

Autoevaluación UT123

1. Indique cuáles de los siguientes identificadores son válidos en Java. Si el identificador no es válido explique por qué no lo es.

1) registrol	2) lregistro	3) archivo_3
4) while	5) \$impuesto	6) año
7) primer apellido	8) primer_apellido	9) primerApellido
10) primer-apellido	11) Tom's	12) C3PO
13) 123#	14) PesoMáximo	15) %descuento
16) Weight	17) \$\$precioMínimo	18) _\$Único
19) tamaño_máximo	20) peso.maximo	21) Precio_____
22) matrícula?	23) cuántoVale	24) high
25) barça	26) piragüista	27) B_011
28) X012AB	29) 70libro	30) else

2. Cree un programa que solicite al usuario tres distancias. La primera estará medida en milímetros, la segunda en centímetros y la tercera en metros. El programa mostrará en centímetros la suma total de las tres medidas.
3. Una empresa que gestiona un parque acuático necesita una aplicación que les ayude a calcular el importe que hay que cobrar en taquilla por la compra de una serie de entradas, teniendo en cuenta que existen dos tipos de entradas: infantiles y de adulto. Las primeras cuestan 15,50 euros y las segundas 20 euros. La cantidad de entradas de cada tipo será introducido por el usuario. En caso de que el importe total sea igual o superior a 100 euros, se aplicará un bono descuento del 5%. EL programa debe imprimir el coste total antes de aplicar el descuento y el coste tras aplicar el descuento si procede.
4. Diseñe una clase CuentaCorriente que almacene el dni, el nombre del titular y el saldo de una cuenta corriente. Las operaciones que debe soportar la clase serán:
 - Crear una cuenta a partir de un dni y un nombre (el saldo será cero).
 - Sacar dinero . Esta operación debe verificar si hay fondos suficientes por lo que retornará como resultado un valor booleano (true si la operación se ha podido realizar y false en caso contrario).
 - Ingresar dinero.
 - Al imprimir un objeto de la clase este se debe mostrar con el formato siguiente: "La cuenta de nombre cuyo dni es xxxxxx tiene actualmente x euros de saldo". Construya un programa principal que utilice todas las operaciones de la clase para verificar su funcionamiento.
5. Los códigos de barras son una técnica que permite representar números (y, en menos ocasiones, letras) mediante una serie de líneas verticales paralelas con diferentes grosores y separaciones entre ellas. Aunque existen diferentes estándares de codificación, el European Article Number (EAN) resulta el más extendido. De este estándar hay dos formatos: EAN-8 y EAN 13, que codifican respectivamente 8 y 13 caracteres.

Desarrolle un programa que lea un número entero correspondiente a un código de barras codificado en EAN-8 y determine si el código de barras es correcto o no. Para determinar si un código de barras EAN-8 es correcto se calcula el código de control numérico que debería coincidir con el último dígito del código. Este último dígito se utiliza para detección de errores, y se calcula a partir de los demás. Para eso, empezando *por la derecha* (sin contar el dígito de control que se está calculando), se suman los dígitos individuales multiplicados por un factor. Los dígitos en posiciones impares (empezando a contar por la derecha saltándonos el de control) se multiplican por 3 y los dígitos en posiciones pares se multiplican por 1.

Por ejemplo, para el código EAN-8 de la figura, la operación a realizar debe ser: $2 \cdot 3 + 5 \cdot 1 + 9 \cdot 3 + 3 \cdot 1 + 8 \cdot 3 + 5 \cdot 1 + 6 \cdot 3 = 88$



El dígito de comprobación es el número que hay que sumar al resultado anterior para llegar a un valor múltiplo de 10. En el ejemplo de EAN-8, para llegar al múltiplo de 10 más cercano por encima del número 88 hay que sumar 2 (y llegar al 90). Tenga en cuenta que si la suma resulta ser ya múltiplo de 10, el dígito de control será 0.

Casos de prueba: 65839522 (Correcto). 65839521 (Incorrecto).

6. Los radares de tramo consisten en colocar dos cámaras en dos puntos alejados de una carretera para poder comprobar cuánto tiempo ha tardado el coche en recorrer ese tramo. Si la velocidad media supera la velocidad máxima permitida significa que en algún punto del trayecto se ha superado necesariamente esa velocidad. Por ejemplo, si colocamos las cámaras separadas 10Km en un tramo cuya velocidad está limitada a 110Km/h y un coche tarda 5 minutos en ser visto por la segunda cámara, sabremos que su velocidad media ha sido de 120Km/h, y por tanto en algún sitio ha superado el límite de velocidad aunque al pasar por debajo de las dos cámaras el coche fuera a 80Km/h.

Desarrolle un programa que reciba tres números: el primero será la distancia (en metros) que separan las dos cámaras, el segundo indicará la velocidad máxima permitida en todo ese tramo (en Km/h) y el tercer y último número indicará el número de segundos que ha tardado un coche en recorrer el tramo. Todos esos números serán enteros.

El programa generará una línea indicando si el coche debe ser multado o no. En concreto, indicará "OK" si el coche no superó la velocidad máxima, indicará "MULTA" si se superó esa velocidad pero en menos de un 20% de la velocidad máxima permitida, y "PUNTOS" si la velocidad fue superada en un 20% o más de esa velocidad pues en ese caso, además de la multa se le quitarán puntos del carnet. El sistema de radar puede fallar y registrar entradas incorrectas (por ejemplo, indicando que el tiempo que ha tardado el coche es negativo). En esos casos, el sistema mostrará la cadena "ERROR".

Casos de prueba: 9165 110 300 (OK). 9165 110 299 (MULTA). 12000 100 433 (OK). 12000 100 431 (MULTA). 12000 100 359 (PUNTOS). -1000 -50 -100 (ERROR).
