

methods

✗ PARAMETROS

✗ RETORNO

0. Define un método que imprima "hola mundo". Llama a este método desde el *main*.
1. Define un método que imprima "hello world". Invócalo desde el *main*.
2. Define un método que invoque los dos métodos anteriores (1 y 2). Invócalo desde el *main*.
3. Analiza los siguientes programas:

a)

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        metodo1();
    }

    static void metodo1(){
        System.out.println(a);
    }
}
```

b)

```
public class Main {
    public static void main(String[] args) {
        metodo1();
        System.out.println(a);
    }

    static void metodo1(){
        int a = 10;
    }
}
```

c)

```
public class Main {
    public static void main(String[] args) {
        metodo1();
    }

    static void metodo1(){
        int a = 10;
        metodo2();
    }

    static void metodo2(){
        System.out.println(a);
    }
}
```

d)

```

public class Main {
    public static void main(String[] args) {
        metodo1();
    }

    static void metodo1() {
        metodo2();
        System.out.println(a);
    }

    static void metodo2() {
        int a = 10;
    }
}

```

✗ PARAMETROS
 ✓ RETORNO

4. Haz un método que retorne el texto **"hola mundo"**. Invoca el método desde el main e imprime el resultado.
5. Haz un método que invoque al método anterior (4) y añada unas exclamaciones al resultado. Retorna el resultado. Invoca este método desde el main e imprime el resultado.
6. Haz un método que retorne un número aleatorio entre 0 y 10. Invócalo desde el main e imprime el resultado.
7. Haz un método que pida un número por teclado y lo retorne. Invócalo desde el main e imprime el resultado.
8. Define un método que retorne el valor aproximado de Pi ~ 3.14159
9. Implementa un método que retorne un texto con el día de la semana actual (**"lunes"**, **"martes"**,...). Utiliza para ello la llamada a `LocalDate.now().getDayOfWeek().getValue()` que devuelve un entero indicando el día (1 ⇒ lunes, 2 ⇒ martes, ...). Invoca este método desde el main e imprime el resultado.
10. Implementa un método que retorne un número aleatorio entre 1 y 6. Luego crea otro método que lo utilice y retorne el carácter adecuado: `□ □ □ □ □ □`. Imprímelo desde el main.
11. Implementa un método que retorne si estamos en periodo navideño. Puedes obtener el mes y día actual con: `LocalDate.now().getMonth().getValue()` y `LocalDate.now().getDayOfMonth()`. Utiliza este método desde el main para imprimir (o no) un árbol de navidad 🌲
12. Haz un método que retorne un array con las opciones de un menú: **"Crear usuario"**, **"Eliminar usuario"**, **"Ver estadísticas"**, **"Salir"**
13. Haz un método que llame al método anterior e imprima las opciones numeradas. Luego pida una opción al usuario, compruebe que la opción está entre las disponibles, y vuelva a pedir la opción en

caso necesario. Cuando la opción sea válida, la debe retornar. Invoca el método desde el main e imprime un texto diferente dependiendo de la opción.

14. Haz un método que retorne un array de strings con datos del sistema obtenidos a partir de `((com.sun.management.OperatingSystemMXBean) ManagementFactory.getOperatingSystemMXBean()))`. Imprime los datos desde el main

✓ PARAMETROS

✓ RETORNO

15. Escribe un método que reciba dos números decimales y retorne su multiplicación. Utilízalo desde el main

16. Escribe un método que reciba una edad y retorne si es o no mayor de edad. Utilízalo desde el main

17. Escribe un método que reciba dos números y retorne el mínimo. *Utilízalo desde el main* ↺ ↻ ↺ ↻

18. Escribe un método que reciba un número y retorne si es positivo, negativo o cero.

19. Escribe un programa que reciba kilómetros y retorne millas.

19.2. Que escriben los siguientes programas?

1)

```
public class Main {  
  
    static int metodo(int a) {  
        return b*2;  
    }  
  
    public static void main(String[] args) {  
        int b = 4;  
        System.out.println(metodo(b));  
    }  
}
```

2)

```
public class Main {  
  
    static int metodo(int z) {  
        return z*2;  
    }  
  
    public static void main(String[] args) {  
        int a = 4;  
        System.out.println(metodo(a));  
        int b = 7;  
        System.out.println(metodo(b));  
    }  
}
```

3)

