

Prepa Tec: Campus Eugenio Garza Lagüera
Actividad 11: Métodos estáticos parte 2

Sección 1. Codifica los siguientes ejercicios. Prueba cada método utilizando los casos adjuntos. Revisa la plantilla adjunta para obtener una idea de cómo probar cada problema.

Nota: Los primeros dos ejercicios deberán ser entregados a tu profesor durante clase, a mano.

1. Diseña un método booleano llamado `compareLastDigit` que reciba como parámetro dos números positivos enteros. Si ambos números tienen el mismo último dígito, el método retornará `TRUE`, de lo contrario, retornará, `FALSE`.
 - `compareLastDigit(7,17) → true`
 - `compareLastDigit(1,0) → false`
 - `compareLastDigit(3,133) → true`
 - `compareLastDigit(0,0) → true`
 - `compareLastDigit(1234, 98) → false`
2. Diseña un método `String` llamado `frontBack` que reciba como parámetro un `String` cualquiera, y retorne un nuevo `String` con el último y el primer carácter intercambiado.
 - `frontBack("code") → "eodc"`
 - `frontBack("a") → "a"`
 - `frontBack("ba") → "ab"`
 - `frontBack("") → ""`
 - `frontBack("12345") → "52341"`
3. Diseña un método booleano llamado `checkPalindrome` que reciba como parámetro de entrada un `String` y retorne `TRUE` cuando el `String` es palíndromo y `FALSE` cuando no lo sea. Un palíndromo es un texto o frase que se lee igual al derecho y al revés, ignorando espacios.
 - `checkPalindrome("pop") → TRUE`
 - `checkPalindrome("Pop") → FALSE`
 - `checkPalindrome("anita lava la tina") → TRUE`
 - `checkPalindrome("123") → FALSE`
 - `checkPalindrome("rotator") → TRUE`
 - `checkPalindrome("Hello world") → FALSE`
 - `checkPalindrome("") → FALSE`
 - `checkPalindrome("1") → TRUE`
4. Diseña un método con la siguiente firma: `void findAndPrint(String s1, String s2)`. El método deberá buscar los caracteres de `s2` en el texto `s1`. Al encontrar la primera coincidencia, imprimirá en consola el carácter encontrado y terminará la ejecución del método.
 - `findAndPrint("hola", "h") → Prints "h"`
 - `findAndPrint("Hello, world!", "l") → Prints "l"`
 - `findAndPrint("ABCDE", "z") →`
 - `findAndPrint("ABCDE", "a") →`
 - `findAndPrint("", "a") → Prints "Error"`
 - `findAndPrint("", "") → Prints "Error"`

5. ¡RETO! Diseña una versión mejorada `checkPalindromeV2`, con los siguientes cambios: Al revisar si un String es un palíndromo o no:
- Deberá ignorar signos de puntuación (!?-,; ' ")
 - Los caracteres como “ñáéíóú” deberá considerarlos como si fueran “naeiou”.
 - Las minúsculas y mayúsculas son intercambiables. Por ejemplo, “a” y “A” son equivalentes.
- `checkPalindromeV2("ñnN") → TRUE`
 - `checkPalindromeV2("año") → FALSE`
 - `checkPalindromeV2("A man, a plan, a canal--Panama!") → TRUE`
 - `checkPalindromeV2("") → FALSE`
 - `checkPalindromeV2("a") → TRUE`
 - `checkPalindromeV2("Madam, I'm Adam") → TRUE`
 - `checkPalindromeV2("Live on time, emit no evil") → TRUE`
 - `checkPalindromeV2("123") → FALSE`
 - `checkPalindromeV2("Hello World") → FALSE`