

Apellidos: _____

Nombre : _____ Grupo: _____

DNI y firma:

Escriba métodos Java para cada una de las siguientes especificaciones

- ~~1.~~ (0'5 puntos) Escriba un método Java para la siguiente especificación
 - **Nombre:** tiraDado
 - **Parámetro de entrada:** un entero $n > 0$
 - **Objetivo:** escribe en la pantalla un número entero aleatorio entre 1 y n (tira un dado de n caras)
 - **Precondición:** el argumento de entrada es mayor que 0

- ~~2.~~ (1'5 puntos) Escriba un método **RECURSIVO** Java para la siguiente especificación
 - **Nombre:** abecedaria
 - **Parámetro de entrada:** un String
 - **Valor devuelto:** de tipo lógico: cierto si la palabra es abecedaria, falso si no.
 - **Objetivo:** determinar si una palabra es o no "abecedaria"; lo es si las letras que aparecen en la palabra lo hacen en orden alfabético. Por ejemplo, "hijos" es "abecedaria" porque las letras "h", "i", "j", "o" y "s" están en orden alfabético, y también lo es "ellos", mientras que "hija" no lo es.
 - **Precondición:** el argumento de entrada solo contiene letras minúsculas del alfabeto inglés, o es vacía
 - **Observaciones:** considere que la cadena vacía y todas las cadenas de longitud 1 son abecedarias.

- ~~3.~~ (1'5 puntos) Escriba un método Java para la siguiente especificación
 - **Nombre:** comunes
 - **Parámetros de entrada:** dos "arrays" de caracteres
 - **Valor devuelto:** valor entero: número de caracteres comunes en ambos "arrays"
 - **Precondición:** los argumentos de entrada solo contienen letras mayúsculas del alfabeto inglés, y no son vacías
 - **Ejemplo:** "VALLADOLID" y "BARCELONA" tienen 4 letras en común (2 aes, una ele y una o)

- ~~4.~~ (1'5 puntos) Escriba un método Java para la siguiente especificación
 - **Nombre:** arrayHistCol
 - **Parámetros de entrada:** un "array" bidimensional de enteros (`int a [] []`) y un entero `col`
 - **Valor devuelto:** un nuevo "array" unidimensional
 - **Objetivo:** obtener un "histograma a 99" de la columna especificada. Este debe contener, como elemento 0, el número de elementos de la columna `col` del "array" de entrada que sean menores que o iguales a 0, como elemento 1, el número de elementos de la columna cuyo valor sea 1, como elemento 2, el número de elementos de la columna cuyo valor sea 2, ... , como elemento 99, el número de elementos de la columna cuyo valor sea 99 y como elemento 100 el número de elementos de la columna cuyo valor sea 100 o mayor.
 - **Precondición:** `col` es un número de columna válido

- ~~5.~~ (1 punto) Escriba un método Java para la siguiente especificación
 - **Nombre:** arrayHist
 - **Parámetros de entrada:** un "array" bidimensional de enteros (`int a [] []`)
 - **Valor devuelto:** un nuevo "array" unidimensional
 - **Objetivo:** obtener un "histograma a 99" de la matriz completa

- ~~6.~~ (1'5 puntos) Se ha definido en Java la clase C con 2 variables de instancia privadas: un número real (double) r y dos números enteros p y q.
- ~~a.~~ Escriba un constructor sin parámetros que establezca los 3 campos al valor 0
 - ~~b.~~ Escriba un constructor que establezca el valor de los campos al especificado por los parámetros
 - ~~c.~~ Escriba métodos "getter" y "setter" para el campo real
 - d. Escriba un método de la clase C que permita obtener, para un objeto de la clase, la distancia entre los campos enteros de un objeto de la misma.
 - e. Escriba un programa que defina un objeto de la clase con los valores 0.5, 0 y 1 respectivamente
 - f. Ponga un ejemplo de llamada al método del apartado d aplicado al objeto de apartado e

- ~~7.~~ (2 puntos) Se dispone de un fichero de texto "Ejemplo.txt" con la siguiente información:

Una cantidad indeterminada de líneas, cada una de las cuales contiene

<una palabra> <espacio> <un int n> <espacio> <n reales (double), tantos como indica n, separados por espacios>

Diseñe y codifique un programa que escriba en pantalla una línea por cada una del fichero, en la que aparezca la palabra y la media de los n reales de la línea, además de una línea final con el número total de líneas procesadas. Un ejemplo de fichero de entrada sería

Álava 2 10.5 20.3	que produciría la salida:	Álava 15.4
Alicante 3 0 30.2 -12.7		Alicante 5.833333333333333
Albacete 0		Albacete 0.0
Almería 1 122.17		Almería 122.17
		4 líneas procesadas

Si el fichero estuviera vacío la salida sería simplemente "0 líneas procesadas". El fichero de entrada es correcto.

8. (0'5 puntos) Explique la diferencia entre variable estática y dinámica

Las siguientes funciones pueden ser útiles.

substring

```
public String substring(int beginIndex)
```

Returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.

Examples: "unhappy".substring(2) returns "happy"; "Harbison".substring(3) returns "bison"

"emptiness".substring(9) returns "" (an empty string)

Parameters: beginIndex - the beginning index, inclusive.

Returns: the specified substring.

Throws:

[IndexOutOfBoundsException](#) - if beginIndex is negative or larger than the length of this String object.

substring

```
public String substring(int beginIndex, int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.

Examples: "hamburger".substring(4, 8) returns "urge"; "smiles".substring(1, 5) returns "mile"

Parameters: beginIndex - the beginning index, inclusive. ; endIndex - the ending index, exclusive.

Returns: the specified substring.

Throws:

[IndexOutOfBoundsException](#) - if the beginIndex is negative, or endIndex is larger than the length of this String object, or beginIndex is larger than endIndex.