuenta que:

- Los coeficientes a y b no pueden ser 0 simultáneamente
- Si el coeficiente a es 0, la resolveremos como una ecuación de 1er grado
- Si el discriminante ($b^2 - 4 \cdot a \cdot c$) es negativo, no tiene soluciones reales
- Si el discriminante es 0, sólo tiene una solución

1.5. Haz un programa para calcular la letra del DNI. Solicitará al usuario la introducción del número de DNI (se comprobará que es de 8 dígitos)

PISTA:

- El cálculo se hace de la forma siguiente: dividimos el número del DNI entre 23 y nos quedamos con el resto, que estará entre 0 y 22. Ese valor lo usaremos para obtener la letra a partir de la siguiente tabla:

RESTO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

- Las clases “envoltorio” (*wrapper*) de los tipos numéricos primitivos (Integer, Short,...) disponen de métodos que nos permiten convertir las cadenas de caracteres en números

1.6. En una granja se compra diariamente una cantidad de alimento, **comidaDiaria**, para alimentar a los animales. El número de animales de la granja, todos de la misma especie, es **numAnimales** y, cada animal, necesita comer diariamente la cantidad **kilosPorAnimal**.

El programa solicitará al usuario la entrada de los valores anteriores y, en función de ellos, determinará según el caso:

- el **excedente** o kilos de comida sobrante
- la **ración** por animal, en caso de que no haya comida suficiente

2. BUCLES

Escribe el PSEUDOCÓDIGO, dibuja su DIAGRAMA DE FLUJO e implementa en JAVA de los siguientes supuestos

2.1. Escribe un programa que pida al usuario un número entre 0 y 9 y muestre su tabla de multiplicar.

2.2. Haz un programa que muestre la tabla de multiplicar de 0 a 10. Su salida será:

```
*****
* TABLA DE MULTIPLICAR *
*****

-----
TABLA DEL 0
-----
0 x 0 = 0
0 x 1 = 0
...
-----
TABLA DEL 10
-----
10 x 0 = 0
10 x 1 = 1
...
10 x 10 = 100
```

2.3. Diseña un programa que solicite al usuario un número entero positivo y determine si es un número primo o no.

2.4. Modifica el programa para que devuelva todos los números primos entre 1 y el número introducido por el usuario

2.5. Escribe un programa que solicite al usuario la introducción de valores numéricos (uno por línea) hasta que se introduzca el valor "00". A continuación, se mostrará la suma, el mayor y el menor de los valores introducidos

2.6. Diseña un programa que, en primer lugar, le pida al usuario que escriba un carácter cualquiera (letra, dígito,...). A continuación, aceptará entradas del usuario hasta que éste escriba la secuencia "<>". Por último, mostrará cuántas ocurrencias del carácter se produjeron en el texto introducido hasta la aparición de dicha secuencia.

```
Introduce un carácter: a
Escribe lo que quieras (<> para Finalizar)
Hola, qué tal?
Yo bien, y tú?
baaahhh... así así
Pues nada, tómatelo con calma!<>
Apariciones de 'a': 12
```

2.7. Si te fijas en la codificación ASCII de las letras, se puede observar cómo la diferencia entre una letra mayúscula y su correspondiente minúscula, está en el bit 6

A	→	65 ₁₀	→	0b1000001
a	→	97 ₁₀	→	0b1100001
B	→	66 ₁₀	→	0b1000010
b	→	98 ₁₀	→	0b1100010
C	→	67 ₁₀	→	0b1000011
c	→	99 ₁₀	→	0b1100011

Haz un programa que acepte un texto por teclado y, usando las operaciones de nivel de bit, convierta cada carácter al contrario. El cambio sólo debería afectar a los caracteres alfabéticos

2.8. Diseña un programa que nos diga si una cadena es palíndromo o no. Una cadena es palíndromo si se lee igual de izquierda a derecha que de derecha a izquierda. Por ejemplo, las cadenas "ana", "Ana" y "Oí lo de mamá: me dolió" son palíndromos. Ten en cuenta que en la frase podrá haber diferentes signos de puntuación y espacios que deben ser descartados. El programa no debería distinguir tampoco entre una vocal con tilde o sin ella ni mayúsculas de minúsculas.

PISTA:

Recuerda que la clase String dispone de métodos para hacer búsquedas, reemplazos y diferentes comprobaciones sobre el contenido de las cadenas de caracteres