

INSTRUCCIONES:

El examen se desarrollará a partir del proyecto base facilitado en la plataforma, para facilitar su realización y corrección. Una vez realizado se subirá en el buzón habilitado al efecto, como cualquier otra tarea ..

- **Puntuación: 10 puntos en total.** (3 ejercicios. La puntuación de cada uno se indica al comienzo)
- El práctico supone un 100%. No hay teórico.
- **La nota de este examen, para hacer media con la nota del 2º cuatrimestre para el cálculo de la nota de “exámenes presenciales”, debe ser como mínimo de 3,5 puntos.**

EXAMEN:

[2,5 puntos] Ejercicio 1).- Crea un programa en Java para leer siete números reales y comprobar que estén en el intervalo [-100, -50] o bien en el [60, 200]

- El programa debe iterar mediante un bucle.
- Se deben controlar las excepciones ante posibles lecturas de datos no válidos como caracteres alfanuméricos en vez de números.
- Si el número real leído está en uno de los dos rangos permitidos, se contabilizará como tal, se escribirán el total de números contabilizados en los intervalos hasta ahora y se escribirá por consola con dos decimales, tal y como se puede ver más abajo.
- Tras salir del bucle se escribirá por consola el total de números válidos (que estén en uno de los rangos permitidos) leídos.

Ejemplo de ejecución:

```
Leer 7 números reales y comprobar que estén en el intervalo [-100, -50] o bien en el [60, 200]
-----
Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
10
Incorrecto. Número descartado. Total de números reales válidos hasta ahora: 0

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
x
Número no válido.
Incorrecto. Número descartado. Total de números reales válidos hasta ahora: 0

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
60
Correcto. Total de números reales válidos hasta ahora: 1
El último número que acabamos de leer es el: 60,00

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
-50
Correcto. Total de números reales válidos hasta ahora: 2
El último número que acabamos de leer es el: -50,00

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
-101
Incorrecto. Número descartado. Total de números reales válidos hasta ahora: 2

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
201
Incorrecto. Número descartado. Total de números reales válidos hasta ahora: 2

Introduzca un número real que esté en el intervalo [-100, -50] o bien en el [60, 200]
200
Correcto. Total de números reales válidos hasta ahora: 3
El último número que acabamos de leer es el: 200,00

Total de números válidos leídos: 3
```



[4 puntos] Ejercicio 2).- Dado el siguiente programa que ya tiene implementada la entrada de datos y la salida de resultados, escribir el código necesario en Java para que se procese el array de entrada con números de serie y se rellene un array de salida indicando para cada número de serie si es válido o no (cadena “válido” o “inválido”).

Los requisitos que debe cumplir un número de serie para ser válido son:

1. Debe comenzar por dos letras mayúsculas. La primera obligatoriamente tiene que ser X o Y. La segunda puede ser cualquiera (sin contar la ñe).
2. A continuación habrá un guión ('-').
3. Ha de terminar por un año que debe encontrarse entre 1970 y el año actual.
4. Si el año es impar, la primera letra del número de serie debe ser 'Y'. Si el año es par, la primera letra debe ser 'X'.

El resultado que debe proporcionar el programa es el siguiente:

VALIDADOR DE NÚMEROS DE SERIE

Lista de números de serie de prueba:

[ZA-2000, XAZ2000, XA2000, XA-1969, YH-1969, XQ-1970, XJ-2002, YV-2021, XB-2022, YV-2022, YA-1970, YH-2002, XB-2021]

RESULTADO

El resultado de la comprobación de la validez de los números de serie es:

[inválido, inválido, inválido, inválido, inválido, inválido, inválido, inválido, inválido, inválido, válido, válido, válido]

Tan solo hay que implementar la parte del procesamiento así como declarar y utilizar todas las variables auxiliares que consideres oportunas. La entrada de datos y la salida de resultados están ya hechas y no hay que tocar nada.

En el procesamiento que implementes podrás hacer uso de **expresiones regulares** para facilitarte el trabajo (es lo más recomendable), aunque habrá comprobaciones que no podrás realizar mediante expresiones regulares y que tendrás que hacer “a mano” mediante código.

¡Y no olvides reservar el espacio apropiado para el array de salida procurando no usar un literal! (El tamaño debe ser el mismo que del array de entrada).