```
public class Main {  
  
    static void metodo(int a) {  
        a = 2;  
    }  
  
    public static void main(String[] args) {  
        int a = 4;  
        metodo(a);  
        System.out.println(a);  
    }  
}
```

20. Escribe un programa que pida el ancho y alto de un rectángulo y muestre por pantalla su área y su perímetro. Crea las funciones que consideres necesarias.

21. Implementa un método que reciba dos números y devuelva el máximo de los dos. Utiliza este método para calcular el máximo de tres números.

22. Escribe un programa que reciba un número como parámetro y retorne un array de ese tamaño inicializado con números aleatorios. Utiliza este método desde el main, y muestra el array retornado.

23. Implementa un método que reciba un array de números como parámetro y lo muestre por pantalla.

24. Utiliza el método (23) en el programa (22)

25. Implementa un método que reciba un array de números y un número N ; el método debe retornar otro array con los N primeros números del array recibido. Si el número N es mayor que el tamaño del array, devolverá una copia completa del array original.

26. Implementa un método que reciba un array de números y un número N ; el método debe retornar otro array con los N primeros números del array recibido. Si el número N es mayor que el tamaño del array, completará el nuevo array con ceros.

27. En el main, utiliza el método (26) pasándole un array generado con el método (22) e imprimiéndolo con el método (23)

28. Crea un método llamado `limit` que reciba un array de enteros y un número límite L . El método debe devolver un nuevo array con los valores del original, pero haciendo que todos los valores que sean mayores a L , se establezcan a L .

El método `limit` sería útil para restringir los valores en un array de enteros a un límite superior especificado, devolviendo un nuevo array con los valores ajustados. Es una forma **no destructiva** de aplicar un límite, ya que el array original no se modifica.

29. Programa un método llamado `limit` que reciba un array de enteros y un número límite L . El método debe modificar el array recibido haciendo que los valores mayores a L se establezcan a L .

30. Implementa un método llamado `filter` que reciba un array de enteros, un límite superior L . Este método debe retornar un nuevo array que contenga todos los elementos del array original menores o iguales a L .

31. Implementa un método llamado `filterAndTransform` que reciba un array de enteros, un límite superior L , y un valor multiplicador M . Este método debe retornar un nuevo array que contenga todos los elementos del array original menores o iguales a L , y multiplicados por M .

32. Implementa un método llamado `booleanCount` que reciba un array de booleanos y retorne dos números indicando cuantos son `true` y cuantos son `false`.

33. `StringUtils`. Programa una librería para manipular Strings con los siguientes métodos:

1. `leftPad`: dado un String S y un número L , retorna un String añadiendo espacios a la izquierda de S para que su longitud sea L .

```
StringUtils.leftPad("hola", 7)  = "    hola"
StringUtils.leftPad("hola", 4)  = "hola"
StringUtils.leftPad("hola", 2)  = "hola"
```

2. `leftPad`: dado un número N y un número L , retorna un String añadiendo ceros a la izquierda de N para que su longitud sea L .

```
StringUtils.leftPad(1, 3)  = "001"
StringUtils.leftPad(17, 5) = "00017"
StringUtils.leftPad(1234, 2) = "1234"
```

3. `wordCount`: dado un String S , retorna el número de palabras de S .

```
StringUtils.wordCount("hola que tal") = 3
StringUtils.wordCount("hola")        = 1
StringUtils.wordCount("")             = 0
```

4. `lineCount`: dado un String S , retorna el número de líneas de S .

```
StringUtils.wordCount("hola que\ntal\nestas") = 3
StringUtils.wordCount("hola")                 = 1
StringUtils.wordCount("")                     = 0
```

5. `characterCount`: dado un String S y un carácter C , retorna el número de veces que aparece C en S .

```
StringUtils.characterCount("oye como va", 'o') = 3
StringUtils.characterCount("hola", 'e')         = 0
```

6. `removeCharacter`: dado un String S y un carácter C , retorna un String eliminando el carácter C de S .

```
StringUtils.removeCharacter("oye como va", 'o') = "ye cm va"
```

```
StringUtils.removeCharacter("hola", 'e') = "hola"
```

7. **removeEnd**: Dados un String *S* y un String *E*, retorna un String eliminando *E* de *S*, solo si *E* está al final de *S*, si no, retorna *S*.

```
StringUtils.removeEnd(null, *)          = null
StringUtils.removeEnd("", *)            = ""
StringUtils.removeEnd(*, null)          = *
StringUtils.removeEnd("www.domain.com", ".com.") = "www.domain.com"
StringUtils.removeEnd("www.domain.com", ".com")  = "www.domain"
StringUtils.removeEnd("www.domain.com", "domain") = "www.domain.com"
StringUtils.removeEnd("abc", "")              = "abc"
```

8. **appendIfMissing**: Dados un String *S* y un String *E*, retorna el String *S*, añadiendo *E* al final, si *S* no terminaba con *E*.

```
StringUtils.appendIfMissing(null, null)      = null
StringUtils.appendIfMissing("abc", null)      = "abc"
StringUtils.appendIfMissing("", "xyz")        = "xyz"
StringUtils.appendIfMissing("abc", "xyz")     = "abcxyz"
StringUtils.appendIfMissing("abcxyz", "xyz")  = "abcxyz"
StringUtils.appendIfMissing("abcXYZ", "xyz")  = "abcXYZxyz"
```

9. **abbreviate**: Dados un String *S* y un número *M*, retorna el String *S* con una longitud máxima de *M*, y añadiendo tres puntos en caso de que haya que cortar *S*. Si el número *M* es menor a 3, retorna el string *S* original.

```
StringUtils.abbreviate(null, *)            = null
StringUtils.abbreviate("", 4)              = ""
StringUtils.abbreviate("abcdefg", 6)       = "abc..."
StringUtils.abbreviate("abcdefg", 7)       = "abcdefg"
StringUtils.abbreviate("abcdefg", 8)       = "abcdefg"
StringUtils.abbreviate("abcdefg", 4)       = "a..."
StringUtils.abbreviate("abcdefg", 3)       = "abcdefg"
```

10. **rotate**. Dados un String *S* y un número *N*, retorna el *S* desplazando sus caracteres a izquierda (si *N* > 0) o derecha (si *N* < 0).

```
StringUtils.rotate(null, *)                = null
StringUtils.rotate("", *)                  = ""
StringUtils.rotate("abcdefg", 0)           = "abcdefg"
StringUtils.rotate("abcdefg", 2)           = "fgabcde"
StringUtils.rotate("abcdefg", -2)          = "cdefgab"
StringUtils.rotate("abcdefg", 7)           = "abcdefg"
StringUtils.rotate("abcdefg", -7)          = "abcdefg"
StringUtils.rotate("abcdefg", 9)           = "fgabcde"
StringUtils.rotate("abcdefg", -9)          = "cdefgab"
```