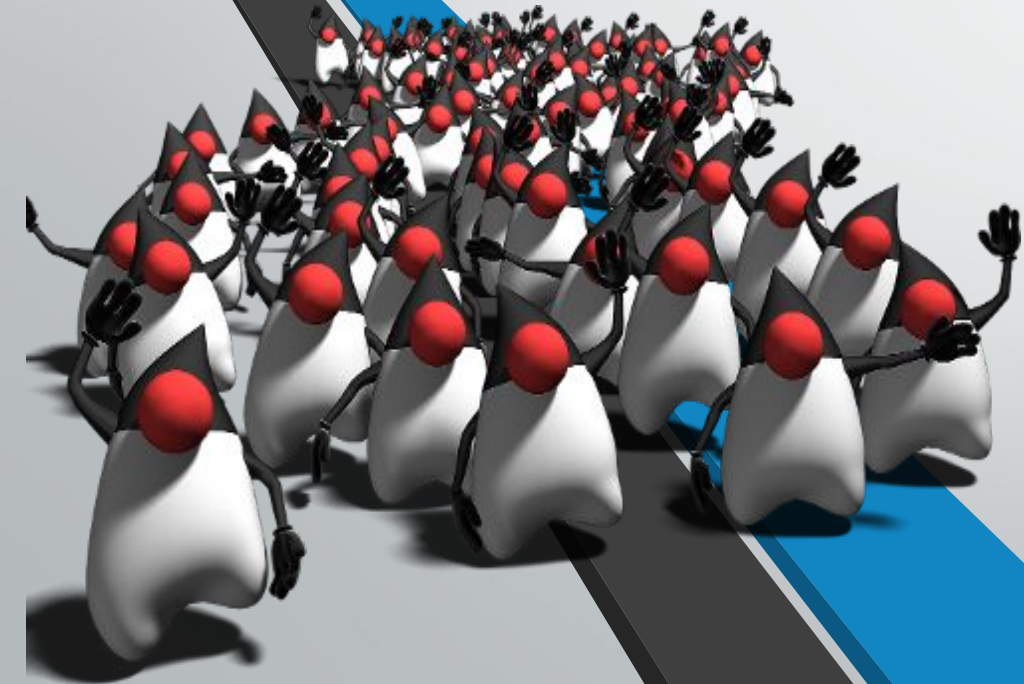


JAVA

TEMA 06:

TAREAS DE PROFUNDIZACIÓN Y/O AMPLIACIÓN



EJERCICIOS

- **Ejercicio 01.- (OPTATIVO)** Realizar un programa en Java que rellene un vector de 6 celdas con números comprendidos entre el 1 y el 10. Los números se le pedirán al usuario. Controla, además, que el usuario introduzca una letra (en cuyo caso, habrá que volver a pedir el número).
- Una vez relleno el array, se mostrará por pantalla.

EJERCICIOS

- **Ejercicio 02.- (OPTATIVO)** Realizar un programa en Java que rellene un vector de 6 celdas con números positivos.
- Una vez introducido el número, se le pedirá al usuario la posición del vector donde insertarlo. Si la posición está ya rellena, le volveremos a pedir la posición hasta que podamos introducirlo.
- Por último, mostraremos el contenido del vector.

EJERCICIOS

- **Ejercicio 03.- (OPTATIVO)** Realizar un programa en Java donde rellenemos un vector de 10 celdas. Estas celdas se rellenarán con números introducidos por el usuario.
- Posteriormente, pediremos al usuario 3 posiciones. Esas posiciones tendremos que "borrarlas" y arrastrar todo el contenido del vector (para que las celdas vacías sean las 3 últimas).
- Al final, mostraremos el contenido del vector.

EJERCICIOS

- Ejemplo:

[3] [7] [9] [4] [5] [12] [3] [1] [7] [19]

Posición: 2.

[3] [7] [4] [5] [12] [3] [1] [7] [19] [-1]

Posición: 8.

[3] [7] [4] [5] [12] [3] [1] [7] [-1] [-1]

Posición: 4.

[3] [7] [4] [5] [3] [1] [7] [-1] [-1] [-1]

Mostrando el vector: [3] [7] [4] [5] [3] [1] [7] [-1] [-1] [-1]

EJERCICIOS

- **Ejercicio 04.- (OPTATIVO)** Realizar un programa en Java que rellene una matriz de 3x3 con números positivos y negativos. (Los números se le pedirán al usuario).
- Después, le pediremos al usuario un número "X": tendremos que crear un método de tipo booleano que nos diga si hemos encontrado "X" números negativos en la matriz.

EJERCICIOS

- **Ejercicio 05.- (OPTATIVO)** Realizar un programa en Java que rellene una matriz de 4×4 con números positivos y negativos. (Los números se le pedirán al usuario).
- Una vez rellena la matriz, le daremos al usuario la opción de buscar un número para cada fila mientras el usuario desee realizar la búsqueda.

EJERCICIOS

- Ejemplo:
[3] [7] [9] [-4]
[1] [-2] [7] [3]
[-3] [7] [9] [4]

¿Desea buscar un número en todas las filas? *Si*
Número a buscar: 1
El 1 no está en todas las filas.

¿Desea buscar un número en todas las filas? *Si*
Número a buscar: 7
El 7 si está en todas las filas.

¿Desea buscar un número en todas las filas? *Si*
Número a buscar: 3
El 3 no está en todas las filas.

¿Desea buscar un número en todas las filas? *No*

EJERCICIOS

- **Ejercicio 06.- (OPTATIVO)** Crea un array de números de 100 posiciones, que contendrá los números del 1 al 100. Obtén la suma de todos ellos y la media.

EJERCICIOS

- **Ejercicio 07.- (OPTATIVO)** Crea un array de caracteres que contenga de la 'A' a la 'Z' (solo las mayúsculas). Después, ve pidiendo posiciones del array por teclado y si la posición es correcta, se añadirá a una cadena que se mostrara al final, se dejará de insertar cuando se introduzca un -1.
- (Sigue en la siguiente diapositiva)

EJERCICIOS

- Por ejemplo, si escribo los siguientes números
0 //Añadira la 'A'
5 //Añadira la 'F'
25 //Añadira la 'Z'
50 //Error, inserte otro número
-1 //fin
Cadena resultante: AFZ
- Nota: Crea el vector en el momento de declararlo, tal y como se explica en la diapositiva 11 del tema.
- PISTA: Para ir añadiendo letras a nuestro String, debemos hacer lo siguiente: `variableString = variableString + char;`

EJERCICIOS

- **Ejercicio 08.- (OPTATIVO)** Crea un array de números de un tamaño pasado por teclado. El array contendrá números aleatorios entre 1 y 300.
- Mostrar aquellos números que acaben en un dígito que el usuario indique por teclado.

EJERCICIOS

- **Ejercicio 09.- (OPTATIVO)** Crea un método que, dados dos vectores del mismo tamaño (puedes crearlos tú mismo en el main), retorne true si los vectores son exactamente iguales y false en caso contrario.

EJERCICIOS

- **Ejercicio 10.- (OPTATIVO)** Teniendo un vector con valores aleatorios entre 1 y 50, insertar en otro vector de la misma longitud (inicializadas todas sus celdas a -1), aquellos números que sean pares y mayores que 25.
- Después, mostrar el vector de origen y el de destino, sólo los número introducidos.