

Ejercicios de Programación orientada a objetos en Java

Programación — DAW

Ricardo Pérez López
IES Doñana

Curso 2024/2025

1. Escribir un programa Java que sume los elementos de un *array* de 5 elementos (de tipo entero) ya inicializado.
2. Escribir un programa Java que solicite al usuario la introducción de 5 números enteros y los almacene en un *array* de tamaño suficiente. Posteriormente deberá mostrar por la salida cada uno de ellos en una línea separada y, a continuación, mostrar su suma.
3. Escribir un programa que salude al usuario cuyo nombre haya sido indicado como primera opción al ejecutar dicho programa desde la línea de órdenes. Por ejemplo, si el programa compilado es `Saluda.class` y se ejecuta como:

```
$ java Saluda Ricardo
```

deberá responder «¡Hola, Ricardo!».

4. Usar el método `Arrays.copyOf` para crear una copia de un *array* de forma que el nuevo *array* tenga $long/2$ elementos más, siendo *long* la longitud del *array* original.
5. Escribir un programa que solicite números al usuario y los vaya introduciendo en un *array* con una longitud inicial de 5. Si se supera la capacidad del *array*, se deberá ampliar usando la solución del ejercicio anterior. Cuando el usuario introduzca un 0, deberá mostrar todos los elementos almacenados en el *array* y finalizar el programa.
6. Escribir un programa que cree un *array* de 5 elementos de tipo `StringBuffer`, luego lo copie creando un nuevo *array* usando `Arrays.copyOf`, modifique una de las cadenas almacenadas a través del primer *array* y finalmente compruebe si se ha modificado en el nuevo *array*. ¿Qué conclusión podemos sacar?

7. Escribir un programa que compare dos *arrays* usando `Arrays.equals`.
8. Escribir un programa donde el método `main` pase un *array* de enteros a un método llamado `cambiar` definido por el programador, el cual modificará uno de los elementos del *array* (pero no devolverá nada, es decir, debe definirse de tipo `void`). Finalmente, se deberá comprobar en el método `main` si el *array* ha sido modificado tras la llamada a `cambiar`. ¿Cambia en algo el resultado si el *array* se declara como `final` en el método `main`? ¿Cambia en algo el resultado si el parámetro del método `cambiar` se declara como `final`? ¿Por qué? (Pista: recordar el concepto de mutabilidad).
9. Escribir un programa que represente la matriz:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

usando un *array* bidimensional inicializado a los valores adecuados. A continuación, debe mostrar la matriz por la salida recorriendo el *array* usando dos bucles anidados.

10. Crear un método `rellenaCuadrado` que reciba un entero n y que devuelva una matriz cuadrada de $n \times n$ que sólo contenga al entero n .

Ejemplos:

```
jshell> rellenaCuadrado(3)
$1 ==> [
    [3, 3, 3],
    [3, 3, 3],
    [3, 3, 3]
]

jshell> rellenaCuadrado(5)
$2 ==> [
    [5, 5, 5, 5, 5],
    [5, 5, 5, 5, 5],
    [5, 5, 5, 5, 5],
    [5, 5, 5, 5, 5],
    [5, 5, 5, 5, 5]
]

jshell> rellenaCuadrado(1)
$3 ==> [
    [1]
]
```

```
jshell> rellenaCuadrado(0)
$4 ==> []
```

Notas:

- $n \geq 0$.
 - Si $n = 0$, devuelve un *array* vacío.
11. Crear un programa que cree un *array* de cinco elementos, donde el primero sea un entero, el segundo una cadena, el tercero un booleano, el cuarto un valor `null` y el quinto un `StringBuilder`. Luego debe enviar ese *array* a un método estático que recorrerá el *array* desde el primer elemento hasta el último y los mostrará por la salida.