

Ejercicios prácticos

Unidades: 8

Guía

- Se plantean diferentes ejercicios para realizarlos en java
- Se recomienda utilizar un IDE como Eclipse o Netbeans.
- Puede haber más de una solución en un ejercicio, buscar la solución óptima, aquella que requiera menos líneas de código y no se abuse de estructuras de control.

Ejemplo

Crea un array con un máximo de 10 elementos. Los elementos serán de tipo entero y se generarán de forma aleatoria.

```
public static void main(String[] args) {  
    int[] array = new int[10];  
    for (int i = 0; i < 10; i++) {  
        array[i] = new java.util.Random().nextInt(100);  
    }  
}
```

Ejercicio 1.

Crea un array de enteros que contenga una secuencia numérica del 1 al 100. Muestra por pantalla todos los elementos pares del array.

Ejercicio 2.

Crea un array para almacenar las 10 frutas que más te gusten. Haz una pequeña aplicación con un menú para solicitar la posición de la fruta se quiere mostrar por pantalla.

Ejercicio 3

Crea un array de enteros de 20 elementos. Una vez inicializado, queremos dar la oportunidad al usuario de ampliar los elementos del array y ordenarlos en forma ascendente. El programa debe trabajar sobre una sola array y debe permitir solicitar al usuario cuantos elementos nuevos se van a guardar en el array, mostrar en cualquier momento los valores del array y la posibilidad de ordenar de forma ascendente todos sus elementos. Cada elemento se generará a partir de un número aleatorio de 100. Nota. Para generar un número aleatorio utiliza el objeto *Random* que aparece en el ejemplo de la práctica.

Genera el menú necesario para que el usuario pueda operar tal y como se indican en los requisitos.

Ejercicio 4.

Supongamos que tienes una lista de conexiones entre varios puntos en un mapa. Cada conexión se representa por un par de números (por ejemplo, la conexión entre el punto 2 y el punto 3 se puede representar como 2,3). Crea una matriz de adyacencia para representar estas conexiones.

Por ejemplo la matriz de adyacencia de los puntos {1, 2}, {2, 3}, {3, 1}, {4, 5}, {5, 1} es:

	1	2	3	4	5
1	0	1	1	0	1
2	1	0	1	0	0
3	1	1	0	0	0
4	0	0	0	0	1
5	1	0	0	1	0

Ejercicio 5.

Crea un programa que reciba un string y encuentre la palabra más larga en el string. Ignora los signos de puntuación y distingue entre mayúsculas y minúsculas. En caso de empate se indicará la primera palabra que se encuentre con más caracteres.

Ejercicio 6.

Escribe un programa que verifique si una frase es un palíndromo, ignorando los espacios, la puntuación y las diferencias entre mayúsculas y minúsculas. Un palíndromo es aquella expresión que se lee de izquierda a derecha que de derecha a izquierda.

Ejercicio 7.

A partir de una cadena de texto cualesquiera que contenga dígitos y caracteres, se pide extraer todos los números enteros y realizar la suma de todos ellos. Por ejemplo, de la siguiente cadena "El tigre vive alrededor de 10 años, suele medir 70 cm y puede llegar a comer 40kg de carne" el resultado tiene que ser $(10+70+40)$ 120.

Nota: Se puede utilizar la función *Character.isDigit* para comprobar si un carácter es un dígito o no.

Ejercicio 8.

Realiza un programa que tome una frase e imprima por pantalla cada palabra en una nueva línea, junto con su longitud y su posición inicial y final en la cadena original

Ejercicio 9.

Imagina que tienes que desarrollar un sistema para manejar las reservas de un pequeño hotel.

El hotel tiene tres pisos, cada piso tiene 10 habitaciones y las reservas se pueden hacer para cualquier día del año (asumiremos 365 días).

Para este ejercicio, el hotel se representa como un array tridimensional donde los tres ejes representan el piso, la habitación y el día del año. Cada celda en el array puede ser true (la habitación está reservada para ese día) o false (la habitación está libre).

Deberás implementar la siguiente funcionalidad:

- Realizar una reserva para un cliente en una habitación específica para un día específico, si la habitación está disponible.
- Cancelar una reserva existente.
- Comprobar si una habitación está disponible en un día específico.
- Comprobar qué habitaciones están disponibles en un día específico.
- Mostrar un informe de la ocupación del hotel para un día específico.

Ejercicio 10.

Imagina que tienes una aplicación de música y quieres implementar una funcionalidad para manejar playlists. Cada playlist se representará como una clase Playlist con las siguientes propiedades:

- nombre: El nombre de la playlist.
- canciones: Un array de strings, donde cada string representa el nombre de una canción. El formato de cada string será el siguiente: "nombre canción - duración". Por ejemplo, "Bohemian Rhapsody - 5.55". La cantidad de canciones que puede contener el playlist la establece el usuario cuando se crea.

Deberás implementar los siguientes métodos en la clase Playlist:

- Un método para agregar una nueva canción a la playlist.
- Un método para eliminar una canción de la playlist por su nombre.
- Un método para imprimir todas las canciones en la playlist.
- Una función que reciba el nombre de una canción y devuelva true si la canción está en la playlist, false en caso contrario.
- Una función que devuelva la canción más larga de la playlist.
- Una función que devuelva el total de la duración de todas las canciones en la playlist.