

Expresiones lambda en Java

Java 8 introduce varias características nuevas del lenguaje diseñadas para facilitar la escritura de dichos bloques de código; la característica clave son las expresiones *lambda*, también conocidas coloquialmente como cierres o métodos anónimos. Una expresión *lambda* es simplemente una forma más breve de escribir una implementación de un método para su posterior ejecución.

1. Suma dos números enteros usando la expresión *lambda*

Escriba un programa Java para implementar una expresión *lambda* para encontrar la suma de dos números enteros.

2. Verifique si una cadena está vacía usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para verificar si una cadena determinada está vacía.

3. Convierta cadenas a mayúsculas/minúsculas usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para convertir una lista de cadenas a mayúsculas y minúsculas.

4. Filtrar números pares e impares con *lambda*

Escriba un programa Java para implementar una expresión *lambda* para filtrar números pares e impares de una lista de números enteros.

5. Ordene las cadenas alfabéticamente usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para ordenar una lista de cadenas en orden alfabético.

6. Encuentra el promedio de dobles usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para encontrar el promedio de una lista de dobles.

7. Eliminar duplicados de enteros usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para eliminar duplicados de una lista de números enteros.

8. Calcular factorial usando expresión *lambda*.

Escribe una expresión *lambda* para implementar una expresión *lambda* para calcular el factorial de un número determinado.

9. Compruebe si el número es primo usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para crear una expresión *lambda* para verificar si un número es primo.

10. Concatenar dos cadenas usando *lambda*

Escriba un programa Java para implementar una expresión *lambda* para concatenar dos cadenas.

11. Hallar el máximo y el mínimo en una lista usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que permita hallar los valores máximo y mínimo en una lista de enteros.

12. Multiplicar y sumar elementos de una lista usando expresiones *lambda*

Escribe un programa en Java para crear una expresión *lambda* que multiplique y sume todos los elementos de una lista de enteros.

13. Contar palabras en una oración con una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que cuente las palabras en una oración.

14. Comprobar si una cadena es un palíndromo usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que compruebe si una cadena dada es un palíndromo.

15. Suma de cuadrados de números pares e impares usando *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que calcule la suma de los cuadrados de todos los números pares e impares de una lista.

16. Comprobar si una palabra existe en una lista usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que compruebe si una lista de cadenas contiene una palabra específica.

17. Encontrar la cadena más larga y la más corta usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que permita encontrar la longitud de la cadena más larga y la más corta en una lista.

18. Comprobar si un número es un cuadrado perfecto usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que permita comprobar si un número dado es un cuadrado perfecto.

Nota: Un cuadrado perfecto es un elemento de estructura algebraica cuyo cuadrado es igual al de otro elemento.

19. Encontrar el segundo elemento más grande / más pequeño en un arreglo usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que permita encontrar el segundo elemento más grande y el segundo más pequeño en un arreglo.

20. Ordenar objetos por atributo usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que ordene una lista de objetos según un atributo específico.

21. Suma de todos los números primos en un rango usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que calcule la suma de todos los números primos en un rango dado.

Nota: Un número primo es un número natural mayor que 1 que no es producto de dos números naturales menores. Un número natural mayor que 1 que no es primo se denomina número compuesto. Por ejemplo, 5 es primo porque las únicas formas de escribirlo como producto (1×5 o 5×1) incluyen el 5. Sin embargo, 4 es compuesto porque es un producto (2×2) en el que ambos números son menores que 4. Los números primos son fundamentales en la teoría de números debido al teorema fundamental de la aritmética: todo número natural mayor que 1 es primo o puede factorizarse como producto de primos, siendo este único salvo por su orden.

22. Comprobar mayúsculas y minúsculas en una lista usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que compruebe si una lista de cadenas está compuesta solo por mayúsculas, solo por minúsculas o si contiene una mezcla de ambas.

23. Calcular la longitud promedio de una cadena usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que calcule la longitud promedio de las cadenas en una lista.

24. Hallar el mayor factor primo usando una expresión *lambda*

Escribe un programa en Java para implementar una expresión *lambda* que permita hallar el mayor factor primo de un número dado.

25. Convertir un entero a binario usando una expresión *lambda*

Escriba un programa en Java para implementar una expresión *lambda* que convierta un entero a su representación binaria.