[3,5 puntos] Ejercicio 3).- Enunciado gestión de stock de bombillas:

Recordarás el ejercicio de la tarea de la unidad 1 en el que te pedíamos que hicieses un programa para gestionar la colocación de bombillas por estanterías en una ferretería. El programa comenzaba pidiendo la cantidad de bombillas a colocar para terminar imprimiendo la cantidad de estanterías completas necesarias y la cantidad de huecos libres en la última estantería no completa. Puedes recordar el enunciado completo en el siguiente enlace: [Tarea Unidad 01](#)

En esta ocasión no te pedimos que calcules la colocación de bombillas en las estanterías, sino que escribas un programa para gestionar las existencias (*stock*) de bombillas de la ferretería utilizando programación dirigida a objetos. Es decir, creando una clase que modele el proceso de gestión de venta y reposición de bombillas en las estanterías de la ferretería.

- Definición de *stock* según la RAE: <https://dle.rae.es/stock>
- Existencias o *stock*: <https://www.fundeu.es/recomendacion/stock-existencias-mercancias-provisiones/>

Nombre de la clase: ExistenciasDeBombillas

Método constructor: ExistenciasDeBombillas()

//Instancia un objeto de la clase ExistenciasDeBombillas e inicializa las existencias a 0.

Métodos de clase:

void reponer(int bombillas)

//Incrementa las existencias.

boolean vender(int bombillas)

//Si es posible vender la cantidad de bombillas indicada se restan de las existencias actuales y se devuelve true, en caso contrario se devuelve false.

int existenciasActuales()

//Devuelve las existencias de bombillas que hay actualmente.

String toString()

//Devuelve una cadena con el texto "Las existencias de bombillas son: <stockActual>"

Para facilitar tu labor y que puedas concentrarte en la codificación de la clase diseñada, hemos escrito el flujo de programa principal que hace uso de la clase, tu misión consiste en añadir la codificación de la clase diseñada.

NO HAY QUE REALIZAR EL JAVADOC DE LA CLASE, ÚNICAMENTE IMPLEMENTAR ÉSTA.

A continuación mostramos dos ejemplos de salida de ejecución del programa de prueba una vez implementada la clase solicitada:

Primer Ejemplo:

```
¿Cuántas bombillas se van a reponer?: 100
EMPIEZA LA VENTA DE BOMBILLAS TRAS HABERSE REPUESTO LA CANTIDAD DE BOMBILLAS EN EL
ALMACÉN
En el almacén hay ahora 100 bombillas
¿Cuántas bombillas se van a vender?: 50
Se vendieron 50 bombillas
En el almacén hay ahora 50 bombillas
¿(T)erminar o (N)ueva venta?: n
En el almacén hay ahora 50 bombillas
¿Cuántas bombillas se van a vender?: 25
Se vendieron 25 bombillas
En el almacén hay ahora 25 bombillas
¿(T)erminar o (N)ueva venta?: t
En el almacén hay ahora: 25 bombillas
```



Módulo: Programación

I.E.S.: Aguadulce

C.F.G.S. : Desarrollo de aplicaciones web (DAW)

Fecha: martes 2 de febrero de 2021

Hora de Comienzo: 18:45

Examen: Práctico

Duración: 2 horas y 45 minutos

Nombre:

D.N.I.:

Centro en el que se realiza el examen:

Segundo Ejemplo:

```
¿Cuántas bombillas se van a reponer?: 100
EMPIEZA LA VENTA DE BOMBILLAS TRAS HABERSE REPUESTO LA CANTIDAD DE BOMBILLAS EN EL
ALMACÉN
En el almacén hay ahora 100 bombillas
¿Cuántas bombillas se van a vender?: 50
Se vendieron 50 bombillas
En el almacén hay ahora 50 bombillas
¿(T)erminar o (N)ueva venta?: n
En el almacén hay ahora 50 bombillas
¿Cuántas bombillas se van a vender?: 25
Se vendieron 25 bombillas
En el almacén hay ahora 25 bombillas
¿(T)erminar o (N)ueva venta?: n
En el almacén hay ahora 25 bombillas
¿Cuántas bombillas se van a vender?: 25
Se vendieron 25 bombillas
En el almacén hay ahora 0 bombillas
¿(T)erminar o (N)ueva venta?: n
En el almacén hay ahora 0 bombillas
¿Cuántas bombillas se van a vender?: 5
Imposible vender, no hay suficientes bombillas en el almacén.
En el almacén hay ahora 0 bombillas
¿(T)erminar o (N)ueva venta?: t
En el almacén hay ahora: 0 bombillas
```

